



INTEGRATED OPERATIONS ARCHITECTURE

ADMINISTRATION

World Leader in Automated Operations Technology



NEW
DIMENSION
SOFTWARE™

© Copyright 1996 by New Dimension Software Ltd. All rights reserved.

First Edition.

The software described in this document is furnished under a license agreement, and may be used or copied only in accordance with the terms of the agreement.

All New Dimension Software products are trademarks or registered trademarks of New Dimension Software Ltd. Other brand and product names are trademarks or registered trademarks of their respective holders.

The information in this document is subject to change without prior notice and does not represent a commitment on the part of New Dimension Software Ltd.

New Dimension Software Ltd. assumes no responsibility for any errors that may appear in this document.

I-500G-A9605B

20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Faint vertical text on the left side of the page, possibly bleed-through from the reverse side.

Faint, illegible text in the lower-left quadrant of the page.

C

C

C

About This Manual

This guide contains the information necessary for IOA administrators who are responsible for customizing and maintaining the IOA family of products.

Basic information about how IOA products operate is provided in user manuals for each product.

Most administration information for each product is found in the section devoted to that product. However, all IOA administrators should read Sections 1 and 2 before continuing with the rest of this guide. Section 1 contains an overview of the key IOA concepts which apply to all IOA products. Section 2 describes how to customize the IOA environment.

The sections in this guide are as follows:

Section 1 IOA: An Overview

Overview of the IOA environment and a description of key IOA concepts and components.

Section 2 IOA

How to customize and maintain the IOA environment.

Sections 3-7 Customizing and Administering CONTROL-x

Information on administrative tasks for each IOA product. Each section includes information on the product's New Day processing and work flow, as well as information about how the relevant product can be used to perform specific tasks. These sections also include information about the structure of relevant files, operator commands, etc.

Notes CONTROL-R is described together with CONTROL-M in Section 3.

CONTROL-V is described together with CONTROL-D in Section 4.

Section 8 IOA Maintenance

Information on maintaining and updating IOA products.

Section 9 Utilities

Special features designed to help you perform specific tasks. This section contains an alphabetized reference guide for all available utilities for each IOA product.

Section 10 Exits

Information on exits available with each IOA product which can be used to modify operations.

Appendix A IOA Application Program Names

Appendix B IOA and CONTROL Products Dataset Formatting Utilities

Appendix C Modifying IOA Online Facility Commands

Appendix D Logical Field Names for the CONTROL-T Repository

Appendix E IOA Online Options Cross-Reference

Related Publications

IOA Installation Guide

A step by step guide to installing IOA products using the IOA Customization and Installation Engine (ICE) application.

IOA Security Guides

Step by step guides to implementing security in IOA products using the IOA Customization and Installation Engine (ICE) application. Security guides are currently available for IOA interaction with RACF, CA-TOP SECRET, and CA-ACF2.

User Manuals

Product-specific manuals containing comprehensive information about the operation and implementation of each IOA product.

IOA Messages and Codes

A comprehensive listing and explanation of all IOA messages and codes.

We Want to Hear From You

If you have any comments or opinions regarding this publication, we want to hear them. Your feedback will help us improve our documentation to our mutual benefit. Send your remarks by email to publications@dddf.com. Please include the name and version number of the publication in your email.

Table of Contents

Section 1: IOA Concepts and Components

Introduction	1-1
IOA Products	1-1
Installation and Maintenance	1-2
Installation and Customization Engine (ICE)	1-2
Product Maintenance	1-3
Online Facility	1-3
Primary Option Menu	1-3
Logic.....	1-4
Automated Processing Definitions.....	1-4
Jobs.....	1-4
Missions.....	1-4
Rules.....	1-4
Monitors	1-5
IOA Monitors.....	1-5
Product-Specific Monitors	1-6
Daily Processing/New Day Processing.....	1-6
File Management	1-7
IOA Access Method.....	1-7
IOA Core.....	1-7
Product Repositories.....	1-9
Miscellaneous	1-11
Dynamic Destination Table.....	1-11
AutoEdit Facility.....	1-12
Security Implementation.....	1-12
User Exits.....	1-12
Utilities.....	1-13
Simulation	1-13
Cross-Product Communication.....	1-13

Section 2: Customizing and Administering IOA

Introduction	2-1
The IOA Online Environment	2-1
Entering the IOA Online Facility.....	2-2
IOA Online Monitor.....	2-3
Principles of Operation.....	2-3
Activating the IOA Online Monitor (IOAOMON).....	2-5
Displaying a List of All Active Users.....	2-5
Deactivating the IOA Online Monitor.....	2-6
Problem Determination.....	2-6
The VTAM Monitor (IOAVMON).....	2-7
Activating the VTAM Monitor (IOAVMON).....	2-7
Deactivating the VTAM Monitor.....	2-7
Displaying a List of All Active Users.....	2-7
Customizing the IOA Online Environment	2-8
Transaction Members.....	2-8
Program List Members.....	2-9
Allocation Members.....	2-10
Modifying IOA Online Facility Commands.....	2-12
Modifying IOA Online Facility PFKey Definitions.....	2-13
The IOA Primary Option Menu.....	2-13
Line Format Display.....	2-14
Box Format Display.....	2-15
Customizing the Menu.....	2-15
Customizing IOA Screens.....	2-17
Modifying IOA Screens and Constants.....	2-17
Constant Blocks.....	2-18
Recommended Steps for Screen Modification.....	2-19
Customizing IOA Display Format Members.....	2-19
@STYLE.....	2-20
@HEADER.....	2-21
@LINE.....	2-22
@FIELD.....	2-22
@VAL.....	2-23
@END.....	2-24
@DLM.....	2-24
Color Parameters.....	2-24
Extended Color Support.....	2-25

ISPF Characteristics	2-25
IMS/DC and IDMS/DC	2-25
COM-LETE	2-26
IRMA PC Terminal Emulator Users	2-26
Customizing Extended Color Support.....	2-26
IOA Access Method.....	2-27
File Structure	2-27
IOA Access Method Naming Conventions.....	2-28
File Utilities	2-29
File Definition Statements.....	2-30
Dual Mirror Image File Support.....	2-30
Dual Mirror Image File Parameters	2-31
Recovering From a Damaged IOA Access Method File.....	2-31
IOA Profiles	2-31
Profile Members	2-32
Profile Contents	2-33
Considerations for Profile Symbol Lines.....	2-33
SHOW Window Filters	2-34
Profile Attribute in the Screen Definition	2-34
Saving a Profile	2-35
Profile Variables.....	2-35
Window Display Variables	2-36
Color Variables	2-39
Work Mode Variables	2-40
Presentation Mode Variables	2-43
Miscellaneous Variables.....	2-45
Other Types of Profile Variables.....	2-46
Modifying IOA Product Defaults	2-46
Modifying IOA Messages	2-47
Recommended Steps for Message Modification.....	2-48
IOA Dynamic Destination Table	2-49
Replacing the Current Dynamic Destination Table	2-49
Expanding the Resource File (RES).....	2-49
Expanding the IOA LOG File (LOG).....	2-50

Section 3: Customizing and Administering CONTROL-M

General	3-1
Activating the CONTROL-M Monitor	3-1
Shutting Down the CONTROL-M Monitor.....	3-1
Modifying the CONTROL-M Sleeping Interval.....	3-2
Starting the CONTROL-M EVENT MANAGER (CMEM) Facility	3-2
Shutting Down the CMEM Facility.....	3-3
Replacing an Active CMEM Monitor.....	3-3
Job Ordering via New Day Processing	3-4
Overview.....	3-4
Sample New Day Processing	3-5
How the Sample Components Perform New Day Processing.....	3-7
Date Control Records and Enhanced Daily Checkpointing.....	3-8
Implementing New Day Processing.....	3-9
Implementation Tasks.....	3-10
Programs Called During New Day Processing.....	3-16
Use of the Date Control Record by User Daily Jobs	3-17
New Day Procedure Flow	3-19
Activation of Started Tasks	3-20
CONTROL-M Monitor and JES Considerations	3-21
Cases of JES Malfunction.....	3-21
Special Considerations.....	3-21
Considerations for Changing Clocks	3-23
Advancing the Clock Forward	3-23
Moving the Clock Backwards	3-24
Sending a Job's SYSDATA to a Held Output Class	3-25
Dual Checkpoint Mode	3-26
Managing the CMEM Facility	3-26
Automatic Loading of Rules	3-27
Deleting (Deactivating) an Active Rule Table.....	3-28
Displaying Active Rules.....	3-29
Controlling CMEM Rule Operation Mode	3-29
Modifying the CMEM Sleeping Interval	3-30
Refreshing the CMEM Security Cache.....	3-30
Problem Determination	3-30

Inter-System Communications Facility	3-32
Method of Operation	3-32
Create the Inter-System Communication Files	3-33
Prepare the Condition Selection List.....	3-33
Modify JCL Procedures and CLISTs.....	3-34
Copy JCL to Procedures Library.....	3-34
Customize the Exits	3-35
Start the Communication Monitors	3-35
Expanding CONTROL-M Files	3-35
Expanding the Active Jobs File (CKP).....	3-35
Expanding the CONTROL-M Jobs Dependency Network File (GRF)	3-36
Expanding the CONTROL-M Statistics File (STAT).....	3-36

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

General	4-1
Activating the CONTROL-D Monitor	4-1
Activating Generic Processing	4-1
Activating the Compressed Dataset Access Method.....	4-2
Modifying the CONTROL-D Sleeping Interval.....	4-2
Reloading the Recipient Tree.....	4-3
Loading the Recipient Tree Into the IOA Online Monitor	4-3
Reloading the Manual Conditions File	4-3
Activating the IOA Archive Server (CONTROL-V)	4-4
Deactivating the CONTROL-D Monitor	4-4
Deactivating Generic Processing	4-4
Automatic Warning	4-5
Deactivating the Compressed Dataset Access Method	4-5
Deactivating the IOA Archive Server (CONTROL-V)	4-5
New Day Processing	4-6
Overview	4-6
Starting the New Day Procedure	4-6
New Day Procedure - Workflow	4-7
User Daily Job.....	4-7
Date Control Record.....	4-7
Use of the Date Control Record by the User Daily Job.....	4-8
Use of the Date Control Record by the New Day Procedure.....	4-10

Table of Contents

Programs Called During New Day Processing	4-10
Parameters of the New Day Procedure	4-12
Mission Scheduling	4-13
Overview	4-13
Scheduling - General	4-13
Using the New Day Procedure (CTDNDA) to Schedule Missions	4-14
Supplied Mission List Members.....	4-15
Using a User Daily Procedure to Schedule Missions	4-16
Scheduling a Mission Manually	4-16
Scheduling Missions -Workflow.....	4-17
Decollation Mission Management	4-19
Overview	4-19
Generic Decollating Missions	4-19
Generic Decollating Mission Workflow.....	4-20
Additional Considerations	4-21
Scheduling Generic Decollating Missions Via the New Day Procedure.....	4-21
Controlling the Generic Process	4-22
Defining a Generic User Name List	4-22
Interfaces to Production Control Systems.....	4-23
Overview of CONTROL-M Scheduling With CONTROL-D.....	4-23
Job-Report Dependency	4-25
Scheduling via the CONTROL-M Production Control System.....	4-26
Scheduling via a Non-CONTROL-M Production Control System.....	4-27
Considerations for When CONTROL-M and CONTROL-D Are Installed	4-29
Printing Mission Management	4-30
Printing Mission Execution Workflow.....	4-31
Advanced Scheduling Issues	4-34
Distribution According to Scheduling Dates	4-34
Report Decollating and Printing Mission Dependency	4-34
Printer Control	4-34
One-Chunk Method.....	4-35
Multi-Chunk Method	4-35
One-Outgroup Method (JES2 Only)	4-36
Opening/Closing Printers	4-36
Printing Process	4-37
Identifying CONTROL-D Chunks on Spool (JES2 Only).....	4-38
Printing on AFP (APA) Printers.....	4-38
Using the CDAM PAGEDEF/FORMDEF Parameters.....	4-39
Using a Specified OUTPUT Statement	4-39

AFP Page Mode Output.....	4-39
Page Markers Under AFP	4-40
In-Stream AFP Control Statements.....	4-40
Printing Using XEROX LCDS (DJDE) Parameters.....	4-42
Advanced ACIF Interface Facility	4-43
What is AFPDS?	4-43
What is ACIF?	4-43
Using OUTPARM for Global Control of Printing Characteristics.....	4-47
Printing to a File.....	4-49
Tailoring CTDX005.....	4-49
Backup Mission Management.....	4-51
Overview	4-51
Advanced Scheduling Issues.....	4-51
Backup According to Scheduling Dates.....	4-51
Report Decollating and Backup Mission Dependency	4-52
Backup Mission Workflow.....	4-53
Exception Handling.....	4-55
Changing Backup Mission Retention Period.....	4-55
Migration Mission Management (CONTROL-V).....	4-55
Overview	4-55
Advanced Scheduling Issues	4-56
Migration Mission Workflow.....	4-56
Dual Migration.....	4-58
Migration Media Definition	4-58
Cartridge (CART) Media Management.....	4-59
Migration Mission Skeletons.....	4-59
Migration to OSS.....	4-59
Migration to Cartridge	4-60
Migration to Disk, Fat-DASD, IBM 3995 Optical Library Dataserver.....	4-60
Migration to FileTek Storage Machine	4-60
Exception Handling.....	4-61
Restore Mission Management	4-61
Overview	4-61
Advanced Scheduling Issues.....	4-61
Restoring According to Scheduling Dates.....	4-61
Restore Mission Workflow.....	4-62
Exception Handling.....	4-64
Restoring With the Original Backup Utility.....	4-64
IOA Archive Server	4-65

Table of Contents

General	4-65
Device Definition and Usage	4-65
Cartridge (CART) Media Definition	4-65
KODAK/DataWare OSS Usage	4-66
FileTek Storage Machine Media Definition.....	4-66
Media and Resource Control.....	4-67
Media Control	4-68
Resource Control.....	4-68
Displaying Media Information.....	4-71
Problem Determination	4-72
User Report List File Management.....	4-72
Overview	4-72
Permanent User Report List File	4-74
Active User Report List File.....	4-75
Migrated User Report List File	4-76
History User Report List File.....	4-77
Summary.....	4-78
Repository Maintenance.....	4-79
General	4-79
Expanding the Active Missions File.....	4-79
Expanding the Active Transfer File	4-80
User Reports List File Housekeeping.....	4-80
Reorganizing the Files	4-80
Rebuilding the Index Component.....	4-81
Recovering a Damaged File	4-82
SMF Accounting	4-82

Section 5: Customizing and Administering CONTROL-O

General	5-1
Starting CONTROL-O	5-1
Shutting Down CONTROL-O	5-1
Replacing an Active CONTROL-O Monitor	5-2
Automatic Loading of Rules	5-2
Manual Loading of Rules	5-3
Manual Loading of CMEM Rules	5-4
Deleting (Deactivating) an Active Rule Table	5-5
Recommended Organization Method.....	5-5
Use of Two Rule List Members	5-7
Replacing the IPL CONTROL-O	5-8
Rule Scheduling	5-8
Displaying Active Rules	5-9
Controlling the Message Statistics Facility	5-9
Preventing Unnecessary Enlargement of Statistics Files	5-9
Handling Near-Full and Full Conditions for the Statistics File	5-10
Controlling the Automation Log Facility	5-10
Determining the Size of the Log	5-11
Preventing Logging of Unnecessary Messages	5-11
Controlling Rule Operation Mode	5-12
Global AutoEdit Variables	5-12
Defining a New Global Member	5-13
Automatic Compression of the Global AutoEdit Library	5-14
Modifying the CONTROL-O Sleeping Interval.....	5-15
Refreshing the CONTROL-O Security Cache	5-16
Problem Determination.....	5-16
Customization of Automation Options	5-18
AOP Overview	5-18
Menus	5-19
Menu Member Syntax	5-19
Parameters	5-19
Format Members	5-23
Client Programs	5-23
Program Input.....	5-24
Execution Process.....	5-24
Client Program Linkage Conventions.....	5-25
Considerations for When CONTROL-M and CONTROL-O Are Installed.....	5-25

Section 6: Customizing and Administering CONTROL-B

General	6-1
CONTROL-B New Day Procedure	6-1
Reformatting the Active Balancing File - Utility CTBFRM.....	6-2
Scheduling Balancing Missions	6-2
Date Control Record	6-2
Format of the Date Control Record	6-3
Use of the Date Control Record by the New Day Procedure.....	6-3
Invoking CONTROL-B (Runtime Environment)	6-4
Passing Arguments While Invoking CONTROL-B.....	6-5
Invoking CONTROL-B Via a Direct Call	6-6
Invoking CONTROL-B From a Job Step or Application Program	6-7
Invoking CONTROL-B by CONTROL-M.....	6-8
Invoking CONTROL-B by CONTROL-D	6-8
Invoking CONTROL-B With Balancing Missions	6-9

Section 7: Customizing and Administering CONTROL-T

General	7-1
The CONTROL-T Real-Time Environment	7-1
Initialization	7-2
Procedure CTTINIT	7-2
Loading of Rules	7-6
Termination	7-6
New Day Procedure	7-7
New Day Functions	7-7
Repository Structure	7-9
Media Database Structure.....	7-9
Data File Contents.....	7-10
Fields in the Data File.....	7-11
Index File Contents	7-12
Structure of the Trace File	7-17
Structure of the Stacking Statistics File	7-17
Repository Maintenance	7-19
Verifying Data Integrity of Media Databases	7-19

Manual Update of the Media Database	7-21
Enlarging the Media Database	7-21
Enlarging the Trace File	7-22
Repository Backup and Recovery	7-23
General.....	7-23
Backup and Recovery	7-23
Disaster Recovery.....	7-24
Selective Recovery.....	7-25
General Considerations.....	7-25
Robotic Library Support	7-25
Robotic Tape Libraries Using System Managed Tape Functions.....	7-25
IBM Tape Library Dataserver Support.....	7-26
StorageTek Silo Support	7-28
MEMOREX TELEX Automatic Tape Library (ATL) Support.....	7-32
EMASS Automatic Tape Library (ATL) Support.....	7-32
Cartridge Message Display Customization.....	7-33
External Data Manager Support.....	7-35
HSM Support.....	7-35
DMS/OS Support.....	7-35
CA-ASM2 Support.....	7-36
DF/SMS Support.....	7-37
Adjusting ACS Routines.....	7-38
Dynamic Dataset Stacking Facility.....	7-39
General Considerations.....	7-40
Activating the Dynamic Dataset Stacking Facility	7-40
Activating the Dynamic Dataset Stacking Facility in Global Test Mode.....	7-41
Testing the Dynamic Dataset Stacking Facility	7-41
Activating the Dynamic Dataset Stacking Facility for the Entire System.....	7-42
CONTROL-R Support to CONTROL-T.....	7-43
CONTROL-R Driver Exit CTRX001G.....	7-43
Installing the CONTROL-R Interface.....	7-43
CONTROL-T Application Programming Interface	7-44
Background Information.....	7-45
CONTROL-T Control Table (TCT).....	7-45
Media Database Structure	7-45
Base Level vs. High Level API.....	7-46
Base Level API	7-47
Record Access by the Base Level API	7-47
Macro CTTIOS	7-47

Table of Contents

Base Level API Access of the Media Database.....	7-51
Examples for Base Level API.....	7-53
High Level API.....	7-68
Macro CTTACCDB.....	7-68
Macro CTTCHKDB.....	7-70
Media Database Error Handling Routine.....	7-70
Example for High Level API.....	7-71

Section 8: Maintaining IOA Products

Introduction.....	8-1
IOA Release 5.0.0 Packaging.....	8-1
Product Packaging.....	8-1
Language Support.....	8-2
Applying Periodic Maintenance.....	8-3
Preliminary Steps.....	8-3
Installation Steps.....	8-3
Step 1: Apply Periodic Maintenance.....	8-3
Applying Ad Hoc Maintenance.....	8-4
Introduction.....	8-4
Installation Steps.....	8-4
Step 1: Run SMP/E to RECEIVE, APPLY CHECK and APPLY the Fix.....	8-4
Step 2: Customize and Propagate Fix Elements to Working Libraries.....	8-5
Step 3: Enable Fixes Applied to ICE.....	8-10
Step 4: Copy JCL Procedures and ISPF Elements to System Libraries (Optional).....	8-11

Section 9: Utilities

General	9-1
Summary of Utilities	9-1
IOA Utilities	9-1
CONTROL-M Utilities	9-2
CONTROL-D/V Utilities	9-2
CONTROL-O Utilities	9-3
CONTROL-B Utilities	9-4
CONTROL-T Utilities	9-5
IOA Utilities	9-7
IOABLCAL - Build Calendars	9-7
IOACLRES - Clean IOA Conditions/Resources File	9-13
IOACND - Perform Condition/Resource Adjustments	9-16
IOACOPRS - Copy/Resize IOA Conditions/Resources File	9-20
IOACPLOG - Copy the IOA Log File	9-22
IOADBF - Allocate/Format an IOA Access Method File	9-24
DD Statement DAFILE	9-24
IOA Access Method File Definition Statements	9-25
IOADBSR - Sort Records in an IOA Access Method Data Component	9-31
IOADCPY - Recover an IOA Access Method File	9-33
IOADIG - Verify Integrity of an IOA Access Method File Data Component	9-35
IOADII - Verify Integrity of an IOA Access Method File Index Component	9-36
IOADLD - Load IOA Access Method File Contents	9-37
IOADPT - Print IOA Access Method File Content	9-38
IOADUL - Unload IOA Access Method File Contents	9-39
IOALDNRS - Load Manual Conditions File	9-41
IOANOTE - Write a Message to the IOA Log File	9-44
IOATEST - Simulate Production Environment Events	9-45
IOAVERFY - Verify IOA File Installation and Status	9-47
CONTROL-M Utilities	9-49
CTMBGRP - Convert Regular Scheduling Tables to Group Scheduling Tables	9-49
CTMCAJF - Maintain the Active Jobs File	9-51
CTMJSA - Job Statistics Accumulation	9-55
CONTROL-B Utilities	9-56
CTBABI - Initialize Active Balancing File	9-56
CTBCMP - Compile Rule Definitions	9-57
CTBDBIB - Rebuild a CONTROL-B Index File	9-58
CTBDBVCG - Change Number of Variable Generations	9-60

Table of Contents

CTBDBVCP - Copy Variables or Groups to Another File 9-62

CTBDBVDL - Delete Variables/Groups from Variable/Group Files..... 9-65

CTBDBVIG - Check the Integrity of Variable/Group Files..... 9-67

CTBFRM - Compress the Active Balancing File 9-68

CTBJAFCP - Copies Entries in Rule Activity/Report Files 9-69

CTBJAFDL - Deletes Entries From Rule Activity/Report Files 9-71

CTBJAFIG - Check the Integrity of Rule Activity/Report Files 9-73

CONTROL-D/V Utilities..... 9-74

 CTDCAMF - Compress Active Missions File 9-74

 Compressing the Active Missions File 9-74

 Changing the Size of the Active Missions File 9-74

 CTDCATF - Compress Active Transfer File 9-76

 Compressing the Active Transfer File 9-76

 Changing the Size of the Active Transfer File 9-76

 CTDCA2P - Copy Active User Report File to Permanent 9-78

 CTDCCU - Delete Unneeded CDAM Files 9-80

 CTDCLHIS - Clean History User Report List File..... 9-82

 MODE Parameter 9-82

 CTDCP2A - Copy Permanent User Report File to Active 9-83

 CTDDELRP - Delete Reports Which Are Not Needed 9-85

 Parameters 9-85

 DAYS Statement 9-86

 INCLUDE/EXCLUDE Statements..... 9-87

 CTDDELRP Deletion Rules 9-88

 Examples..... 9-89

 CTDDIB - Rebuild Index Component of a User Report List File..... 9-91

 CTDUPBKP - Update Backup Details..... 9-92

 Selection Parameters 9-93

 CHANGETO Parameters 9-94

 Report Parameters 9-95

 CTVCLMIG - Clean Migrated CDAM Files 9-98

 CTVDELI - Delete Index Files from DASD 9-99

 EXEC Statement 9-99

 DAYS Statement 9-100

 INCLUDE/EXCLUDE Statements..... 9-100

CONTROL-O Utilities	9-103
CTOCSF - Copying the Message Statistics File.....	9-103
CTORSTM - Statistics Report Utility	9-104
SORT Parameter	9-104
INCLUDE/EXCLUDE Statements	9-104
CTOALOCP - Backing Up the Automation Log.....	9-105
CONTROL-T Utilities	9-107
Record Selection Logic (INCLUDE/EXCLUDE Statements)	9-107
General.....	9-107
Expressions.....	9-107
INCLUDE Statements.....	9-107
EXCLUDE Statements	9-108
INCLUDE/EXCLUDE Block.....	9-108
Single INCLUDE/EXCLUDE Block Processing.....	9-108
Multiple INCLUDE/EXCLUDE Blocks	9-109
INCLUDE/EXCLUDE Examples.....	9-109
CTTACP - Copy Trace File.....	9-111
CTTAFR - Format and Initialize Trace File	9-113
CTTBIX - Rebuild Media Database Index	9-114
CTTCRDF - Allocate and Format Data/Index Files.....	9-116
CTTCTRC - Allocate and Format Trace File	9-119
CTTDBDLD - Load Media Database/Stacking Statistics File Data Components	9-120
CTTDBDUL - Unload Media Database/Stacking Statistics File Data Components ..	9-121
CTTDBF - Format Existing Data/Index Files.....	9-122
CTTDBIB - Rebuild Indexes of Stacking Statistics File.....	9-123
CTTDBID - Check Integrity of Data File.....	9-124
CTTDBII - Check Integrity of Index File.....	9-125
CTTDBPRT - Print Data Records	9-126
CTTDLDD - Add/Delete/Convert Volumes	9-127
CTTGVL - User Interface to Stacking Facility	9-130
CTTIDB - Check Integrity of the Media Database	9-132
CTTMER - Merge Records Into the Media Database.....	9-134
CTTMUP - Manually Update the Media Database.....	9-136
CTTRCV - Recover the Media Database Using Trace File.....	9-144
CTTRPT - Database Extraction and Report Utility.....	9-148
CTTRTM - Perform Retention Management.....	9-167
CTTSPL - Split the Media Database.....	9-177
CTTSTK - Build and Update Stacking Statistics File	9-182

Table of Contents

CTTTPI - Tape Initialization, Information Mapping, and Tape Erasure9-184
 Safe Tape Initialization - Function INITT9-184
 Media Information Mapping - Function TAPEMAP9-185
 Safe Tape Erasure - Function TAPERAS9-186
 CTTTPI and I/O Error Recovery9-187
 Input Parameters - SYSIN DD File9-189
 Invoking CTTTPI From Another Program9-195
CTTTRB - Synchronize Trace File and Media Database Backup9-197
CTTUDB - Update the Media Database From Transaction File9-198
CTTVTM - Perform Vault Management9-199

Section 10: Exits

General 10-1
IOA Exits 10-2
CONTROL-M Exits 10-5
CONTROL-R Exit 10-8
CONTROL-D/V Exits 10-9
 Replacing CONTROL-D User Exits10-14
 Tailoring CONTROL-D Banner Exit10-14
 Banner Pages10-15
CONTROL-B Exits 10-21
CONTROL-T Exits 10-22
CONTROL-O Exits 10-23

Appendix A: IOA Application Program Names

Appendix B: Dataset Formatting Utilities for IOA Products

Appendix C: Modifying IOA Online Facility Commands

Modifying IOA Online Facility PFKey Definitions.....	C-1
--	-----

Appendix D: Logical Field Names for the CONTROL-T Repository

Table 1: Keywords to Be Used With Dataset Type Records	D-2
Table 2: Keywords to Be Used With Volume Type Records.....	D-6
Table 3: Keywords to Be Used With Stacking Type Records.....	D-10
Table 4: Keywords to Be Used With Trace Type Records.....	D-11
Table 5: Keywords to Be Used With Trace Data.....	D-12

Appendix E: IOA Online Options Cross-Reference

IOA.....	E-1
CONTROL-M.....	E-2
CONTROL-D.....	E-3
CONTROL-V.....	E-5
CONTROL-O.....	E-8
CONTROL-B.....	E-9
CONTROL-T.....	E-10
ISPF.....	E-12

Index



Section 1: IOA Concepts and Components

Introduction	1-1
IOA Products	1-1
Installation and Maintenance	1-2
Installation and Customization Engine (ICE).....	1-2
Product Maintenance	1-3
Online Facility	1-3
Primary Option Menu.....	1-3
Logic	1-4
Automated Processing Definitions	1-4
Jobs	1-4
Missions	1-4
Rules	1-4
Monitors	1-5
IOA Monitors	1-5
Product-Specific Monitors	1-6
Daily Processing/New Day Processing.....	1-6
File Management	1-7
IOA Access Method.....	1-7
IOA Core	1-7
Product Repositories	1-9
Miscellaneous	1-11
Dynamic Destination Table.....	1-11
AutoEdit Facility.....	1-12
Security Implementation.....	1-12
User Exits	1-12
Utilities	1-13
Simulation	1-13
Cross-Product Communication	1-13



Introduction

IOA (Integrated Operations Architecture) is a fully integrated family of products designed to help you streamline and automate your mainframe operations.

IOA provides you with the capability of implementing unattended, "lights out" operations.

The IOA family of products share a set of common components and concepts. This section contains an introduction to the IOA products and the basic concepts and components used in IOA processing.

IOA Products

The IOA products, which can be used to automatically manage the functioning of an MVS data center are:

- | | |
|--------------|---|
| CONTROL-M | Production Control System

Manages and automates setup, scheduling and execution of jobs in your MVS data center. |
| CONTROL-R | Automated Job Restart

Automates activities which must be performed when restarting a failed job, including scratching and uncataloging of datasets created by the failed job. |
| CONTROL-D | Output Management System

Automatically schedules and controls every aspect of report processing and distribution, including report decollating, bundling, printing, Online Viewing and backup. |
| CONTROL-D/PC | Online Viewing and Manipulation of Mainframe and Reports

Comprehensive output viewing system which enables viewing, editing and printing of mainframe and non-mainframe reports on a PC. |
| CONTROL-V | Quick Access Archive Viewing System

Replaces microfiche and provides online access to mainframe reports and archived data via indexed data retrieval. |
| CONTROL-O | Console Management System

Increases system availability and reliability by monitoring all system messages, responding to messages/commands automatically, etc. CONTROL-O also highlights messages requiring human intervention and automates real-time automation tasks. |

Section 1: IOA Concepts and Components

CONTROL-O/PC Remote Console Automation

Allows remote console automation of an MVS mainframe via a PC, and provides consolidation and management of messages from multiple consoles via a single point of control.

CONTROL-T Removable Media Management

Optimizes removable media utilization and provides effective library control. CONTROL-T controls retention periods, prevents misuse of media, and provides improved vaulting control.

CONTROL-B Information Integrity System

Performs in-stream validation, and accuracy/reasonability checks on information used by data center production tasks (e.g., reports, files, databases).

Related to the IOA family of products is the Enterprise Controlstation family of products which enables automation of MVS and non-MVS platforms.

A number of common components are shared by IOA products. Furthermore, various IOA products use many similar operating procedures.

This remainder of this section contains a brief description of components and features common to IOA products, followed by references indicating where more detailed information can be found.

Installation and Maintenance

IOA installation is performed using an ISPF menu-driven interface (Installation and Customization Engine).

Maintenance is performed periodically using SMP/E.

Installation and Customization Engine (ICE)

ICE is an ISPF application used to install and customize IOA products. It consists of a series of dialog boxes that provide a simple method for entering data, creating and submitting installation jobs.

Tasks in ICE are organized in a hierarchical manner.

- ICE displays the IOA installation online activities in the IOA Installation Main menu. Each menu option represents a key activity related to installing and maintaining IOA (e.g., INSTALL IOA, INSTALL CTx, SECURITY, and CUSTOMIZE).
- Each activity is organized as a high level list of major steps. Each major step consists of a one or more minor steps.
- Each minor step allows the user to perform tasks such as parameter specification, submission of jobs, or other tasks related to the installation process.

The ICE application is described in detail in the IOA Installation Guide.

Product Maintenance

Product Maintenance is the process by which updates are implemented in the products between releases. In IOA Release 5.0.0 these updates are implemented via SMP/E only.

Product Maintenance consists of - Periodic Maintenance and Ad hoc Maintenance:

- Periodic Maintenance contains improvements to the software and solutions to problems that were discovered during day-to-day product use. As the name suggests, this kind of maintenance is distributed periodically to the users.
- Ad hoc Maintenance contains one or more fixes to a specific problem. It is distributed to the user as the need arises.

See Section 8 for a detailed description of maintenance implementation.

Online Facility

The IOA Online facility is a set of interactive applications which provides users with the basic means of communication with IOA, thus, simplifying implementation of IOA operations. The IOA Online facility generates all IOA screens and handles all IOA online functions.

The IOA Online facility can be accessed directly under TSO, ISPF, TSO/ISPF and ROSCOE. In addition, the IOA Online facility (IOAOMON) can be accessed via CICS, IMS, IDMS, VTAM, ROSCOE, TSO, COM-PLETE or VM under the IOA Online monitor.

Upon entering the IOA Online environment (the IOA Online facility accessed under a communication monitor or TSO/ISPF), the user is presented with a menu listing available options (called the Primary Option menu).

The IOA Online facility is described later in this section and in Section 2 of this guide.

Primary Option Menu

The Primary Option menu displays IOA product options and facilities available to the user.

The Primary Option menu is dynamically generated according to the products and/or options available to the user. The IOA administrator can optionally customize the environment so that specific products and/or options are available only to certain users or user groups.

The IOA Primary Option menu is displayed in one of two possible formats, depending on the number of options available to the user.

Line Format Options are listed one per line with a description of each option.

Box Format Options are grouped according to product. No option description is displayed.

Reducing or increasing the number of available options may change the format in which the Primary Option menu is displayed.

Menu format and customization are described in detail in "Customizing the IOA Primary Option Menu" in Section 2 of this guide.

Section 1: IOA Concepts and Components

Logic

Listed below are the different components of IOA products.

Automated Processing Definitions

Automated processing definitions are the means by which the user communicates to the IOA products what tasks to perform and when to perform them. Automated processing definitions contain action statements that describe actions to be performed and criteria which define when and/or under what conditions the defined actions should be performed. Each IOA product contains a definition facility in which automated processing is defined.

Automated processing definitions are referred to as jobs, rules, or missions - depending on the IOA product and the type of action indicated in the definition.

Job/rule/mission definitions are stored in tables (members) in partitioned datasets. Definitions are activated by scheduling (ordering) them to an active environment (file or memory, depending on the product). Only then can the product process these definitions.

Automated processing definitions are described in detail in Section 2 of each IOA product user manual.

The different types of automated processing definitions are described in the following paragraphs.

Jobs

Job scheduling definitions are used by CONTROL-M to handle job processing. They can be defined for jobs, started tasks, etc. Each job scheduling definition indicates scheduling criteria and/or conditions to be met before CONTROL-M schedules or submits a particular job. Each definition also indicates actions to be performed by CONTROL-M after the job terminates. CONTROL-M can perform different post-processing actions, depending on the execution results.

CONTROL-R parameters and statements are also defined in CONTROL-M job scheduling definitions.

Missions

Missions are used by CONTROL-D/CONTROL-V, to determine actions to be performed on job output. When a mission's scheduling criteria are met, the actions indicated by the statements in the mission definition are performed.

Missions are used by CONTROL-B to define scheduling criteria to be applied to CONTROL-B rules which indicate actions to be performed.

Rules

Rules are used by CONTROL-O, CONTROL-T, the CONTROL-M Event Manager (CMEM), and CONTROL-B to respond to specified events (e.g., the detection of a message by CONTROL-O).

The occurrence of the specified event triggers the rule which, results in the performance of the actions specified in the rule.

Scheduled rules remain active in memory after the performance of the rule. They can be triggered each time an occurrence of the specified event is detected. There is no limit to the number of times a rule can be triggered.

Note CONTROL-B rule definitions contain only action statements. CONTROL-B rules are either executed upon request without regard to scheduling criteria, or according to scheduling criteria specified in a CONTROL-B mission described above.

Monitors

IOA monitors are started tasks which perform IOA tasks. There are two types of monitors, IOA monitors and product monitors.

Instructions for how to activate and deactivate each monitor are provided in the sections for the relevant products.

A brief description of each monitor and its functions follows.

IOA Monitors

IOA monitors perform tasks which may be initiated from any of the CONTROL products.

IOA Online Monitor (IOAOMON)

The IOA Online monitor interacts with different environments (e.g., CICS) to provide an online interface for the different IOA applications. The online monitor generates the IOA product screens which are available to users signed on to the online monitor, and handles all IOA Online functions.

The IOA Online monitor usually operates 24 hours a day as a started task. The IOA Online monitor is usually activated automatically as part of the IPL process.

See "The IOA Online Environment" in Section 2 of this guide for more information.

IOA VTAM Monitor (IOAVMON)

The VTAM monitor (IOAVMON) enables access to the IOA Online monitor for VTAM users without passing through any TP monitors (e.g., CICS, IMS/DC).

IOA Functional Monitor (IOAFMON)

CONTROL-T uses the IOA Functional monitor to process certain action statements (e.g., DO CONDITION, and DO SHOUT).

When an event occurs (e.g., volume checked in, dataset created) which requires execution of an action statement via the Functional monitor, the CONTROL-T component writes a trace record that describes the needed action. The IOA Functional monitor reads the CONTROL-T Trace file and processes the specified statements.

Section 1: IOA Concepts and Components

The Functional monitor is described in detail in the CONTROL-T User Manual and the IOA Installation Guide.

IOA Archive Server Monitor (IOASMON)

The Archive server provides access to all reports migrated to non-DASD media, thus enabling CONTROL-V users to view and print reports.

Product-Specific Monitors

Specific product monitors are responsible for many of the automated operations performed by each product.

CONTROL-M

The CONTROL-M monitor scans its active environment (file) and other files and determines when jobs should be submitted. It submits jobs, tracks their execution and analyzes the results.

If CONTROL-O is not installed, and the CONTROL-M CMEM facility is active, a CMEM monitor is activated which processes CMEM rules that were triggered by events in the system.

CONTROL-O

The CONTROL-O monitor processes rules that were triggered by events in the system and records statistics on rule activation and message detection.

If CONTROL-M is installed, the CONTROL-O monitor also assumes responsibility for processing of CMEM rules.

CONTROL-D/V

CONTROL-D activates two monitors, the main monitor (the CONTROL-D monitor) and the Printers Control monitor. The CONTROL-D monitor scans its active environment (file) and other files, and determines when missions should be processed. It processes missions, tracks mission execution and analyzes the results.

If CONTROL-V is installed, the monitor also handles indexing of reports and their migration to other storage media (e.g., Optical Storage, cartridges).

The CONTROL-D Printers Control monitor creates subtasks for each print mission, thus enabling parallel running of print missions.

Daily Processing/New Day Processing

For most IOA products, there are a number of tasks that should be performed each day. Typical daily tasks are:

- Updating the active environment (file or memory) with definitions for the new day.
- Checking files for integrity and validity.
- Producing general reports that describe the actions taken during the previous day.
- Housekeeping. Cleaning unneeded information from product and IOA files.

The above tasks are performed automatically via the New Day procedure for each product. The New Day procedure is a program used to automate daily maintenance tasks and automatically schedule processing definitions for each day.

New Day procedures and User Dailies are described in the beginning of the sections for the relevant products in this guide (e.g., Section 3 for CONTROL-M).

File Management

There are two major ways to differentiate between files which are used by IOA products.

- According to their access method
 - Files managed using a common internal access process called the IOA Access Method.
 - Files managed using standard access methods.
- According to their usage
 - IOA Core files which are shared among IOA products.
 - Product repository files which are exclusively used by a particular IOA product.

IOA Access Method

Certain IOA products have files that are managed using an internal file access process called the IOA Access Method.

IOA Access Method files are sequential datasets in an indexed (keyed) file structure that offers enhanced data integrity. IOA Access Method files are managed (e.g., allocated, formatted and accessed) via a special set of IOA utilities (described in Section 9 of this guide).

Most IOA Access Method files consist of two separate sequential datasets:

- A data component, where the actual information resides.
- An index component, which provides keyed access to records in the data component.

Certain IOA Access Method files contain only an index or data component.

Both index and data information are stored in compressed format, resulting in reduced I/O processing overhead and improved disk space utilization.

The IOA Access Method is described in greater detail in Section 2 of this guide.

IOA Core

The IOA Core is a collection of files which are shared by all IOA products at your site. The following IOA files comprise the IOA Core:

Log File

Contains messages generated by IOA products. Messages can be viewed via option 5 in the IOA Primary Option menu. The IOA Log file is cyclic and stores a limited number of messages (specified via an installation parameter). Each new entry in the IOA Log file overwrites the oldest existing entry if required.

The IOA Log file is described in detail in Section 2 of each product-specific user manual.

Section 1: IOA Concepts and Components

Conditions/Resources File

The IOA Conditions/Resources file contains information on the status and availability of resources system wide. IOA conditions and resources can be viewed via option 4 in the IOA Primary Option menu. The following types of information are included in this file:

Prerequisite conditions User-defined conditions set as "in conditions" (via IN statements) or "out conditions" (via OUT and/or DO COND statements) in job/rule/mission definitions.

- When a condition is defined as an "out condition," it is set (in the IOA Conditions/Resources file) during or following the processing of the job/rule/mission.
- When a condition is specified as an "in condition," the condition must exist in the IOA Conditions/Resources file before the job/rule/mission can be processed.
- Prerequisite conditions can be used in this way to make the execution of one job/rule/mission dependent on the execution of another. For example, a job with a particular "in condition" cannot be submitted until a job with the same condition specified as an "out condition" was executed and set the condition.

Prerequisite conditions can also be used to indicate that a required manual operation has been performed.

Quantitative resources Quantity of a resource in the system. Different jobs may require different quantities of specific resources (e.g., a job may require two tape drives).

Specification of Quantitative resource requirements for a job provides a solution for the allocation of quantitative computer resources (e.g., cartridge drives, CPU utilization, database access rate). Specification of Quantitative resource requirements increases computer throughput by controlling access to these resources, thus preventing execution bottlenecks.

Control resources Usage (Shared/Exclusive) of a specific resource. Some jobs may require a resource to be in a specific mode (e.g., a backup job may require exclusive access to a specific dataset).

Specification of Control resource requirements for a job/mission provides a solution for the problem of resource sharing between different jobs. The Exclusive or Shared mode in which a Control resource is required by a job is also specified.

IOA considers the mode of resource usage required when allocating Control resources and prevents jobs whose resource usage is incompatible from executing simultaneously.

See Sections 1 and 2 in the appropriate user manual for more information regarding the use of the IOA Conditions/Resources file.

Manual Conditions File

File that contains prerequisite conditions that must be added manually (e.g., they are required by jobs in the active environment but are not automatically added by other jobs in the active environment).

Calendar Tables

Calendars determine on which days automated process definitions (jobs/rules/missions) should be processed. Calendars can be used to simplify the definition of scheduling criteria.

IOA calendars are organized on a yearly basis, each calendar listing all the days in a specified year, with an indication stating if an automated definition should be processed that day.

Each calendar is assigned a unique name, which can be specified via Basic Scheduling parameters of an automated processing definition (subparameters DCAL, WCAL, and/or parameter CONFCAL).

Calendars can be defined as either regular or periodic:

- Regular calendars contain schedules that can be easily defined using Basic Scheduling parameters. These calendars consist of scheduling dates or days (of the week) which can be fixed according to monthly patterns.

Regular calendars are especially useful when a large number of jobs have the same schedule. By defining the schedule once in a calendar, and specifying the calendar name in the automated processing definitions with that schedule, it is unnecessary to individually define that schedule in each definition.

- Periodic calendars are especially useful when basic scheduling criteria would be difficult to define because they do not easily conform to fixed date/day of the week/month breakdowns. For example, you may need to schedule a particular job every fourteen days regardless of date.

The IOA Calendar facility is described in Section 2 of each product-specific user manual.

Product Repositories

Each IOA product uses a variety of different files to accumulate information about its operations. These files are referred to collectively as the product repositories (regardless of their access method). A list of some of the more important files in each IOA product's repository and a brief description of the contents of each file. More information about each product's repository can be found in Section 1 of each product-specific user manual.

Note In addition to the following list, libraries and tables which contain automated process definitions for each IOA product are also part of each IOA product repository.

CONTROL-M

Active Jobs file

The CONTROL-M active environment contains copies of the ordered job scheduling definitions and the status of those jobs. Accessible via option 3 in the IOA Primary Option menu.

Section 1: IOA Concepts and Components

Job Statistics file Job execution statistics (e.g., elapsed time for each job). Updated via utility CTMJSA. Accessible via option S in the Status screen (which is accessed via option 3 in the IOA Primary Option menu).

Job Network file Dependency information about jobs in the Active Jobs file. Accessible via option N in the Status screen (which is accessed via option 3 in the IOA Primary Option menu).

CONTROL-D/V

Active Missions file The CONTROL-D/V active environment contains copies of pending, executing and recently executed missions, and mission status. Accessible via option A in the IOA Primary Option menu. Managed by the IOA Access Method.

User Report List files Information about reports sent to a specific user. Reports can be selected for Online Viewing, deleted, and transferred to other users. Accessible via option U in the IOA Primary Option menu. These files are managed via the IOA Access Method utilities.

The following types of user report list files exist:

- | | |
|-----------|---|
| Permanent | A list of all reports defined to CONTROL-D. |
| Active | Recently (and/or soon to be) processed reports. |
| Migrated | Reports migrated from DASD to various storage devices (CONTROL-V only). |
| History | Reports backed up on tape or cartridge. |

Active Transfer file Information about packets that are scheduled for transfer to a PC. Only relevant for sites with CONTROL-D/PC. Accessible via option F in the IOA Primary Option menu.

CONTROL-O

Message Statistics file List of all messages and commands detected in the system, and various information about each message (e.g., frequency, and whether or not they were suppressed). Accessible via option OM in the IOA Primary Option menu.

Automation log Automation information about all input available to CONTROL-O:

- Console, IMS, and CICS messages and commands.
- CONTROL-O internal messages.
- CONTROL-O rule traces.

The Automation log can be accessed via option OL in the IOA Primary Option menu.

Global Variables library A list of all user-defined Global variables. The members of this library are loaded when CONTROL-O is started, and these members can be updated, defined, or updated via various operator commands.

CONTROL-B

Active Balancing file	List of all CONTROL-B missions loaded during the CONTROL-B New Day procedure and information about rule status. Accessible via option BB in the IOA Primary Option menu.
CONTROL-B Report file	Detailed information (Balancing report) about each invoked CONTROL-B rule. Accessed via option R in the Rule Activity screen (BA).

CONTROL-T

Media Database	Information on all datasets and volumes managed by CONTROL-T.
Trace file	Record of activities in the CONTROL-T environment. Used primarily for recovery of the Media Database.
Stacking Statistics file	Statistical information on each dataset which is used by CONTROL-T to estimate the amount of space required for that dataset. Information about previous generations of the dataset is used to calculate space requirements for storage of each dataset.

Statistical information can be extracted from repository files via various utilities and KeyStroke Language (KSL) reports:

- Utilities are described in Section 9 of this guide.
- Information extraction via KSL reports and/or sample KSL reports are described in detail in the product-specific user manuals.

Miscellaneous

Dynamic Destination Table

The IOA Shout facility allows the user to specify messages to be sent to particular destinations upon the occurrence of specified events. However, it may be necessary to send a particular message to several destinations, and/or a particular physical destination may vary depending on time of day or other factor (e.g., the TSO logon ID of the shift manager may be different for each shift).

These situations are handled by defining groups of destinations in a Dynamic Destination table. The Dynamic Destination table consists of a site-defined list of logical destination names. For each logical (i.e., group) destination name, a list of physical destinations is defined.

A logical destination name can be specified for a Shout message. The message, when shouted, is sent to each physical destination defined to the logical location. If the physical location is logged on, it will receive the message.

Multiple Dynamic Destination tables can be defined, one of which is defined as the default. This Dynamic Destination Table is loaded when the product-specific monitor (e.g., the CONTROL-M monitor) is started. A different Dynamic Destination table can be loaded via operator command. (Shout messages are directed by the product-specific monitor to the Dynamic Destination table in memory.)

Section 1: IOA Concepts and Components

When a change is made to the Dynamic Destination Table in memory, and it is desired that the change be implemented immediately, the table should be reloaded into each product's memory.

(For information regarding the loading and reloading of Dynamic Destination tables in memory, see "IOA Dynamic Destination Table" in Section 2 of this guide.)

AutoEdit Facility

The AutoEdit facility allows symbolic representation (via AutoEdit variables) of dynamic values (e.g., values which change from execution to execution, or from day to day). Each variable is resolved at the appropriate time, and the resolved value is substituted for the variable. Special AutoEdit functions and control statements enable manipulation of values specified for these variables. The variables can be used in job/rule/mission definitions in various ways allowing flexibility and high level logic in automation of your work environment.

In addition, AutoEdit variables and functions can be used in different ways for each IOA product. The AutoEdit facility can be especially useful in the following:

- JCL setup of jobs submitted under CONTROL-M.
- Messages issued with the IOA Shout facility.
- Report scripts written in the KeyStroke Language (KSL).
- KeyStroke OpenAccess scripts used by CONTROL-O to interact with VTAM applications.

See relevant user manuals for how the AutoEdit facility can be used with the products at your site.

Security Implementation

IOA products can be protected just like any other data center application. IOA has built-in interfaces to widely used security environments (e.g., RACF, CA-ACF2 and CA-TOP SECRET). Each IOA function is associated with a security module. These security modules are used to check user authorization for requested actions, and to permit or deny the actions accordingly. In addition, user exits are invoked before the security modules to allow the user to perform required user functions that are not related to security.

IOA security is described in detail in the IOA Security Guide.

User Exits

IOA user exits can be used to influence the continued operation of an IOA product at specific points in processing.

For example, an exit can be used to check user authorization for a specific action or to modify report banners and indexing before printing.

Exits are described in Section 10 of this guide. Detailed information about each exit is provided in the IOA SECUDATA library.

Utilities

Utilities are specialized programs designed to perform specific tasks in the IOA product environment. Some utilities are provided with all IOA products, and some are product specific.

A comprehensive list of all IOA product utilities is provided in the beginning of Section 9. Unless otherwise indicated, these utilities are described in detail in the remainder of Section 9.

Utilities of special interest to product users are described in the user manual for the relevant product.

Simulation

Prior to running a complex task for the first time, or during a conversion from another product, it is often useful to be able to first perform the task (or conversion) in simulation mode. For this reason, certain IOA products and utilities can be run in simulation (TEST) mode.

In simulation mode, the IOA product has no effect on your system. However, certain products (or utilities) can generate files containing information which is useful for checking performance.

Cross-Product Communication

Certain IOA products can perform actions in a response to information received from another IOA product.

A description of some ways in which IOA products can interact follows:

File Access

The IOA Core files are accessed by more than one product. A change can be made in one of these files by one product, and the file (and the change) can then be read by another IOA product.

Example

A prerequisite condition set by CONTROL-O is detected by CONTROL-M during its next scan of the IOA Conditions/Resources file. As a result of the detected condition, CONTROL-M can then run a job whose execution is dependent on this condition (i.e., the condition is specified in the job's IN parameter).

See the description of the IOA Core earlier in this section for more information on files shared by IOA products.

Automated Processing Definition Statements

Some statements in automated processing definitions (e.g., job scheduling definitions) can be used by one IOA product to directly influence the operation of another IOA product.

Section 1: IOA Concepts and Components

Examples

The DO FORCEJOB statement can be used by other IOA products to instruct CONTROL-M to force (unconditionally schedule) a specified job.

Parameter CTB STEP specified in a job scheduling definition can be used to trigger a CONTROL-B rule.

Special Cross-Product Interfaces

Some IOA products have special interfaces to other IOA products installed at the same site. Special interfaces that exist for specific combinations of IOA products are discussed in the relevant product-specific sections of this guide.

Example

CONTROL-R has a special Driver exit which it uses to interface with CONTROL-T.

Section 2: Customizing and Administering IOA

Introduction	2-1
The IOA Online Environment.....	2-1
Entering the IOA Online Facility	2-2
IOA Online Monitor	2-3
Principles of Operation.....	2-3
Activating the IOA Online Monitor (IOAOMON)	2-5
Displaying a List of All Active Users	2-5
Deactivating the IOA Online Monitor.....	2-6
Problem Determination.....	2-6
The VTAM Monitor (IOAVMON)	2-7
Activating the VTAM Monitor (IOAVMON).....	2-7
Deactivating the VTAM Monitor.....	2-7
Displaying a List of All Active Users	2-7
Customizing the IOA Online Environment.....	2-8
Transaction Members.....	2-8
Program List Members.....	2-9
Allocation Members.....	2-10
Modifying IOA Online Facility Commands.....	2-12
Modifying IOA Online Facility PFKey Definitions.....	2-13
The IOA Primary Option Menu	2-13
Line Format Display.....	2-14
Box Format Display.....	2-15
Customizing the Menu	2-15
Customizing IOA Screens	2-17
Modifying IOA Screens and Constants	2-17
Constant Blocks.....	2-18
Recommended Steps for Screen Modification	2-19
Customizing IOA Display Format Members.....	2-19
@STYLE.....	2-20
@HEADER.....	2-21
@LINE.....	2-22
@FIELD	2-22
@VAL	2-23
@END.....	2-24
@DLM	2-24
Color Parameters.....	2-24
Extended Color Support.....	2-25

ISPF Characteristics	2-25
IMS/DC and IDMS/DC	2-25
COM-LETE	2-26
IRMA PC Terminal Emulator Users	2-26
Customizing Extended Color Support	2-26
IOA Access Method	2-27
File Structure	2-27
IOA Access Method Naming Conventions	2-28
File Utilities	2-29
File Definition Statements	2-30
Dual Mirror Image File Support	2-30
Dual Mirror Image File Parameters	2-31
Recovering From a Damaged IOA Access Method File	2-31
IOA Profiles	2-31
Profile Members	2-32
Profile Contents	2-33
Considerations for Profile Symbol Lines	2-33
SHOW Window Filters	2-34
Profile Attribute in the Screen Definition	2-34
Saving a Profile	2-35
Profile Variables	2-35
Window Display Variables	2-36
Color Variables	2-39
Work Mode Variables	2-40
Presentation Mode Variables	2-43
Miscellaneous Variables	2-45
Other Types of Profile Variables	2-46
Modifying IOA Product Defaults	2-46
Modifying IOA Messages	2-47
Recommended Steps for Message Modification	2-48
IOA Dynamic Destination Table	2-49
Replacing the Current Dynamic Destination Table	2-49
Expanding the Resource File (RES)	2-49
Expanding the IOA LOG File (LOG)	2-50

Introduction

This section describes the customization facilities available within IOA. The following topics are described:

- The IOA Online Environment
- Customizing the Online Environment
- The IOA Access Method
- IOA Profiles
- Modifying IOA Product Defaults
- Modifying IOA Messages
- The IOA Dynamic Destination Table
- Expanding the IOA Resource File (RES)
- Expanding the IOA Log File (LOG)

A detailed description of each topic follows.

The IOA Online Environment

The Online facility can be activated under the following environments:

TSO (native)	VTAM
TSO/ISPF	IMS/DC
ROSCOE/ETSO	IDMS/DC
CICS	CA-COM-LETE

Options for Cross-Memory interfaces to the Online monitor address space are available under native TSO and ROSCOE/ETSO.

There are slight differences in the operation of the Online facility under different environments.

The following paragraphs describe how the IOA Online monitor facility provides support for various environments using CICS as a model scenario. The description also applies to the following environments:

IMS/DC

VTAM

COM-LETE

IDMS/DC

ROSCOE Cross-Memory Interface

TSO Cross-Memory Interface

Entering the IOA Online Facility

In TSO and ISPF, the IOA Online facility is entered by running a CLIST (located in the IOA CLIST library). The CLIST calls IOAONL, which is the main CLIST that activates the Online facility in the specified environment.

The following CLISTs are provided with IOA:

- IOATSO For activating IOA Online in a TSO environment.
- IOAISPF For activating IOA Online in an ISPF environment.
- IOAXTSO For activating IOA Online under TSO through Cross-Memory services.

The format for the CLIST used to activate the IOA Online environment under ISPF or TSO is as follows:

```
IOAONL APPLTYPE(atype) [TRANID(transmem)]
```

Where:

- atype Type of environment under which IOA is run. Valid values are:
 - I ISPF.
 - S TSO.
 - X Online monitor.
- transmem Name of the relevant transaction member in the IOA PARM library. The name of this member must be no more than four characters long. Optional.
 - If no transaction member is specified a full list of all options for the installed products is displayed.
 - For more information see "Transaction Members" later in this section.

Example

At a site where CONTROL-M and CONTROL-D are installed, the following CLIST enables all options for CONTROL-M and CONTROL-D under TSO:

```
%IOAONL APPLTYPE(S)
```

Note *Dataset allocation for the IOA Online environment can be customized via specification of overriding or alternative allocation members in the CLIST used to activate the IOA Online environment. For more information, see "Allocation Members" later in this section.*

It is possible to pass a transaction ID directly as a positional parameter to either the IOATSO or IOAISPF CLISTs. For example:

```
IOATSO DMAN
```

will activate the IOA environment under TSO, setting up only the IOA and CONTROL-D options regardless of other products installed at the site (where transaction member DMAN contains the appropriate statements).

Example

The following CLIST and transaction member combination enables only option U of CONTROL-D under ISPF (where transaction member DOLV contains the appropriate statements).

In the CLIST library:

```
%IOAONL APPLTYPE(I) TRANID(DOLV)
```

See the explanation of transaction members later in this section for more information.

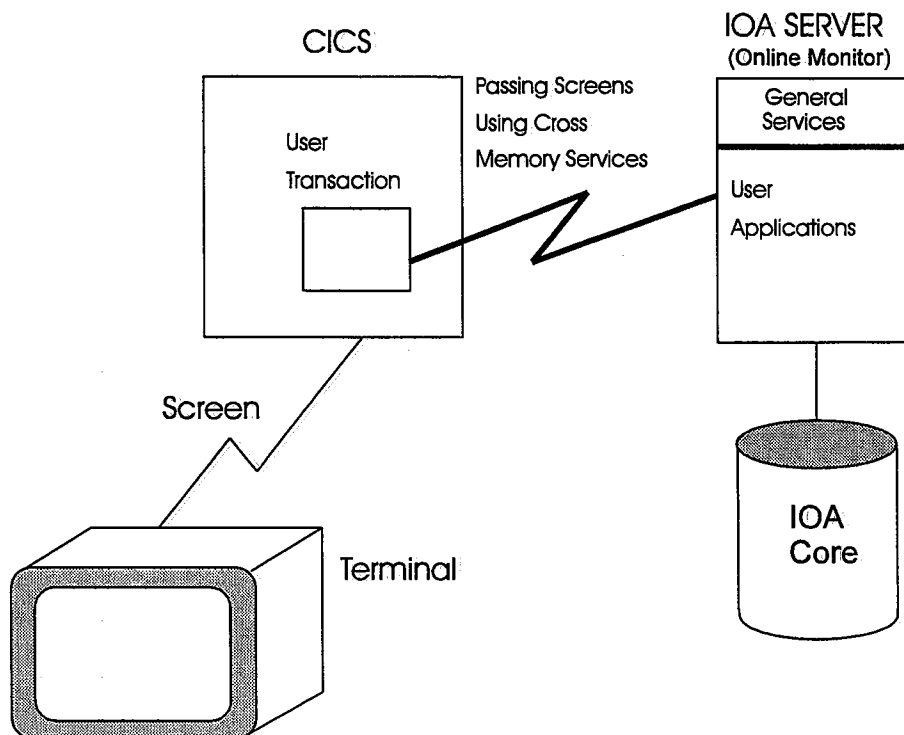
IOA Online Monitor

The IOA Online monitor interacts with different environments (e.g., CICS) to provide an online interface for the different IOA applications. The online monitor generates the IOA product screens which are available to users signed on to the online monitor, and handles all IOA online functions.

Each IOA Online monitor usually operates 24 hours a day as a started task. The IOA Online monitor is usually activated automatically as part of the IPL process.

Principles of Operation

To enable the same user interface to be activated by users under different environments, a virtual environment is built for each user, which handles all actions a user requests to perform in the IOA environment. Thus, any environment being used at a site can be supported by adding a customized routine to convert the virtual environment to the site's environment.



Section 2: Customizing and Administering IOA

A special started task, named the IOA Online monitor, provides an environment for executing IOA applications. One or more online monitors can be started, and each monitor can support several virtual environments (signed-on users).

Each monitor can customize IOA product screens, constants, messages, colors, commands and PFKey definitions to adapt them to the site's requirements. It is recommended that the same prefix be used for each group of monitors which are customized for specific functions. See the explanation on parameter IOAXPARM in the IOA Installation Guide for more information.

In addition to site-wide Global profile customization, each IOA product can be customized to respond differently to individual users signed on to the online monitor through variables specified in User profiles.

In the CICS environment, a small conversational transaction communicates with the IOA Online monitor using cross-memory services. The CICS transaction receives a screen from the user terminal, and passes the screen to the IOA Online monitor for processing. The IOA Online monitor returns a screen (to be displayed) to the CICS transaction which in turn sends this screen to the CICS user at the terminal. Using this method, the IOA application is performed outside CICS, and the normal CICS function is uninterrupted. The memory requirement of each user in the CICS region is approximately the size of the screen (i.e., $2 \times 24 \times 80 = 3840$ bytes).

The IOA Online monitor activates each user as a subtask. Therefore, in the event of an error or abend of a user application, other users working under the Online monitor are not affected.

IOA applications work above and below the 16MB line. Therefore, region size limits may restrict the number of users who will be able to work concurrently under the same Online monitor. To solve this potential problem, it is possible to open any number of additional IOA Online monitors. When the user enters the IOA CICS transaction code, the IOA CICS application searches for a free space in one of the IOA Online monitors which is currently active. The process of choosing the correct monitor is transparent to the CICS user. Using this method, there is no limitation on the number of users signing on to IOA.

It is possible to balance the workload of two or more IOA Online monitors. Refer to parameter BALANCE in the "Install IOA Online Monitor" step of the IOA installation procedure.

The actual number of users under one monitor varies according to their memory requirements (influenced by how many screens they use concurrently, what type of IOA options they are using, etc.). Many options exist in the IOA Primary Option menu. However, there is no reason for every user to access all options of each product. It may be desirable to limit access to certain options to only a selected group of users. This is done by passing a transaction code to the Online monitor. See "IOA Online Monitor" later in this section for more information. It is also possible to determine that certain monitors allow users to sign on only with specific transactions thus enabling monitors to be specialized for different groups and numbers of users.

Note *The CONTROL-D Online Viewing facility runs mostly above the 16MB line. Therefore, it is possible to activate a maximum of approximately 70 Online Viewing users under one monitor. However, these users should be limited to Online Viewing transactions only. This can be done by forcing users to sign on with the transaction ID DOLV.*

A data center can run more than one CICS in the same computer. From the IOA point of view, this is transparent to the user. The IOA Online monitor always returns the screen to the CICS which issued the request. It is possible to define additional transaction IDs for local use (e.g., a transaction ID offering all options of the menu except for the Conditions/Resources screen). See “Customizing the IOA Online Environment” in this section for more information.

Activating the IOA Online Monitor (IOAOMON)

Each IOA Online monitor usually operates 24 hours a day as a started task. Usually the IOA Online monitor is automatically activated as part of the IPL process by the operator command. For example:

```
S monitor_name
```

The names of the online monitors available in a site are defined in member IOAXPARM in the IOA PARM library (described later in this section).

If the IOA Online monitor is successfully activated, the following message is displayed on the operator console:

```
CTM777I monitor-name ONLINE MONITOR INITIALIZATION STARTED
```

After a few seconds the following message is displayed:

```
CTM778I monitor-name ONLINE MONITOR INITIALIZATION COMPLETED
```

If you try to activate more than one monitor with the same monitor ID in the same computer, the newly-activated monitor immediately shuts down and a relevant message is issued.

Note *The IOA Online monitor uses cross-memory services to communicate with other address spaces requesting services. Like other address spaces using cross-memory services, whenever the IOA Online monitor is shut down, the address space entry in the MVS Address Space Vector Table (ASVT) remains non-reusable until the next IPL, and the message IEF352I is issued (in some MVS releases). If the IOA Online monitor is brought up and down many times, the ASVT may become full. New address spaces will not start, and an immediate IPL may be required.*

To prevent this problem, specify a large enough value in MVS initialization parameters MAXUSER, RSVSTRT and RSVNONR in member IEASYSXX in the SYS1.PARMLIB library.

Information about these parameters can be found in the “MVS Initialization and Tuning Reference” manual.

Displaying a List of All Active Users

Specify the following operator command:

```
F monitor_name, DISPLAY
```

The following list appears on the operator console from which the modify command was issued:

```
CTM645I MONITOR USER PGM TRANID TERMINAL START LASTUSED ST
CTM646I monitor user pgm tranid terminal start last      status
```

Section 2: Customizing and Administering IOA

Message CTM646I contains the following data for each user in the IOA Online monitor:

MONITOR	Name of the IOA Online monitor
USER	User ID
PGM	Name of the program which is active
TRANID	Transaction ID. (See the IOA Installation Guide for an explanation.)
TERMINAL	Terminal ID under the IOA Online monitor
START	Start time (hour and minute the user entered the Online facility)
LASTUSED	The last time the user performed an action with the Online facility (hour.minute.second)
ST	Status:
	W Waiting for input (from the terminal)
	A Active - working

Deactivating the IOA Online Monitor

To deactivate the IOA Online monitor, use operator command:

P monitor_name

Problem Determination

A problem may be encountered when the Online monitor is active but users cannot access the monitor, and error message CTM011E is displayed by program CTMCINT. The reason may be that IOA has either determined that no Online monitor is active, or that IOA cannot access the monitor which is active due to one of the following reasons:

- The name of the Online monitor is not found in the list of active Online monitors. This may occur if:
 - Exit 9 is not passing a correct name. Exit 9 can determine the name of the Online monitor to be accessed. It is possible that the exit was not customized as required, and the wrong version of exit 9 is being used, or that faulty versions of member IOAPARM and/or IOAXPARM exist in the IOA LOAD library.
 - There is no active Online monitor. There may be other Online monitors active, but they are not the correct ones.
 - Whenever the user attempts to connect under TSO Cross-Memory, both the Online monitor, and the TSO user must be running on the same CPU. Otherwise TSO Cross-Memory services cannot recognize that there is an Active Online monitor running on a different CPU.
- The subsystem does not match the subsystem name specified in member IOAXPARM. This may occur if a different subsystem is active, or a wrong version of member IOAXPARM is being used.

- A problem is identified in the subsystem chains. The address of the Online monitor chain cannot be found. This may occur if:
 - Modules from different IOA releases were activated together (e.g., a wrong or bad library is concatenated, or the library itself contains faulty programs).
 - When signing on from TSO Cross-Memory the Online monitor is not running on the same CPU as the one from which an attempt was made to sign on.

The VTAM Monitor (IOAVMON)

The VTAM monitor (IOAVMON) enables access to the IOA Online monitor for VTAM users without passing through any TP monitors (e.g., CICS, IMS/DC).

Activating the VTAM Monitor (IOAVMON)

The VTAM monitor usually operates 24 hours a day as a started task. The monitor is usually automatically activated as part of the IPL process by operator command:

```
S IOAVMON
```

The VTAM application should be active before starting the VTAM monitor. To activate the VTAM application manually, use VTAM command:

```
V NET,ACT,ID=ioaappl
```

Where `ioaappl` is the VTAM major node containing the IOAVMON acbname.

If the VTAM monitor is successfully activated, the following message appears on the operator console:

```
CTM7A0I monitor name VTAM MONITOR INITIALIZATION STARTED
```

After a few seconds, the following messages are displayed:

```
CTM7A1I monitor name VTAM MONITOR INITIALIZATION COMPLETED
```

```
CTM7ACI monitor name ACCEPTING LOGONS TO acbname
```

Only one VTAM monitor can be activated at a time.

Deactivating the VTAM Monitor

To deactivate the VTAM monitor, use operator command:

```
P IOAVMON
```

Displaying a List of All Active Users

To display the list of users logged on to IOAVMON, enter operator command:

```
F IOAVMON,D
```

The following list appears on the console from which the modify command was issued:

```
CTM7B0I monitor - TERMINAL TRANSID
```

```
CTM7B1I monitor - LUsername transid
```

Customizing the IOA Online Environment

All IOA products are accessed via the IOA Online environment. This environment can be customized to meet your site's needs. Using the methods described below, you can, for example, modify menus and allocation tables, and control users' access to the various features of the IOA products installed at your site.

Transaction Members

A transaction member is a member in which each line contains customization information for one of the IOA products installed at the site. Transaction members are line-oriented.

Transaction members can be used to determine:

- Which products are available to the user.
- Which options of each product are available to the user.
- What program member is to be used by the transaction.

The valid format for a line in a transaction member is:

```
product [optiona, optionb, ...optionx] [PGM=pgmlist]
```

Where:

product IOA product described in the line. Valid values are: IOA, CTM, CTD, CTT, CTO, and CTB.

Products do not need to be listed in any specific order. Only products listed in the transaction member are made available to the user.

optiona, optionb, x Option codes of options to be displayed, or ALL. The option codes are the one or two character codes that are entered in the command line to indicate choice of a specific option. If ALL is specified, all options of the product described in the line are displayed.

pgmlist Name of the Program List member to be used for activation of the IOA product described in this line. See "Program List Members" later in this section for more information.

Notes *Transaction members corresponding to transactions (application IDs) used in releases prior to IOA Release 5.0.0 are included in the IOA PARM library for your convenience.*

Certain transaction names cannot be used for user-defined transaction members. Reserved transaction names are: IOAX, DOLV, KOA, VK, IALL, and ECS.

Example

Member DOLV in the PARM library specifies option U of CONTROL-D:

```
CTD U
```

Member DMAN in the PARM library specifies all options of IOA and CONTROL-D:

```
IOA ALL
```

```
CTD ALL
```

A transaction member that will contain only option 3 of CONTROL-M and options 4 and 5 of IOA will look like this:

```
CTM 3
```

```
IOA 4,5
```

If a transaction member specifies only one option, once activated, the option will be accessed directly without displaying a menu panel. If more than one option is specified, the following options will automatically be added to the menu: 0,1 and X. Therefore, the minimum number of options that will appear in a menu member is 5. Options 0,1 and X do not need to be specified in transaction members and should not be specified in PGM members (described later in this section).

Program List Members

A Program list specifies a member containing a list of programs to be activated for the options of an IOA product. A Program List member exists for each of the IOA products (including IOA).

The naming convention for Program List members is PGMxxx, where xxx is the product suffix (e.g., PGMCTM is the Program List member for CONTROL-M).

Each product has its own program list member, which contains all the options of the product. Unless specified otherwise in a transaction member, the product's program list member will be used to specify which options of the product are to be available for users. The PGM list member will be used only if the product is installed as specified in member IOAPARM in the IOA PARM library.

Program List members are line-oriented members where each line describes a program in the relevant product.

The valid format for a line in a Program List member is as follows:

```
program DELETE envir [=]optioncode
```

Where:

program Program described in the line. If the program has been modified, the name of the new (modified) program should be specified here.

envir One-character designation of the environment(s) in which the program option is to be made available. Valid values for envir are:

I	ISPF
R	KSL
S	TSO

Section 2: Customizing and Administering IOA

X	Online monitor
*	All environments
blank	No environments

= If specified, the equal sign indicates that this option will be accessible from all screens in the IOA environment using the transfer command (e.g., =5 transfers the user to the IOA Log screen). If no equal sign is specified, the option is only available from the IOA Primary Option menu.

optioncode A one- or two-character option code used to activate the specified program. To specify a different option code for a program, specify the new option code here. More than one option code can be specified for each program. No two programs may have the same option code.

Note Changing the option code listed in the Program List member affects only the operation of the program. To ensure that the display shows the new option code, it is necessary to update member MENU in the MSG library (described later in this section).

Option codes for options 0,1, and X are reserved and cannot be changed.

The following default Program List members are located in the IOA PARM library:

PGMCTB	PGMCTD
PGMCTM	PGMCTT
PGMCTO	PGMCTR
PGMCTV	PGMIOA

Sample Program List member for CONTROL-O (member PGMCTO):

```
CTOTOMP DELETE * =OR
CTOTMSC DELETE * =OM
CTOTARF DELETE * =OS
CTOTAOP DELETE * =OA
CTOTALO DELETE * =OL
```

An alternate Program List member can be used to specify a modified version of a specific program (or programs) to be activated for a specific option(s).

Alternate program lists must contain programs for all available options of the relevant product available to the user.

Allocation Members

Allocation of files and datasets for each IOA product are specified in Allocation members.

The naming convention for Allocation members is ALCxxx, where xxx is the name of the product. Each product has its own Allocation List member which contains all the allocations that need to be made to access the product using the online facility. The datasets which are specified in a product allocation member list will only be allocated if the product is installed as specified in member IOAPARM in the IOA PARM library.

The following allocation members are located in the IOA PARM library:

ALCIOA	ALCCTM
ALCCTD	ALCCTB
ALCCTT	ALCCTO
ALCCTR	ALCCTV

The valid format of an Allocation member is as follows:

product/s DD=ddname,DSN=dsname,M=member_name,DISP=disposition

Where:

product	One letter product name specified per product in columns 1-9. If a blank is specified, the format applies to all products.
ddname	Should appear between columns 10-80. Mandatory.
dsname	Should appear between columns 10-80. Mandatory.
member_name	Should appear between columns 10-80. Optional.

Note If you do not use member name, you should not specify M= in an Allocation member.

disposition	Should appear between columns 10-80. The default is always SHR, unless specified otherwise.
-------------	---

Note If you do not use disposition, you should not specify DISP= in an Allocation member.

* In column 1, represents a comment line.

Example

CTMHELP should be allocated if any CONTROL product is installed:

DD=CTMHELP,DSN=IOA.V500.MSG

To customize allocation at your site, specify one of the following DD statements in the CLIST (or the Online monitor procedure or the KSL) used to activate the IOA Online facility:

DAOVRALC Specifies a member containing allocations that override corresponding allocations in the Allocation member list for the relevant product.

The datasets specified in the appropriate products in the allocation member are all allocated. However, any allocations for datasets specified in the overriding member are also allocated and they will override the corresponding allocations.

CLIST with an overriding Allocation member:

```

PROC 0  APPLTYPE(S)  -
          TRANID(TRANSACTION_ID_MEMBER)
ALLOC DD(DAOVRALC)  -
          DSN('IOA.V500.PARM(ALCOVR)') SHR REU
%IOAONL APPLTYPE(&APPLTYPE) TRANID(&TRANID)
FREE DD(DAOVRALC)
END:END
    
```

Section 2: Customizing and Administering IOA

DAALTALC Specifies a member containing allocations to replace the allocations in the Allocation members for all products at the site.

The alternate list must contain all necessary allocations to the environment as none of the default Allocation members will be used.

CLIST with an alternate Allocation member:

```
PROC 0 APPLTYPE(S) -
      TRANID(TRANSACTION_ID_MEMBER)
ALLOC DD(DAALTALC) -
      DSN('IOA.V500.PARM(ALCALT)') SHR REU
%IOAONL APPLTYPE(&APPLTYPE) TRANID(&TRANID)
FREE DD(DAALTALC)
END:END
```

If neither an overriding Allocation member nor an alternate Allocation member is specified, the default Allocation members are used. Overriding and alternate Allocation members can be stored in the IOA PARM library.

Only one overriding or alternate Allocation member can be specified in a procedure (e.g., a CLIST). However, multiple overriding and alternate Allocation members can be created at a site for use with different CLISTs and Online monitor procedures.

Sample ALC Member (ALCCTO)

```
DD=DAPASCTO,DSN=%ILPREFA%.PARM,M=PASCTO
DD=DALOAD,DSN=%STEPLIB%%LANG%
```

Note If a concatenated allocation is required, delete the original allocation from the appropriate ALCxxx member and add it to the user's procedure, CLIST, JCL, etc.

Modifying IOA Online Facility Commands

Every screen of the IOA Online facility supports a set of commands. It is possible to change the names of these commands or to create synonyms. The commands reside in the IOA PARM library in members with the following naming conventions:

TxxxCMD1 Active commands member.

Where:

xxx is the screen identifier. Appendix E contains a detailed list of all options, screens and their corresponding command members.

A command member is composed of one header line and any number of command lines. The number at the left of the header line is the total number of command lines in the member. It must be updated when lines are added to or deleted from the command member.

The structure of the command line is:

Columns	Description
1- 8	Command name.
25-28	Reserved hexadecimal (unprintable) value. Do not modify.
29-68	Description of the command.
69-72	Reserved hexadecimal (unprintable) value representing the internal command number. Do not modify.

It is possible to change the command name and/or the description by typing in the change. To add a synonym (a command that is to have the same effect as an existing command), duplicate the command line, modify the command name on the duplicated line, and add one to the command line counter at the left side of the header line. To delete a command, delete the command line in the member and decrease the counter by one.

Modifying IOA Online Facility PFKey Definitions

Every screen of the IOA Online facility is associated with a set of PFKeys with pre-assigned commands. It is possible to change these PFKey definitions. The PFKey assignments reside in the IOA PARM library in members with the following naming convention:

TxxxPF1 Active PFKey definition member

Where xxx is the screen identifier. Appendix E contains a detailed list of all screens and their corresponding PFKey members.

A PFKey member is composed of one header line and any number of PFKey definition lines.

The structure of the PFKey definition line is:

Columns	Description
1- 8	The PFKey or <ENTER>.
9-22	The command assigned to the PFKey.

At least one PFKey/Enter key must be defined as the <ENTER> command.

The IOA Primary Option Menu

The IOA Primary Option menu displays IOA product options and facilities available to the user. Customizing the menu is usually not required since the menu is dynamically generated according to the options available to the user.

The IOA Primary Option menu is displayed in one of the following formats:

Section 2: Customizing and Administering IOA

- When less than a threshold number of options is available, options are displayed in Line format. Options are listed one per line on the screen. For each option an option code, the name of the option, and a brief description of the option is listed. Information on the IOA products installed at the site is displayed in the upper right corner of the screen.
- When more than the threshold number of options is available, options are grouped according to product in Box format. In Box format only option codes and option names are displayed. This format enables displaying all IOA products options on one screen. Information on IOA products installed at the site is available via option 0.

Reducing or increasing the number of available options may change the format in which the Primary Option menu is displayed.

Line Format Display

```
----- IOA PRIMARY OPTION MENU -----
(1)
OPTION ===>                                USER      M21
                                             DATE      05.05.96
                                             M Version 5.0.0-9604
                                             R Version 5.0.0-9604

0  IOA INFO:                               IOA Version Information
1  MAIN MENU:                              IOA Primary Option Menu
2  JOB SCHEDULE DEF:                       CTM Job Scheduling Definition
3  JOB STATUS:                             CTM Job Status Display
4  COND/RES:                               IOA Conditions/Resources Display
5  LOG:                                     IOA Log Display
6  UTILITIES:                              IOA On-Line Utilities
7  MANUAL COND:                            IOA Manual Conditions Display
8  CALENDAR DEF:                           IOA Calendar Definition
C  CMEM DEFINITION:                        CTM Event Manager Rule Definition
X  EXIT:                                   Terminate Session

PLEASE SELECT OPTION AND PRESS ENTER OR PRESS THE OPTION'S PFK      12.20.43
```

Box Format Display

IOA PRIMARY OPTION MENU		
(1)		
OPTION ==>		USER M21
IOA	CONTROL-D/V	CONTROL-O
0 IOA INFO	A MISSION STATUS	OR RULE DEFINITION
1 MAIN MENU	M MISSION DEF	OM MSG STATISTICS
4 COND/RES	R REPORT DEF	OS RULE STATUS
5 LOG	T RECIPIENT TREE	OL AUTOMATION LOG
6 TSO	U USER REPORTS	OA AUTOMATION OPTS
7 MANUAL COND	F PC PACKET STATUS	
8 CALENDAR DEF		
X EXIT		
CONTROL-M/R	CONTROL-B	CONTROL-T
2 JOB SCHEDULE DEF	BB BALANCING STATUS	TR RULE DEFINITION
3 JOB STATUS	BM MISSION DEF	TP POOL DEFINITION
C CMEM DEFINITION	BV DB VARIABLE DEF	TV VAULT DEFINITION
	BR RULE DEFINITION	TI INQ/UPD MEDIA DB
	BA RULE ACTIVITY	TC CHECK IN EXT VOL
PLEASE SELECT OPTION AND PRESS ENTER OR PRESS THE OPTION'S PFK		12.20.43

The menu is displayed according to the order of appearance of each product in the menu table (described later in this section). The IOA menu is always displayed on the left side of the screen (displayed only in Box format). However, changes to other menus affect the arrangement of the options of other products in the menu.

Customizing the Menu

Options for the Primary Option menu are listed in member MENU in the IOA MSG library. This member can be used to modify the information that appears in the menu display. The lines in this member look like this:

```

*      Comment
T*     IOA
TMR    CONTROL-M/R
TM     CONTROL-M
TD     CONTROL-D
TDV    CONTROL-D/V
TO     CONTROL-O
TB     CONTROL-B
TT     CONTROL-T
L*     * 0IOA INFO           IOA Version Information
L*     * 1MAIN MENU         This Menu
LM     * 2SCHEDULE          Setup Schedule
LM     * 3STATUS            Job Status Display
LI     * 4RES                Conditions/Resources file
LI     * 5LOG                IOA Audit Trail
LI     S 6TSO                Enter TSO Command

```

Section 2: Customizing and Administering IOA

Where:

T Titles which appear at the top of each box in the Box Display Type for the IOA Primary Option menu. Following the letter T is the name of the product whose options are to be contained in the box with this title.

Example

The line shown below indicates the title to be displayed at the top of the box containing CONTROL-M options.

```
TM CONTROL-M
```

L Lines beginning with the letter L indicate menu options. The format for an option line in the menu member is as follows:

```
Lproduct environment optcode optname opt-description
```

Where:

product IOA product that must be installed for this option to be displayed in the menu. For example, M indicates that the option is displayed when CONTROL-M is installed at the site. I is for IOA.

environment Environment that IOA must be operating under for this option to be displayed. Valid environment specifications are:

T TSO

I ISPF

0 no environment

* all environments

optcode Option code used to activate (choose) the option. The option code can be one or two characters in length and must be the same as the option code indicated for this option in the Program List member in use at your site.

optname Option name. Displayed adjacent to the option code in Box format of the menu, and preceding the option description in the Line format.

opt-description Option description. Displayed only in Line format. This can be modified to aid option identification at your site (e.g., it can be translated).

For more information, see the description that appears at the top of member MENU in the IOA MSG library.

Customizing IOA Screens

Customization of IOA screens can be accomplished in a number of different ways. In this section the following topics are described:

- Modifying IOA Screens and Constants
- Customizing IOA Display Format Members
- Customizing Color in IOA
- Customizing of the IOA Primary Option Menu

Note To avoid incompatibilities with IOA scripts, it is recommended to back up members modified during customization of IOA screens, and to use these backups for KSL scripts.

Modifying IOA Screens and Constants

IOA screens and constants can be adapted by the user for local language, local site terminology, ease of use, etc.

IOA screen and constant members are stored in the IOA MSG library. The naming convention for these members is:

CTxSyyy

Where

x A single character indicating the product (e.g., M for a CONTROL-M screen).

yyy A 3-character screen identifier.

Each member contains definitions for either a screen and its constants, or a set of constants. Definitions are specified as Assembler macro instructions. Each member must be assembled and link-edited to create a CSECT and module with a name identical to the screen member name. The load module containing the screen replaces an existing CSECT in one or more load modules in the IOA LOAD library.

A screen is defined by the following macros:

CTMSSTA	Start screen/constants definition (one CTMSSTA macro for each screen definition).
CTMSBLK	Start a block of fields/constants.
CTMSFLD	Define a field in the screen/constant.
CTMSEND	End screen/constant definition (one CTMSEND macro for each screen definition).

Each unit of data in an IOA screen is called a block. A block may contain one or more fields.

Section 2: Customizing and Administering IOA

Example

- The header block is composed of two or more lines which contain the header line, the command field, the scroll field, etc.
- The following line which appears at the top of the CONTROL-M Job List screen is a block.

```
'====>>>>> TOP OF ACTIVE JOBS LIST <<<<<====='
```

Within each block, the location of the fields can be changed. It is possible to change the length of fields which are defined as skip protected and contain constants. Do not change the length of unprotected data fields unless specifically mentioned otherwise in the member. The length is the length of the field, not including the attribute byte. The total accumulated length of the fields within the block which describes screen line(s) should remain the same after modifications have been made. This restriction does not apply to blocks used for defining constants.

Each block is line-oriented so that the total length of each block must be the number of lines in the block multiplied by the length of the line.

Do not change the location and contents of macros CTMSSTA, CTMSBLK, and CTMSEND. Changes can be applied only in the contents (parameters) of macro CTMSFLD.

Macro CTMSFLD defines a field in the screen. The format is:

CTMSFLD

NAME = field name	If a name has be specified, do not change it.
LTH = m,	Length without attribute byte. If not specified, the length of the DATA text will be used. The real length of the field is always one more than that specified in the LTH field to account for the attribute byte.
ATTR = code,	Field attributes. Refer to macro CTMSATT in the IOA MAC library.
DATA = 'text'	The data in the field (if any) in quotes.
PROF = Y	The value of the field upon exiting the online environment will be saved in the user's profile member and will be used at the next invocation of the environment. Default: N. Valid values are Y or N.

Note *Not all fields may use profiled values.*

Constant Blocks

Blocks used to define constants in SYSIN input DD statements, as part of a message, or as statuses on screens, are different from screen field blocks. You can change the length of the fields, disregarding the "total block length rule." An example is member CTMSAES, which is used to define parameter syntax of the CONTROL-M AutoEdit Simulation facility.

Recommended Steps for Screen Modification

Sources (screen members) of IOA products are modified via an SMP/E USERMOD. Use the following steps to perform this task:

1. Choose a name for a new USERMOD (to be built in the following steps). The name of the USERMOD must be 7 characters in length, must begin with an alphabetic character (i.e., not a number or symbol), and must be unique.
2. Create a new member in the IOA MAINTLIB library with the same name as the USERMOD created in Step 1.
3. Copy the contents of member UMODSRC in the IOA JCL library to the newly-created member. This member contains a skeleton sample for the USERMOD (to be customized in the following steps).
4. Specify the name of the new USERMOD in the ++USERMOD statement.
5. Determine the FMID and the RMID of the screen source via SMP/E Online option 3.2. Specify SRC as the entry-type, and the name of the screen source as the entry-name.

Specify the FMID in the ++VER statement.

If the RMID value is not the same as the FMID value, add the RMID value to the ++VER statement via the PRE(rmid-value) parameter.

6. Specify the screen source-name in the ++SRC statement.
7. Copy the contents of the source member to the line immediately after the ++SRC statement. Update the copy to meet your needs.
8. Specify the name of the USERMOD in job UMODRACK in the IOA JCL library. Run the job. This job RECEIVES and APPLY-CHECKS the USERMOD.

Job UMODRACK must end with a completion code of 0.

9. Specify the name of the USERMOD in job UMODAPP in the IOA JCL library. Run the job. This job applies the USERMOD.

Job UMODAPP must end with a completion code of 0.

Customizing IOA Display Format Members

In certain IOA product screens, it is possible to specify the display type (or format) of the screen. A wide selection of predefined display types is provided. For example, in the CONTROL-D User Report list, display type S is similar to IBM's SDSF, while display type J can be used for production control and programmers. Refer to the appropriate User Manual for details.

If the provided predefined types are not adequate, users may define their own display types using the Display Type Editing facility. The following pages describe parameters which define new display types or modify existing display types.

Section 2: Customizing and Administering IOA

All of the display formats are defined in members in the IOA MSG library. For a list of these members, refer to Appendix E which contains a list of all the options of IOA and the screen member and/or format members which are used to build the IOA panels. For example, in CONTROL-D, the display formats of option U are defined in members \$\$FRM and \$\$USR in the IOA MSG library. These members contain sets of predefined display formats. The user can add new formats in these members, or modify existing ones. When a display type is activated, it first checks for any errors, which are displayed in an appropriate screen.

Each line in a format member is of a certain type. The line type must be written in positions 1-10 of the line.

The permitted line types are:

@STYLE

@HEADER

@LINE

@FIELD

@VAL

@END

@DLM

A line starting with an asterisk (*) is a comment line.

The parameters on the line characterize the line's attributes. Parameters can be written in positions 11-72 of the line. A continuation line can be created by leaving the "line type" area blank, and by using positions 11-72 on the new line for parameters describing the line's attributes.

Each line type, and its parameters, are described below:

@STYLE

@STYLE is a line type used to start a new display format. The valid format for a **@STYLE** line is:

```
@STYLE      ID=id; TYPE=type; CLASS=class; [IDLIKE=idlike;  
              TYPELIKE=typelike; CLASSLIKE=classlike;] LTH=n;
```

Where:

ID Name of the display type. This value can be used to identify the DISPLAY TYPE field in an option's entry panel (if one exists) or when changing the Display Type by specifying the command DISPLAY x (where x is the display type) from the command line. The value can be any printable character. A blank is not valid. This field must be one character in length only. The character '1' is reserved for internal use and should not be used.

TYPE Logical type of the display type (second level qualifier). Valid values depend on the screen in which the display type is used. For example, display types of the CONTROL-D User Report List screen use this field for the following valid file types:

- P Permanent file
- A Active file
- H History file
- * All file types

CLASS Class of record for which this display type is used. Valid values depend on the screen in which the display type is used. For example, display types of the CONTROL-D User Report List screen have the following valid record classes:

- U User records
- S Sysdata records
- R Ruler records
- * All record classes

IDLIKE Name of the display type to which the current ID is identical. **TYPELIKE** and **CLASSLIKE** must also be specified. The definition of the display type specified via parameter **IDLIKE** must precede the current line. When this parameter is specified the current display format should have only a **@STYLE** line, and take its definition from the specified display format. This field must be one character in length only.

TYPELIKE Type of the display type to which the current ID is identical. Valid values are the same as in the **TYPE** parameter.

CLASSLIKE Class of record for which this display type is used. Valid values are the same as in the **CLASS** parameter (described above).

LTH Length of the line to be displayed. Valid values are 80 or 132.

Color parameters can also be modified. Refer to "Color Parameters" below.

@HEADER

@HEADER is a line type used to describe the header line of the current display format. The valid format for a **@HEADER** line is:

```
@HEADER DATA=data;
```

Where:

DATA Actual header data, to be displayed as the header line of the screen in which the display type is shown. The header data can be a maximum of 80 (or 132) characters in length. If less data is specified, it is padded with blanks to fill the physical line.

Section 2: Customizing and Administering IOA

Color parameters can also be modified. Refer to "Color Parameters" below.

@LINE

@LINE is a line type used to start a new line in the current display format. The valid format for a @LINE is:

```
@LINE      LONG={Y|N}; SHOWBLINE={Y|N};
```

Where:

LONG	Whether or not the line should be displayed in regular screen format (i.e., when Additional Information is not requested on the line).
	N (No) Always display the line. Default.
	Y (Yes) Displayed the line only when ADDITIONAL INFO is selected on the displayed line. (ADDITIONAL INFO is requested by entering A in the line's option field).
SHOWBLINE	Whether or not the line should be displayed when all fields in it are blank.
	Y (Yes) Show the line. Default.
	N (No) Do not show the line.

Color parameters can also be modified. Refer to "Color Parameters" below.

@FIELD

@FIELD is a line type used to start a new field in the current line in the current display format. The valid format for a @FIELD is:

```
@FIELD:    NAME=name; DATA=data; LTH=ln; PREF=pref; KANJI={Y|N};  
           OFFSET=ofst; EDIT={Y|N}; DEFAULTEDIT={Y|N}; DFLT=dflt;
```

Where:

NAME	Field to be displayed. Valid names are listed in the description of the display types of each product.
DATA	Constant data to be displayed. This is useful, for example, for displaying a field descriptor. Either NAME or DATA must be present, but not both.
LTH	Display length of the information of the field specified in the NAME parameter, or the display length of the constant information in the DATA parameter. If the actual information provided by NAME or DATA is too short, it is padded with blanks on the right. If the information is too long, it is truncated on the right.
PREF	Prefix to be added to the left of the displayed data specified by NAME or DATA.

KANJI	Specifies if Kanji (Japanese writing using Chinese characters) input is allowed.
	Y (Yes) The current field allows input in Kanji format.
	N (No) The current field does not allow input in Kanji format. Default.
OFFSET	Offset in the data field from which data should be taken for screen display purposes. This mechanism allows the user to split a long data field between several display fields (maybe on several different lines). n is the offset from which to start.
EDIT	Whether or not the displayed field should be updatable. Valid only if NAME is specified.
	N (No) The field is displayed in a protected (cannot be modified) form. Default.
	Y (Yes) The user can change the displayed field.
DEFAULTEDIT	Whether or not the user should be able to edit an insert record. Valid only if NAME is specified.
	N (No) The field is displayed in a protected (cannot be modified) form. Default.
	Y (Yes) For an Insert record the field is displayed so that the user can enter initial information for it.
DFLT	Default value for the field in an insert line. Valid only if NAME is specified. The data specified is displayed in the field. If DFLT is omitted, the initial value of the field is blank.

Color parameters can also be modified. Refer to "Color Parameters" later in this section.

@VAL

@VAL is a line type used to set the color of the current field according to the value of the field. More than one @VAL line type can be specified for different values of the current field. The first match found between the field's value and the value specified in the @VAL line determines the color of the field.

The valid format of a @VAL line is:

@VAL DATA=data

Where:

DATA Specifies data which should be compared to the current value of the field. The data may contain * and ? mask characters.

Color parameters must also be specified. In case of a match between the current field's value and data in the @VAL line, the colors specified in the @VAL parameter will override the colors specified in the @FIELD line.

Section 2: Customizing and Administering IOA

@END

The @END line type is used to end the definition of the current display format.

@DLM

The @DLM line type is used to change the currently used delimiter in this member to a new one. The default delimiter is ; This line can be inserted anywhere in the member. The valid format for a @DLM line is:

```
@DLM=newold
```

Where:

new New delimiter.

old Old delimiter.

In the @DLM statement the new delimiter is specified as the value for the DLM keyword, and is delimited by the old delimiter.

Example

```
@DLM=,;           <<----- delimiter first set to ,  
@DLM=&,           <<----- delimiter then set to &  
@DLM=; &          <<----- delimiter set back to ;
```

Color Parameters

The @STYLE, @LINE, @FIELD and @HEADER line types support COLOR and HILITE parameters. Colors take effect on color monitors only. On monochromatic monitors, the highlight takes effect.

Valid colors are:

NONE (Default)

WHITE

BLUE

GREEN

TURQ

RED

PINK

YELLOW

Valid highlights are:

NONE (Default)

BLINK

USCORE

REVERSE

DARK

Users can create a combination of color and highlight for each field by specifying COLOR and HILITE parameters on the @FIELD line.

In case of monochrome monitors, users can add the INTENS parameter to the @FIELD line type.

Valid intensities are:

LOW (Default)

HIGH

The COLOR, HILITE, and INTENS parameters can all be specified. Each takes effect when appropriate.

If the above parameters are omitted on the @FIELD line, the field inherits those parameters from the @LINE preceding it.

If the above parameters are omitted on the @LINE line, the line inherits those parameters from the @STYLE preceding it.

If the above parameters are omitted on the @STYLE line, they are set to their default values.

@STYLE, @LINE and @FIELD also accept the COLORA and HILITEA parameters that accept a range of values similar to COLOR and HILITE, respectively. These parameters can be used to specify a set of alternate colors when displaying records: the first record on the screen is displayed with COLOR and HILITE; the second record is displayed with COLORA and HILITEA; the third record with COLOR and HILITE, and so on. This can cause an alternating color effect. If COLORA and HILITEA are omitted, the alternate colors are set to be identical to the standard color values.

Extended Color Support

The following should be considered when using the extended color feature of the IOA Online facility:

ISPF Characteristics

Due to ISPF characteristics, adjacent fields with different colors are separated by a byte without color (black). Therefore, some IOA screens have a different appearance under ISPF than under other Online environments (e.g., native TSO, CICS).

IMS/DC and IDMS/DC

IOA does not automatically recognize an IMS/DC or IDMS/DC terminal as supporting extended color attributes. If the IMS/DC or IDMS/DC terminal supports extended color attributes, the ICOLOR=YES parameter must be added to the user's profile. For details on modifying the user's profile, refer to "Profile Variables" later in this section.

It is important not to use parameter ICOLOR=YES for non-color terminals. Doing so will cause an error and the terminal will be inhibited.

COM-PLETE

At this time, IOA does not support extended color attributes under COM-PLETE. In a future release of IOA, color support for COM-PLETE will be added.

IRMA PC Terminal Emulator Users

Early versions of IRMA (from DCA[®]) have a problem interpreting the 3270 color stream. As a result, IOA screens may appear with incorrect coloration. If you encounter this problem, call your local IRMA support representative and request a later version of IRMA™ that corrects this problem.

Customizing Extended Color Support

The IOA environment uses extended color support if the terminal is capable of supporting it. Various defaults are set so that if no changes are made, extended color is used wherever possible.

Two VTAM commands can be used to help determine whether or not a particular terminal supports color:

- RPQ

This is a VTAM command which can be issued to a terminal, and which causes the terminal to respond with an indication whether or not it supports extended color.

Note This command is a physical command, and is therefore not supported by certain older control units.

- LOGMODE

This is an entry in one of the VTAM tables, which logically defines the capabilities of a terminal. The following table indicates that the capability of the terminal to support extended color is determined by the contents of the VTAM LOGMODE entry:

Environment	RPQ Default	Suppress Issued By	Force Issuing RPQ	No Extended Color
Native TSO and ROSCOE	RPQ	IOA, if TCOLOR=TERMINAL	TCOLOR=YES or NO	TCOLOR=NO
TSO and ROSCOE (Cross-Memory)	RPQ	IOA		ICOLOR=NO
ISPF	RPQ	ISPF	N/A	LOGMODE
IOAVMON	RPQ	IOAVMON		LOGMODE
IMS/DC and IDMS	ICOLOR no issued.			ICOLOR=NO
CICS	RPQ	IOACICS		ICOLOR=NO

In the previous table, TCOLOR and ICOLOR are IOA Profile variables. See "IOA Profiles" in this section for more information about these variables.

Note For TSO/ROSCOE Cross-Memory and CICS, the appropriate change must be applied before the ICOLOR variable is activated.

IOA Access Method

Several IOA products include files that are processed using an internal component called the IOA Access Method.

The IOA Access Method is used to create indexed (keyed) file structures that offer enhanced data integrity. IOA Access Method files are physical sets of sequential datasets which are allocated, formatted and accessed by IOA products via IOA Access Method utilities.

Most IOA Access Method files consist of two separate sequential datasets:

- A data component, where the actual record information resides.
- An index component, which provides keyed access to records in the data component.

Certain IOA Access Method files may contain only an index or data component.

Both index and data information are stored in compressed format, resulting in reduced I/O processing overhead and optimum disk space utilization.

The IOA Access Method provides Dual Mirror Image file support. For example, this mechanism enables to quickly recover from a disk crash, if the crash occurred on a disk that contains the IOA Access Method primary file.

File Structure

IOA Access Method index components contain fixed length index records. IOA Access Method file data component records can be either fixed or variable length.

The first block in each IOA Access Method file extent consists of a control record, which contains a bit table indicating the blocks within the extent that contain free space for additional records.

IOA Access Method index files function as balanced tree index structures.

In addition to index (key) and data record pointer fields, IOA Access Method file index component records can contain non-key data, enabling non-key selection criteria to be used without accessing the associated Access Method file data component. Non-key data fields in index records can be stored in compressed format, providing maximum space utilization benefits.

Section 2: Customizing and Administering IOA

Examples

- CONTROL-D User Report list files are indexed by recipient name, job name and record creation timestamp.
- CONTROL-B Group Definition files are indexed by group name.

Data and index components may extend beyond one physical dataset. Initially, however, a component is allocated as a single physical dataset called the primary extent.

To increase an IOA Access Method file component's capacity, a maximum of 255 secondary extent datasets can be allocated and formatted. Using an IOA Access Method definition option, secondary extent allocation and formatting can automatically be performed when an out-of-space condition occurs.

Note Currently, secondary extents are supported only in CONTROL-D and CONTROL-V.

IOA Access Method Naming Conventions

IOA Access Method files which are allocated during CONTROL-D, CONTROL-V and CONTROL-B installation are listed below and use the following naming convention:

prefix.dbname.Ennn

Where:

prefix IOA Access Method file name prefix. The prefix is specified during product installation by setting relevant installation variables (e.g., DBPREFB for CONTROL-B, and DBPREFD for CONTROL-D and CONTROL-V).

Example: CTD.PROD

Note The prefix may contain more than one qualifier.

dbname IOA Access Method file identifier name. The identifier name is specified during product installation as part of the name assigned by the parameter DSN in the Access Method file's definition statements. For example, identifier name ACT is assigned to the CONTROL-D Active User Report list data component while ACTI is assigned to the CONTROL-D Active User Report list index component. The identifier name of an Access Method file should never be modified.

Ennn Extent sequence number suffix generated internally by IOA Access Method management routines whenever an Access Method file component is allocated. The suffix consists of the letter E followed by nnn, a number from 000 to 255, indicating an Access Method file's extent sequence number. A maximum of 256 physical file extents can be associated with each IOA Access Method file.

CONTROL-B Files Supported by the IOA Access Method

CTB.PROD.GRPD.E000	CONTROL-B Group Definition file data component.
CTB.PROD.GRPI.E000	CONTROL-B Group Definition file index component.
CTB.PROD.JAFD.E000	CONTROL-B Rule Activity file data component.
CTB.PROD.JAFI.E000	CONTROL-B Rule Activity file index component.
CTB.PROD.MODD.E000	CONTROL-B Basic Parameters file data component.
CTB.PROD.MODI.E000	CONTROL-B Basic Parameters file index component.
CTB.PROD.REPD.E000	CONTROL-B Report file data component.
CTB.PROD.REPI.E000	CONTROL-B Report file index component.
CTB.PROD.VARD.E000	CONTROL-B Variables Generation file data component.
CTB.PROD.VARI.E000	CONTROL-B Variables Generation file index component.
CTB.PROD.ABF.E000	CONTROL-B Active Balancing file data component.
CTB.PROD.ABFBKP.E000	CONTROL-B Backup Active Balancing file data component.

CONTROL-D/V Files Supported by the IOA Access Method

CTD.PROD.HST.E000	CONTROL-D History User Report list data component.
CTD.PROD.HSTI.E000	CONTROL-D History User Report list index component.
CTD.PROD.ACT.E000	CONTROL-D Active User Report list data component.
CTD.PROD.ACTI.E000	CONTROL-D Active User Report list index component
CTD.PROD.PRM.E000	CONTROL-D Permanent User Report list data component.
CTD.PROD.PRMI.E000	CONTROL-D Permanent User Report list index component.
CTD.PROD.MIG.E000	CONTROL-V Migrated User Report list data component.
CTD.PROD.MIGI.E000	CONTROL-V Migrated User Report list index component.

Note *Jobs or procedures using IOA Access Method files should contain a DD statement only for the first extent of the primary file. All other extents are dynamically allocated when the IOA Access Method file is opened.*

File Utilities

A comprehensive set of utilities is provided for IOA products which use IOA Access Method files. These utilities are used to create and maintain IOA Access Method files.

IOADBF	Allocates and formats an IOA Access Method file's index or data component.
IOADIG	Verifies the integrity of an IOA Access Method file's data component.
IOADII	Verifies the integrity of an IOA Access Method file's index component.
IOADPT	Prints an IOA Access Method file's data and index component records in SNAP dump format.
IOADCPY	Copies an IOA Access Method file from the dual mirror image file to the primary, as well as from the primary file to the dual mirror image file.

Section 2: Customizing and Administering IOA

IOADLD	Loads an IOA Access Method file's contents from a sequential file.
IOADUL	Unloads an IOA Access Method file's contents to a sequential file.

Note *Products which use the IOA Access Method contain additional utilities which are specific to the product. For more details about additional IOA Access Method utilities, refer to the summary list of utilities in Section 9.*

File Definition Statements

When allocating and formatting an IOA Access Method file (i.e., via utility IOADBF), a library member containing IOA Access Method File Definition statements is read.

These statements define an IOA Access Method file's physical and logical attributes, such as dataset name, unit, volume parameters, space parameters, record key length (for IOA Access Method file index components), and whether additional file extents are allocated automatically or manually when the current file extent is full. The members reside in the IOA INSTCTx library, where x is a one-letter product ID.

Examples

- The IOA Access Method File Definition statement for CONTROL-D's History User Report list components can be found in the DEFHST and DEFHSTI members in the IOA INSTCTD library.
- The IOA Access Method File Definition statement for CONTROL-B's Group Definition components are the DEFGRPD and DEFGRPI members in the IOA INSTCTB library.

The IOA Access Method file's attributes are kept in the file's control block. If you need to change these parameters after the file is created, modify the Definition Statements member, and run utility IOADBF (with FUNC=CHANGE).

For more information on IOA Access Method File Definition statements, refer to Utility IOADBF in Section 9.

Dual Mirror Image File Support

Each IOA Access Method file can be created with dual mirror image file support. For each logical extent of the file, the IOA Access Method automatically allocates a dual extent of the same size. The name of the dual extent is identical to that of the corresponding primary extent, except for the first character of the last qualifier which is set to D instead of E (e.g., prefix.dbname.Dnnn).

The dual mirror image file is created if parameter DUAL in the IOA Access Method definition member is set to Y. See the explanation on utility IOADBF in Section 9 of this guide. The definition member also specifies the unit and volume names for the dual file allocation. It is highly recommended that different volumes be used for the primary and dual files for improved performance and recovery.

The IOA Access Method synchronizes the primary and dual files. All read operations are performed only from the primary file, and all write operations are performed in parallel to both files. The dual file is allocated dynamically. It is recommended to activate the dual mirror image support only for the IOA Access Method data component, and not for the index component.

Dual Mirror Image File Parameters

The following parameters are used to manage the dual mirror image file:

- **DUALM**

If the DUALM parameter is set to Y and an error occurs in the dual mirror image file, the current operation fails for the primary file as well, and a manual correction should be made before any further processing is done. If the DUALM parameter is set to N, the processing continues for the primary file only, without dual mirror image file support.

- **DUALST**

An additional option of the dual mirror image is an internal timestamp checkpoint to ensure that the primary and dual copies are fully synchronized with each other. This ensures greater data integrity but at the same time increases the I/O processing overhead.

This option is activated if parameter DUALST in the IOA Access Method definition member is set to Y during file creation. If the timestamp values are not synchronized between the primary and dual files, the dual file should be rebuilt. See the explanation for utility IOADBF in Section 9 of this guide for additional information.

Recovering From a Damaged IOA Access Method File

If the IOA Access Method is used with dual file support, and one of the two copies is corrupt or becomes unavailable, the damaged file can be restored by using utility IOADCPY. This utility copies all the extents of the working copy to the corresponding extents of the corrupted copy. See Section 9 of this guide for more information about this utility.

IOA Profiles

A profile is a set of values to be used for certain fields in IOA screens, that are used as default values during sessions of a specific user or group of users (e.g., the library name field in the Job Scheduling Definition entry panel, or a show screen filter).

The IOA Online environment provides the following types of profiles.

- The Global profile, which can be customized by the system administrator only. This profile serves as the default setting for all users in the Online environment. If the user profile does not indicate a setting for a specific field, or does not provide a value for a specific variable, the value specified in the Global profile is used.

The Global Profile also contains screen filter definitions that can be used by any user in addition to the profiles that are defined in his/her user profile.

The Global profile is a combination of the default values in member \$PROFFLD, and the overriding specifications in member \$PROFMOD. (See "Profile Members" in this section).

- User profiles are defined for users or specific groups of users. These profiles can include menu definitions and user-defined filters (described below). Settings in the User profile override the settings in the Global profile.

Profile Members

Global profile information is stored in the following members found in the IOA PARM library.

\$PROFFLD Contains the default settings for the IOA Online environment. This member is supplied with IOA and should not be used for customization. This member also contains a line that indicates the location of the User Profile members.

\$PROFMOD Contains global customization information. The member can contain the following types of information.

- Global filter definitions.
- User profile locations. If specified, these override the location of the User profiles specified in member \$PROFFLD.
- Values to override the default values for fields specified in member \$PROFFLD.

This member can be copied so that customization need not be redefined when upgrading from a previous release.

\$PROFVAR Contains the values specified for profile variables (runtime parameters). The values of all profile variables are maintained via ICE (Installation and Customization Engine). See "Profile Variables" in this section for more information on the variables in this member.

User profile members are located in libraries specified in member \$PROFFLD, and optionally, in \$PROFMOD. These libraries are listed in the lines beginning with a dollar sign (\$) in the \$PROFFLD member. It is possible to divide the user profiles at your site into more than one library. By default the following line is included in member \$PROFFLD:

```
$*           =%OLPREFA%. PROF
```

This line indicates that all user profiles are found in the PROF library with the prefix of the dataset in which IOA was installed.

User profiles can be divided into more than one library. For example:

```
$M*         =%OLPREFA%. PROFM
```

```
$N*         =%OLPREFA%. PROFN
```

May be used to direct all User profiles beginning with M to be stored in the PROFM library and User profiles beginning with N to be stored in the PROFN library.

As shown, mask characters (* and ?) can be used in the specification of the user names (profiles) to be stored in a specific library.

Profile Contents

The following types of lines can appear in a profile member:

- Lines that contain an asterisk (*) in column one are remark lines.
- Lines that contain a dollar sign (\$) in column one indicate user profile libraries (described above).
- Lines describing either variables, fields or filter fields. These are called Profile Symbol lines.

The format of a symbol line is described below:

Columns 1-8	Profile symbol (field) name. Mandatory.
Column 9	Period. (Mandatory only if a Filter name follows.)
Columns 10-17	Filter name. Optional. Default: blank.
Column 18	Equal sign (=). Mandatory.
Columns 19-72	Profile symbol value. (Blank is a valid value.)

Example

The line shown below is a regular field line (not a filter).

```
SRLBLIB          =%OLPREFB%.RULES
```

Example of a Filter

The following lines describe a global filter ENDNOTOK, which, when activated, will display all ended NOTOK jobs from option 3 (Job Status).

```
SSTA700 .ENDNOTOK=N
SSTA704 .ENDNOTOK=N
```

Considerations for Profile Symbol Lines

- Values for Profile variables are specified in ICE (using the CUSTOMIZE option for IOA in the IOA Installation Main Menu). The values are stored in member \$PROFVAR in the PARM library. See "Profile Variables" later in this section for details on what values to specify for the Profile variables via ICE.
- Field lines contain values to be used for fields in IOA that are not part of a SHOW window (e.g., a library name in a rule/job/mission definition entry panel). Field lines may be located in any of the following members:

\$PROFFLD	The field line indicates the default value supplied with the IOA installation.
\$PROFMOD	The field line indicates a value specified as part of on-site customization of the Global profile. Field line specifications in \$PROFMOD override specifications in \$PROFFLD.
\$userid	The field line indicates a value specified as part of a user profile.

Section 2: Customizing and Administering IOA

- Filter field lines specify values for selection criteria fields in Show Window filters (e.g., Y or N for field UTILITIES in the IOA Log Show Screen Filter window, which indicates whether or not messages resulting from CONTROL-T utilities should be displayed).

SHOW Window Filters

A filter is a set of values for fields in a show window. The user can name a set of values and activate a filter at a later time by specifying the following command in the command line of the relevant screen:

SHOW filtername

Filters that are saved by a specific user are stored in that user's User Profile member. To add a filter to the Global profile, perform the following steps:

1. Create and save the filter in the normal way (i.e., in your User Profile member).
2. Copy the lines containing the name of the filter from your User profile to the Global Profile member (\$PROFMOD).

Some default filters are provided in member \$PROFFLD.

Note *IOA filters are currently available for options 3 and 5 in the IOA Primary Option menu. See the descriptions of these options in the product-specific user manual(s) for more information on IOA filters.*

Profile Attribute in the Screen Definition

IOA is installed with profile attributes assigned to various fields. If you wish to change these defaults, read the following paragraphs.

A field can be defined with a profile attribute by adding the parameter PROF=YES to the field macro definition in the screen definition member in the MSG library. For example:

CTMSFLD NAME=SSCHROL,ATTR=UH,LTH=4,PROF=YES

For instructions on how to apply a change to the screen format, see "Recommended Stages of Screen Modification" earlier in this section.

Under certain screens, parameter PROF takes a different form. Instead of YES, it designates the name of the profile variable. (When YES is used, the field name is used as the Profile Variable name.) If you want to change the profile attributes of these screens, consult your IOA representative.

Note *The IOA profile capability can be applied to fields in the header, footer, and windows of each screen, but not to fields which appear as part of a scrollable area on the screen. If you wish to apply the profile capability to a field in a scrollable area, consult your IOA representative.*

Saving a Profile

A profile is saved by writing it to the user profile library in a member name identical to the user ID (the Online environment session ID, e.g., TSO user ID) only when the user exits the IOA Online facility, and only in case of normal termination.

Note *If the profile library is full, the attempt to save a User profile fails with a D37abend. It is therefore recommended that the profile library be compressed regularly.*

Profile Variables

Follow the Major and Minor steps in ICE to specify values for the Profile Variables. If you are not familiar with the ICE application, refer to the IOA Installation Guide.

To change the value of a Profile variable, use the CUSTOMIZE option in the IOA Installation Main menu.

When IOA is installed, all Profile variables receive a value. Values for the profile are first stored in ICE tables (ISPF tables). These values are copied from the ICE tables into the parameter members once all ICE steps are performed.

ICE allows referencing of previous settings of Profile variables specified in other (previous or current) environments of IOA. When installing a new IOA environment, it is possible to automatically retrieve the predefined settings at your site. A default value is provided for any new profile variables that are not referenced.

When updating the parameter values, ICE verifies that the values stored in the ICE tables are the most current values. The values are verified using an internal comparison mechanism that compares the values in ICE and the values in the target parameter members.

If all values are identical, processing continues. If conflicts are identified, a screen is displayed listing the parameters for which the values in ICE are different than the values specified in the parameter members. You can select which version of the parameters is required, and either retain the current values in ICE or copy the values from the parameter member into ICE.

Notes *Differences between ICE tables and parameter members can occur when the user updates values in ICE and exits without updating the parameter member, or when the user updates the parameter member outside ICE.*

If no value is specified, a default value is provided for each profile variable. A full list of the Profile variables and their defaults appears later in this section.

Section 2: Customizing and Administering IOA

To change the value of a Profile variable, perform the following steps in ICE:

1. Specify IOA as the Product ID.
2. Select CUSTOMIZE (Option 7) from the Main menu.
3. Select Major Step 3, "Profile Variables."
4. Select the type of variable.
5. Locate desired variable and specify the required value.

Note Online help is available for each variable specified via ICE. Detailed explanation for each variable is provided later in this section.

Window Display Variables

SABKWND Controls whether or not to display the Save confirmation window when a request is made to update a backup mission from the CONTROL-D Active Mission screen (screen A.Z).

Y (Yes) A confirmation window is displayed.

N (No) The backup mission is processed without prompting the user for confirmation. Default.

✓ **SAPRWND** Controls whether or not to display the Save confirmation window when a request is made to update a printing mission from the CONTROL-D Active Mission screen (screen A.Z).

Y (Yes) A confirmation window is displayed.

N (No) The printing mission is processed without prompting the user for confirmation. Default.

✓ **SARSWND** Controls whether or not to display the Save confirmation window when a request is made to update a restore mission from the CONTROL-D Active Mission screen (screen A.Z).

Y (Yes) A confirmation window is displayed.

N (No) The restore mission is processed without prompting the user for confirmation. Default.

✓ **SDECWND** Controls whether or not to display the Save confirmation window when a request is made to update a decollation mission from the CONTROL-D Active Mission screen (screen A.Z).

Y (Yes) A confirmation window is displayed.

N (No) The decollation mission is processed without prompting the user for confirmation. Default.

Section 2: Customizing and Administering IOA

SAMAWND Controls whether or not a confirmation window is displayed when a request is made to rerun a job from the CONTROL-D Active Missions screen (screen A).

Y (Yes) A confirmation window is displayed. Default.

N (No) The request to rerun a job is processed without prompting the user for confirmation.

SAMDEL1 Controls whether or not a confirmation window is opened when a request is made to delete an entry from the CONTROL-D Active Missions screen (screen A).

Y (Yes) A confirmation window is displayed.

N (No) The request to delete an entry from the Active Missions screen is processed without prompting the user for confirmation. Default.

SDWYCNA Controls whether or not a confirmation window is displayed when a request is made to add a prerequisite condition from the CONTROL-D Why screen (screen A.?).

Y (Yes) A confirmation window is displayed.

N (No) The add request is processed without prompting the user for confirmation. Default.

SMSCWND Shows the Reset confirmation window in the CONTROL-O Message Statistics screen (screen OM).

Y (Yes) A confirmation window is displayed. Default.

N (No) The Reset is processed without prompting the user for confirmation.

SSTACNS Controls whether or not a confirmation window is opened when a request is made to confirm a rerun for a job placed in the Active Jobs file (screen 3) with the status of WAIT CONFIRMATION (CONTROL-M without CONTROL-R).

Y (Yes) A confirmation window is displayed. Default.

N (No) An attempt is made to rerun the job without prompting the user for confirmation.

SSTAWND Controls whether or not a confirmation window is opened when a request is made to confirm a rerun for a job (CONTROL-M without CONTROL-R).

Y (Yes) A confirmation window is displayed. Default.

N (No) An attempt is made to rerun the job without prompting the user for confirmation.

Section 2: Customizing and Administering IOA

- 11 SSTDEL1 Controls whether or not a confirmation window is opened when a request is made to delete an entry from the CONTROL-M Job Status screen (screen 3).
- Y (Yes) A confirmation window is displayed.
- N (No) The delete request is processed without prompting the user for confirmation. Default.
- 12 SUSRDELW Controls whether or not a confirmation window is displayed when a request is made to delete a report in screen U of CONTROL-D.
- Y (Yes) A confirmation window is displayed.
- N (No) The delete request is processed without prompting the user for confirmation. Default.
- 13 SSZMWND Controls the method of exiting the Zoom screen (screen 3.Z) of the CONTROL-M Online interface when the user has made changes.
- Y (Yes) The PF03/END command is interpreted as a SAVE request and a SAVE confirmation window is displayed to prompt the user for confirmation.
- If an explicit SAVE command is issued in the command line, changes to the job entry is made without displaying the SAVE confirmation window.
- N (No) An explicit SAVE command must be issued in the command line to request that changes be kept. No SAVE confirmation window is displayed. Default.
- 14 SWHYCNA Controls whether or not a confirmation window is opened when a request is made to add a prerequisite condition from the CONTROL-M Why screen (screen 3.?).
- Y (Yes) A confirmation window is displayed. Default.
- N (No) The add request is processed without prompting the user for confirmation.
- 15 SNRSCNE Controls whether or not a confirmation window is displayed when a request is made to erase a condition from the IOA Manual Conditions list (screen 7).
- Y (Yes) A confirmation window is displayed.
- N (No) The erase request is processed without prompting the user for confirmation. Default.
- 16 SRESCND Controls whether or not a confirmation window is displayed when a request is made to delete a condition/resource from the IOA Conditions/Resources file (screen 4).
- Y (Yes) A confirmation window is displayed.
- N (No) The delete request is processed without prompting the user for confirmation. Default.

Color Variables

A color and highlight value can be specified for each of the profile variables described below. Each profile variable has its own default color and highlight value. The format for specification of color and highlight is:

profile-variable=color,highlight

Where:

profile-variable	Name of the profile variable for which color and highlight values are being specified. Profile variables used to specify color for elements in the IOA environment are listed below.
color	Color of the relevant element. Valid values are: BLUE RED PINK GREEN TURQ YELLOW WHITE
highlight	Type of display for the element. Valid values are: BLINK USCORE REVERSE

The following profile variables are used to determine the color and highlight attributes of specific elements in the IOA environment:

MSGERR	Attribute for ERROR messages (whose codes end with E). Default: PINK,REVERSE.
MSGINF	Attribute for INFORMATION messages (whose codes end with I). Default: TURQ,REVERSE.
MSGSVR	Attribute for SEVERE error messages (whose codes end with S). Default: RED,REVERSE.
MSGWRN	Attribute for WARNING messages (whose codes end with W). Default: YELLOW,REVERSE.
SOLVHFC	Attribute for the header area in Online Viewing screens (screens U.V and 3.V). Default: PINK,REVERSE.

Section 2: Customizing and Administering IOA

SLNFPAG	Attribute for the sample lines when editing a report to create a ruler in the CONTROL-D Online Viewing screen (screen U.E). Default: RED,REVERSE.
SOLVOCL	Attribute for the overlay line in Online Viewing screens (screens U.V and 3.V). Default: RED,REVERSE.
SOLVSCL	Attribute for the scrollable area in Online Viewing screens (screens U.V and 3.V). Default: BLUE,REVERSE.
SSYVSCL	Attribute for the SYSOUT viewed from screens U.V and 3.V. This parameter is effective only for sites with CONTROL-R installed. Default:WHITE,REVERSE.

Work Mode Variables

PDATELTH	Date format length (does not affect all IOA date references). Valid values are: 6 digits. The format is ddmmyy. Default. 8 digits. The format is ddmmyyyy. <hr/> <i>Note This variable effects only to Western date formats.</i> <hr/>
PSCHDTYP	Defines the default value of the date field in the confirmation window opened as a result of ordering or forcing a job/rule/mission from a Definition List screen (screens 2, R, M, OR, and BM). DATE Gregorian date corresponding to DATE (the MVS system date) is inserted in the window. Default. ODAT Gregorian date corresponding to ODAT (the site working date) is inserted in the window.
SSCHBRO	Controls whether or not a user is forced into BROWSE mode when a schedule definition selected from any of the IOA definition screens is in use by another user. Y (Yes) Force the user into BROWSE mode. N (No) Display message CTM717E and reject the option. Default.
SSYVORD	Controls the order of jobs displayed under Option 3.V of the CONTROL-M/R Online interface (SYSOUT viewing). This parameter is relevant only for sites with CONTROL-R installed. FIFO SYSOUT is viewed in FIFO (first in, first out) order. The oldest job's SYSOUT is displayed first. Default. LIFO SYSOUT is viewed in LIFO (last in, first out) order. The most recent job's SYSOUT is displayed first.

- SINSLMT** Limits the lines to be read in the Media Database list (screen TI).
The search limit is a number from 1 to 999,999. Default: 100,000.
- SUS2LMT** Limits the lines to be read in screen U.
The search limit is a number from 1 to 999,999. Default: 500.
A value of 999999 specifies an unlimited search.
- SUS2SRTI** Automatically activate the SORT command in screen U.
Valid values are:
Y (Yes) Automatically activates the SORT command upon accessing screen U.
N (No) The SORT command is not activated when accessing screen U. Default.
See the CONTROL-D User Manual for more details about the SORT command in screen U.
- SVEWFFI** Used in online viewing screens U.V and 3.V to speed up the FIND command.
N (No) The regular FIND command is activated when viewing reports/sysouts. Default.
Y (Yes) Activates a fast find algorithm when viewing reports/sysouts.
The line where the string is found is always displayed as the first line on the screen. In CONTROL-D, the fast find algorithm applies when viewing reports without rulers.
- PZUMFAPP** In the Job Scheduling Definition screen (screen 2), controls whether or not the user is required to enter a value in the APPLICATION field.
Y (Yes) The user is required to enter a value in the APPLICATION field.
N (No) The user is not required to enter a value in the APPLICATION field. Default.
- PZUMFGRP** In the Job Scheduling Definition screen (screen 2), controls whether or not the user is required to enter a value in the GROUP field.
Y (Yes) The user is required to enter a value in the GROUP field.
N (No) The user is not required to enter a value in the GROUP field. Default.

Section 2: Customizing and Administering IOA

- IFLTSTA** Specified the name of an optional initial filter when entering the CONTROL-M Job Status screen (screen 3). If a valid filter name is specified, it becomes the default of the initial SHOW filter of screen 3.
- EXEC** Filter EXEC is displayed as the initial SHOW filter, until the user changes it using the SHOW xxx command, regardless of the last filter that was used.
- blank** No initial filter is used. The filter that was used the last time screen 3 was entered is displayed. Otherwise, the default filter is used as the initial filter. Default.
- RESWPF3** Specifies whether PF3 will be used in addition to PF4 in the Show window of the Job Status screen (screen 3) to reset (i.e., exit the window with no action). Valid values are:
- Y (Yes)** PF3 will reset the window.
- N (No)** PF3 will not reset the window. Default.
- SRESDRNG** Determines date range for the Conditions/Resources file (screen 4). Valid values are:
- ALL**
- YEAR**
- MONTH**
- nnn** Sets FROM TO -nnn days from today, and UNTIL TO the current date. Default: -3.
- SRCHLMT** Specifies the search limit for the IOA Log file and the Automation Log file (screens 5 and OA). This parameter must contain six valid digits (including leading zeroes if necessary). Default: 10,000.
- PNRSDTYP** Defines the default value of the date field in the window opened as a result of adding a new manual condition (ADD NEW command) in the IOA Manual Conditions file (screen 7). Valid Values are:
- DATE** Gregorian date corresponding to DATE (the MVS system date) is inserted in the window. Default.
- ODAT** Gregorian date corresponding to ODAT (the IOA operating date) is inserted in the window.
- SNRSDRNG** Determines the DATE range for the IOA Manual Conditions file (screen 7). Valid values are:
- ALL**
- YEAR**
- MONTH**
- nnn** Sets FROM TO -nnn days from today, and UNTIL TO the current date. Default: -3.

Presentation Mode Variables

ISPFBOTL	Controls the position of the last screen line when the IOA Online interface operates under ISPF.
L1	The last IOA screen line is displayed on the bottom line of the physical screen. Default.
L2	The last IOA screen line is displayed on the second line from the bottom of the physical screen.
SBALTBO	Controls whether a confirmation window or a double confirmation window is opened when an entire mission is ordered or forced in the CONTROL-B Mission List screen (screen BM).
N (No)	A confirmation window is opened when a mission is ordered or forced. Default.
Y (Yes)	A double confirmation window is opened when a mission is ordered or forced.
SRCMTBO	Controls whether a confirmation window or a double confirmation window is opened when a rule is ordered or forced in the CMEM Rule list (screen C).
N (No)	A confirmation window is opened when a mission is ordered or forced. Default.
Y (Yes)	A double confirmation window is opened when a mission is ordered or forced.
SMISTBO	Controls whether a confirmation window or a double confirmation window is opened when an entire mission is ordered or forced in the CONTROL-D Printing/Backup/Restore Mission list (screen M).
N (No)	A confirmation window is opened when a mission is ordered or forced. Default.
Y (Yes)	A double confirmation window is opened when a mission is ordered or forced.
SOMPTBO	Controls whether a confirmation window or a double confirmation window is opened when an entire table is ordered or forced in the CONTROL-O Rule Table list (screen OR).
N (No)	A confirmation window is opened when a table is ordered or forced. Default.
Y (Yes)	A double confirmation window is opened when a table is ordered or forced.
SREPTBO	Controls whether a confirmation window or a double confirmation window is opened when an entire mission is ordered or forced in the CONTROL-D Mission list (screen R).
N (No)	A confirmation window is opened when a job is ordered or forced. Default.
Y (Yes)	A double confirmation window is opened when a job is ordered or forced.

Section 2: Customizing and Administering IOA

SRLDATO	Controls whether Rule Description or Rule Data is displayed by default in the CONTROL-T Rule list (screen TR). N (No) Display the Rule Data by default. Y (Yes) Display the Rule Description by default.
SUSRHDR	Controls the display type of the header of the User Report entry panel. Default: blank. When a blank is encountered, the following values will be used: When CONTROL-D is installed display type D. When CONTROL-V is installed display type V.
SUSRFTR	Controls the display type of the footer of the User Report entry panel. Default: blank. When a blank is encountered, the following values will be used: When CONTROL-D is installed display type D. When CONTROL-V is installed display type V.
SUSRVIEW	Controls whether or not the view indicator in the Report List screen (screen U) is displayed, and if it is displayed, what the format is. N (No) The view indicator is not displayed and not saved. Y (Yes) The view indicator is displayed as a blank if the report has not yet been viewed, and as V if the report has been viewed. Default. A The view indicator indicates the number of times (a maximum of 255) the report has been viewed. If the number is too large to fit into the display field, an asterisk (*) is displayed. <hr/> <i>Note</i> <i>By default, the view indicator is one column wide.</i> <hr/>
SSCHTBO	Controls whether a confirmation window or a double confirmation window is opened when an entire table is ordered or forced in the CONTROL-M Table list (screen 2). N (No) A confirmation window is opened when a table is ordered or forced. Default. Y (Yes) A double confirmation window is opened when a table is ordered or forced.
SSTALTO	Controls whether or not late jobs are combined with all other jobs in the Show Screen Filter window of the CONTROL-M Job Status screen (screen 3). Valid values are: ONLY Late jobs are not combined with all other jobs (i.e., only late jobs are displayed, regardless of what other selection criteria are specified). Default. blank Late jobs are combined with jobs that conform to other selection criteria.

Miscellaneous Variables

ICOLOR	Specifies whether or not extended color support should be used for terminals accessing the IOA Online facility from IMS, IDMS, and other systems that do not allow determination of terminal characteristics.
YES	Use extended color support for this terminal.
NO	Do not use extended color support for this terminal. Default.
IDBCS	Specifies whether or not Double Byte Character Set support should be used for terminals accessing the IOA Online facility from IMS, IDMS, and other systems that do not allow determination of terminal characteristics.
YES	Use Double Byte Character Set support for this terminal.
NO	Do not use Double Byte Character Set support for this terminal. Default.
TCOLOR	Specifies whether or not extended color support should be used for terminals accessing the IOA Online facility from TSO or ROSCOE.
TERMINAL	Query the hardware to determine whether or not to use extended color support for this terminal. Default.
YES	Use extended color support for this terminal.
NO	Do not use extended color support for this terminal.
TDBCS	Specifies whether or not Double Byte Character Set support should be used for terminals accessing the IOA Online facility from TSO or ROSCOE.
TERMINAL	Query the hardware to determine whether or not to use Double Byte Character Set support for this terminal. Default.
YES	Use Double Byte Character Set support for this terminal.
NO	Do not use Double Byte Character Set support for this terminal.
USCORE	Controls whether or the UNDERSCORE attribute for fields should be suppressed or honored. Some terminals and terminal emulator programs may display underscored fields with incorrect colors and/or reverse video.
YES	Honor the UNDERSCORE attribute for terminal fields.
NO	Suppress the UNDERSCORE attribute for terminal fields. Default.

Section 2: Customizing and Administering IOA

EDIMACNM	Name of the optional ISPF EDIT IMACRO to be used when editing of a job's JCL in screens 2.J and 3.J.
macro name	The specified IMACRO is invoked at entry to ISPF EDIT.
blank	No IMACRO is invoked at entry to ISPF EDIT. Default.

Other Types of Profile Variables

TERMTYPE	Specifies the terminal type used to access the IOA Online environment. Valid value is:
3270	IBM 3270 or 3270 Emulation terminal. Default value for MVS users.

Modifying IOA Product Defaults

After one or more CONTROL products are installed and running for a period of time, certain defaults can be modified within the product. For example, increase the number of CONTROL-M attempts to read a job's sysout.

1. Review the documentation member IOADFLTS, explaining the defaults which can be modified for IOA. This member resides in the IOA SECUDATA library.
2. Update member IOADFLTC in the IOA MAC library using member UMIDFLTU in the SECUDATA library. Modify one or more default values as detailed in the documentation member xxxDFLTS (where xxx is a product code). For the above example, you can modify the default value of APPLY to YES (turn off uppercase). Note that the parameters are in Assembler Macro format, and are therefore restricted to Assembler Macro syntax (continuation marker in column 72, continuation line starts at column 16, etc.).

Note Line number columns 73-80 must remain the same as in IOADFLTC in the IOA MAC library.

3. In addition to member IOADFLTC, there may be user exits in the IOA SECUDATA library which can be applied to modify default processing within a specific product. See member \$DOISECU in the IOA SECUDATA library for more information.

Note Once all the required changes are made to the CONTROL product defaults, it is necessary to restart all the monitors and users of the modified products by shutting down the monitors and restarting them, and logging users off and on.

Modifying IOA Messages

IOA messages can be adapted for local languages, site-specific terminology, etc.

IOA messages are stored in the IOA MSG library. The messages are organized in members with the following naming convention:

CTxAMyyy

CTxMyyy

CTxRMyyy

CTxTMyyy

CTxCMyyy

Where

x A single character indicating the product (e.g., M for a CONTROL-M screen).

yyy A 3-character message member identifier.

Each member contains definitions for one or more messages. The definitions are specified as Assembler macro instructions. Each member must be assembled and link-edited to create a CSECT and module with a name identical to the message member name. The load module containing the messages replaces an existing CSECT in one or more load modules in the IOA LOAD library.

The message member must contain a CTMLEVEL macro prior to the first message definition. If this macro is omitted, results can be unpredictable.

Each message is defined by macro CTMMMSG. This macro has the following parameters:

CODE Code of the message. Do not change this code, since it is used to connect the message to the application program.

MESSAGE Text of the message (in quotes).

PARM Length and position of the text inserted into the message by the IOA application. The format of this parameter is:

PARM = (pos1,length1,)

pos is the position of the inserted text in the message +6.

length is the length of the text to be inserted. Do not change this length.

Parameter PARM is optional, but must be used whenever it already appears in the message definition. When the contents of a message are modified, the position of the application generated area in the message may change, so the PARM field must be modified accordingly. The order of the fields in the PARM should not be changed, but it is possible to change the order of appearance in the message by changing the positions.

Section 2: Customizing and Administering IOA

Example

The original message (from member CTMMSEL):

```
CTMMMSG CODE=SEL206W,PARM=(18,5,29,16),  
MESSAGE='ABENDED CC XXXXX STEP YYYYYYYYYYYYYYYY'
```

The modified message:

```
CTMMMSG CODE=SEL206W,PARM=(45,5,23,16),  
MESSAGE='ABENDED ON STEP YYYYYYYYYYYYYYYY CODE XXXXX'
```

Recommended Steps for Message Modification

Sources (message members) of IOA products are modified via an SMP/E USERMOD. Use the following steps to perform this task:

1. Choose a name for the USERMOD (to be built in the following steps). The name of the USERMOD must be 7 characters in length, must begin with an alphabetic character (i.e., not a number or symbol), and must be unique.
2. Create a new member in the IOA MAINTLIB library with the same name as the USERMOD created in Step 1.
3. Copy the contents of member UMODSRC in the IOA JCL library to the newly-created member. This member contains a skeleton sample for the USERMOD (to be customized in the following steps).
4. Specify the name of the new USERMOD in the ++USERMOD statement.
5. Determine the FMID and the RMID of the message source via SMP/E Option 3.2. Specify SRC as the entry-type and the name of the message source as the entry-name.

Specify the FMID in the ++VER statement.

If the RMID value is not the same as the FMID value, add the RMID value to the ++VER statement via the PRE(rmid-value) parameter.
6. Specify the source-name in the ++SRC statement.
7. Copy the contents of the source member to the line immediately after the ++SRC statement. Update the copy to meet your needs.
8. Specify the name of the USERMOD in job UMODRACK in the IOA JCL library. Run the job. This job RECEIVES and APPLY-CHECKS the USERMOD.

Job UMODRACK must end with a completion code of 0.
9. Specify the name of the USERMOD in job UMODAPP in the IOA JCL library. Run the job. This job applies the USERMOD.

Job UMODAPP must end with a completion code of 0.

IOA Dynamic Destination Table

One of the IOA facilities is the Shout facility. It allows the user to specify messages to be sent to specified destinations upon the occurrence of various events.

Destinations in a production environment are not necessarily fixed. For example, the TSO logon ID of the shift manager may be different for every shift. The Dynamic Destination table allows the user to specify a logical destination name for a group of destinations. When the logical destination name is specified as the destination of a Shouted message, the message is sent to each destination in the group.

An example of such a table can be found in the member IOADEST in the IOA INSTALL library. To update the Destination table, you must assemble and link-edit load module CTMDEST using member IOADESTJ in the IOA INSTALL library, and subsequently reload the table into memory for each of the IOA products installed at your site. For more information regarding Dynamic Destination tables, see "IOA Dynamic Destination Table" in Section 1 of this guide.

Replacing the Current Dynamic Destination Table

The Dynamic Destination table is loaded when the IOA product monitor is started. To replace the current Destination table with a new one, use the appropriate operator command for each IOA product installed at your site.

F CONTROLM,NEWDEST (for CONTROL-M)

F CONTROLD,NEWDEST (for CONTROL-D)

F CONTROL-O,NEWDEST (for CONTROL-O)

F IOAFMON,NEWDEST (for the IOA Functional monitor used by CONTROL-T)

After a few seconds, a message regarding the result of the operation is displayed on the operator console from which the modify command was issued.

Example

The following are sample lines from a Dynamic Destination table:

```
CTMGDEST TSO-SHIFT-MNGR,DEST=(TSO-Q04,OPER,U-Q04)
```

```
CTMGDEST TSO-DBA,DEST=(TSO-S08,TSO-S09)
```

Expanding the Resource File (RES)

1. Close all the monitors and IOA activities.
2. Rename the old RES and SNC files to a different name.
3. Select major step "Customize Dataset Parameters" from the CUSTOMIZE activity in ICE. Change the value of parameters RESREC# and RESLREC to the required values (A maximum of 32 for parameter RESREC#, and a maximum of 32760 for parameter RESLREC). Complete the remainder of the minor steps for this step to update and compile member IOAPARM, and customize the formatting jobs.

Section 2: Customizing and Administering IOA

4. Edit member FORMRES in the IOA INSTALL library. Check that the value of parameter LRECL is identical to the value of parameter RESLREC in member IOAPARM, and that the value of parameter RECNUM is calculated as follows: $RESREC\#*32+2$. Submit this job. This job will allocate and format a new set of RES/SNC files with the new size.
5. If you do not use a dual resource file, skip to step 8 below.
6. Edit member FORMDRES in the IOA INSTALL library. Check that the value of parameter LRECL is identical to the value of RESLREC in IOAPARM, and that parameter RECNUM is calculated as follows: $RESREC\#*32+2$. Submit this job. This job will allocate and format a new set of ALTRES/ALTSNC files with the new size.
7. Copy the old resource file into the new one using utility IOACOPRS.
8. Start the monitors.

Expanding the IOA LOG File (LOG)

1. Close all the monitors and IOA activities.
2. Rename the old LOG file to have a different name
3. Select major step "Customize Dataset Parameters" from the CUSTOMIZE option in ICE. Recalculate the LOGSIZE parameter. Complete the remainder of the minor steps in this step to update and compile member IOAPARM, and customize the formatting jobs.
4. Submit member FORMLOG in the IOA INSTALL library. This job allocates and formats a new LOG file with the new size.
5. Copy the old LOG file into the new file using job IOACPLOG in the IOA JCL library.
6. Start the monitors.

Section 3: Customizing and Administering CONTROL-M

General	3-1
Activating the CONTROL-M Monitor	3-1
Shutting Down the CONTROL-M Monitor	3-1
Modifying the CONTROL-M Sleeping Interval	3-2
Starting the CONTROL-M EVENT MANAGER (CMEM) Facility	3-2
Shutting Down the CMEM Facility	3-3
Replacing an Active CMEM Monitor	3-3
Job Ordering via New Day Processing	3-4
Overview	3-4
Sample New Day Processing	3-5
How the Sample Components Perform New Day Processing	3-7
Date Control Records and Enhanced Daily Checkpointing	3-8
Implementing New Day Processing	3-9
Implementation Tasks	3-10
Programs Called During New Day Processing	3-16
Use of the Date Control Record by User Daily Jobs	3-17
New Day Procedure Flow	3-19
Activation of Started Tasks	3-20
CONTROL-M Monitor and JES Considerations	3-21
Cases of JES Malfunction	3-21
Special Considerations	3-21
Considerations for Changing Clocks	3-23
Advancing the Clock Forward	3-23
Moving the Clock Backwards	3-24
Sending a Job's SYSDATA to a Held Output Class	3-25
Dual Checkpoint Mode	3-26
Managing the CMEM Facility	3-26
Automatic Loading of Rules	3-27
Deleting (Deactivating) an Active Rule Table	3-28
Displaying Active Rules	3-29
Controlling CMEM Rule Operation Mode	3-29
Modifying the CMEM Sleeping Interval	3-30
Refreshing the CMEM Security Cache	3-30
Problem Determination	3-30
Inter-System Communications Facility	3-32

Problem Determination.....	3-30
Inter-System Communications Facility	3-32
Method of Operation	3-32
Create the Inter-System Communication Files.....	3-33
Prepare the Condition Selection List.....	3-33
Modify JCL Procedures and CLISTs.....	3-34
Copy JCL to Procedures Library	3-34
Customize the Exits	3-35
Start the Communication Monitors	3-35
Expanding CONTROL-M Files.....	3-35
Expanding the Active Jobs File (CKP).....	3-35
Expanding the CONTROL-M Jobs Dependency Network File (GRF).....	3-36
Expanding the CONTROL-M Statistics File (STAT)	3-36

General

This section describes the initialization, customization and administration features that are available for CONTROL-M.

Activating the CONTROL-M Monitor

The CONTROL-M monitor usually operates 24 hours a day as a started task (STC). Usually the monitor is automatically activated as part of the IPL process. To activate the monitor manually, use the operator command:

S CONTROLM

If the monitor is successfully activated, the following message appears on the operator console:

CTM100I CONTROL-M MONITOR STARTED

Once the CONTROL-M monitor is active, if you try to activate an additional CONTROL-M monitor with the same IOA components in the same computer, the new (i.e., additional) monitor immediately shuts down and an appropriate message is issued.

Note *It is possible to activate more than one CONTROL-M monitor in the same computer environment (e.g., PROD and TEST version) by defining a different IOA environment (and a different QNAME) for each monitor. See "Installation Considerations" in the CONTROL-M installation procedure in the IOA Installation Guide.*

Shutting Down the CONTROL-M Monitor

If for some reason it is necessary to shut down the CONTROL-M monitor, use the operator command:

P CONTROLM

After a few seconds (a maximum of a minute), the CONTROL-M monitor shuts down and the following messages appear on the operator console:

CTM107I SHUT DOWN UPON REQUEST FROM OPERATOR (not highlighted)

CTM120I CONTROL-M MONITOR SHUTTING DOWN (highlighted, unrollable)

In case of emergency, the CONTROL-M monitor can be canceled. However, this is not recommended. If the monitor is canceled, reports produced from the IOA Log file may not be complete.

Note *When you shut down the CONTROL-M monitor, all other CONTROL-M facilities (e.g., CMEM), IOA Online monitors, and Online facility sessions can remain active.*

Modifying the CONTROL-M Sleeping Interval

Periodically, at a predefined interval, CONTROL-M “wakes up” and checks what it has to do. This interval is set via a CONTROL-M installation parameter and can be changed by the system administrator. In addition, the interval can be altered by the operator command:

```
F CONTROLM, INTERVAL=xx
```

where xx represents the interval in seconds.

It is recommended that the interval be modified by automatic commands invoked by the CONTROL-M monitor itself according to set conditions and time ranges, and not manually by the operator.

At most sites, the interval should be longer during the day (when fewer batch production jobs are executing) and shorter during the night.

The optimal sleeping interval depends on the processing power of the machine. Depending on the processing power of the machine indicated below, the sleeping interval should usually not be less than the number of seconds indicated:

- For machines with less than 20 MIPS 10 seconds
- For machines with 20-50 MIPS 5-6 seconds
- For machines with over 50 MIPS 4 seconds

There is no practical benefit in setting the interval to less than these minimums. Doing so could slow down CONTROL-M operation.

When the modification is received by CONTROL-M, the following message is displayed on the operator console from which the modify command was issued:

```
CTM123I CONTROL-M INTERVAL IS SET TO nn SECONDS
```

Starting the CONTROL-M EVENT MANAGER (CMEM) Facility

The CONTROL-M Event manager (CMEM) is designed to handle events occurring outside the control of the CONTROL-M monitor. CMEM consists of a monitor which uses the IOA subsystem and performs predefined actions in response to system events (e.g., arrival of a specified job on the JES spool).

Note *When CONTROL-O is installed at your site, the CONTROL-O monitor takes over control of the CMEM facility. In this case, this description of the CMEM facility is not relevant for your site. Instead, refer to Section 5 in this guide for information on how to manage the CMEM facility at your site.*

The CMEM monitor must operate 24 hours a day. The usual way to ensure this is to automatically initialize the CMEM monitor during the IPL process.

To start CMEM as part of the IPL, add the following command to member COMMNDnn in library SYS1.PARMLIB (where nn is either the number specified in member IEASYS in library SYS1.PARMLIB, or defaults to 00 if not specified in IEASYS):

```
S CTMCMEM
```

The same operator command can be used to activate the CMEM monitor manually.

Shutting Down the CMEM Facility

Under normal circumstances, the CMEM monitor is not shut down. However, CMEM shutdown might be necessary for the following reasons.

- To resolve a problem that cannot otherwise be resolved. In this case, the monitor should be immediately restarted after the problem is solved to minimize the impact of the shutdown on the work environment.
- To clean up (erase) all loaded CMEM tables from memory, or stop all CMEM functionality (e.g., for a system shutdown).

To stop and immediately restart the CMEM facility, replace the active CMEM monitor by starting a new CMEM monitor. (See "Replacing an Active CMEM Monitor" below.)

When the monitor replacement method is not applicable, and a complete shutdown is required, issue one of the following operator commands:

```
F CTMCMEM, STOP
```

or

```
P CTMCMEM
```

CMEM shuts down after a few minutes.

Note CMEM rules are not triggered for dataset events and step termination events caused by jobs that start while CMEM is down.

Replacing an Active CMEM Monitor

If a CMEM monitor is currently active, and a new CMEM monitor is started (via operator command S CTMCMEM), the current CMEM monitor passes execution control to the new CMEM monitor and then shuts down. It is not necessary to reload the rule tables. They are passed from the current monitor to the new one. Therefore, to stop and immediately restart the CMEM monitor with minimum interference to on-going work, issue the following operator command:

```
S CTMCMEM
```

Job Ordering via New Day Processing

Overview

The CONTROL-M monitor is usually activated as a started task and remains active 24 hours a day. At a set time each day (defined via installation parameters), New Day processing is performed by the CONTROL-M monitor.

New Day processing consists of both automatic cleanup from the previous day's job ordering and automatic ordering of jobs for the new day.

The main components related to New Day processing are:

- Scheduling tables and job scheduling definitions
- New Day procedure and User Daily jobs
- Date Control records
- Active Jobs file

New Day processing is completely automated through the use of the New Day procedure¹ and User Daily jobs. The main purpose of the New Day procedure and User Daily jobs is to call programs that:

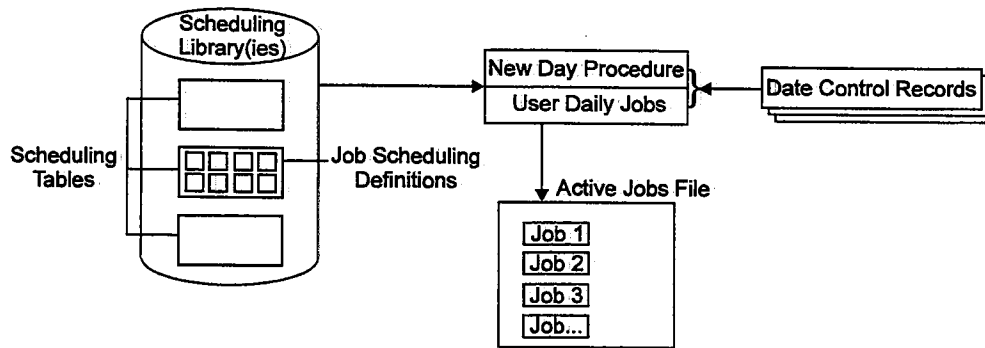
- Scan scheduling tables to select jobs for scheduling.
- Schedule the selected jobs (i.e., place copies of the selected job scheduling definitions as job orders in the Active Jobs file).

The New Day procedure and each User Daily job must have its own Date Control record. A Date Control record is a member in the CONTROL-M PARM library in which relevant date information is placed during New Day processing. This date information is used to manage job orders.

Selection of jobs is based on the Date Control record, the current date, and the Basic Scheduling parameters of the jobs in the scheduling tables. Any time the User Daily job is run, the current working date is placed in the Date Control record. The Basic Scheduling parameters of each job in the scheduling table are checked against this date to see if the job should be placed in the Active Jobs file.

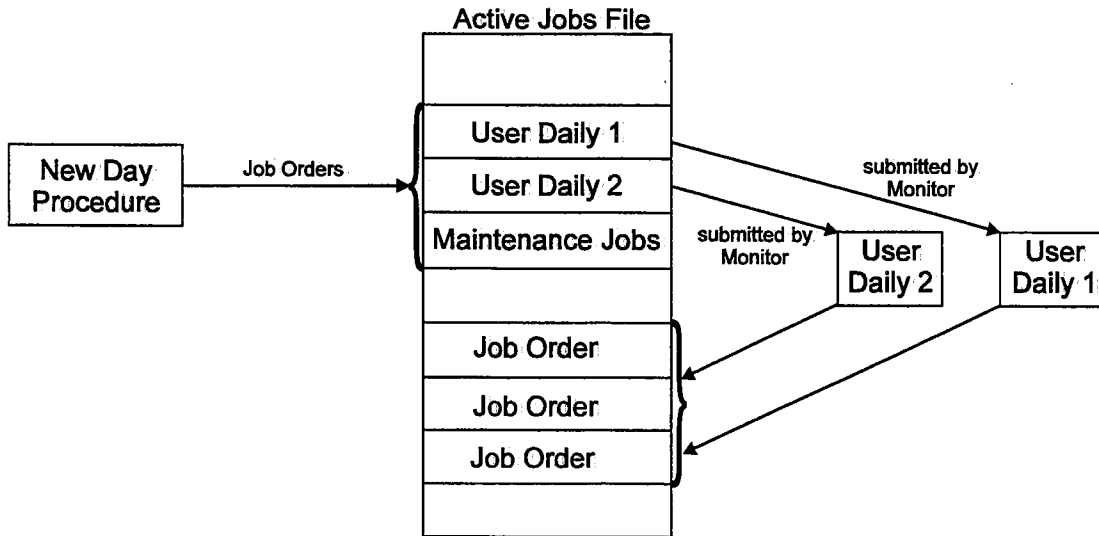
¹ Called the "General Daily" prior to IOA Release 5.0.0

Section 3: Customizing and Administering CONTROL-M



New Day processing generally works as follows:

- The New Day procedure is performed each day at its predefined time. It:
 - Schedules User Daily jobs.
 - Schedules maintenance jobs. These jobs call programs which perform cleanup after the previous day's processing.
- User Daily jobs (scheduled by the New Day procedure) select and schedule all other jobs for that day.



Sample New Day Processing

CONTROL-M is supplied with samples of several of the above mentioned components.

To effectively implement New Day processing at your site, you must first understand how the sample components operate. Once the operation of the sample components is understood, you can then customize New Day processing based on site requirements. Sample New Day processing components are described below.

Section 3: Customizing and Administering CONTROL-M

Sample Components Provided With CONTROL-M

At time of installation, each site is provided with the following:

- New Day Procedure

A single New Day procedure is provided. Its default name is CONTDAY (the name can be changed). This procedure should have wide authorization for accessing scheduling tables and jobs.

- User Daily Jobs

The following sample User Daily jobs are provided:

DAILYPRD Contains sample JCL for scheduling production jobs.

DAILYSYS Contains sample JCL for scheduling system jobs.

These sample User Daily jobs are defined in table MAINDAY in the SCHEDULE Library. These jobs activate User Daily procedure CTMDAILY which is responsible for ordering the production jobs.

It is generally advisable to use these sample User Daily jobs to create separate User Daily jobs according to department (or other functional entity) and according to authorization. This is described later in this section.

- Maintenance Jobs

The following maintenance jobs are provided:

CTMCLRES Activates utility IOACLRES which cleans the IOA Conditions/Resources file.

CTMLDNRS Activates utility IOALDNRS which creates and loads the Manual Conditions file. (This utility is normally run after all User Daily jobs have executed.)

These maintenance jobs are defined in table MAINDAY in the SCHEDULE library. Utilities IOALDNRS and IOACLRES are described in Section 9 of this guide.

- Scheduling Table: MAINDAY

A scheduling table called MAINDAY is provided in the SCHEDULE library. This table contains the above mentioned User Daily jobs (DAILYSYS and DAILYPRD) and the above mentioned maintenance jobs (CTMCLRES and CTMLDNRS).

- Date Control Records

The following Date Control records (members) are supplied in the CONTROL-M PARM library:

DATEREC Defined for New Day procedure CONTDAY.

DATERECU Sample Date Control record for User Daily jobs.

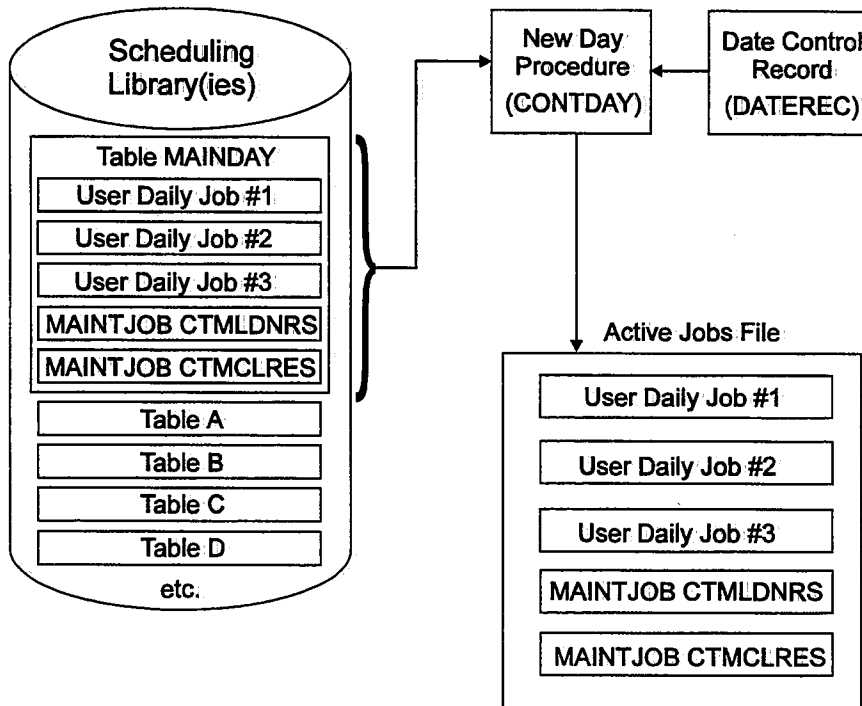
- Called Programs

The New Day procedure and User Daily jobs call programs which perform various steps of New Day processing (checking the Date Control record, selecting job orders, etc.). These programs are described in "Programs Called During New Day Processing" later in this section.

How the Sample Components Perform New Day Processing

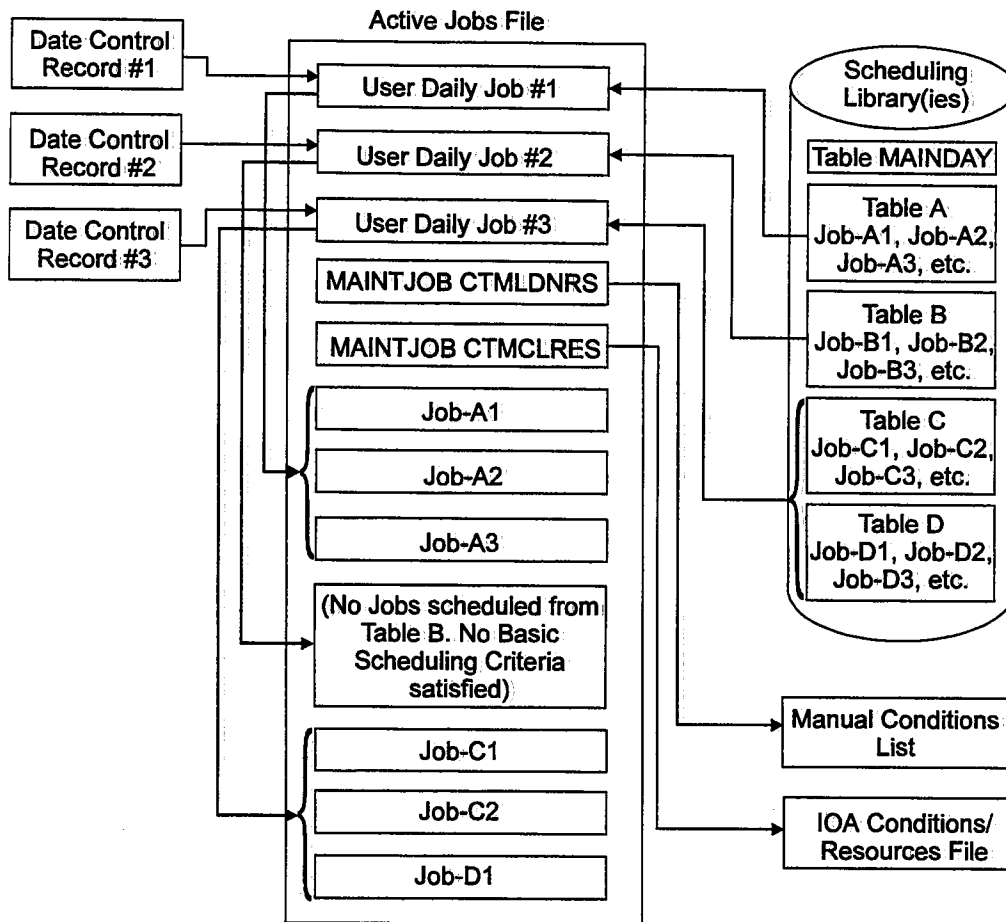
New Day processing performed with the sample components works as follows:

During New Day processing, the New Day procedure accesses its Date Control record, scans table MAINDAY, and selects and loads the maintenance jobs and User Daily jobs to the Active Jobs file.



The User Daily and maintenance jobs placed in the Active Jobs file are submitted by CONTROL-M according to their runtime scheduling criteria. When a User Daily job is executed, it accesses its own Date Control record, scans the scheduling tables defined to it, selects jobs, and places the selected job orders in the Active Jobs file. (User Daily jobs can also schedule maintenance jobs as required.)

Section 3: Customizing and Administering CONTROL-M



It should be noted that because User Dailies are jobs, their use is not restricted to New Day processing. Although User Daily jobs are normally executed immediately after they are ordered by the New Day procedure, they can be executed at any time defined in their Runtime Scheduling parameters. Furthermore, they can be ordered at any time by any of the methods described in "Alternative Methods of Job Ordering" in Section 6 of the CONTROL-M User Manual.

Date Control Records and Enhanced Daily Checkpointing

A Date Control record must be defined for each User Daily job. This record is usually defined by the IOA administrator.

The Date Control record for User Daily jobs consists of six fields. At different stages in New Day processing (i.e., before or after the execution of specific called programs which perform New Day processing), the current original scheduling date is placed in one of these date fields.

This enables CONTROL-M to manage the process of job ordering. Furthermore, if New Day processing is interrupted at any point, the values in these fields can indicate which called program was in use when the interruption occurred.

If the interruption occurred during job ordering, it is also desirable to identify the last job ordered so that ordering can continue from that point. An optional second record in the Date Control member, the Enhanced Daily Checkpoint record, enables this identification.

The Enhanced Daily Checkpoint record contains fields which store information about the last ordered job: JCL member name, internal sequence number, order ID, and the group to which the job belongs:

- If Enhanced Daily Checkpointing is implemented, this information “pinpoints” (identifies) from where job ordering left off, and job ordering continues from that point during recovery.
- If Enhanced Daily Checkpointing is not implemented, when the User Daily is rerun, all jobs in the scheduling table are considered for scheduling (including those jobs already scheduled by the interrupted run).

Both the Date Control record and the Enhanced Daily Checkpoint record have a fixed format, discussed later in “Implementing New Day Processing.”

Note If input scheduling tables are modified prior to rerunning User Daily jobs (or the New Day procedure), the checkpointed job and internal sequence number might not match. In this case, rerun of the User Daily job(s) is terminated and manual intervention is required.

Implementing New Day Processing

As indicated earlier, sample User Daily jobs (DAILYSYS and DAILYPRD) are supplied with CONTROL-M in scheduling table MAINDAY.

In theory, it is not necessary to use User Daily jobs. It is possible (but not recommended) to place all job scheduling definitions in one or more tables and have them scheduled by the New Day procedure.

It is also possible (and also not recommended) to maintain only the two sample User Daily jobs provided with CONTROL-M, and to order all user jobs through the User Daily DAILYPRD.

The recommended way of automating the production environment via New Day Processing is by:

- Defining a different scheduling table for each group of related jobs.
- Defining a different User Daily job for each department, application, or comparable entity.

Such implementation provides the following advantages:

Improved performance	Many User Daily jobs running in parallel can order the full day's production jobs in significantly less time than it would take one User Daily to order all jobs one at a time.
Ease of administration	The IOA administrator can make each department responsible for its own User Daily job(s) and scheduling table(s) and for controlling its own job ordering.
Increased security	While maintaining exclusive authorization for the New Day procedure, the IOA administrator can limit each department's authorization to its own User Daily jobs.

Section 3: Customizing and Administering CONTROL-M

Minimization of problems Problems encountered in one User Daily do not necessarily affect the job ordering of other User Daily jobs.

It is recommended that the following tasks (described below) be performed when implementing New Day processing:

- Decide which User Daily jobs are needed (and for which tables).
- Customize the sample New Day procedure (CONTDAY).
- Use the sample JCL to create JCL for each User Daily job.
- Create User Daily job scheduling definitions and customize table MAINDAY.
- Decide whether or not Enhanced Daily Checkpointing will be used.
- Create Date Control records.

Note *The New Day procedure and its accompanying Date Control record are defined at time of installation. They require no further implementation.*

- Ensure subsequent runs of utility IOALDNRS if necessary.

Implementation Tasks

Decide Which User Daily Jobs Are Needed (and for Which Tables)

A job scheduling definition is defined for each job, and each job scheduling definition is placed within a scheduling table. Usually, related job scheduling definitions are grouped into their own scheduling table.

Based on the scheduling tables defined at your site, and the jobs they contain, decide what User Daily jobs you require, and which scheduling tables each User Daily job will scan.

Customize the Sample New Day Procedure (CONTDAY)

Prerequisite conditions are always assigned a date reference (day and month). If old prerequisite conditions are not deleted from the IOA Conditions/Resources file, after a year jobs can be incorrectly triggered by the old prerequisite conditions with the same date reference. To prevent this problem, program CTMFRM is activated as part of the New Day procedure to erase all prerequisite conditions of the coming execution date from the IOA Conditions/Resources file.

IGNORE/SELECT Statements

Often, prerequisite conditions which are not date-related (e.g., IMS-IS-UP, AR-FILE-OK) must not be erased. A list of conditions which are not to be erased can be supplied via DD statement DAFRMIN. AutoEdit statements can be used as input to DAFRMIN. For more information see the description of utility CTMCAJF in Section 9 of this guide.

DD statement DAFRMIN is used to specify IGNORE and SELECT statements, which impact as follows:

- Jobs meeting SELECT criteria are deleted.
- Jobs meeting IGNORE criteria are bypassed (not deleted).

The valid format is:

```

IGNORE | SELECT {
COND      cond - prefix
GROUP     GROUP groupname
JOBNAME   JOBNAME jobname | MEMBER membername
STATUS    STATUS status [FROM date] [TO date]
}
    
```

Note *Conditions that are not date-related can also be defined with a date reference of STAT which eliminates the need for including IGNORE/SELECT statements in procedure CONTDAY.*

A maximum of 500 IGNORE/SELECT statements can be supplied. Jobs are handled according to the first statement for which criteria are met.

IGNORE/SELECT statements can come before or after IGNORE COND but should not be mixed with IGNORE COND.

All conditions beginning with the prefix (cond-prefix) specified in an IGNORE COND statement will not be erased.

Example

To suppress the erasure of the next day's conditions by the New Day procedure, specify one definition:

```
IGNORE COND *
```

When suppressing the function, remember to delete conditions (via utility CTMCLRES). If this is not done, jobs in the next year's schedule will be triggered because of today's conditions.

Parameters

One or more of the following parameters can be specified in any IGNORE/SELECT statement (in any order):

GROUP groupname	Name of the group appearing in the job scheduling definition.
JOBNAME jobname/ MEMBER membername	jobname is the job name appearing in both the job statement and the JOB NAME field of the Status screen. membername is the Job Scheduling Definition name (name of the JCL file). Job name or membername (but not both) can be specified in one statement.

Note *A group name, job name or member name prefix may be indicated by an asterisk (*) at the end of the name (e.g., PROD*).*

Section 3: Customizing and Administering CONTROL-M

STATUS status Specify one of the following job statuses:
ENDOK
ENDNOTOK
HELD
WAITSCHEDED
FROM date Original scheduling date (ODATE) in yymmdd format.
TO date Original scheduling date (ODATE) in yymmdd format.

Example

```
IGNORE JOB NAME OPER*
IGNORE JOB NAME PROD* STATUS ENDNOTOK
SELECT GROUP TEST
```

In this example, no jobs whose names begin with prefix OPER are deleted. Also no jobs whose names begin with prefix PROD that ended NOTOK are deleted. Of the remaining jobs, those belonging to group TEST are deleted.

Use the Sample JCL to Create JCL for Each User Daily Job

Create the JCL for each User Daily job by customizing the sample JCL shown below:

```
//DAILY EXEC CTMDAILY
//DACHK DD DISP=SHR,DSN=parm_lib(date_control_record)
//DAJOB DD DISP=SHR,DSN=sched_library(table1)
// DD DISP=SHR,DSN=sched_library(table2)
// DD DISP=SHR,DSN=sched_library(table3)
// DD DISP=SHR,DSN=parm_library(member)
```

The above sample allows two alternative methods of identifying scheduling tables to the User Daily jobs. The user can choose either or both methods:

- Normal Method DSN=sched_library(table)

This method requires that the user specify the name of a scheduling table and library directly in the JCL.

- Alternative Method DSN=parm_library(member)

This method requires that the user specify a PARM library and member. The specified member contains order requests which identify scheduling libraries, scheduling tables and jobs to schedule.

When using this method, the format of the records placed in specified member of the PARM library is:

```
ORDER {DSN=sched_library | DDNAME=ddname},MEMBER=table
[,JOB=jobname][,ODATE=date][,FORCE]
```

Where:

DSN=sched_library Name of the scheduling library. Either this parameter or parameter DDNAME, but not both, must be specified.

Section 3: Customizing and Administering CONTROL-M

DDNAME=ddname	Pre-allocated DD name which points to the scheduling library. Either this parameter or parameter DSN, but not both, must be specified.
MEMBER=table	Name of the scheduling table which contains the job scheduling definitions. Mandatory.
JOB={jobname *}	Name of a job scheduling definition in the scheduling table. Optional. If a job name is not specified, or if an asterisk (*) is specified, the entire table is processed.
ODATE=date	Order date of the job(s) in the table (in mmddy, ddmyy, or yymmdd format, depending on the site standard). This parameter overrides the checkpoint date (specified in DD statement DACHK). Optional. The default is the date from the checkpoint record.
FORCE	Places the job(s) order in the Active Jobs file regardless of the job's Basic Scheduling parameters. Optional.

This method provides the following advantages over the normal method:

1. Individual jobs can be specified.
2. Changes required can be made to the member in the PARM library without changing the JCL of the User Daily job.

Note that the Date Control record is referenced by DD statement DACHK.

Create User Daily Job Scheduling Definitions and Customize Table MAINDAY

The supplied sample User Daily jobs, DAILYPRD and DAILYSYS, scan the scheduling tables referenced by DD statement DAJOB. However, different authorization is granted to each of these User Daily jobs.

Use these sample User Daily jobs to create a User Daily job for each department in table MAINDAY. Assign the authorizations accordingly. Each User Daily job should scan a different set of scheduling tables than the other User Daily jobs.

It is common in many sites for the IOA administrator to create a customized User Daily job for each department and then turn the scheduling table over to the control of the department. The department can then modify the scheduling table (and job scheduling definitions) as necessary.

Although User Daily jobs can execute immediately after the jobs have been placed in the Active Jobs file, a site may choose to delay execution of a User Daily. To delay submission of a User Daily job, the User Daily's runtime scheduling criteria should be defined accordingly.

Note If User Daily jobs are executed at different times, IOALDNRS should be run after each User Daily job is run.

Additional maintenance jobs can be added to table MAINDAY as necessary.

Section 3: Customizing and Administering CONTROL-M

Decide Whether or Not Enhanced Daily Checkpointing Will Be Used

Enhanced Daily Checkpointing enables automatic rerun of the job ordering process to resume from the unsuccessfully ordered job if New Day processing terminates during job ordering (due to system crash, etc.).

The decision of whether or not to implement Enhanced Daily Checkpointing can be made separately for each User Daily job (i.e., the same decision does not have to be made for all User Daily jobs).

Implementation of Enhanced Daily Checkpointing requires definition of a second Date Control record in the Date Control Record member. (The format of this record is described below in "Create Date Control Records.") This second record stores information which identifies jobs already ordered to the Active Jobs file. This, in turn, allows New Day processing to continue ordering jobs without redundancy.

Create Date Control Records

Date Control records are members in the CONTROL-M PARM library. A Date Control record must be defined for the New Day procedure, and for each User Daily job a different Date Control record must be defined.

If Enhanced Daily Checkpointing is implemented, a second Date Control record must be defined for each User Daily job.

A User Daily Date Control record has an 80-character length. It is usually defined only once for each job, and from then on it is usually updated by the User Daily job. Format of the dates added to the record is mmddy, ddmmy or yymmdd, depending on the site standard.

The New Day procedure Date Control record differs from the User Daily Date Control record in that it has one additional date: Date7. The following chart shows the format of the Date Control record, and indicates when the User Daily adds the original scheduling date values to the fields in the record:

Column	Value added	Description
01-06	date1	User Daily adds the ODATE before the User Daily procedure is begun.
18-23	date2	User Daily adds the ODATE before the job ordering process begins for jobs being scheduled based on DATES, DAYS and/or DCAL parameters.
25-30	date3	User Daily adds the ODATE after the job ordering process ends for jobs being scheduled based on DATES, DAYS and/or DCAL parameters.
43-48	date4	User Daily adds the ODATE before the job ordering process begins for jobs being scheduled based on WDAYs and/or WCAL parameters.
50-55	date5	User Daily adds the ODATE after the job ordering process ends for jobs being scheduled based on WDAYs and/or WCAL parameters.

Section 3: Customizing and Administering CONTROL-M

60-65	blank	In the User Daily Date Control records, these columns are blank.
	(date7)	In the New Day procedure Date Control record, these columns are the last formatting date, date7, of the Active Jobs file (used by program CTMFRM). This field prevents formatting from being carried out twice on the same day. When this date is in a record, program CTMCHK recognizes the record as a New Day procedure Date Control record. If there are any problems concerning the date, the program presents the operator with a series of prompts.
67-72	date6	User Daily adds the ODATE upon completion of all processing.

The format of the second Date Control record, which is required only if Enhanced Daily Checkpointing will be used, is shown below.

Columns	Constant or Value added	
01-04	JOB=	Constant
05-12	blank	In this area, CONTROL-M stores the MEMNAME value of the last ordered job.
13-23	.SERIAL_NO=	Constant (note the period before the "S").
24-28	blank	In this area, CONTROL-M stores the internal sequence number it assigned to the last ordered job.
29-37	.ORDERID=	Constant (note the period before the "O").
38-42	blank	In this area, CONTROL-M stores the order ID of the last ordered job.
43-49	.GROUP=	Constant (note the period before the "G").
50-69	blank	In this area, CONTROL-M stores the group name of the last ordered job.

When creating this record, the user should:

- Specify the indicated constants (e.g., JOB) in the appropriate columns.
- Leave blank the columns indicated as blank. These columns are filled in by the User Daily during processing.

Ensure Subsequent Runs of Utility IOALDNRS if Necessary

If all User Daily jobs are scheduled to run in parallel, utility IOALDNRS only needs to run once, after the User Daily jobs have finished execution. If, however, the User Daily jobs are executed in groups at different times in the day, utility IOALDNRS should run after each group of User Daily jobs is executed. This can be ensured by having each group of User Daily jobs set the appropriate prerequisite conditions to ensure the execution of IOALDNRS.

Programs Called During New Day Processing

The most important programs in New Day processing are CTMILY and CTMILU.

- The New Day procedure executes program CTMILY.
- Each User Daily calls procedure CTMDAILY which executes program CTMILU.

Programs CTMILY and CTMILU both execute other programs which implement New Day processing. The programs called by CTMILY and CTMILU are listed in the table below. Both CTMILY and CTMILU read the member referenced by DD statement DAPROG and activate the programs listed in the member.

The format for each record in the program list is:

Columns

- 01-08 Program name.
- 10-11 Maximum return code allowable in the preceding program. If a higher return code is encountered in the preceding program, the current program is not executed.

The programs called by the program CTMILY (the New Day procedure) and by program CTMILU (User Daily jobs) are shown in the following chart:

Program	Called By:		Purpose
	CTMILY	CTMILU	
CTMCHK	✓	✓	Checks the current date and its relation to the Date Control record (see below). The program may communicate with the computer operator to verify that CONTROL-M is activated on the correct date.
CTMFRM	✓		<p>Reformats the CONTROL-M Active Jobs file and the IOA Conditions/Resources file:</p> <p>CONTROL-M Active Jobs File</p> <p>By default (i.e., if not SELECT/IGNORE statements are specified), the following jobs are erased from the Active Jobs file and the file is compressed:</p> <ul style="list-style-type: none"> • Jobs which have already executed and ended OK or NOTOK (regardless of parameter MAXWAIT). • Job whose MAXWAIT parameter has been exceeded. • Emergency jobs which are not needed.

Called By:			
Program	CTMILY	CTMILU	Purpose
			<p>IOA Conditions/Resources File</p> <p>This program erases all prerequisite conditions of the coming execution date.</p> <p>(Prerequisite conditions should not be kept in the IOA Conditions/Resources file for more than one year. Otherwise, jobs may be submitted for execution due to prerequisite conditions set on the same date in the preceding year.)</p> <p>At start of execution, this program creates a copy of the Active Jobs File for recovery purposes.</p>
CTMEDA	✓		Signals to the CONTROL-M monitor that the Active Jobs file was successfully formatted.
CTMJOB	✓	✓	Places job orders in the Active Jobs file according to the date in the Date Control record and the data in the scheduling table(s) supplied.
CTMPDA	✓	✓	Marks the end of the Daily run.

If CONTROL-R is installed, step DELARCH is run after the conclusion of program CTMILY. This step runs the following:

Called By:			
Program	CTMILY	CTMILU	Purpose
CTMDAS	✓		Deletes archived SYSDATA of jobs that were deleted from the Active Jobs file by program CTMFRM.

Use of the Date Control Record by User Daily Jobs

The work flow of User Daily jobs is dependent on the Date Control record. The main steps of a User Daily job are:

- Checking the last running date of the User Daily job (via internal program CTMCHK).

The first date in the Date Control record (column 1-6) is compared to the current working date (at the time of the run).

- If they match, the User Daily job has already run today. An appropriate message is issued and the condition code is set to 0004.
- If the current working date is earlier than the first date of the Date Control record, a User Daily job run has been attempted before its time. The User Daily job stops executing and notifies the user accordingly.

Section 3: Customizing and Administering CONTROL-M

- If the current working date is later than the first date of the Date Control record (the normal situation), the first date of the Date Control record (columns 1-6) is updated automatically to the current working date. This date is then used as the current scheduling date.

If the User Daily job did not run for more than one day, a warning message is issued and the User Daily job tries to schedule the jobs for all of the days which have passed since the last scheduling date (according to the production parameters).

The New Day procedure asks the operator a series of questions regarding the current system date (see "New Day Procedure Flow," below). This verifies that an incorrect date was not inadvertently entered during the IPL process.

- Placing job orders in the Active Jobs file according to the current scheduling date and the last running date (via internal program CTMJOB).

Program CTMJOB acts on the user scheduling table(s) referenced by DD statement DAJOB. For each job in the table(s), the program checks whether the job should be scheduled on one or all the days that have passed since the last original scheduling date (date3 or date5) until the working date in the record (date1). If the job should be scheduled, a job order is placed in the Active Jobs file.

For example, if a computer did not operate from the 20th to the 23rd, a job which was scheduled to run on the 20th did not run. Program CTMJOB decides whether or not the job should be retroactively scheduled to run on the logical date of the 20th. (See parameter RETRO in Section 3 of the CONTROL-M User Manual.)

When the program finishes processing the user scheduling table(s), the finish indicator dates (date3 and date5) are updated to the working date (date1) calculated by program CTMCHK.

Before program CTMJOB starts operating, it compares date2 with date3 (and date4 with date5). If they do not match, a previous run of program CTMJOB of the same User Daily job has probably abended. The user is notified and the program terminates. To correct the error, adjust the date values in the user Date Control record (using a standard editor).

Note: When manually modifying the Date Control record, make sure that jobs are not scheduled to run twice on the same day.

- Indicating that the User Daily job has ended (via program CTMPDA).

Program CTMPDA updates the finish indicator date (date6) by inserting in it the value of the running date (date1). This indicates that the User Daily job finished successfully.

- Rerunning the User Daily job after a failure.

See "Enhanced Daily Checkpointing" below.

Enhanced Daily Checkpointing

Enhanced Daily Checkpointing enables automatic rerun of the job ordering process, starting from the unsuccessfully ordered job, if the User Daily job terminates during job ordering (due to a system crash, etc.).

Before the ordering process starts, the program checks if the checkpoint fields in Record 2 are blank.

If the checkpoint fields are blank, the User Daily job continues normal processing. Before each job is ordered, the fields in Record 2 are updated (overwritten) with information identifying the current job being ordered. Only upon successful completion of the User Daily job is the information in the checkpoint fields erased.

If the checkpoint fields are not blank, the recovery procedure is activated.

Recovery Procedure

The program passes over the jobs in the input tables, internally counting the jobs and comparing the count to the value in the SERIAL_NO field, until the count and serial number match. The matching job is selected.

The program then compares the values in the JOB and GROUP fields to the values belonging to the selected job. If the fields do not match, error message CTMD67S is issued and processing terminates.

- If the fields match, the program checks the Active Jobs file for a job with an order ID matching the order ID recorded in Record 2. If the match is found, an additional check is performed to verify that the job's MEMNAME and GROUP values match the checkpoint JOB and GROUP values.
- If the Active Jobs file already contains the job, the job is not ordered again and the program switches to normal processing starting with the next job.
- If the Active Jobs file does not contain the job, the job is ordered. The program then switches to normal processing.

Note If relevant scheduling tables are modified prior to rerunning the User Daily job, the checkpointed job and serial number may not match. In this case, automatic rerun is terminated, but rerun can be initiated manually.

New Day Procedure Flow

Once a day, at a time set by the IOA administrator, the CONTROL-M monitor begins New Day processing by going into a suspended state and issuing the following messages (the first is a highlighted, unrollable message):

```
CTM113I CONTROL-M MONITOR monitor NEW DAY PROCESSING STARTED
CTML00I CONTROL-M MONITOR monitor PROCESSING SUSPENDED
CTML07W CONTROL-M MONITOR monitor WAITING FOR NEW DAY PROCEDURE
```

Shortly after that last message is issued, started task CONTDAY (the New Day procedure) is automatically activated.

- When CONTDAY finishes executing without any problems, the following messages are issued, and the CONTROL-M monitor ends its suspended state and resumes processing:

```
CTML01I CONTROL-M MONITOR monitor PROCESSING RESUMED
CTML02I CONTROL-M MONITOR monitor NEW DAY PROCESSING COMPLETE
```

- If a problem occurs during CONTDAY processing, the CONTROL-M monitor prompts the operator for an appropriate response via the following messages:

```
CTML05W NEW DAY PROCESSING ERROR DETECTED
CTML06W REPLY "R" FOR RETRY OR "E" FOR END
```

Section 3: Customizing and Administering CONTROL-M

The operator should try to correct the problem and request that CONTROL-M retry New Day processing. Halting New Day processing (option E) should only be requested if the problem cannot be corrected.

- During New Day processing, CONTDAY checks the system date and time against what it expects to find in the CONTROL-M control files. If for some reason they do not match, the operator is prompted with the following messages:

```
CTM426W GENERAL "DAILY" DID NOT RUN FOR nnnnnn DAYS
```

```
CTM427W IS IT TRUE? (ANSWER "YES" OR "NO")
```

```
CTM428W YOUR ANSWER IS:
```

This situation may occur after the computer has not been working for a few days (e.g., a hardware failure or holiday); your response should be YES. If this is not the case, it is probable that the computer was IPLed with the wrong date. Your response should be NO, and the date on the computer should be checked and corrected. Then start CONTDAY again. If the date on the computer is correct, the IOA administrator should be called to check the cause of the problem.

Notes To rerun User daily jobs from the beginning (after a failure) all dates (1-6) must be restored to their previous value before attempting the rerun.

If the CONTROL-M monitor has been down for more than 28 days, the previous working date (i.e., the current working date minus 1) should be manually specified as values 1-6.

Activation of Started Tasks

CONTROL-M can activate started tasks as well as jobs. The JES2/JES3 definitions required to support started tasks are described in the CONTROL-M section of the IOA Installation Guide.

When working in a multi-CPU environment, CONTROL-M can also activate started tasks in CPUs other than the one in which the CONTROL-M monitor is active.

Under JES2, the CONTROL-M monitor activates started tasks in other CPUs by using command \$Mm, where m is the appropriate system ID. This system ID is defined in the JES2 initialization check as follows:

```
MASDEF SID(n)=cccc
```

or

```
sn SID=cccc (under older releases)
```

For further details, refer to IBM's publication "JES2 Initialization and Tuning" (SC23-0046 or equivalent).

Note that JES2 will fail a \$Mm command if m is the ID of the system ID in which the CONTROL-M monitor itself is working. Therefore, when CONTROL-M is ordered to activate a started task in a specific system, it checks whether a \$Mm command or a regular MVS START command should be issued. To ensure that this check is performed correctly, you must define in the CPUS installation parameter (in CTMPARM) all the CPUs in your computer complex.

Under JES3, the CONTROL-M monitor activates started tasks in other CPUs by issuing a `*Tcccc' JES3 command, where "*" is the JES3 command prefix, and cccc is the required system ID. This system ID is defined in the JES3 initialization deck (INISHDECK) as follows:

```
MAINPROC,NAME=cccc,SYSTEM=JES3,...
```

CONTROL-M Monitor and JES Considerations

Cases of JES Malfunction

The CONTROL-M monitor uses JES services to receive information about the status of the jobs running in the system. If CONTROL-M detects a critical error in JES operation, it shuts itself down. This prevents the incorrect submission of jobs due to a JES malfunction. One of the following highlighted, unrollable messages is displayed on the operator console:

```
CTM168S CONTROL-M SHUTTING DOWN - COMMUNICATION TO "JES" NOT AVAILABLE
CTM256S CONTROL-M SHUTTING DOWN - COMMUNICATION TO "JES" NOT AVAILABLE
```

Special Considerations

To analyze how a job finished executing, CONTROL-M uses JES services to read the job's output. Therefore, the following limitations should be considered:

- Jobs submitted by CONTROL-M can be canceled by the operator. It is important, however, not to purge their outputs (i.e., JES commands \$PJnnn, \$CJnnn, P and other similar commands must not be used).
- Job output for jobs submitted by CONTROL-M should not be released for printing other than by CONTROL-M (i.e., MVS JES2 command \$TO, MSP JES command \$0, or other similar commands should not be activated on the job's output).
- If JES operator command \$HJ is issued for the job, the job must be released from held status before CONTROL-M can read the job's output. Otherwise, the job status is changed to EXECUTING (SYSOUT IN HOLD STATUS).

Whenever the CONTROL-M Monitor does not succeed in reading job sysout, the following rollable message appears on the operator console:

```
CTM262W  n UNSUCCESSFUL ATTEMPTS TO READ JOB DATA BY SUBSYSTEM REQUEST. RETRY
CONTINUES
```

Message CTM262W does not necessarily indicate a serious problem.

Examples

- When a job is not run due to a JCL error, only two sysout datasets exist for the job. Therefore CONTROL-M cannot read the third sysout dataset, and the above message is displayed.
- When JES is very busy, a period of up to a minute (in extreme cases) may pass between the time the job has finished executing and the time JES enables CONTROL-M to read its sysout (i.e., JES is "stuck" in the Output Processing stage).

Section 3: Customizing and Administering CONTROL-M

By default, CTM262W is displayed every 5 times the CONTROL-M monitor attempts to read the job sysout and does not succeed. If after 20 attempts the CONTROL-M monitor still does not succeed in reading the sysout, the following message is displayed:

```
CTMD50S  READING JOB DATA BY SUBSYSTEM REQUEST FAILED AFTER n ATTEMPTS. LAST  
RC code FILE n jobname/jobid
```

The above two default values can be changed using installation defaults.

On the other hand, message CTM262W can indicate serious problems with the job's sysout. Below is a list of typical problems which can cause this message to appear.

- When a job's output has been released for print (i.e., no longer in held output), the job's output should be printed or purged.
- In a multi-computer environment the following chain of events may occur:
 - CONTROL-M monitor submits the job from computer A
 - Computer A crashes (or is shut down)
 - CONTROL-M monitor is activated on computer B and the job executes in computer B. When the job finishes executing CONTROL-M cannot read the job's output and message CTM262W appears.

This is caused by the job waiting to be "handled" by the JES of computer A.

This problem can be overcome by assigning the job to computer B (using the JES command: \$TJnn,S=sysid). CONTROL-M then reads the output, and the message disappears from the operator console.

Message CTM262W Summary

Whenever message CTM262W appears, wait a minute or two. If the message continues to appear every few seconds for the same job, try to resolve the problem, by following the following steps:

Note *To stop the message from appearing on the operator console while you are checking the problem, hold the job order in the CONTROL-M Job Status screen (screen 3). Remember to release it once you have resolved the problem.*

- Issue the JES2 command \$DJnn, and scan the results.
- Check whether or not the job's output is in held class (the job waits for print). If yes, the CONTROL-M monitor cannot analyze the output, so you must analyze it manually. Print or purge the output of the job. Make sure that the job order in CONTROL-M is not HELD. Wait for a minute until the status of the job changes to DISAPPEARED. Do not forget to ADD/DELETE conditions according to the result of the run. Do this manually in the IOA Conditions/Resources screen (Screen 4).
- If the job is "waiting for output processing," check whether or not the job (not the output) is held (by a previously issued \$HJ command). If the job is held, release it using the JES2 command \$AJnn.
- If the job is "waiting for output processing" by a system ID which is currently not active, try to resolve the problem using the \$TJnn command method described above.

Considerations for Changing Clocks

Daylight savings time considerations are described in the following paragraphs. All examples assume 02:00 a.m. as the time of change.

Advancing the Clock Forward

The CONTROL-M monitor should not be brought down when the clock is being advanced. This applies to computers that are capable of changing the clock without restarting the system.

New Day Procedure

No special action should be taken once the clock is advanced.

- If the New Day procedure starts before you reset the clock, the New Day procedure will start working before the clock is advanced, and will continue normally (even if the clock is advanced while the New Day procedure is in process).
- The same consideration applies if the New Day procedure is scheduled to begin at exactly 2:00 a.m. It is possible that the New Day procedure will start execution before the clock is manually changed. Otherwise, changing the clock will initiate New Day processing.
- If the New Day procedure is to begin between 2:00 a.m. and 3:00 a.m., once the computer clock is advanced the monitor will start the normal New Day processing.
- If the New Day procedure will begin after 3:00 a.m., no action is required. The monitor will start the standard New Day procedure.

Time-Dependent Shouts

- Shouts scheduled before 2:00 a.m. do not require any action.
- Shouts scheduled at the time between 2:00 a.m. and 3:00 a.m. will be issued, even though there might not be a real delay in production, since the time frame for the production is smaller.
- This also applies to jobs which have a shout scheduled at a later time (e.g., 6:00 a.m.). These jobs may be considered late jobs because of the tighter production time frame.

Time-Dependent Schedules (FROM UNTIL)

Jobs whose scheduled time overlaps the time gap created by the clock shift may need manual intervention. For example, it is possible that a job with a FROM value of 2:15 a.m., and an UNTIL value of 2:45 a.m. might not be submitted at all. These jobs should be manually adjusted.

IOA Log File

Just like the system log, the IOA Log file will not contain entries with a timestamp between 2:00 a.m. and 3:00 a.m. Any KSL scripts and/or programs that rely on the time of the log entry should be checked for a possible discrepancy as a result of advancing the clock.

Moving the Clock Backwards

The following examples assume that the clock is moved back at 2:00 a.m., i.e., 2:00 a.m. becomes 1:00 a.m.

New Day Procedure

- If the installation New Day time is before 1:00 a.m., then no special action should be taken. The New Day procedure will run only once (between 00:00 and 00:59).
- If the newday time is exactly at 1:00 a.m., then the computer time should not be turned back to 1:00 a.m., as this will initiate another newday process, and a second newday process requires a manual intervention. It is advisable to wait until 2:01 a.m., for example, and then turn the clock back to 1:01 a.m.
- If the New Day time is between 1:00 a.m. and 2:00 a.m., do one of the following:
 1. Wait until at least a full hour has elapsed from the daily run, and then turn the clock back to a time which is past New Day time.
 2. Update the clock before New Day processing begins.

For example, if the New Day time is 1:45 a.m., the clock should be moved back one hour no later than 1:44 a.m. If this was not done by 1:44 a.m., then the user should wait until 2:46 a.m. and then shift the time back.

- If New Day time is after 2:00 a.m., no special action should be taken.

Time-Dependent Shouts

Certain shouts scheduled between 1:00 a.m. and 2:00 a.m., may be issued twice.

Time-Dependent Schedules (FROM-UNTIL)

No special action should be taken for jobs with FROM-UNTIL schedule. Jobs scheduled to start at a time between 1:00 a.m. and 2:00 a.m. will start at the first occurrence of that hour (provided that other conditions, such as input conditions, resources, etc. are met). However, they may be restarted once the clock is moved back.

IOA Log File

The IOA Log file will contain entries with a time which is earlier than previous entries, due to the time shift. The same considerations which apply to forward clock shift should be applied to the IOA Log file.

Sending a Job's SYSDATA to a Held Output Class

CONTROL-M analyzes job execution results by reading the job's sysout. Therefore, the SYSDATA² portion of the job must be sent to a held output class to prevent it from being printed accidentally before the CONTROL-M monitor manages to read it.

HLDCLAS is the automatic held output class to which CONTROL-M sends the job's SYSDATA output.

When a job is submitted by CONTROL-M, CONTROL-M either adds a MSGCLASS parameter to the job card or it overrides the existing MSGCLASS parameter. The sysout is directed to the class designated in the HLDCLAS installation parameter. It is required that a regular automatically held output class be specified during installation.

Special precautions should be taken so that the job's sysout is not accidentally purged or released for printing. In some installations, all jobs in the automatic held class are purged by an operator command at least once a day. To prevent deletion before CONTROL-M reads the job's sysout, the following procedure can be used:

- A special automatically held output class is dedicated to CONTROL-M.
- The sysouts in this class are protected from any change (e.g., by SDSF exits).

This procedure ensures that CONTROL-M will be able to read the sysout.

If other users are to access the sysout, the following sample parameters should be defined in any job production parameters definition:

```
ON PGMSTEP ANYSTEP   CODES *****  
DO SYSOUT   OPT C   PRM A   FRM X
```

When CONTROL-M finishes reading the job's sysout, it transfers the sysout to the designated output class where any user can access it.

Special care should be taken when a system programmer formats the JES queue. This procedure usually involves unloading the output queues to a tape, formatting the spool, and then reloading the queues. The monitor should be shut down before the queue is unloaded to a tape, and brought up again only after the queues have been loaded from the tape back to the spool. If this procedure is not followed, CONTROL-M can assume that the job is lost (purged by an operator) and act accordingly.

When you change the HLDCLAS parameter after CONTROL-M installation, you should also change the class in DD statement INTRDR of the CONTROL-M monitor, and of JCL procedures CTMAESIM and CTMTAPUL.

² SYSDATA is an IOA term used to designate the data in the following job sysout datasets:

1. The job log (console messages)
2. The expanded JCL
3. The system output messages

Dual Checkpoint Mode

The CONTROL-M monitor can work in dual checkpoint mode. Under this mode, the CONTROL-M monitor maintains duplicate copies (mirror files) of the CONTROL-M Active Jobs file and of the IOA Conditions/Resources file. In case of a disk crash which makes the regular files inaccessible, these files can immediately be restored from the mirror files (assuming, of course, that the mirror files are located on a different disk).

The Active Jobs Mirror file is kept synchronized with the main Active Jobs file by the CONTROL-M monitor. The Mirror file should be protected from any update from another source. If either of the two files is damaged during CONTROL-M operation, the CONTROL-M monitor will shut down with an appropriate message. To resume CONTROL-M monitor operation, the damaged file must be restored.

Restoring the Active Jobs file involves the following stages:

- Allocate and format a new Active Jobs file (via utility CTMFRCKP).
- Run utility CTMCAJF. Use the undamaged Active Jobs file as input and the newly created one as output.
- Rename the damaged file and rename the newly created Active Jobs file with the original file name (the Main or the Mirror).
- Restart the CONTROL-M monitor.

Restoring the Conditions/Resources file involves the following stages:

- Allocate and format a new Conditions/Resources file (via utility IOAFRRES).
- Run utility IOACOPRS. Use the undamaged Conditions/Resources file as input and the newly created one as output.
- Rename the damaged file and rename the newly created Conditions/Resources file (RES and SNC) with the original file name (the Main or the Mirror).
- Start the CONTROL-M monitor again.

Managing the CMEM Facility

The CONTROL-M Event manager (CMEM) is designed to handle events occurring outside the control of the CONTROL-M monitor. CMEM consists of a monitor which uses the IOA subsystem, and performs predefined actions in response to system events (e.g., arrival of a specified job on the job spool).

Note When CONTROL-O is installed at your site, the CONTROL-O monitor assumes control of the CMEM facility. Therefore, this description of the CMEM facility is not relevant for your site. Instead, refer to Section 5 in this guide for information on how to manage the CMEM facility at your site.

Automatic Loading of Rules

When the CMEM facility is started (and it is not replacing an active CMEM monitor) it loads CMEM rule tables according to a list referenced by DD statement DACTMLST. This list of CMEM rule tables is called the CMEM list. A default list is located in member IOACMEML in the IOA PARM library. The CMEM list is a partitioned dataset (PDS) containing the names of the tables to be ordered. The default list can be overridden by specifying the ORDER parameter in command S CTMCMEM, which references a different CMEM list.

Each line in the CMEM List has the following format:

```
*      library table
```

Where:

```
*                Must be included as a constant.
library          Rule library name.
table           Rule table name (or mask).
```

CMEM loads the specified rule table from the specified library for each line in the CMEM list.

Manual Loading of Rules

The CMEM list specified during startup contains a list of rule tables to be activated by CMEM when it is started.

To load additional tables, or to replace a currently active table with a new (updated) copy of the rules in the table, use one of the following options:

Option 1

Enter the CMEM Online facility (=C), and use the FORCE option in the Table List screen.

Option 2

Issue the following operator command:

```
F CTMCMEM,C=library(table)
```

Where:

```
C=                Load a CMEM rule. Each rule is loaded by the CMEM monitor and is
                  activated.
library          Rule library name.
table           Rule table name (or mask).
```

Examples

```
F CTMCMEM,C=CTM.PROD.RULES(DATASET)
```

Loads table DATASET from CTM.PROD.RULES

```
F CTMCMEM,C=CTM.PROD.RULES(*)
```

Loads all tables from CTM.PROD.RULES

```
F CTMCMEM,C=CTM.PROD.RULES(PROD*)
```

Loads tables starting with PROD from CTM.PROD.RULES

Section 3: Customizing and Administering CONTROL-M

Replacing All CMEM Rule Tables in One CPU

To replace all loaded CMEM tables with those in the CMEM list (specified in DD statement DACTMLST) use the following operator command:

```
F CTMCMEM,C=ALL[,REBUILD]
```

If the REBUILD option is specified, CMEM rule tables not listed in the CMEM list are deleted.

If the REBUILD option is not specified, previously loaded CMEM rule tables are replaced by a new copy of the rule table, and unchanged tables are left intact.

Replacing All CMEM Rule Tables in All CPUs

All CMEM rules in all CPUs where the CMEM monitor is active can be reloaded at the same time. The reload process is performed in the same way as the automatic loading is performed during startup of the CMEM monitor. All active rules are deleted, and all rule tables specified in the CMEM list referenced by DD statement DACTMLST are loaded.

To replace all rules in all CPUs issue the following command:

```
F CONTROLM,NEWCONLIST
```

Specifying this command is the same as specifying F CTMCMEM,C=ALL,REBUILD in all CPUs.

CONTROL-M informs the CMEM monitor running in each CPU about this command request.

Rule tables that were manually loaded and/or do not appear in the CMEM list are deleted during execution of the above operator command.

Deleting (Deactivating) an Active Rule Table

An active CMEM rule table can be manually deactivated via the following operator command:

```
F CTMCMEM,D=library(table)
```

Where:

D= Deactivate a CMEM rule. Each rule is deactivated by the CMEM monitor.

library Rule library name.

table Rule table name (or mask).

Example

```
F CTMCMEM,D=CTM.PROD.RULES(PRODTAB1)
```

Displaying Active Rules

A list of the active CMEM rules (up to a maximum of 1000 rules) can be displayed on the operator console. To display the list enter operator command:

F CTMCMEM, DISPLAY

This command displays a list of all active rules in the CMEM facility with the following information:

Rule	Rule name (i.e., the name which appears in the first ON statement in the rule definition).
Type	Rule type. Valid types are: R JOBARRIVAL X JOBEND D DSNEVENT Z STEP
Table	Name of the table containing the rule.
Status	Rule status. Valid status is ACTIVE.
Library	Name of the library containing the rule member.
Priority	Internal CMEM rule scanning priority.

Controlling CMEM Rule Operation Mode

The mode of operation (i.e., the trace mode) for a rule is determined by parameter MODE in its rule definition. Sometimes it is useful to override the operation mode of all active rules and verify that events and actions are recorded in a particular way. For example:

- Ensure a full trace of all rules (i.e., all events and actions are recorded) to facilitate analysis of the interaction between rules.
- Record (trace) only the triggering of every rule.

Global trace operations are requested via operator commands, as follows:

1. Activate a full trace:

F CTMCMEM, LOG=ALL

All rules are fully traced as if they were defined with mode LOG. This operator command should only be used temporarily for specific tests because extended use of full trace mode can adversely affect CMEM performance.

2. Trace rule triggering only:

F CTMCMEM, LOG=TRIGGER

Only rule triggering is traced for all rules. However, rules defined with mode LOG are fully recorded.

3. Restore the default operation mode (as defined in the rule definition) for each rule:

F CTMCMEM, LOG=DEFAULT

Modifying the CMEM Sleeping Interval

CMEM “wakes up” every few seconds. This time interval is defined in the CONTROL-M installation parameters and can be changed by the system administrator. In addition, the interval can be modified by the operator command:

```
F CTMCMEM, INTERVAL=xx
```

Where xx represents the interval in seconds.

When the modification is accepted by CMEM, the following message appears on the operator console:

```
CME123I CMEM INTERVAL IS SET TO nn SECONDS
```

Refreshing the CMEM Security Cache

CMEM security modules use a security block to identify each user for which an authority check is performed. The first time a user’s security authorization is checked CMEM creates a security block for that user. The security block can then optionally be saved for the next time the user’s security authorization is checked.

Security blocks saved for subsequent checks are kept in the CMEM Security Cache.

The CMEM Security Cache holds security blocks for the last 30 users to have their security authorization checked.

Changes made to a user’s security authorization (since the last time that user’s security block was created) are not automatically included in the information in the user’s security block in the CMEM Security Cache. However if a user’s security authorization has been changed, and there is no security block in the CMEM Security Cache for that user, changes made to the user’s security authorization will be in effect the next time that user’s security authorization is checked.

To immediately include new user authorization information in the CMEM security cache, refresh the security cache via the following operator command:

```
F CTMCMEM, NEWSECDEF
```

This command refreshes all user authorization information in the CMEM security cache.

When the modification is accepted, the following message appears on the operator console:

```
CME251I RUNTIME SECURITY REFRESH ENDED OK
```

Problem Determination

CMEM is supplied with internal debugging facilities: The ability to print an internal debugging trace and the ability to print the contents of the CMEM internal data areas. Under normal circumstances, the debugging facilities are dormant. However, if required (i.e., your IOA representative has requested debugging information), it is possible to activate the debugging facilities as follows:

Activate the CMEM internal trace:

1. Perform either step A or B below.

Section 3: Customizing and Administering CONTROL-M

- A. Start a new CMEM monitor with the following operator command:

```
S CTMCMEM,DEBUG=xxx
```

The current CMEM monitor passes control to the new CMEM monitor and shuts down.

or

- B. Issue the following operator command:

```
F CTMCMEM,DEBUG=level
```

The required debugging level will be supplied by your IOA representative. It can be any value from 00 to 255. (00 specifies no debugging.)

Note It is highly recommended not to activate CMEM with the DEBUG parameter on a regular basis because in case of a JES problem, CMEM may get hung up waiting for JES.

2. The debugging information is printed to DD statements PRTDBG and DADUMP of the CMEM procedure.
3. When you have finished your problem determination procedures, start a new CMEM using the following operator command:

```
S CTMCMEM
```

or specify the operator command:

```
F CTMCMEM,DEBUG=00
```

Print CMEM Internal Data Areas

To print CMEM internal data areas, issue the following operator command:

```
F CTMCMEM,SNAP[=name1,name2 . . . . .,namen]
```

Where:

name1, name2,.... namen are the names of the CMEM internal data areas.

When no name is specified, all data areas are printed. Your IOA representative can provide you with the list of the data area names and can specify which areas should be printed depending on the problem encountered.

Valid values are:

ALL	LINK	MTOPLB	OMT	RQCEXO	RQC	SSVT
ALO	MAIN	MTOPND	OPR	RQCFREE	RQH	STO
CAS	MCT	MTOPNX	PARM	RQCMTO	RULES	SWT
CONLIST	MTO	MTOSRV	PND	RQCRFR	SEC	UCM
CONS	MTOINX	MTOSRVA	RFR	RQCSLO	SLO	VARs
CONSOLE	MTOLNK	MTOWSC	RQCALO	RQCSRV	SRV	WISHES
DLY	MTOMIX	MVS	RQCDLY	RQCSTO	SSCT	WSC
EXO	MTOMPT					

Section 3: Customizing and Administering CONTROL-M

When the snap is completed, the following message appears on the console:

```
CME150I SNAP COMMAND WAS PERFORMED SNAPID=XXXX
```

Where:

XXXX is the snap identifying number which appears on the lower right of the screen after the snap is completed.

Displaying Internal Resource Utilization

To obtain information on internal resource utilization, issue the following operator command:

```
F CTMCMEM, WATERMARKS
```

The response to this command is displayed as message CME15DI. See the IOA Messages and Codes manual for an explanation of this message.

Inter-System Communications Facility

The CONTROL-M Inter-System Communications facility can be used in multi-CPU sites which run the following environment:

- CPUs which do not employ a shared Spool (i.e., each CPU runs its own CONTROL-M monitor)
- CPUs which employ a shared DASD, with a global ENQ manager (such as GRS, SDSI, MSX, MIM, etc.).

This facility enables communication between multiple IOA monitors in different computers, providing there is a shared disk between them.

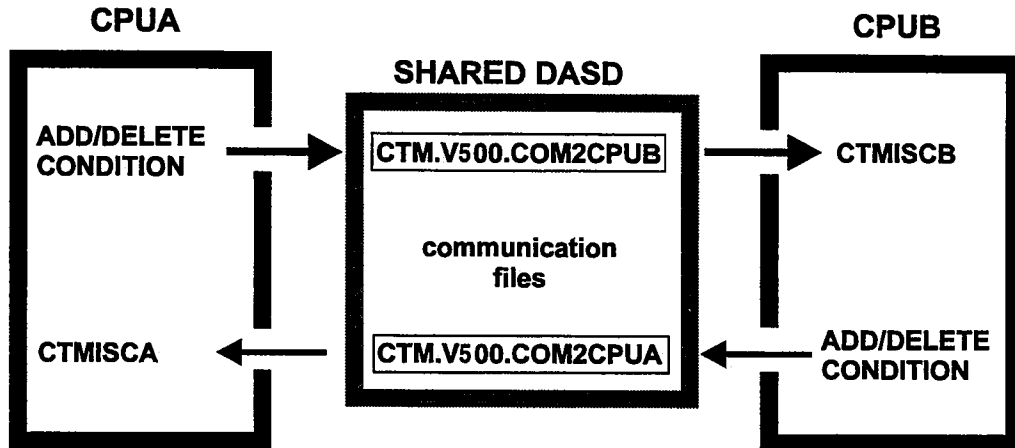
Method of Operation

When a prerequisite condition is added to or deleted from the Conditions/Resources file, a request to add/delete that condition is also written to an Inter-System Communications file (a sequential file residing on a shared disk).

The CONTROL-M Inter-System Communications monitor (residing in the other CPU) reads the conditions from the Communications file and updates the Conditions/Resources file of the other computer. After the Conditions/Resources file is updated, the Inter-System Communications file is emptied and is ready to receive new conditions.

An Inter-System Communications monitor should reside in every CPU which is to be connected. A Communications file must be created for each communication direction.

In the example below, conditions from CPUA are written to CPUB via Communications file CTM.V500.COM2CPUB. Communication is also available in the reverse direction via the second Communications file CTM.V500.COM2CPUA.



To write requests to the Communications file, additional program code must be applied to exit IOAX007. A routine invoked by exit IOAX007 receives control when adding or deleting a condition, and writes the condition name to a sequential file residing on a shared DASD. Not all conditions need to be communicated between computers. Therefore, the routine uses a condition selection list which is created by the user (described below). Only conditions which fit the selection list are passed to the communications file.

Create the Inter-System Communication Files

Edit member ISCALLOC in the IOA INSTCTM library. Correct the job card, dataset names, etc., submit the job, and save the member. The job allocates two sequential datasets (one for each CPU).

Note The files should be allocated on shared DASD that is accessible for update from both CPUs.

Prepare the Condition Selection List

Create a member named ISCPARM in the CONTROL-M PARM library. The member should contain a list of prerequisite conditions which should be communicated (passed) between the computers. The condition name can contain the following special characters:

* can be used to substitute a string of any length.

? can be used to substitute for any one character.

Example

```

COND=PAYROLL
COND=J * ENDED
  
```

Section 3: Customizing and Administering CONTROL-M

The following is passed:

- Condition PAYROLL
- All conditions whose names start with J and end with ENDED

Modify JCL Procedures and CLISTs

The following DDNAMEs should be allocated in both computers according to the following rules:

For all procedures/CLISTs running in CPUA:

```
//DAISCOUT DD DISP=SHR,DSN=prefix.version.COM2CPUB
//DAISCPRM DD DISP=SHR,DSN=prefix.version.PARM(ISCPRM)
```

For all procedures/CLISTs running in CPUB:

```
//DAISCOUT DD DISP=SHR,DSN=prefix.version.COM2CPUB
//DAISCPRM DD DISP=SHR,DSN=prefix.version.PARM(ISCPRM)
```

Add two new DDNAMEs (described above) to the following JCL procedures:

CONTROLM	The CONTROL-M monitor
IOAOMON1	The Online monitor(s)
IOACND	The IOACND Utility
IOARKSLx	For all KeyStroke Language procedures
Logon Procedures	For TSO and ISPF users

Add two new DDNAMEs (described above) to the following CLISTs:

IOACND	CTMCND Utility under TSO
IOACCND	CTMCND Utility under TSO/ISPF
IOAONL	Main IOA Online facility

Copy JCL to Procedures Library

Edit members CTMISCA and CTMISCB in the IOA PROCLIB library, adapt them to your data center naming conventions, copy them to your procedures library, and save the members.

CTMISCA should be copied to CPUA. CTMISCB should be copied to CPUB.

The PARM value in members CTMISCA and CTMISCB specifies the interval (in seconds) after which the Inter-System Communications monitor should wake up to receive prerequisite conditions from the second CPU.

Verify that the CTMISCA procedure (running in CPUA) has the communications file (for conditions coming from CPUB) defined to DDNAME=DAISCIN1.

```
//DAISCIN1 DD DISP=SHR,DSN=prefix.version.COM2CPUB
```

Verify that the CTMISCB procedure (running in CPUB) has the communications file (for conditions coming from CPUA) defined to DDNAME=DAISCIN1.

```
//DAISCIN1 DD DISP=SHR,DSN=prefix.version.COM2CPUB
```

The access to the communications files from both CPUs is serialized by the ENQ mechanism with SCOPE=SYSTEMS.

The QNAME is CTM.

The RNAME is the file name allocated to DAISCINn or DAISCOUT.

Verify that the above names are included in the list of QNAMES/RNAMES managed by your global resource serialization product (GRS, SDSI, MSX, MIM, etc.).

Customize the Exits

Member IOAX007I in the IOA SECUDATA library is a sample of IOA exit IOAX007 which invokes the CTMISR routine (the routine used to write the conditions to the communication file). If you do not use exit IOAX007 at your site (i.e., the default dummy exit is used), you can use the provided IOAX007 sample as is. If you already use exit IOAX007, you can use the sample exit as an example of how to include a call to CTMISR in the exit.

Modify your IOAX007 exit or use the provided sample. Assemble and link-edit the exit. If different IOA Load libraries are used in each computer, the assemble and link-edit should also be performed in the second CPU.

Start the Communication Monitors

Start the CTMISCA communication monitor in CPUA and the CTMISCB communication monitor in CPUB. Check that they are up and running, and that there are no error messages.

Shut down and start the CONTROL-M monitor, the IOA Online monitor, and TSO sessions.

Enter option 4 of the IOA Online facility. Add a condition which should be communicated between the computers. Verify that it was added in the other computer's IOA Conditions/Resources file.

Expanding CONTROL-M Files

Follow the steps below to expand CONTROL-M files.

Expanding the Active Jobs File (CKP)

1. Shut the monitor. Specify the following operator command:

```
P CONTROLM
```
2. Rename the old CKP and BKP files to have a different name.

Section 3: Customizing and Administering CONTROL-M

3. Select step "Customize Dataset Parameters" from the CUSTOMIZE activity in ICE. Change the value of parameter CKPSIZE to the required value. Complete the remainder of the minor steps for this step to update and compile member CTMPARM, and customize the formatting jobs.
4. Edit member FORMCKP in the IOA INSTCTM library. Check CKPSIZE, and submit the job. This job will allocate and format a new set of CKP or BKP files with the new size.
5. Run the CTMCAJF utility with the COPY parameter to copy the old CKP file into the new one.
6. Start the monitor. Specify the following operator command:

```
S CONTROLM
```

Expanding the CONTROL-M Jobs Dependency Network File (GRF)

1. Close the monitor using the following operator command:
- ```
P CONTROLM
```
2. Rename the old GRF file to have a different name.
  3. Edit member FORMGRF on the IOA INSTCTM library, change parameter SIZE to the new requested size and submit the job for run. This job will allocate and format a new GRF file.
  4. Bring the monitor up using the command:

```
S CONTROLM
```

5. Once the monitor is up issue a modify command to re-calculate files dependency:

```
F CONTROLM,NET
```

### Expanding the CONTROL-M Statistics File (STAT)

1. Shut down the monitor. Specify the following operator command:
- ```
P CONTROLM
```
2. Rename the old STAT file to a different name.
 3. Edit member DEFSTAT in the IOA INSTCTM library. Change parameter cylinders to the new requested size CI size.
 4. Submit job FORMSTT in the INSTCTM library.
 5. Copy the old STAT file into the new file using IDCAMS REPRO.
 6. Start the monitor. Specify the following command:

```
S CONTROLM
```

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

General	4-1
Activating the CONTROL-D Monitor	4-1
Activating Generic Processing	4-1
Activating the Compressed Dataset Access Method	4-2
Modifying the CONTROL-D Sleeping Interval	4-2
Reloading the Recipient Tree	4-3
Loading the Recipient Tree Into the IOA Online Monitor	4-3
Reloading the Manual Conditions File	4-3
Activating the IOA Archive Server (CONTROL-V)	4-4
Deactivating the CONTROL-D Monitor	4-4
Deactivating Generic Processing	4-4
Automatic Warning	4-5
Deactivating the Compressed Dataset Access Method	4-5
Deactivating the IOA Archive Server (CONTROL-V)	4-5
New Day Processing	4-6
Overview	4-6
Starting the New Day Procedure	4-6
New Day Procedure - Workflow	4-7
User Daily Job	4-7
Date Control Record	4-7
Use of the Date Control Record by the User Daily Job	4-8
Use of the Date Control Record by the New Day Procedure	4-10
Programs Called During New Day Processing	4-10
Parameters of the New Day Procedure	4-12
Mission Scheduling	4-13
Overview	4-13
Scheduling - General	4-13
Using the New Day Procedure (CTDNDAY) to Schedule Missions	4-14
Supplied Mission List Members	4-15
Using a User Daily Procedure to Schedule Missions	4-16
Scheduling a Mission Manually	4-16
Scheduling Missions -Workflow	4-17
Decollation Mission Management	4-19
Overview	4-19
Generic Decollating Missions	4-19

Generic Decollating Mission Workflow	4-20
Additional Considerations	4-21
Scheduling Generic Decollating Missions Via the New Day Procedure	4-21
Controlling the Generic Process	4-22
Defining a Generic User Name List.....	4-22
Interfaces to Production Control Systems	4-23
Overview of CONTROL-M Scheduling With CONTROL-D	4-23
Job-Report Dependency	4-25
Scheduling via the CONTROL-M Production Control System	4-26
Scheduling via a Non-CONTROL-M Production Control System.....	4-27
Considerations for When CONTROL-M and CONTROL-D Are Installed	4-29
Printing Mission Management	4-30
Printing Mission Execution Workflow	4-31
Advanced Scheduling Issues.....	4-34
Distribution According to Scheduling Dates	4-34
Report Decollating and Printing Mission Dependency	4-34
Printer Control	4-34
One-Chunk Method.....	4-35
Multi-Chunk Method	4-35
One-Outgroup Method (JES2 Only).....	4-36
Opening/Closing Printers	4-36
Printing Process	4-37
Identifying CONTROL-D Chunks on Spool (JES2 Only)	4-38
Printing on AFP (APA) Printers.....	4-38
Using the CDAM PAGEDEF/FORMDEF Parameters	4-39
Using a Specified OUTPUT Statement	4-39
AFP Page Mode Output	4-39
Page Markers Under AFP.....	4-40
In-Stream AFP Control Statements.....	4-40
Printing Using XEROX LCDS (DJDE) Parameters.....	4-42
Advanced ACIF Interface Facility.....	4-43
What is AFPDS?	4-43
What is ACIF?	4-43
Using OUTPARM for Global Control of Printing Characteristics	4-47
Printing to a File	4-49
Tailoring CTDX005.....	4-49
Backup Mission Management.....	4-51
Overview	4-51
Advanced Scheduling Issues.....	4-51

Backup According to Scheduling Dates	4-51
Report Decollating and Backup Mission Dependency	4-52
Backup Mission Workflow	4-53
Exception Handling	4-55
Changing Backup Mission Retention Period.....	4-55
Migration Mission Management (CONTROL-V).....	4-55
Overview.....	4-55
Advanced Scheduling Issues	4-56
Migration Mission Workflow	4-56
Dual Migration	4-58
Migration Media Definition.....	4-58
Cartridge (CART) Media Management.....	4-59
Migration Mission Skeletons	4-59
Migration to OSS.....	4-59
Migration to Cartridge.....	4-60
Migration to Disk, Fat-DASD, IBM 3995 Optical Library Dataserver.....	4-60
Migration to FileTek Storage Machine	4-60
Exception Handling	4-61
Restore Mission Management	4-61
Overview	4-61
Advanced Scheduling Issues	4-61
Restoring According to Scheduling Dates	4-61
Restore Mission Workflow	4-62
Exception Handling	4-64
Restoring With the Original Backup Utility	4-64
IOA Archive Server	4-65
General	4-65
Device Definition and Usage	4-65
Cartridge (CART) Media Definition.....	4-65
KODAK/DataWare OSS Usage	4-66
FileTek Storage Machine Media Definition	4-66
Media and Resource Control	4-67
Media Control.....	4-68
Resource Control.....	4-68
Displaying Media Information	4-71
Problem Determination.....	4-72
User Report List File Management	4-72
Overview	4-72
Permanent User Report List File.....	4-74

Active User Report List File	4-75
Migrated User Report List File	4-76
History User Report List File	4-77
Summary	4-78
Repository Maintenance	4-79
General	4-79
Expanding the Active Missions File	4-79
Expanding the Active Transfer File	4-80
User Reports List File Housekeeping	4-80
Reorganizing the Files	4-80
Rebuilding the Index Component	4-81
Recovering a Damaged File	4-82
SMF Accounting	4-82

General

This section describes the initialization and customization features that are available for CONTROL-D and CONTROL-V.

CONTROL-D and CONTROL-V share the same customization features, except the IOA Archive Server (which is a CONTROL-V feature), and the migration mission (which is a CONTROL-V feature). Whenever the text refers to CONTROL-D, the same customization methods apply to CONTROL-V.

Activating the CONTROL-D Monitor

CONTROL-D and CONTROL-V share the same monitor when both products are installed.

The CONTROL-D monitor usually operates 24 hours a day as a started task (STC) in one and only one computer at an installation. Generally, the monitor is automatically activated as part of the IPL process. To activate the monitor manually, specify the operator command:

S CONTROLD

If the monitor is successfully activated, the following message appears on the operator console:

CTD100I CONTROL-D MONITOR STARTED

If you try to activate more than one CONTROL-D monitor with the same database in the same computer environment, the second monitor immediately shuts down and an appropriate message is issued.

Note See "Activating More Than One Monitor" in the *CONTROL-D Installation Considerations* section in the *IOA Installation Guide*.

The Printers Control monitor also operates 24 hours a day, but is activated and controlled by the main CONTROL-D monitor. When the Printers Control monitor is successfully activated, the following message appears on the operator console:

CTD775I monitor-name CONTROL-D PRINTERS CONTROL MONITOR STARTED

Activating Generic Processing

When the CONTROL-D monitor is brought up, Generic Classes processing is automatically activated. You may need to activate Generic Classes processing manually after CONTROL-D is brought up. Caution should be used if you manually activate the Generic Option when it has not been activated automatically.

To manually automate Generic Classes processing, enter the operator command:

F CONTROLD, STARTGEN

The following message appears on the operator console from which the modify command has been issued:

CTD139I GENERIC JOB DECOLLATION IS ACTIVE ON CLASSES (class-list)

Activating the Compressed Dataset Access Method

The Compressed Dataset Access Method (CDAM) should operate 24 hours a day because CDAM is used both by the CONTROL-D monitor and by jobs within the system to read, write and view compressed reports. For information about how a job can invoke the Compressed Dataset Access Method, refer to Section 5 in the CONTROL-D User Manual.

Usually, CDAM is automatically initialized as part of the IPL process. To activate the Compressed Dataset Access Method manually, use the operator command:

```
S IOASINIT,OPTIONS=D
```

If CDAM is successfully initialized, the following message appears on the operator console:

```
CTM227I IOA SUBSYSTEM "subsystem-name" INITIALIZATION OF CONTROL-D FUNCTIONS COMPLETED
```

Modifying the CONTROL-D Sleeping Interval

CONTROL-D "wakes up" every few seconds and checks what it has to do. This interval is set via a CONTROL-D installation parameter and can be changed by the system administrator. In addition, the interval can be altered by the operator command:

```
F CONTROLD, INTERVAL=xx
```

where xx represents the interval in seconds.

It is recommended that the interval be modified by automatic commands invoked by the CONTROL-O or CONTROL-M monitor (if present) according to set conditions and time ranges, and not manually by the operator. For sites which do not use the CONTROL-M Production Control System, the commands can be issued via the JES Automatic Commands facility.

At most sites, the interval should be longer during the day (when fewer batch production jobs are executing) and shorter during the night.

The optimal sleeping interval depends on the processing power of the machine. Depending on the processing power of the machine indicated, the sleeping interval should usually not be less than the number of seconds indicated:

- For machines with less than 20 MIPS 10 seconds
- For machines with 20-50 MIPS 5-6 seconds
- For machines with over 50 MIPS 4 seconds

There is no practical benefit in setting the interval to less than these minimums. Doing so could slow down CONTROL-D operation.

When the modification is received by CONTROL-D, the following message is displayed on the operator console from which the modify command was issued:

```
CTD123I CONTROL-D INTERVAL IS SET TO nn SECONDS
```

Reloading the Recipient Tree

The Recipient Tree resides in a partitioned dataset. It is loaded when the CONTROL-D monitor is started. If you want to replace the current copy of the Recipient Tree in the CONTROL-D monitor, type the following operator command:

```
F CONTROLD,LOADTREE
```

Notice the messages which are sent to the operator console. The following message signifies that the tree was loaded successfully:

```
CTD160I CONTROL-D RECIPIENT TREE LOADED- nnnnnn RECIPIENTS
```

where nnnnnn is the number of recipients successfully loaded.

Loading the Recipient Tree Into the IOA Online Monitor

The recipient tree is also loaded into the IOA Online monitor.

If you wish to load a new (modified) Recipient Tree under the IOA Online monitor, enter the following operator command:

```
F IOAOMONx,LOADTREE
```

where x is the unique monitor ID.

If the tree is loaded successfully, the following message is displayed on the operator console from which the modify command was issued:

```
CTM786I monitor-name NEW TREE LOADED. NEW USERS WILL BE SIGNED ON TO THE NEW TREE
```

Watch the console for error messages while loading the Recipient Tree. Notify the IOA administrator in case of errors.

When a new Recipient Tree is loaded, every user who enters the CONTROL-D User screen will start working with the new tree. However, users who were using the Online facility when the new tree was loaded will continue to use the old tree. The old tree is deleted from memory after all users who were accessing it have exited the Online facility.

Reloading the Manual Conditions File

The Manual Conditions file (screen 7) is created by utility IOALDNRS. A full description of this utility and its parameters is in Section 9 of this guide. This utility scans the Active Missions file and builds a list of all the prerequisite conditions that should be set manually.

The Manual Conditions file is refreshed (i.e., recreated) by each run of utility IOALDNRS. Normally, the utility is activated as the second step of the CTDNDAY procedure. However, the utility can be run more than once a day if desired.

Activating the IOA Archive Server (CONTROL-V)

To activate the IOA Archive Server issue the following command:

```
S IOASMON
```

Note *The IOA Archive Server uses cross-memory services to communicate with other address space requesting services. Like other address spaces using cross-memory services, whenever the IOA Archive Server is shut down, the address space entry in the MVS Address Space Vector Table (ASVT) remains non-reusable until the next IPL, and the message IEF352I is issued (in some MVS releases). If the IOA Archive Server is brought up and down many times, the ASVT may become full. New address spaces will not start, and an immediate IPL may be required.*

To prevent this problem, specify a large enough value in MVS initialization parameters MAXUSER, RSVSTRT and RSVNONR in member IEASYSXX in the SYS1.PARMLIB library.

Information about these parameters can be found in the "MVS Initialization and Tuning Reference" manual.

Deactivating the CONTROL-D Monitor

To shut down the CONTROL-D monitor, use following operator command:

```
F CONTROLD
```

After a few seconds (a maximum of a minute), the monitor shuts down and the following messages appear on the operator console:

```
CTD107I SHUT DOWN UPON REQUEST FROM OPERATOR
CTD136I STOPPING THE PRINTERS CONTROL MONITOR monitor-name
CTD779I monitor-name CONTROL-D PRINTERS CONTROL MONITOR ENDED
CTD120I CONTROL-D MONITOR SHUTTING DOWN
```

Message CTD779I is highlighted. Message CTD120I is highlighted and unrollable.

In case of emergency, the CONTROL-D monitor can be canceled. However, this is not recommended. If the monitor is canceled, the Printers Control monitor(s) and the secondary monitors should also be canceled.

When you shut down the CONTROL-D monitor, all other CONTROL-D facilities (the Compressed Dataset Access Method, IOA Online monitors, IOA Archive Server) and every Online facility session can remain active.

Deactivating Generic Processing

It is sometimes desirable to stop CONTROL-D from processing the outputs in the generic classes (usually to replace the current report decollating mission definitions).

Enter the operator command:

```
F CONTROLD,STOPGEN
```

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

The following message is displayed on the operator console from which the modify command was issued:

```
CTD140I GENERIC JOB DECOLLATION IS BEING DEACTIVATED
```

New decollation of generic class output does not start. Currently executing (decollating) missions finishes processing those jobs whose output processing already started.

Automatic Warning

When generic decollation is deactivated, the CONTROL-D monitor issues the following warning on the operator console:

```
CTD271I GENERIC JOB DECOLLATION IS INACTIVE - CHECK WHY
```

This message is highlighted, and reappears every ten minutes if there is a job waiting to be decollated in one of the generic classes.

Generic mission deactivation can occur in one of the following ways:

- If started task CTDNDAY fails during the ordering of a generic decollating mission, deactivation is automatic. Check if all the generic decollating missions are in the Active Missions file before reactivating generic processing.
- The operator issues the STOPGEN command (discussed above). Check why generic decollating was deactivated and whether it should remain deactivated.
- The Active User Report List file becomes full. Run utility CTDDLRP to delete unnecessary entries from this file.

The spool can become full if there are many generic jobs waiting to be decollated.

Deactivating the Compressed Dataset Access Method

There is rarely any reason to deactivate the Compressed Dataset Access Method. It should be done only in conjunction with problem-solving efforts as directed by your IOA representative. Shut down the CONTROL-D monitor and issue the operator command:

```
S IOASTERM,OPTIONS=D
```

The following message is displayed on the operator console:

```
CTM231I IOA SUBSYSTEM "subsystem-name" DEACTIVATION OF CONTROL-D FUNCTIONS COMPLETED
```

If the same subsystem supports other IOA products, their functions remain active.

Deactivating the IOA Archive Server (CONTROL-V)

To deactivate the IOA Archive Server, use one of the following operator commands:

```
P IOASMON
```

or

```
F IOASMON,STOP
```

When either of these commands is issued, the IOA Archive Server terminates and an appropriate message is displayed on the operator console.

```
IOA107I IOASMON - SHUT DOWN UPON REQUEST FROM OPERATOR
```

```
IOA108I IOASMON - IOA ARCHIVE SERVER SHUTTING DOWN
```

New Day Processing

Overview

The CONTROL-D monitor is usually activated as a started task and remains active 24 hours a day. New Day processing consists of automatic cleanup from the previous day's mission ordering and automatic ordering of missions for the current day.

The main components related to New Day processing are:

- New Day procedure
- User Daily job
- Date Control records
- Active Missions file

Starting the New Day Procedure

Once a day, at a predefined time determined by the IOA system administrator, the CONTROL-D or CONTROL-V monitor and the Printers Control monitor are shut down and New Day procedure CTDNDAY (this name is defined in the installation parameters and may be different at your site) is activated. The following highlighted, unrollable messages are sent to the operator's console:

```
CTD779I monitor-name CONTROL-D PRINTERS CONTROL MONITOR ENDED
```

```
CTD113W CONTROL-D MONITOR SHUTTING DOWN FOR A NEW DAY
```

The CONTROL-D monitor signals the Printers Control monitor to shut down and waits until it does so. Then the CONTROL-D monitor shuts itself down. When the Printers Control monitor shuts down, it suspends all active printing missions. Printing resumes after the Printers Control monitor is restarted.

Before CONTROL-D shuts down, started task CTDNDAY is activated. This started task can ask the operator several questions. After a few minutes CTDNDAY finishes executing and automatically reactivates CONTROL-D using the following command:

```
S CONTROLD
```

If CTDNDAY abends or fails for any reason, a highlighted, unrollable message is issued to the operator console and the monitor is not reactivated automatically.

CTDNDAY compares the date and time of the computer with the CONTROL-D control files. If they do not match, the following questions are displayed on the operator console:

```
CTD426W CONTROL-D (CTDNDAY) DID NOT RUN FOR nnnnnn DAYS
```

```
CTD427W IS THIS TRUE? (ANSWER "YES" OR "NO")
```

```
CTD428W YOUR ANSWER IS:
```

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

This situation may occur after the computer has not been working for a few days (e.g., a hardware failure or a holiday) in which case the response should be YES. If this is not the case, the computer was probably IPLed with the wrong date. In this case the response should be NO, the date on the computer should be checked and corrected, and CTDNDAY should be restarted. If the date on the computer is correct, the IOA system administrator should be called to check the cause of the problem.

New Day Procedure - Workflow

The New Day procedure performs the following daily maintenance actions and then reactivates the CONTROL-D monitor:

1. Cleans unnecessary missions from the Active Missions file. This includes missions that ended OK, missions in wait schedule state whose MAXWAIT parameter has been exceeded, emergency missions which are no longer needed, etc.

Note Missions that ended OK before the time specified in parameter MAXWAIT was reached, can optionally be kept in the Active Missions file until the MAXWAIT period is exceeded for these missions.

2. Schedules regular and generic decollating missions, printing missions, backup missions, restore missions, and CONTROL-V migration missions. See "Mission Scheduling" later in this section for more information.
3. Reactivates the CONTROL-D monitor by issuing the operator command S CONTROLD.
4. Updates the IOA Manual Conditions file. See the description of utility IOALDNRS in Section 9 of this guide.

User Daily Job

The User Daily job is used to place new missions in the Active Missions file. Each User Daily job usually runs once a day on one or more missions definition members. The missions are selected according to the working date specified to the User Daily job. Therefore, the User Daily job is date-dependent, and certain special situations must be dealt with, such as:

- The computer has not been working for a few days (e.g., due to holidays, or a hardware/software failure).
- The user wants to run a job or a group of jobs and to process their reports prior to or later than the current working date.

Date Control Record

Each User Daily job uses a special Date Control record to store the last running date for the User Daily job. A Date Control record is a member in the CONTROL-D PARM library in which relevant date information is placed during New Day processing.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

The Date Control record is analyzed by the User Daily job to determine the current running date, the last running date, and possible error situations. This date information is used to manage the ordering of missions during New Day processing.

When a User Daily job is run, the current working date is placed in the Date Control record, and the Basic Scheduling parameters of each mission in the mission lists being ordered are compared to this date to determine if the mission should be placed in the Active Missions file.

A Date Control record is 80 characters in length, and contains the following fields:

Columns	Value Added	
1-6	date-1	Current (or last) original scheduling date.
8-13	date-2	Current (or last) original scheduling date of non-generic report decollating missions.
15-20	date-3	Current (or last) original scheduling date of non-generic report decollating missions finish indicator.
22-27	date-4	Current (or last) original scheduling date of printing missions.
29-34	date-5	Current (or last) original scheduling date of printing missions finish indicator.
36-41	date-6	Current (or last) original scheduling date of backup missions/migration missions.
42-47	date-7	Current (or last) original scheduling date of backup missions/migration missions finish indicator.
48-53	date-8	Current (or last) original scheduling date of restore missions.
54-59	date-9	Current (or last) original scheduling date of restore missions finish indicator.
67-72	date-10	Finish indicator date of the User Daily job.

Date format is mmddy, ddmmy or yymmdd, depending on the site standard.

Use of the Date Control Record by the User Daily Job

In certain cases, it is possible to manually update the Date Control record of a User Daily job via a regular editor. The Date Control record is referenced by DD statement DACHK (in the Daily procedure or CLIST).

The work flow of the User Daily job is dependent on the Date Control record. The main steps of the User Daily job are:

- Checking the last running date of the User Daily job (the internal program CTDCHK).

The first date in the Date Control record (column 1-6) is compared to the current installation working date (at the time of the run). If they match the User Daily job has already run today, a message is issued, and the condition code is set to 0004.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

If the current installation working date is earlier than the first date of the Date Control record, a run of the User Daily job has been attempted before its time. The User Daily job stops executing and notifies the user accordingly.

If the current installation working date is later than the first date of the Date Control record (the normal situation), the first date of the Date Control record (columns 1-6) is automatically updated to the current installation working date. This date is then used by all components of the User Daily job as the current scheduling date.

If the User Daily job has not run for more than one day, a warning message is issued and the User Daily job tries to schedule the missions for all of the days which have passed since the last scheduling date (according to the production parameters).

- Placing missions in the Active Missions file according to the current scheduling date and the last running date (via internal programs CTDRRQ, CTDPRQ, CTDBRQ, and CTDSRQ). See the table in "Programs Called During New Day Processing" for more information.

Program CTDxRQ works on mission definitions referenced by DD statement DAXXXLST. For each category in the mission, the program checks whether the category should be scheduled on one or all the days that have passed since the last original scheduling date (date-2, date-4, date-6 or date-8) until the working date in the record (date-1). If the mission should be scheduled, the mission is placed in the Active Missions file.

For example, if a computer did not operate from the 20th to the 23rd of the month, a mission originally scheduled to execute on the 20th is not executed. It is the function of program CTDxRQ to decide whether or not the mission should be retroactively scheduled to run on the logical date of the 20th (see parameter RETRO in Sections 3 and 4 of the CONTROL-D User Manual).

Note Basic Scheduling parameters are considered only if the FORCE option is not specified.

When the program finishes processing the mission definitions, the finish indicator dates (date-3, date-5, date-7 and date-9) are updated to the working date calculated by program CTDCHK (date-1).

Before a CTDxRQ program starts operating, it compares date-2 with date-3 (date-4 with date-5, etc.). If do not match, a previous run of the CTDxRQ program of the same User Daily job has probably abended. The user is notified and the program terminates. To correct the error, you must modify the user Date Control record to the correct date values (using a standard editor).

Note When manually modifying the Date Control record, make sure that the same missions are not scheduled to run twice on the same day.

- Indicating that the User Daily job has ended (via program CTDPDA).

Program CTDPDA updates the finish indicator date (date-10) by setting it to the running date (date-1). This indicates that the User Daily job finished successfully.

Use of the Date Control Record by the New Day Procedure

The CONTROL-D New Day procedure uses its own Date Control record. This Date Control record differs from the User Daily job's Date Control record in that it has two additional dates: date-11 and date-12.

Columns	Value Added	
60-65	date-11	The last formatting date of the Active Missions file (used by the program CTDFRM) is required to prevent formatting from being performed twice on the same day. When this date is in the record, the program CTDCHK recognizes that this is a General Date Control record. If there are problems concerning the date, the program prompts the operator with a series of questions.
74-79	date-12	The current original scheduling date of Generic decollating missions.

When working under the New Day procedure (CTDNDAY), the program asks the operator a question regarding the computer's current date. This is to ensure that an incorrect date was not inadvertently entered during the IPL process.

Programs Called During New Day Processing

Two important programs in New Day Processing are CTDILY and CTDILU.

- The New Day procedure executes program CTDILY.
- Each User Daily job executes program CTDILU.

Programs CTDILY and CTDILU are very similar. Both execute other programs which implement New Day processing. However, program CTDILU (used by User Daily jobs) has less authorization (i.e., is not APF-authorized) and calls fewer programs than program CTDILY.

Both CTDILY and CTDILU read the member referenced by DD statement DAPROG, and activate the programs listed in the member.

The format for each record in the program list is:

Columns

01-08	Program name.
10-11	Maximum condition code allowable in the preceding program. If a higher condition code is encountered in the preceding program, the current program is not executed.

The programs called by the program CTDILY (called by the New Day procedure) and by program CTDILU (User Daily jobs) are shown in the following chart:

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Program	Called By		Purpose
	CTDILY	CTDILU	
CTDCHK	✓	✓	Checks the current date and its relation to the General Date Control record (see above). The program may communicate with the computer operator to verify that CONTROL-D is activated on the correct date.
CTDFRM	✓		Program reformats the Active Missions file. Missions that have already executed and ended OK, missions whose MAXWAIT parameter has been exceeded, or emergency missions which are not needed, are erased from the file and the file is compressed. If this program abends, it recovers automatically after startup, using an automatically-generated backup copy. See also "Parameters of the New Day Procedure."
CTDCATF	✓		Reformats the Active Transfer file. Packets that have already transferred successfully, packets whose MAXWAIT parameter has been exceeded, and the corresponding files for these packets are erased from the Active Transfer file. The file is then compressed. If this program abends, it recovers automatically after startup, using an automatically-generated backup copy.
CTDPRQ	✓	✓	Places printing missions in the Active Missions file according to the date in the General Date Control record and the scheduling criteria in the supplied printing mission definitions. The program receives parameters via DD statement DAPRTLST.
CTDBRQ	✓	✓	Places backup and migration missions in the Active Missions file according to the date in the General Date Control record and the scheduling criteria in the supplied backup mission and migration mission definitions. The program receives parameters via DD statement DABKPLST.
CTDSRQ	✓	✓	Places restore missions in the Active Missions file according to the date in the General Date Control record and the scheduling criteria in the supplied restore mission definitions. The program receives parameters via DD statement DARSTLST.
CTDGRQ	✓		Places generic decollating missions in the Active Missions file according to the date in the General Date Control record and the scheduling criteria in the supplied generic decollating mission definitions. The program receives parameters via DD statement DAGENLST.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Program	Called By		Purpose
	CTDILY	CTDILU	
CTDRRQ	✓	✓	Places non-generic decollating missions in the Active Missions file according to the date in the General Date Control record and the scheduling criteria in the supplied report decollating mission definition. The program receives parameters via DD statement DAREPLST.
CTD34F	✓		Executes operator commands (system, JES, VTAM, etc.) from a list (ddname DA34F). The default command is S CONTROLD which starts the CONTROL-D monitor.
CTDPDA	✓	✓	Records the end of the Daily run.

Parameters of the New Day Procedure

Program CTDFRM, activated as part of the New Day procedure, is responsible (among other duties) for erasing all CONTROL-D related conditions from the IOA Conditions/Resources file for the current day. Prerequisite conditions are always assigned a date reference (day and month). They can be kept in the IOA Conditions/Resources file for a entire year. Therefore, it is necessary to erase them at the beginning of each working day. Otherwise, missions may be triggered because of conditions remaining from the previous year.

It is common to use prerequisite conditions which are not date-dependent (e.g., IMS-IS-UP, or AR-FILE-OK). It is important that these conditions not be erased. You can supply a list of conditions which are not to be erased via DD statement DAFRMIN. The valid format is:

IGNORE COND prefix

Where

prefix Name (or mask) of conditions not to be erased. If an asterisk (*) is specified, no conditions are erased. Multiple IGNORE statements can be specified.

Note When the CONTROL-M Production Control System is installed at your site the above mentioned updates of the IOA Conditions/Resources file are performed by CONTROL-M.

Mission Scheduling

Overview

CONTROL-D and CONTROL-V manage report distribution at your site via missions defined by the user. Several different types of missions are available, each for a different type of task.

- Decollation missions
- Printing missions
- Restore missions
- Backup missions
- Migration missions (CONTROL-V only)

The following pages explain how to manage missions for the topics:

- Scheduling - General (an overview of the basic mission scheduling options)
- Using the New Day Procedure (CTDNDAY) to Schedule Missions
 - Supplied Mission List Members
- Using a User Daily Procedure to Schedule Missions
- Scheduling a Mission Manually
- Scheduling Missions - Workflow

Following these topics is a description of special considerations for each mission type.

Scheduling - General

Missions can be scheduled via the following methods:

- Automatically by the CONTROL-D New Day procedure (CTDNDAY). This is the usual method. (See “Using Procedure CTDNDAY to Schedule Missions”.)
- Using a batch job (e.g., procedure CTDRPDAY for decollating missions). The job can be submitted manually or automatically by a scheduler. (See “Using a User Daily Job to Schedule Missions”.)
- Manually from the Mission Definition screen, using the O (Order) and F (Force) options. This is the most convenient manual method.
- Manually under ISPF using the Online Utilities Panel. The panel can be invoked directly using a CLIST (e.g., CTDNISRQ MIS(REP) for a decollating mission).
- Manually under ROSCOE (e.g., using RPF CTDRQPRT for a printing mission).
- Using a KeyStroke Language utility found in the SAMPLE library (e.g., BKPORDER for backup missions).

Using the New Day Procedure (CTDNDAY) to Schedule Missions

This is the recommended method for scheduling missions because the New Day procedure is activated automatically once a day. A member in the CONTROL-D PARM library is referenced by a DD statement in the New Day procedure. This member contains a list of missions that should be scheduled. The Basic Scheduling parameters of these missions are analyzed against the requested scheduling date. If the mission should be scheduled for that day, it is placed on the Active Missions file. The table contains the names of the mission list members, and the DD statements used to reference each mission type in the New Day procedure.

Mission Type	Member	DD Statement
Regular Decollating	REPLIST	DAREPLST
Generic Decollating	GENLIST	DAGENLST
Printing	PRTLST	DAPRTLST
Restore	RSTLIST	DARSTLST
Backup	BKPLIST	DABKPLST
Migration	BKPLIST	DABKPLST

Each line in the member containing the list of missions has the following format:

```
date libname [missionname catname] [FORCE]
```

date ODATE (original scheduling date) of the mission.

Valid formats are:

mmddy, ddmmy or yymmdd (depending on the site standard)

The Basic Scheduling parameters of the mission are compared with this date. The Date Control record is not updated.

* (asterisk) can be specified to indicate that date management be handled automatically by CONTROL-D using the Date Control record. Parameter RETRO affects the result obtained when the mission's basic scheduling criteria are compared with the last scheduling date of the mission.

libname Name of the library in which the mission parameters are found.

Valid values are:

- Specific library name. In this case, the library is dynamically allocated.
- A DD name preceded by an asterisk. The library must be a partitioned dataset with a record length of 80.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

missionname Name of the mission (a member in the library). Optional.
* (asterisk) can be specified to indicate that all the missions (members) in the specified library are analyzed. Use of an asterisk eliminates the need to update mission lists to include new missions.

catname The specific category of the mission is analyzed. If an asterisk is specified, all categories of the mission are analyzed.

FORCE Optional parameter. When FORCE is not specified, the Basic Scheduling parameters of the category of the missions are analyzed against the specified original scheduling date. If the mission should be scheduled on that date, it is placed on the Active Missions file.
When FORCE is specified, mission categories are placed on the Active Missions file with the specified original scheduling date (which must be an actual date, not an asterisk) regardless of basic scheduling criteria.

The following is an example of the JCL (relevant DD statements of procedure CTDNDAY only):

```
//CTDNDAY EXEC PGM=CTDILY
//DACHK DD DISP=SHR,DSN=ctd_parm_library(DDATEREC)
//DAREPLST DD DISP=SHR,DSN=ctd_parm_library(REPLIST)
```

The Date Control record used in this case is the General CONTROL-D Date Control record.

Supplied Mission List Members

The following are the default mission list members supplied with CONTROL-D:

PRTLST Schedules one printing mission (for form STD).

BKPLIST Schedules the following backup missions:

- BKP0007D Back up for 7 days.
- BKP0031D Back up for 31 days.
- BKP0180D Back up for 180 days.
- BKP0365D Back up for 365 days.

RSTLIST Schedules the following restore missions:

- RST0060M Restore reports every 60 minutes
- RSTADHOC Restore reports immediately (for authorized users only)

Note *Generic Decollating Missions*

It is recommended to schedule Generic Decollating missions by the New Day procedure. See "Decollating Mission Management" later in this section.

Using a User Daily Procedure to Schedule Missions

The following JCL should be used:

```
//          EXEC   procedure-name
//DACHK    DD     DISP=SHR,DSN=ctd_parm_library(daterec)
//DAXXXLST DD     DISP=SHR,DSN=ctd_parm_library(mlistmem)
```

Where:

- procedure-name CTDRPDAY should be specified for decollating missions.
 CTDPRDAY should be specified for printing missions.
 CTDRSDAY should be specified for restore missions.
 CTDBKDAY should be specified for backup and migration missions.
- DAXXXLST Reference to the member containing the mission list. Same as specified above for the New Day procedure.
- daterec Name of the Date Control record for the procedure.
 Each procedure must use one Date Control record. A Date Control record cannot be shared by two procedures. The Date Control record is referenced by DD statement DACHK and is used to determine the correct scheduling date for selecting missions according to their scheduling criteria. See the description of the Date Control record earlier in this section for more information.
- mlistmem Name of the member containing the mission list. Same as specified above for the New Day procedure.

Scheduling a Mission Manually

Missions are normally scheduled automatically via the New Day processing. However, it is sometimes necessary to schedule a mission manually (for testing, ad hoc missions, etc.). In addition, it may be necessary to schedule a mission for different scheduling dates (e.g., scheduling a mission which was to run on the 1st of next month on the 30th of this month, or rescheduling a mission which ran on the 4th because the entire run has to be performed again on the 5th).

The recommended method of manually scheduling a mission is using the O (Order) or F (Force) options of the Mission List screen or via an ISPF online utility. However, if you wish to schedule a mission without entering the IOA Online facility, use one of the following CLISTS, KSL utilities or RPFs (for ROSCOE):

Mission Type	CLIST	RPF	KSL Utility
Decollating (Regular and Generic)	CTDMISRQ MIS (REP)	CTDRQREP	REORDER
Printing	CTDMISRQ MIS (PRT)	CTDRQPRT	PRTORDER

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Mission Type	CLIST	RPF	KSL Utility
Restore	CTDMISRQ MIS (RST)	CTDRQRST	RSTORDER
Backup	CTDMISRQ MIS (BKP)	CTDRQBKP	BKORDER
Migration	CTDMISRQ MIS (MIG)	CTVRQMIG	MIGORDER

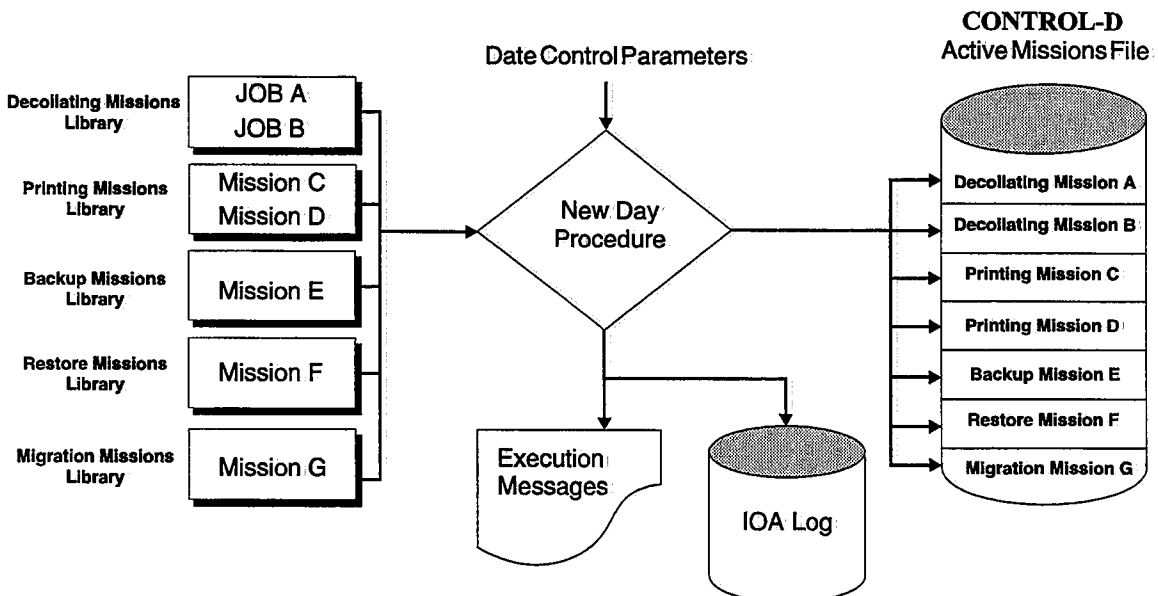
See Section 2 of the CONTROL-D User Manual for more information about ISPF Online utilities.

Scheduling Missions -Workflow

A mission is defined using the Online facility (option R or M). Each mission contains parameters describing actions to be performed and when and under what conditions they are to be performed. Mission definitions are stored in regular libraries (partitioned datasets).

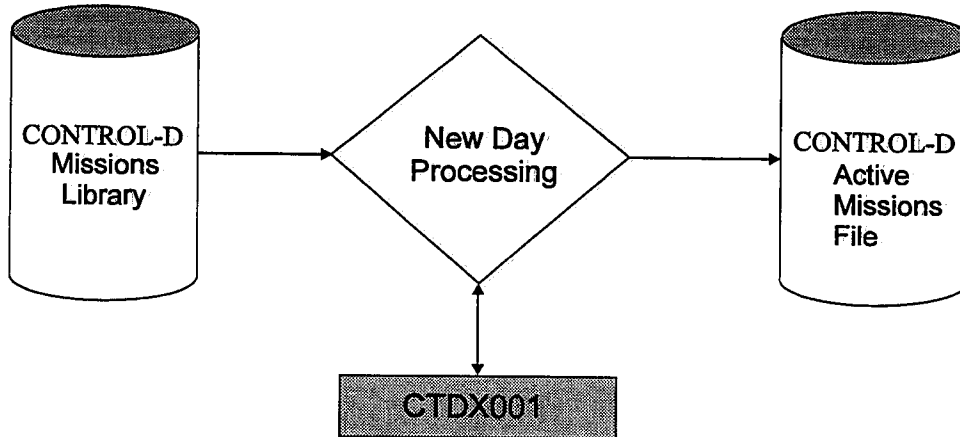
Mission definitions in the library are not active instructions to the CONTROL-D monitor. To make a mission active, it must be placed on the Active Missions file. The process of selecting a mission from the definition library and placing it on the Active Missions file is performed by the CONTROL-D New Day procedure during New Day processing.

When invoked, the New Day procedure analyzes definitions of missions in a library. According to the basic scheduling criteria specified, appropriate missions are selected and placed in the Active Missions file. Each mission is automatically assigned an original scheduling date (ODATE) when it is placed in the Active Missions file.



When a decision is made to schedule a mission, its parameters are passed to CONTROL-D exit CTDX001. This exit can modify the contents of the parameters or cancel the mission. If the mission is not canceled by the exit, the mission is placed in the CONTROL-D Active Missions file.

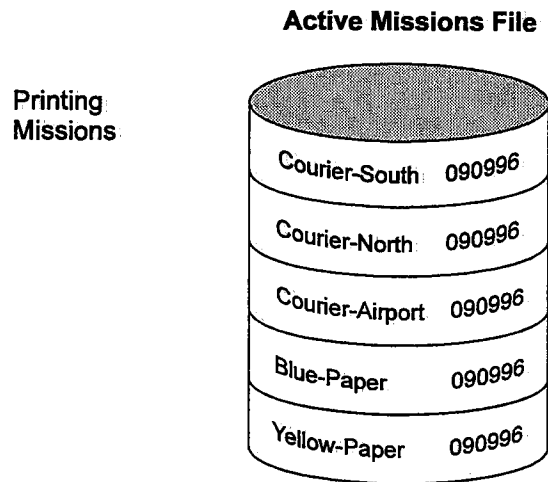
Section 4: Customizing and Administering CONTROL-D and CONTROL-V



The Active Missions file can contain more than one mission with the same name.

Examples

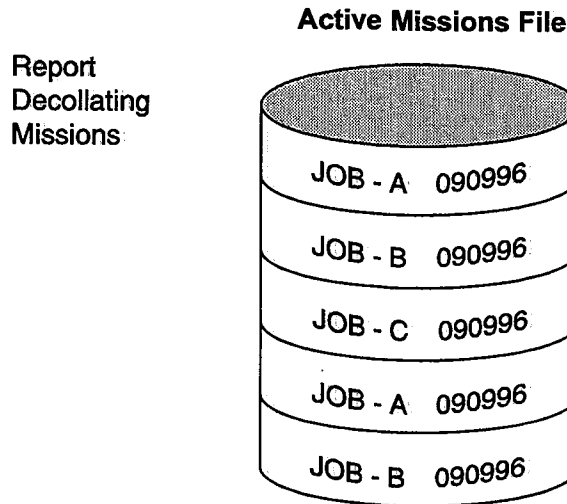
- Several categories of the same printing mission: Consider the organization of bundling by a delivery network. A few categories are used, each category describing a bundle with different report recipients. The categories COURIER-NORTH, COURIER-SOUTH, COURIER-AIRPORT are used for the same printing mission. A different list of recipients are bundled under each category. Using this method, a separate bundle is prepared for each delivery facility (e.g., COURIER).



- A printing mission is late by more than one day (e.g., prerequisite reports have not yet finished). In this case, the Active Missions file contains two printing missions with the same name, but each with a different original scheduling date.
- When a daily job is late by more than one day (e.g., input tape has not arrived yet). In this case, the Active Missions file contains two report decollating missions for the same job, each with a different original scheduling date.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

- When the same job is being run a few times a day.



Note *Scheduling of decollating missions can be made dependent on the operation mode of the production control system at your site. It is possible to use all the different modes of operation at the same site. This is usually done to solve special scheduling problems.*

The different modes of operation generally vary according to the type of production control system in use. The following situations are possible:

- *The production control system is CONTROL-M.*
- *The production control system is not CONTROL-M.*
- *There is no production control system.*

See "CONTROL-D and Production Control System Scheduling Work Flow" later in this section for a description of how CONTROL-D interacts with production control systems.

Decollation Mission Management

Overview

Regular report decollating missions can handle output from the spool or from CDAM datasets. They are executed once to handle the output produced by a specific job name (unless defined as cyclic report decollating missions, with a task type of CRP).

Generic Decollating Missions

CONTROL-D provides the option of decollating output from dedicated output classes. Whenever a non-held output appears on the spool in one of the classes defined for generic processing, the CONTROL-D monitor looks for a generic decollating mission which matches the job name, CLASS, DEST, FORM, and EXTWTR of the selected SYSOUT. The search is performed on all generic missions currently in the Active Missions file (which are not held).

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Generic decollation is intended for processing ad hoc sysouts (e.g., MSGCLASS), or reports generated dynamically from a CICS (or similar) environment. A generic decollating mission can process sysouts from a job which is still running (e.g., SYSLOG files) at time of decollation.

Generic Decollating Mission Workflow

A generic decollating mission can be defined for a specific job name but is usually used in conjunction with a generic job name. A generic job name is specified using mask characters. Execution of the generic decollating mission is triggered by the appearance of a job (whose name matches the job name mask) in a specific output class (in non-held status). These output classes are defined in the CONTROL-D installation parameters. The appearance of a matching job name in other output classes (or in held status in the generic classes) does not trigger the execution of a generic decollating mission.

A generic decollating mission can decollate only from the generic classes defined in the CONTROL-D installation parameters. Any output which is decollated by a generic decollating mission is purged from spool. After decollation, the mission's post-processing parameters (OUT, SHOUT) are executed, but the mission does not stay in ENDED status. It is recycled for re-execution (i.e., it is in WAIT SCHEDULE status).

All runtime scheduling criteria (IN, TIME, PRIORITY) are applicable to generic decollating missions. Usually the recycled generic mission is immediately eligible for execution and is assigned the WAITING FOR JOB status.

Sysouts of jobs that appear on a generic class but do not have a matching generic decollating mission name in the Active Missions file are removed from the generic class in one of the following ways, depending on the value specified in CONTROL-D installation parameter GENOTFND:

- Values which are valid under JES2:
 - The output's spool priority is set to one. This allows other output with higher priority to be processed. Default.
 - The output is deleted from the spool.
 - The output's spool status is altered to hold, preventing it from being processed again by CONTROL-D generic decollation class monitoring.
 - The output's class is altered to the specified class. This class should not be one of the classes specified in parameter GENCLAS.
- Values which are valid under JES3:
 - The output for which there is no associated scheduled generic decollation mission is deleted from the spool.
 - The output's class is altered to the specified class. This class should be defined as HOLD=EXTWTR. This class should not be one of the classes specified in parameter GENCLAS.

Additional Considerations

It is possible that more than one generic decollating mission matches the same job. In this case, they are processed according to their priority levels. If their priorities are equal, they are processed in the order in which they appear in the Active Missions file (i.e., in the order in which they were ordered/scheduled).

When GENERIC=Y and MONITOR is blank, the mission is ordered separately for each monitor and each copy of the mission is assigned a different monitor number (ID). This enables concurrent decollating of generic jobs under more than one monitor.

It is also possible that a match could occur between a generic definition and a regular report decollating mission. This duplication should not create problems because it is impossible to specify a generic class in an ON CLASS statement of a regular report decollating mission. However, if a regular (ON DSN) report decollating mission is specified with parameter WHEN IN QUEUE=Y, it might never execute because its job's output is deleted by a generic decollating mission. If WHEN IN QUEUE=Y is specified for a regular report decollating mission, make sure that the job's output will not be deleted by a generic mission.

In an environment using a production control system there is a significant advantage in decollating job reports using regular report decollating missions. The decollating of the reports can be made dependent on a prerequisite condition which is added by the production control system after checking that the job has finished executing OK. Using this method, erroneous reports are not decollated. On the other hand, the MSGCLASS output of the job (on another class) is best handled by a generic decollating mission.

Examples

Generic decollating missions can be used to:

- Handle all MSGCLASS outputs of jobs in the same way (e.g., move them from spool to compressed datasets, allow Online Viewing of output, and retain backups for two weeks).
- Handle ad hoc jobs (e.g., decollate output to users based on job prefix).

Note CONTROL-D first handles regular report decollating missions according to their priority and then scans the generic output classes for outputs. However, this does not guarantee that the regular mission of the same job is executed before a generic one.

Scheduling Generic Decollating Missions Via the New Day Procedure

Special scheduling of generic decollating missions ensures that all generic missions are scheduled successfully before the CONTROL-D monitor starts generic mission processing. If procedure CTDNDAY fails (e.g., system crash or invalid parameters in generic decollating mission definitions) while processing a report decollating mission from member GENLIST, when the CONTROL-D monitor is started, it will not process generic decollating missions without a manual operator command. If all generic missions are scheduled successfully, decollating of generic missions resumes when the CONTROL-D monitor is brought up.

If generic mission decollating is not active when the CONTROL-D monitor shuts down for a new day (e.g., because the operator deactivated it manually), it remains inactive when the CONTROL-D monitor is brought up again.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Generic decollating missions can also be scheduled manually, or by any of the other scheduling methods. However, it is highly recommended not to schedule the missions via the REPLIST member of the New Day procedure with non-generic decollating missions.

Controlling the Generic Process

You can control generic processing by turning it off and on. This process is performed via operator commands. By default generic processing is started when CONTROL-D is started (unless it was previously deactivated by an operator command).

One reason that generic processing may be deactivated is if the CONTROL-D New Day procedure (CTDNDAY) fails to schedule the generic missions to the Active Missions file. When generic processing is deactivated, the generic missions do not select output from the defined generic classes.

Generic processing may also be deactivated if an error occurs while updating the Active User file during decollation. In this case generic processing is deactivated to prevent the deletion of sysouts from the spool without the creation of relevant reports in the Active User file.

To stop generic processing, issue the following operator command:

```
F CONTROLD,STOPGEN
```

To start generic processing, issue the following operator command:

```
F CONTROLD,STARTGEN
```

Defining a Generic User Name List

CONTROL-D provides an option to specify a generic user name in the report decollation definition. (See parameter DO USER in Section 3 of the CONTROL-D/V User Manual for a detailed description.) This option defines a generic name that describes a group of users (e.g., all the branches of the bank, senior management, or financial controllers.)

A generic user name list resides in a PDS member. The name of the member is referenced in the DO USER statement, preceded by an asterisk (e.g., *BRANCHES.) The member must be in a library which should be referenced by DD statement DAGENUSR of the User Daily job (or CTDNDAY.) Although, it is possible to use a few generic user name libraries in a distributed environment, it is generally recommended to use only one to simplify the administration.

Each member in the library represents a generic user name. The contents of the member are lines in the following format:

Column	Data
01 - 05	TUSER (i.e., constant value TUSER)
06 - 09	blank
10-11	Level code of the user (e.g., recipient) as defined in the Recipient Tree.
12-31	User name

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

One line is required for each user. Additional lines in the member are not permitted. Do not use columns 73-80. Each line must describe a valid user name in the Recipient Tree. Use the main user name, not a synonym.

Example of the member BRANCHES:

```
TUSER 30BR101
TUSER 30BR103
TUSER 30BR106
TUSER 30BR112
TUSER 30BR114
TUSER 30BR123
TUSER 30BR127
TUSER 30BR128
TUSER 30BR130
```

Interfaces to Production Control Systems

Various aspects of CONTROL-D/Production Control System interfaces are explained under the following topics:

- Overview of CONTROL-M Scheduling With CONTROL-D
- Scheduling via the CONTROL-M Production Control System
- Scheduling via a Non-CONTROL-M Production Control System

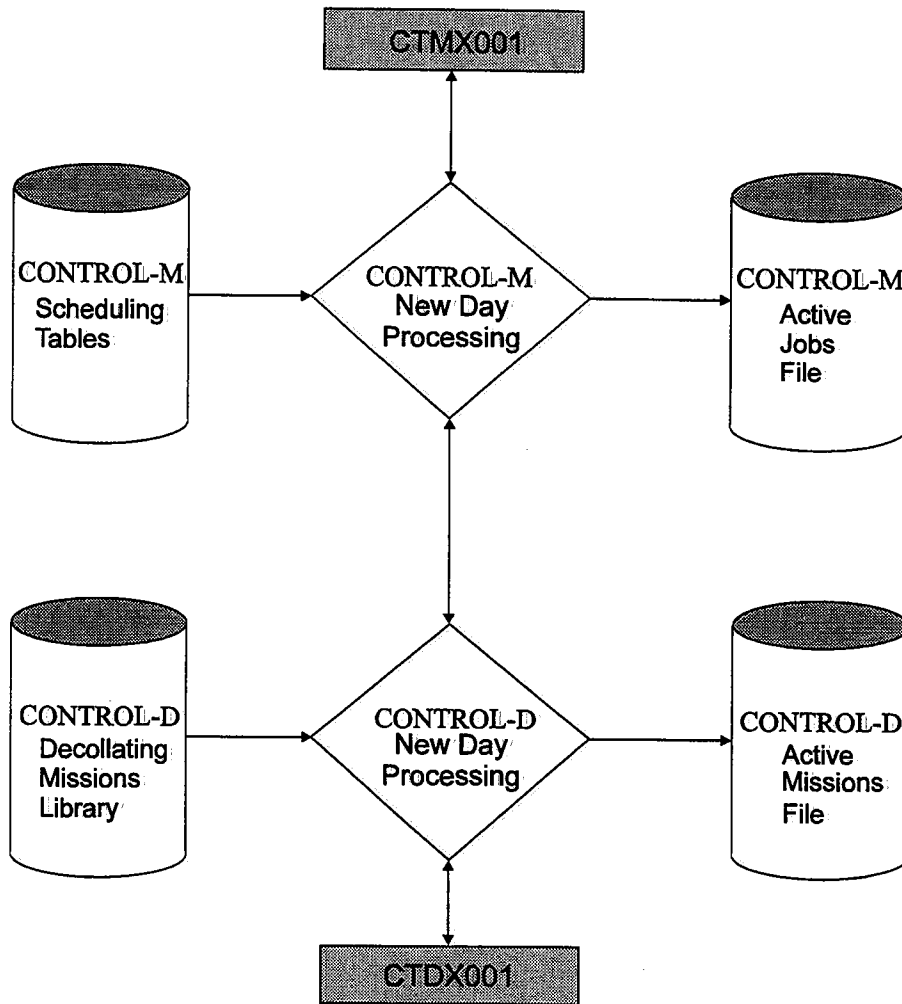
Overview of CONTROL-M Scheduling With CONTROL-D

CONTROL-M jobs are scheduled and placed in the Active Jobs file during CONTROL-M New Day Processing. (The job order is submitted when the runtime requirements are satisfied.) The main function of CONTROL-M New Day Processing is to determine whether or not the job should be executed on a specific day. Once that decision is made, the job order is placed in the CONTROL-M Active Jobs file.

The CONTROL-D category field (in the CONTROL-M job order) is checked. If the category field is not blank, a search is made for the report decollating mission in the library referenced by DD statement DAREPMIS. The search is for a member with the same name as the CONTROL-M MEMNAME parameter (the CONTROL-D job name), and for the same category as specified in the CONTROL-M category field. If category * was used, the search is for all categories of the specified job name.

The scheduling criteria of the CONTROL-D report decollating missions (if any) are ignored (unless optional wish WM0983 is applied). The report decollating missions parameters are passed to CONTROL-D user exit CTDX001. This exit may alter the contents of the parameters or cancel the decollating mission. If the report decollating mission is not canceled by the exit, the report decollating mission is placed in the CONTROL-D Active Missions file. The original scheduling date assigned to the report decollating mission is the same as that of the CONTROL-M job order.

Section 4: Customizing and Administering CONTROL-D and CONTROL-M



Usage Instructions

The library containing report decollating mission definitions is referenced by DD statement DAREPMIS. This DD statement is normally included in the CONTROL-M User Daily job. However, if at your site jobs ordered via the New Day procedure have decollating missions, DD statement DAREPMIS must also be included in CONTROL-M New Day procedure. Add DD statement DAREPMIS to procedures CONTDAY and/or CTMDAILY, the CLIST CTMCJOBS, the Online monitor, etc.

More than one partitioned dataset may be referenced by each DD statement DAREPMIS (in a job or CLIST). If a library is not referenced, an error message is produced and the CONTROL-M New Day procedure or the User Daily job skips to the next job.

If you wish to use more than one library to store definitions of report decollating missions, more than one CONTROL-M Daily may be used. This usually corresponds with using more than one CONTROL-M scheduling library (for security reasons).

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Example (Relevant DD Statements Only):

CONTROL-M User Daily Job 1

```
//DACHK DD DISP=SHR,DSN=parm-library-1(date-CONTROL-record-1)
//DAJOB DD DISP=SHR,DSN=scheduling-library-1(table-1)
// DD DISP=SHR,DSN=scheduling-library-1(table-2)
// DD DISP=SHR,DSN=scheduling-library-1(table-3)
//DAREPMIS DD DISP=SHR,DSN=report-definitions-library-1
//DAAMF DD DISP=SHR,DSN=the-CONTROL-D-active-missions-file
//
```

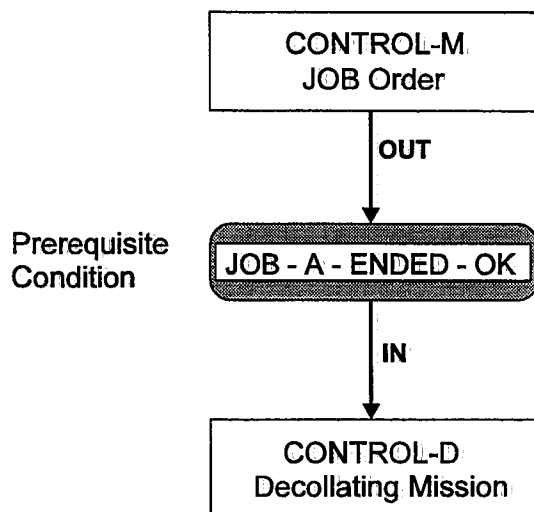
CONTROL-M User Daily Job 2

```
//USER02 EXEC CTMDAILY
//DACHK DD DISP=SHR,DSN=parm-library-2(date-CONTROL-record-2)
//DAJOB DD DISP=SHR,DSN=scheduling-library-2(table-1)
// DD DISP=SHR,DSN=scheduling-library-2(table-2)
//DAREPMIS DD DISP=SHR,DSN=report-definitions-library-2
//DAAMF DD DISP=SHR,DSN=the-CONTROL-D-active-missions-file
```

For a full description of the CONTROL-M New Day processing, see the beginning of Section 3 in this guide.

Job-Report Dependency

It is recommended to establish dependency between the successful execution of the job under CONTROL-M and the processing of the job's reports by the CONTROL-D report decollating mission. The dependency can be established using a prerequisite condition. Parameter OUT of the job order under CONTROL-M should specify a prerequisite condition to be referenced by parameter IN of the CONTROL-D report decollating mission.



Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Using CONTROL-M Scheduling Data - Summary

The advantage of invoking the CONTROL-D report decollating missions via CONTROL-M scheduling criteria is that the scheduling criteria only have to be defined once (under CONTROL-M). It is then unnecessary to update the scheduling criteria of CONTROL-D each time the criteria are updated in CONTROL-M. In addition, ad hoc job orders under CONTROL-M automatically place report decollating missions in the CONTROL-D Active Missions file.

The job scheduling information is managed by one source of data and control.

Scheduling via the CONTROL-M Production Control System

When the CONTROL-M production control system is in use, the integrated environment of CONTROL-D and CONTROL-M can produce optimum results.

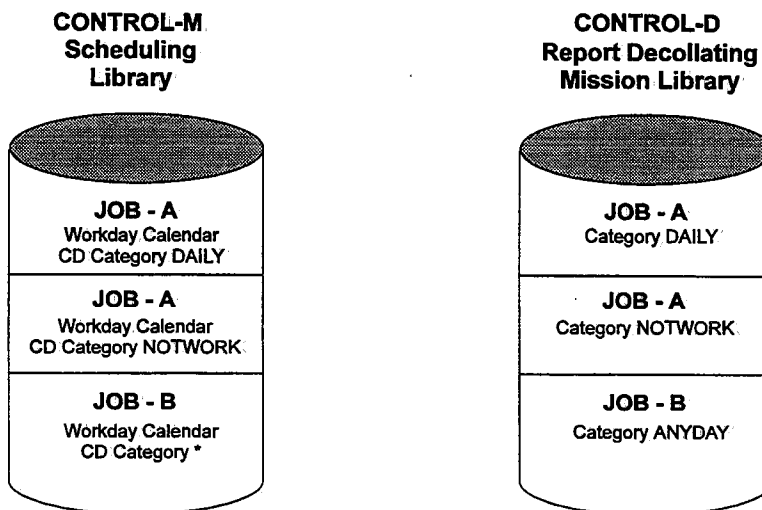
In such an environment, the recommended method of scheduling a non-generic report decollating mission is via CONTROL-M scheduling criteria.

The CONTROL-M Job Scheduling Definition screen contains a field (CATEGORY) to indicate that a CONTROL-D report decollating mission should be scheduled whenever the job is scheduled to run under CONTROL-M.

In this field, the user specifies the CONTROL-D category of the report decollating mission that should be selected from the CONTROL-D report decollating mission library. To select all categories of the job, an asterisk (*) should be specified in the CATEGORY field. (If the CATEGORY field is empty, a report decollating mission is not scheduled for the job.)

Each report decollating mission can be composed of a few categories. The category is usually used as a mechanism to specify different report decollating (processing) parameters for different execution days (e.g., a regular work day, end of the month).

Based on different scheduling criteria under the CONTROL-M monitor, the same CONTROL-M job order can be used to select different report decollating missions of different categories.



Section 4: Customizing and Administering CONTROL-D and CONTROL-V

When CONTROL-M and CONTROL-D operate together in the same environment, the following situations may arise:

- A job to be run by CONTROL-M does not generate any reports. Therefore, there is no need to process the job by CONTROL-D. In this case, the category field in the CONTROL-M Job Scheduling Definition screen should remain blank.
- A job to be run by CONTROL-M generates a report but does not (yet) have a report decollating mission defined for the job. (CONTROL-D can be implemented in phases). In this case, the category field in the CONTROL-M Job Scheduling Definition screen should remain blank.
- A job generates a report which should be decollated by a CONTROL-D report decollating mission. In this case, the name of the CONTROL-D category of the report decollating mission should be defined in the CATEGORY field in the CONTROL-M job scheduling definition. An asterisk (*) may be used to select all categories of the job.

Scheduling via a Non-CONTROL-M Production Control System

The following methods can be used to implement CONTROL-D in data centers which use a non-CONTROL-M production control system:

- CONTROL-D scheduling is managed independently of the production control system. This method is the same as managing the scheduling without any production control system.
- The scheduling of report decollating missions under CONTROL-D is controlled by the production control system (described in the following paragraphs).

Most production control systems include some means of passing control to a user exit before a job is submitted. Some products have user exits in earlier stages (when the decision to schedule a job for a specific day is made). The basic idea is to invoke the CONTROL-D report decollating mission from the user exit of the production control system.

When a production control system and CONTROL-D operate together in the same environment, the following situations may arise:

- A job which is to be run by the production control system does not have any reports. It does not need to be processed by CONTROL-D.
- A job which is to be run by the production control system does not yet have a report decollating mission defined for the job.
- A job is to be run by the production control system and decollated by CONTROL-D.

To handle the situations mentioned above, the user should use some method of specifying to the production control system user exit whether or not a report decollating mission should be issued for the job. The decision of how to mark a job is local for each installation and can be performed using one of the following methods:

- Specify some of the job scheduling parameters (of the production control system) that are otherwise unused, or are used for other purposes (e.g., the job description).
- Use a list internal to the user exit. The list must be modified whenever a new report decollating mission is defined.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

When the production control system user exit receives control and a decision is made to invoke a report decollating mission, the user exit should activate program CTDRRQ. This program performs a search for the report decollating mission. For more information about program CTDRRQ and its parameters, see "Programs Called During New Day Processing" in Section 4 of this guide.

The search is made for a member name with the same name as the job name and for the specified category. The scheduling criteria of the CONTROL-D report decollating mission (if any) are ignored. The report decollating mission parameters are passed to CONTROL-D user exit CTDX001. This exit may alter the contents of the decollating parameters or cancel the Decollating mission. If the report decollating mission is not canceled by the exit, it is placed in the CONTROL-D Active Missions file.

Usage Notes

When a report decollating mission is scheduled by invoking program CTDRRQ from a production control system user exit, the following DD statements must be used:

STEPLIB	IOA LOAD library (if not included in the MVS Linklist).
DAAMF	CONTROL-D Active Missions file.
DAOUT	Output print file.
DALOG	IOA Log file.
DAGENUSR	Generic user names library (if this option is used by your site).

If the production control system does not have an appropriate exit or you do not want to use this method to schedule report decollating missions, there is an alternative. Most production control systems are capable of producing a night plan report - a list of jobs to be executed during the night. This list can be used as input to a program that will schedule report decollating missions by invoking program CTDRRQ.

Examples of production control system interfaces can be found in the IOA SAMPLE library. Member \$\$INTER in the SAMPLE library contains an index of all information related to each production control system which is currently available in the SAMPLE library. A user-written interface to CA-7 (UCC7), which is not officially supported, is available. Detailed interfaces to other production control systems, based on the methods described above, will be available in the future. (For more information on the CA-7 (UCC7) interface, call your IOA representative.)

Job-Report Dependency

It is recommended to establish dependency between the successful execution of the job under the production control system and the processing of the job's reports by the CONTROL-D report decollating mission. This dependency can be established using the prerequisite condition concept.

A prerequisite condition (e.g., REP-M01AUPD-READY) should be specified in the IN parameter of the CONTROL-D report decollating mission. This condition is added to the IOA Conditions/Resources file by the production control system after the job finishes executing successfully.

Use utility IOACND to add a condition. Utility IOACND can be activated from a production control system user exit, from a specially defined successor task of the job under the production control system, or from a job step.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

It is not mandatory to establish prerequisite conditions dependency between the production control system and CONTROL-D. If dependency is not established, CONTROL-D will process the reports when the job appears on the output queue. It is also possible to use manual prerequisite conditions for report processing when required.

Using a Non-CONTROL-M Production Control System - Summary

The advantage of invoking the CONTROL-D report decollating mission from the user exit of the production control system is that scheduling criteria only have to be defined once (under the production control system). The user does not have to update scheduling criteria of report decollating missions each time they are updated in the production control system.

Considerations for When CONTROL-M and CONTROL-D Are Installed

The following issues should be considered if both CONTROL-M and CONTROL-D are being installed:

- In previous releases of CONTROL-M and CONTROL-D, it was necessary to modify some CONTROL-D daily JCL procedures and CLISTS. In the current release, no modifications are required; the appropriate CONTROL-M/CONTROL-D files are dynamically allocated. DD statement DAALOCIN points to a member in a partitioned dataset which contains a list of files that will be dynamically allocated.

- Utility IOACLRES cleans the IOA Conditions/Resources file.

It is recommended that only utility IOACLRES be used to clean the IOA Conditions/Resources file, because the same file is used for both CONTROL-M and CONTROL-D. We also recommend that these utilities all be assigned the same parameter control member via DD statement DACRSIN.

- CONTROL-M New Day procedure CONTDAY updates the IOA Conditions/Resources file. It should delete all the conditions which were left from last year (conditions can be kept up to a year). The CONTROL-M New Day procedure can receive parameters which control the delete operation via DD statement DAFRMIN. When both CONTROL-M and CONTROL-D are installed, IOA installation parameter CTM must be set to Y (otherwise, results may be unpredictable). When CTM=Y, the CONTROL-D New Day Procedure does not handle the conditions in the Conditions/Resources file; maintenance of conditions is handled by the CONTROL-M New Day procedure. Therefore, it is recommended that the CONTROL-M New Day procedure always be run before the CONTROL-D New Day procedure.

It is recommended to clean the IOA Conditions/Resources file periodically using utility IOACLRES.

- Utility IOALDNRS builds the Manual Conditions file from information in the CONTROL-M Active Jobs file, the CONTROL-D Active Missions file, and the IOA Conditions/Resources file. However, "older" utilities CTMLDNRS and CTDLNRS, which perform the same function, are also supplied for compatibility with previous releases of CONTROL-M and CONTROL-D.

We strongly recommend that only utility IOALDNRS be used to build the Manual Conditions file because the same file is used for both CONTROL-M and CONTROL-D. We also recommend that these utilities all be assigned the same parameter control member via DD statement DALNRIN.

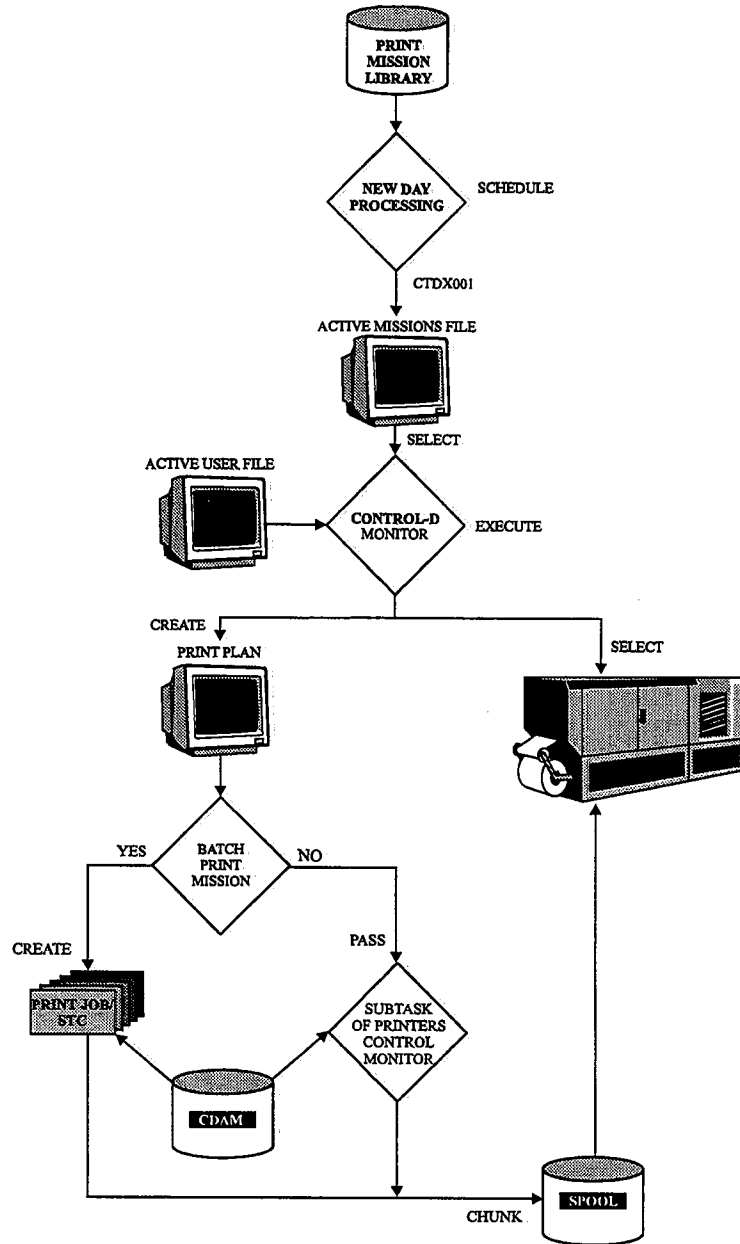
Printing Mission Management

Various aspects of managing printing missions are explained under the following topics:

- **Printing Mission Execution Workflow**
 - **Printing Mission Definition Screen**
- **Advanced Scheduling Issues:**
 - **Distribution According to Scheduling Dates**
 - **Report Decollating and Printing Mission Dependency**
- **Printer Control**
- **Identifying Chunks on Spool**
- **Printing on AFP (APA) Printers**
- **Printing Using XEROX DJDE Parameters**
- **Advanced ACIF Interface Facility**
- **Using OUTPARM for Global Control of Printing Characteristics**
- **Printing to a File**

Printing Mission Execution Workflow

The Printing Mission workflow of CONTROL-D is shown. It illustrates the stages involved from initially scheduling the printing mission through printing reports.



The execution workflow of a printing mission is performed as follows:

The Active User file is scanned by the CONTROL-D monitor for all reports requiring printing by the specified printing mission. (This is a fast scan, not a scan of the entire file.) The result of this process is a Print Plan file which includes entries for reports to be printed. In addition, the CONTROL-D monitor selects an available logical printer for this printing mission.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

When the printer is selected, the control is passed either:

- To the Printers Control monitor (CTDPRINT), if the BATCH parameter was set to N,
or

- To a batch job if the BATCH parameter was set to Y.

The print process continues either by the Printers Control monitor or by the batch job. Chunks are sent to the spool depending on specific parameters such as CHUNKSIZE and OVERRIDE. See "Printer Control" later in this section for further details.

After printing is completed, the status of each report entry from the Print Plan file is updated in the Active User Report List file according to the printing completion status (e.g., Printed/Not printed).

Processing of Batch Printing Mission

When the report is printed by a batch job, the following process is performed:

The CONTROL-D monitor reads a skeleton job from the library referenced by DD statement DADSKL of the CONTROL-D monitor. The member name is defined in the printing mission (e.g., PRTSKL).

Printing Mission Definition Screen

BATCH	Specifies whether a printing mission is submitted as a subtask of a Printers Control monitor, or as a batch job. Y (Yes) Current printing mission is submitted as a batch job instead of executing it as a subtask of the Printers Control monitor. N (No) Printing mission is executed as a subtask of the Printers Control monitor. Default.
SKELETON	Name of an existing member in the CONTROL-D SKL library. Mandatory if BATCH is set to Y.
FREE	Sysout allocation mode. Determines whether or not chunking is implemented and when sysout files will be freed. Optional. Valid values are: CLOSE Regular chunking mechanism is used. Default. END Sysout files are allocated as usual but are not freed until the end of the batch job. This value is available only if BATCH=Y is specified.
TIMEOUT	mm Number of minutes within which the mission should start. If the mission does not start during the specified time interval, it terminates NOTOK. Default: 5 minutes.

See Section 4 of the CONTROL-D User Manual for more information about the BATCH parameter.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

To use a printing mission in batch mode, a skeleton member must be prepared in the CONTROL-D SKL library. The skeleton member can be either a job skeleton, or a started task skeleton. See sample member PRTSKL in the CONTROL-D SKL library.

- The skeleton member can contain certain parameters which are interpreted by the CONTROL-D monitor. The parameters are:

%COM#%	Number of the COM file record assigned to this mission.
%MISSION%	Mission name.
%CATEGORY%	Mission category.
%GROUP%	Mission group.
%OWNER%	Mission owner ID.
%PRTY%	Mission priority.
%DEST%	Mission destination.

- When CONTROL-D finishes preparing the member for submission, control is passed to CONTROL-D user exit CTDX009. This exit can modify the contents of the submitted job (by adding, deleting or modifying the JCL). The exit can return the following return codes:

0 Submit the job.

4 Do not submit the job. User exit CTDX009 can request CONTROL-M to submit the job. Job submission can be done via CTMAJO routines to ensure correct handling of the print job under CONTROL-M.

8 Do not submit the job. Terminate the mission with NOTOK status.

Synchronization between the running job and the CONTROL-D monitor is achieved in the same manner as synchronization between CONTROL-D and the Printers Control monitor, i.e., a special control subtask exists for each printing mission in the CONTROL-D address space. This subtask is synchronized with the corresponding print task via the ENQ mechanism. The QNAME for this ENQ is taken from the CONTROL-D installation parameter PRTSTC (in member CTDPARM in the IOA INSTCTD library). To run a batch printing mission on a CPU which is different from the CPU in which the CONTROL-D monitor runs, this QNAME should be shared through GRS, MIM, or other enqueue manager products.

If parameter PRTMON# is set to 0 in member CTDPARM, no Printers Control monitors are started. In this case, parameter BATCH in the printing mission must be set to Y.

See "One-Outgroup Method" under "Printer Control" later in this section for additional considerations on running printing missions in batch mode.

Note Printing missionabend or timeout does not cause termination of CONTROL-D monitors. In such cases, the printing mission terminates with a status of NOTOK.

Advanced Scheduling Issues

Distribution According to Scheduling Dates

It is possible to organize printing missions so that each day's printing management (i.e., the printing workflow) is the same. The simplest solution is to specify DAYS=ALL in the Basic Scheduling parameters of all printing missions. Once a day, when the printing missions New Day procedure is invoked, all printing missions scheduled for the next 24 hours are placed in the Active Missions file.

This printing organization does not maximize the capabilities of CONTROL-D. You can organize the printing order in different ways for different days (e.g., a weekday, the end of the week, the end of the month, holidays). Using CONTROL-D it is possible to define several copies of the same printing mission. Each copy is an independent definition of the printing mission and should have different scheduling criteria, different runtime criteria, different bundling instructions, different printing configurations, etc.

Example

The (form) STD printing mission will start not later than 05:00 on every workday, will start not later than 04:00 at the end of the month, and will not be scheduled for holidays. Since there is no limit to the number of printing missions which can be defined, the user can organize the printing process to optimally suit every production day of the year.

Report Decollating and Printing Mission Dependency

It is recommended to establish dependency between the successful decollation of reports and the printing of bundles by CONTROL-D printing missions. Dependency can be established using the prerequisite condition concept. Parameter OUT of the report decollating mission should specify a prerequisite condition that is referenced by parameter IN of the printing mission. Using this method, bundles start printing only when all the reports which must be included in them have been decollated.

It is possible to define different report-printing dependencies for different scheduling dates (e.g., regular dates vs. end of the month). This can be done, as described above, by using multiple copies of the same printing mission, each one distinguished by different basic scheduling criteria and different runtime scheduling criteria (prerequisite conditions, time, priority, etc).

Printer Control

CONTROL-D provides the following methods for printing bundles:

- | | |
|---------------------|---|
| One-Chunk Method | Send entire bundle to spool for printing. |
| Multi-Chunk Method | Send a specified number of lines at time to the spool for printing. |
| One-Outgroup Method | Send one outgroup at the end of a printing mission that contains many sysout files. |

One-Chunk Method

The entire bundle is sent to the spool for printing at one time. This method is activated by specifying `CHUNKSIZE=0` in the printing mission definition. If not specified in the printing mission definition, `CHUNKSIZE` defaults to the value defined in parameter `PRINTER` during the installation process. Specifying `CHUNKSIZE=0` suppresses installation parameter `OUTGRP`.

This method can be used for printing reports that contain identical printing characteristics. If reports containing different printing characteristics are processed in this manner, the characteristics of the first report are used for all the reports in the bundle.

Note If the bundle contains reports for different printing destinations, a separate chunk is created for each destination but `CONTROL-D` does not monitor the printing of the chunk. `CONTROL-D` continues to send additional chunks of the bundle to the spool.

Multi-Chunk Method

`CONTROL-D` creates a new chunk each time the number of lines specified in the `CHUNKSIZE` parameter is exceeded, or when printing characteristics of the reports change, whichever comes first (unless `CHUNKSIZE` is specified as "0" as described in the One-Chunk method above). `CHUNKSIZE` can be specified in the printing mission definition. If it is not specified in the printing mission definition, `CHUNKSIZE` defaults to the value defined during the installation process. For the Multi-Chunk method, the value for the `CHUNKSIZE` parameter must be greater than 1; the recommended value is 10000.

Multi-Chunk processing provides the following major advantages:

- The ability to print reports with different printing characteristics in the same bundle.
- The ability to control the size of the chunk so the spool does not become overloaded.

Under the Multi-Chunk method, the printer must be set to `CONTROL-D` using the following JES commands.

Under JES2

A printer is normally defined to JES as `DEST=LOCAL` with one or more printing classes. To "assign" the printer to `CONTROL-D`, use a JES operator command to initialize the printer with the following parameters:

<code>NOSEP</code>	No separator; <code>CONTROL-D</code> creates its own banner pages.
<code>DEST</code>	A value set during <code>CONTROL-D</code> installation for each printer.
<code>SEPDS</code>	No dataset separator.
<code>CLASS</code>	All output classes on which <code>CONTROL-D</code> can print.

Sample JES2 operator command:

```
$TPRT1,S=N,R=U1001,SEPDS=N,WS=(-W,-Q,-PRM,-LIM/R,-F,-UCS,-FCB)
```

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Under JES3

Change the printer options using an operator command. Initialize the printer with the following parameters:

NOSEP No separator; CONTROL-D creates its own banner pages.
CLASS The class defined in CONTROL-D installation parameter PRINTCL.

Sample JES3 operator commands: (PRINTCL=Q)

```
*WTR OUT=04E,H=N,B=N,WC=Q,WS=CL
```

```
*S 04E
```

where * is the JES3 command prefix.

Verify that the printer has been modified successfully with all the above attributes.

One-Outgroup Method (JES2 Only)

It is possible to combine all the output files produced by a printing mission in one output group. Such a method is useful to ensure that all files will be printed together even if the JES printer is not dedicated to CONTROL-D and is used for other jobs as well as for CONTROL-D reports. It is also useful for output files that are sent through NJE to a remote computer.

To combine the output files, perform the following steps:

1. Run the printing mission as a batch job. (Set parameter BATCH to Y in the Printing Mission Definition screen.)
2. Free all sysout files at the end of the job (Set parameter FREE to END in the Printing Mission Definition screen.)
3. Note the following:
 - Parameters CLASS, DEST, WTRNAME, PRMODE and OUTGROUP should be the same for all reports included in the printing mission.
 - Parameter OVERRIDE in the Printing Mission Definition screen can be used to force parameters CLASS, DEST and WTRNAME.
 - Parameters PRMODE=PAGE and GROUPID=CONTROL-D should be defined in all OUTPUT statements in the Printers Control monitor JCL procedure (CTDPRINT).
4. Set parameter USERSET to YES in the OUTDEF statement in member JES2PARM in the SYS1.PARMLIB library.
5. Verify that the output class is a non-held class.

Opening/Closing Printers

During installation, a value of OPEN or CLOSE was assigned to parameter PRINTER for each printer. If the printer was defined as OPEN, skip to "Printing Process". If the printer was defined as CLOSE, use the following command to "open" a CONTROL-D printer:

```
F CONTROL-D,STARTPRT,printername
```

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Where `printername` is the name assigned during CONTROL-D installation. The following message is issued to the operator console from which the modify command was issued:

```
CTD130I COMMAND "STARTPRT" ACCEPTED FOR PRINTER printername
```

To "close" a CONTROL-D printer, enter the following operator command:

```
F CONTROL-D,STOPPRT,printername
```

When there are no CONTROL-D bundles to print, you can reset the printer's original parameters with a JES command.

Note It is not necessary to "close" CONTROL-D printers. A printer remains "open" but nothing is printed if the `JES2-DEST` or `JES3-CLASS` setting of the printer is not assigned to CONTROL-D.

Printing Process

When a printing mission is activated (either manually or automatically), the printing mission creates an index of all reports which are ready to print and which were specified to be printed by this particular printing mission. This index can be viewed and controlled online using the Print Control option (P) in the Active Missions screen. The bundle is then prepared for printing.

CONTROL-D searches for an available printer that was defined in parameter `PRINTER` of the printing mission definition. If parameter `PRINTER` has not been specified, CONTROL-D searches for any available printer that was defined in installation parameter `PRINTER`.

If a printer was defined as `CLOSE`, an operator command is required to `OPEN` the printer to CONTROL-D (as described earlier in this section). If a printer was defined as `OPEN` or if the operator command was issued, CONTROL-D sends the bundle to the spool for printing.

When using the One-Chunk method, the full bundle is created on the spool and waits in queue to print (along with any other output that may be on the spool - i.e., non-CONTROL-D output).

When using the Multi-Chunk method, it is recommended to send only CONTROL-D bundles to the printer at a specified time. If the printer is not assigned exclusively to CONTROL-D when using the Multi-Chunk method, non-CONTROL-D output may be printed in a CONTROL-D bundle. (Assigning a printer to CONTROL-D is described earlier in this section.) After the bundles finish printing, the printer can be assigned to print non-CONTROL-D output.

Bundle printing can be interrupted using the Print Control option in the Active Missions Status screen. This may be required to immediately print an important non-CONTROL-D output, to change priority of a report printed by CONTROL-D, or to perform any other desired action during the printing process. Refer to Section 2 of the CONTROL-D User Manual for a detailed description of this option.

Identifying CONTROL-D Chunks on Spool (JES2 Only)

CONTROL-D provides the ability to identify each chunk that is sent to the JES output queue in a unique way. This is done by setting the OUTGROUP field (GROUPID in JCL, O-GRP-N in SDSF) on the chunks that CONTROL-D sends to the spool.

The following are types of identifications which CONTROL-D can put into the OUTGROUP field:

1. The original job name of the report.
2. The user name of the user (e.g., recipient) who is to receive the report.
3. The printing mission name.
4. A unique time stamp.
5. Specific string specified in the printing mission.

To activate this feature, do the following:

1. Add a special OUTPUT statement to the CONTROL-D Printers Control monitor procedure (CTDPRINT). It must be coded as follows:

```
//OUTGROUP OUTPUT GROUPID=CONTROL-D
```

If the site runs more than one Printers Control monitor (CTDPRIN2, etc.), add this DD statement to all the Printers Control monitor procedures.

2. Add operand GROUPID=CONTROL-D to all OUTPUT statements in the Printers Control monitor procedure(s). This operand must be coded in the first operand in the OUTPUT statements.
3. Set the OUTGRP CONTROL-D installation parameter in member CTDPARM in the IOA INSTCTD library as detailed in the IOA Installation Guide.

We recommend specifying OUTGRP=Y, and using the GROUP field in the printing mission definition to define OUTGROUP processing. Refer to the CONTROL-D User Manual for details on using the GROUP field.

Printing on AFP (APA) Printers

AFP (Advanced Function Printing presentation) is the standard IBM laser printing technology which supports printers such as the 39xx family of printers and the 38xx family of printers, and uses APA (All Points Addressable) technology. Throughout this guide and in the CONTROL-D User Manual, these printers are referred to as AFP printers. The way in which CONTROL-D supports AFP printers is described. Additional information including instructions for local adaptations can be found in the step that implements AFP Support in the IOA Installation Guide, and in the publication "Implementing AFP in the CONTROL-D Environment."

When a sysout is printed on AFP printers, two additional printing control parameters are added: FORMDEF and PAGEDEF. These parameters cannot be defined in a standard DD statement. Instead, they are specified in an OUTPUT statement. This output statement is referenced by a DD statement.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Support for PAGEDEF and FORMDEF in CONTROL-D is implemented in two ways - one is parallel to the JCL approach (using a specified OUTPUT statement) and one is more flexible and easier to use (using the CDAM PAGEDEF/FORMDEF parameters).

Using the CDAM PAGEDEF/FORMDEF Parameters

Under CONTROL-D you can specify parameters FORMDEF and PAGEDEF without actually using an OUTPUT statement. When writing directly to a CDAM file, use the CDAM parameters PAGEDEF and FORMDEF. When you decollate a report from spool, you must specify the requested PAGEDEF/FORMDEF in the decollation mission definition PRINT/CDAM PARMS field.

Note This option is not supported when using the One-Chunk method. See "Printer Control" earlier in this section.

When using this option, during the printing process CONTROL-D looks for an OUTPUT statement named CONTROLD or CONTROLF and automatically inserts inline FORMDEF and PAGEDEF. This option utilizes the AFP Inline Resource Option.

A CONTROLD and CONTROLF OUTPUT statement must exist under all printing environments (the Printers Control monitor, the IOA Online monitor, the TSO user, etc.). See "AFP Support" in the IOA Installation Guide for site adaptations required for AFP support.

Using a Specified OUTPUT Statement

The name of an OUTPUT statement containing the FORMDEF/PAGEDEF combination can be specified. When writing directly to a CDAM file, use CDAM parameter OUTPUT. When decollating a report from spool, specify the requested OUTPUT statement in the decollating mission definition PRINT/CDAM PARMS field. Any OUTPUT specified in the PRINT/CDAM PARMS field overrides any such parameter specified in the DD statement during CDAM file creation.

Note The name of the OUTPUT statement used is the name under the Printers Control monitor and not the name of the OUTPUT statement in the job that created the sysout.

When using this option, CONTROL-D searches for the specified OUTPUT statement and uses it to print the report.

With this method, an OUTPUT statement must exist under all printing environments (the Printers Control monitor, the IOA Online monitor, the TSO user, etc.) for every combination of FORMDEF/PAGEDEF which is in use. See "AFP Support" in the IOA Installation Guide for site adaptations required for AFP support.

AFP Page Mode Output

Any AFP report (including Page Mode reports) can be decollated by CONTROL-D. Decollating includes identifying each and every page of a report with identifying strings and assigning the pages of the report to recipients accordingly. CONTROL-D recognizes page breaks in AFP Page Mode output (CATEGORY 5).

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Searching for a string in Page Mode output can be performed on the entire line of data, as supported by AFP (from column 1 to column 32767).

Page Mode output is usually produced by DCF (IBM's mainframe word processor) when making Composed AFP documents, by GDDM and other software tools (such as SLR) that use GDDM for producing graphic reports and PostScript output (translated to AFP Structured Fields), and by application programs.

Page Markers Under AFP

Page Markers for differentiating between user bundles (print perforation markers on page) can be initiated using standard methods with most printers. (See the "Implementing AFP in a CONTROL-D Environment" guide for details.) The exception is the 3800-3 printer. For special 3800-3 printer Page Markers, refer to member DOCDPAGM in the IOA SECUDATA library.

In-Stream AFP Control Statements

A major feature of CONTROL-D is the automatic insertion of AFP control parameters in-stream in reports. This feature is especially useful for the following situations:

- Implementation of advanced AFP options:

When using this feature, application programs do not need modification to enable insertion of AFP control parameters (structured fields) within reports.

- When AFP control parameters are contained in the sysout itself, at the beginning, and the sysout is decollated into different users' bundles:

In such a situation, the AFP control parameters would normally be missing from all but the first user bundle. This feature can verify that the AFP control parameters are automatically inserted in each user's bundle.

This feature operates as follows:

A special library is used to define the AFP control parameters of each report. This library is referenced by DD statement DAAPA of the Printers Control monitor. The original name supplied is OLPREFD.APAPARM.

The library can be defined as RECFM=V or RECFM=F. By default, it is defined as RECFM=V. RECFM=V is highly recommended, because if you are using an APAPARM library with RECFM=F, you should add four bytes of length (RDW) before every AFP control parameter and before command IMM/IDM.

The APAPARM library contains one member for each job which produces output to be printed on AFP printers. The member name must be identical to the job name. Each member contains AFP control parameters for all reports produced by the job.

Note *If you use the APAPARM option for Online printing, be sure your online environment (e.g., IOA Online Monitor, TSO Logon procedure) is referenced by DD statement DAAPA.*

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

- In each member, there must be one line in the following format for each AFP report produced by this job:

```
+++repname  (+++ starts in column 1)
```

Where:

+++	Identifies the line as a report name line.
repname	Name (or mask) of the report (maximum of 20 characters). repname must be the same as the name of the report specified in Report Decollating parameter DO NAME.

Note Any number of +++repname lines can be present in one member (i.e., a few reports can be produced by the same job).

- Following the +++repname line, one or more AFP control parameters records may follow. Any AFP command can be specified in its original AFP hexadecimal format. The following special commands can also appear as AFP control parameter records:

```
IMM=copy-group-name  (a maximum of eight characters)
```

```
IDM=page-format-name  (a maximum of eight characters)
```

These commands are then translated to the appropriate hexadecimal AFP structured fields IMM and IDM.

Note If you are using APAPARM libraries with RECFM=F, you should add four bytes of length (RDW) before every AFP control parameter and before command IMM/IDM.

Example

Consider the following report decollating mission parameters:

```
JOBNAME=ARINS1
  DO NAME=AR-REPORT-1
  DO NAME=AR-REPORT-2
```

Member ARINS1 in the CONTROL-D APARARM library can contain the following:

```
+++AR-REPORT-1
IMM=GROUP1
IDM=PAGFM1
+++AR-REPORT-2
IMM=GROUP2
```

When CONTROL-D prints the report named ar-report-1, the two lines following the +++ line are written at the beginning of the report.

When CONTROL-D prints the report named ar-report-2, the last line in the member is written to the report.

Sample member APARARMS is located in the CONTROL-D APARARM library.

Not every job has special AFP control parameters. The library should only contain members for jobs with in-stream AFP control parameters and not for other jobs.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

When the Printers Control monitor or a batch printing mission is about to print a report for a user on an AFP printer, it searches for a member name with the same name as the job which produced the report. It looks for the report name in the member, translates any IMM/IDM commands to hexadecimal, and writes all the report's AFP control parameters to the printer (via the spool). AFP printers must be defined as APA in member CTDPARM. (See "Performing Post-Installation Tasks" in the IOA Installation Guide for more information.)

Printing Using XEROX LCDS (DJDE) Parameters

For many XEROX printer models (9700, 8700, etc.), it may be necessary to specify different printing options using special control records which are inserted in the sysout directed to the printer. These parameters are usually referred to as DJDE parameters. The way in which CONTROL-D supports DJDE parameters is described below. Additional information including instructions for local adaptations can be found in the IOA Installation Guide.

A special library is used to define the DJDE parameters of each report. This library is referenced by DD statement DADJDE of the Printers Control monitor. The original name supplied is OLPREFD.DJDEPARM.

The CONTROL-D DJDEPARM library contains one member for each job which produces output to be printed on XEROX printers. The member name must be identical to the job name. Each member contains DJDE parameters for all reports produced by the job.

Note *If you use the DJDEPARM option for online printing, be sure your online environment (e.g., IOA Online monitor, TSO Logon procedure) is referenced by DD statement DADJDE.*

In each member, there must be one line in the following format for each report produced by this job:

+++repname	(+++ starts in column 1)
+++	Identifies the line as a report name line
repname	This is the name or mask of the report (maximum 20 characters). repname must be the same as the name of the report specified in report decollating parameter DO NAME.

One or more DJDE control parameters records may follow the +++repname line.

Note *Any number of +++repname lines can be present in one member (i.e., a few reports can be produced by the same job).*

Example

Consider the following report decollating mission parameters:

```
JOBNAME=ARINS1
  DO NAME=AR-REPORT-1
  DO NAME=AR-REPORT-2
```

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Member ARINS1 in the CONTROL-D DJDEPARM library can contain the following:

```
+++AR-REPORT-1
DJDE  C MULTI RECORD DJDE EXAMPLE
DJDE  FORMS=(XEROX1,1,3),FORMAT=XPDE12,FONTINDEX=1,NUMBER=(3,15,55),;
DJDE  COLLATE=YES,ASSIGN=(1,5),ASSIGN=(5,32),;
DJDE  FONTS=((P0612A),(P0812A)),;
DJDE  ASSIGN=(12,63),TOF=5,BOF=66,END;
+++AR-REPORT-2
DJDE  JDE=JDEAR3,JDL=JDLAR3,END;
```

When CONTROL-D prints the report named ar-report-1 (produced by the job ARINS1), the multi-record DJDE example lines are written to the report.

When CONTROL-D prints the report named ar-report-2, the DJDE line (JDE=JDEAR3) is written to the report.

A sample member JDEPARMS can be found in the CONTROL-D DJDEPARM library.

Note Not every job has special DJDE parameters. Many jobs are printed using the default DJDE parameters. The library should contain only DJDE parameters (i.e., members) for jobs with special DJDE parameters and not for other jobs.

When the Printers Control monitor is about to print a report for a user on a XEROX printer, it searches for a member name with the same name as the job which produced the report. It looks for the report name in the member, and writes all the report's DJDE parameters to the printer (via the spool). The DJDE prefix and offset should be specified in macro CTDUDJP. The printer must be defined as XER in member CTDPARM. (See "Performing Post-Installation Tasks" in the IOA Installation Guide for more information.)

Advanced ACIF Interface Facility

The CONTROL-D Advanced ACIF Interface facility automatically converts selected reports destined for PC file transfer to AFP Category 5 data stream (AFPDS) format, making them available for display with the CONTROL-D/PC for Windows AFP Viewing component.

What is AFPDS?

AFPDS (Advanced Function Presentation Data Stream - also known as an AFP Category 5 data stream) is a data stream format that allows for graphical WYSIWYG viewing, and printing of reports on AFP supported devices and platforms.

AFPDS can contain graphics, text, image and bar code objects. AFPDS also supports print resources such as fonts, overlays, page segments, form definitions and page definitions.

What is ACIF?

ACIF (AFP Conversion and Indexing facility) is a utility that is supplied as a standard component of PSF/MVS (Print Services facility) Version 2.1.1 and above.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Make ACIF Accessible to CONTROL-D

It is recommended that ACIF program module APKACIF be located in the LPA or in an MVS LINKLIST library. Otherwise, all CONTROL-D Printers Control monitor JCL procedures must be manually edited to concatenate the ACIF distribution LOAD library to the STEPLIB DD statement.

The default name of the ACIF distribution LOAD library (as documented in the IBM installation instructions for PSF) is PSF.ACIF.AAPKMOD1.

See the CONTROL-D installation procedure for MVS and PSF release prerequisites.

The major benefits of the ACIF utility are:

- **Converting Line Mode Output to Page Mode Output**

ACIF can convert line mode output to full AFP Category 5 data stream (AFPDS) page mode output. AFPDS output can be displayed under CONTROL-D/PC for Windows with the AFP Viewing component (technology from the IBM Printing Systems Company).

- **Indexing Reports**

ACIF can optionally index reports according to character strings at specified lines and positions within a report. For example, ACIF can produce an index according to account number and client name data contained in a customer statement. The index enables direct access to specific report pages when displayed with CONTROL-D/PC for Windows.

- **Saving/Archiving External Resources**

ACIF can create encapsulated AFPDS files that contain all the resources required for AFP printing and presentation. These resources include FORMDEFs, electronic overlays, page segments and fonts. The resources can be combined with the report and index data and saved as a standard sequential file. This file can be printed or viewed at a later date in its original format, without regard to resource availability or subsequent changes to the resource definitions. The file can also be transferred to other sites or platforms for viewing and printing. If transferred to the PC environment, the output can be viewed online using CONTROL-D/PC for Windows.

For more detailed information on ACIF, refer to the ACIF Application Programming Guide (IBM publication number G544-3824) or subsequent documentation.

Activating the Advanced ACIF Interface

By specifying CDAM parameter ACIF=YES in a report's decollating mission definition, CONTROL-D automatically calls ACIF during printing mission processing if the report is destined for PC file transfer. ACIF converts the report to AFPDS format. The report can then be transferred to CONTROL-D/PC for Windows for display.

ACIF Execution Parameters

ACIF requires parameters at execution time to control the processing of resources and conversion of report data to AFPDS format.

Many parameters specified to ACIF are the same as printing parameters specified in JCL statements when printing a job. Some parameters are used to specify the names of datasets where resource definition members can be found. Other parameters are available to define the report index that ACIF can optionally compile.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

These parameters must be read by ACIF from a sequential file or a library member. While some reports can share common parameter settings, most reports require settings that are in some way different from those specified for other reports. Without the CONTROL-D ACIF Interface facility, tracking hundreds or thousands of ACIF parameter files or members becomes an impossible task.

The ACIFPARM Library

For the CONTROL-D ACIF Interface facility, ACIF execution parameters are placed in the CONTROL-D ACIFPARM library.

Unlike standard executions of ACIF, the CONTROL-D ACIF Interface facility allows the specification of different parameters for different reports associated with the same job name, the same printing mission name, and/or the same recipient (user ID) name, to be kept in a single member of the CONTROL-D ACIFPARM library.

By default, ACIFPARM member names must be identical to a report's original job name. However, as an optional enhancement, the ACIF Interface facility can be modified to associate member names with a report's printing mission name or recipient name.

In each member, one or more sets of ACIF execution parameters may exist for different reports. Each such group must be preceded by a report name (repname) control statement in the following format:

```
-----1-----2-----3-----4-----5-----6-----+
+++repname          uuuuuuuu xpxp xsxs rprp rsrs dpdp dsds c
```

+++ Indication that this is a report name (repname) statement.

repname Name of the report (a maximum of 20 characters can be specified). The name must be the same as the name of the report specified in parameter DO NAME of the decollation mission that created the report.

The report name can be specified explicitly, or may contain the following mask characters:

* Any character, group of characters, or no character.

? Any single character.

Examples

ABC* The report name must begin with ABC (i.e., prefix).

*D The report name must end with D (i.e., suffix).

ABC*D The report name must begin with the prefix ABC and end with the suffix D. Any or no character(s) may be present between the prefix and the suffix.

* All report names are selected.

A?B1 The report name must begin with a prefix of A and end with a suffix of B1. One character of any value must be present between the prefix and the suffix.

unitname Work unit name to be used by the CONTROL-D ACIF Interface facility to allocate temporary ACIF index, resource and report data files.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

p-ind	Primary space allocation, in tracks, for the temporary ACIF index file.
s-ind	Secondary space allocation, in tracks, for the temporary ACIF index file.
p-res	Primary space allocation, in tracks, for the temporary ACIF resource file.
s-res	Secondary space allocation, in tracks, for the temporary ACIF resource file.
p-rep	Primary space allocation, in tracks, for the temporary ACIF report data file.
s-rep	Secondary space allocation, in tracks, for the temporary ACIF report data file.
c	JES output class to which ACIF prints its informational or error message report. If left blank or specified as an asterisk (*), the default MSGCLASS of the Printers Control monitor is assigned.

ACIF execution parameters must immediately follow the rename statement.

For detailed information on available ACIF execution parameters, refer to the ACIF Application Programming Guide (IBM publication number G544-3824) or subsequent documentation.

Any number of rename and ACIF execution parameter groups can be present in an ACIPARM library member.

Example

Consider the following decollation mission parameters for two reports:

```
JOBNAME=XXXXXXXXX
PRINT/CDAM PARMS=ACIF=YES
  DO NAME=REPORT-NAME1
  DO NAME=REPORT-NAME2
```

Member XXXXXXXXXX in the ACIFPARM library can contain the following:

```
+++REPORT-NAME1 .....
ACIF EXECUTION PARAMETER STATEMENTS
.
.
.
+++REPORT-NAME2 .....
ACIF EXECUTION PARAMETER STATEMENTS
```

Default member \$\$\$DFLT in the CONTROL-D ACIFPARM library contains default ACIF execution parameters. These parameters control how reports with no specifically associated member in the ACIFPARM library are processed by the ACIF Interface facility.

Using OUTPARM for Global Control of Printing Characteristics

The CONTROL-D Global Control of the Printing Characteristics option provides the ability to override the default printing characteristics (specified at decollation time) for all (or some) of the reports and banners to be printed by CONTROL-D. This option is for deferred printing as well as for immediate printing. The following printing characteristics can be controlled:

OUTPUT

FCB

FORMS

CLASS

DEST

MODIFY

FLASH

PAGEDEF

FORMDEF

CHARS

CHAR1

CHAR2

CHAR3

CHAR4

OPTCD

TRC

UCS

FLASHC

BURST

CHAR1, CHAR2, CHAR3, and CHAR4, define specific character sets. CHARS defines the character set referenced by CHAR1 and nullifies CHAR2, CHAR3, and CHAR4.

Setting parameter TRC (Table Reference Character) to YES produces the same result as including parameter OPTCD=J in the JCL.

The CONTROL-D OUTPARMS library is used to implement global control of printing characteristics. Each member in the OUTPARMS library refers to a job name (e.g., the member name is the job name), or each member refers to a user ID. All member names must either refer to job names or to user IDs - the two cannot be mixed. The members contain the printing characteristics that will be used for the reports. In addition, a special member named \$\$BANCHR is referred to for printing characteristics of banners (described later in this section).

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Refer to the sample members in this library for details on how to specify the printing characteristics of the reports and banners produced by CONTROL-D.

Note *Reports which are not specified in the OUTPARMS library will print with the default characteristics specified in the decollating mission or taken from the Permanent User file.*

As an example of how this option can be used, the user can send all the reports that belong to the accounting department to the local printer at the accounting department.

To control the printing characteristics of reports add parameters OUTPARM and BANNER to the banner exits (CTDX003 and CTDX014) in the IOA SECUDATA library. (See "OUTPARM Support" in Section 10 of this guide for more information on these parameters.)

To control the printing characteristics of banners, follow these steps:

1. Refer to the sample member SAMPBANN in the OUTPARMS library for details on how to specify the printing characteristics that will be used for banners. All printing characteristics listed above in "Using OUTPARM for Global Control of Printing Characteristics" can be used.

Create member \$\$BANCHR in the OUTPARMS library. This member is used to specify the printing characteristics that will be used for banners. The following is an example of this member:

```
+++STD*  
OUTPUT=BANNER1
```

In this example, all printing missions whose names start with STD will use the OUTPUT statement BANNER1 to determine the printing characteristics of the banners. If an asterisk is specified instead of STD *, all printing missions will use OUTPUT statement BANNER1 to determine the printing characteristics of banners.

2. Add OUTPUT DD statement(s) to the CONTROL-D Printers Control monitor procedure (CTDPRINT). For each OUTPUT statement add the appropriate printing parameters (PAGEDEF, FORMDEF, etc.). For example, assume that member \$\$BANCHR contains the following lines:

```
+++STD*  
OUTPUT=BANNER1  
+++STD?G  
OUTPUT=BANNER2
```

In this case, OUTPUT statements BANNER1 and BANNER2 must be added to the CTDPRINT procedure.

3. If the site runs more than one Printers Control monitor (CTDPRIN2, etc.), these OUTPUT statement(s) must be added each Printers Control monitor procedure.

Printing to a File

It is sometimes advantageous to direct output (a bundle) to a sequential file instead of to the printer. This function is supported under CONTROL-D using a supplied user exit (CTDX005) which is described in the following pages. This capability will be implemented as a standard function of the CONTROL-D Printing Mission Definition screen in a future release of CONTROL-D.

Activating the Facility

There are two recommended methods to route output to a sequential file. The first allocates a new file using a dynamic filename; the second writes the output to an existing file.

1. Allocating a New File

Type SEQL-FILE in the CATEGORY field of the Printing Mission Definition screen. A new dataset is created with the following dataset name format:

`prefix.mission-name.Dddmmyy.Tthmss`

where prefix is the value assigned to CONTROL-D exit 5 parameter PREFIX. (See "Tailoring CTDX005".) If no value is assigned to this parameter, prefix defaults to CTDPRT.

Note If a dataset name has been designated in the DSN parameter of CTDX005, then the name specified in the exit overrides the above default name. However, it is not recommended to specify a fixed name in the exit, since the user must then delete the file after using it to reallocate it the next time.

2. Using an Existing File

Type SEQL-filename in the CATEGORY field of the Printing Mission Definition screen. filename is the name (up to 15 characters in length) of an existing cataloged dataset.

The exit will concatenate the output to the existing file (i.e., it will write the output at the end of the existing file) (DISP=MOD). This option ignores the default parameters of the CTDX005 exit and utilizes the characteristics of the existing file, affording the user flexibility in specifying file characteristics. If the file does not exist, a new file is created (DISP=NEW) utilizing the macro default parameters (as in Option 1).

Tailoring CTDX005

Edit member CTDX005 in the IOA SECUDATA library. The following parameters related to printing to a file should be tailored to site requirements. Following the parameter list is a description of the parameters.

```
CTDUX005    UNIT=SYSDA,
            DSN=,
            PREFIX=,
            PRIVATE=NO,
            VOLCNT=,
            VOL1=XXXXXX,
            VOL2=,
            LABEL=,
            SPACETY=,
```

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

PRIMARY=20,
SECOND=,
RLSE=,
DEFER=,
DEN=,
LRECL=133,
BLKSIZE=1330,
RECFM=FBA,
RULER=NO,
DUMMY=NO,
FOB=NO,
EXPDT=,
TRTCH=

Parameters of Exit CTDX005

UNIT	Unit type - e.g., TAPE, 3380 (default is SYSDA).
DSN	Dataset name for the sequential file (used when the CATAGORY field of the printing mission Definition does not specify a dataset name in a SEQL= string). See the note included in option 1 ("Allocating a New File" above).
PREFIX	The default prefix of the sequential file name (the first qualifier in the dataset name). 1-7 characters in length.
PRIVATE	YES The file will be written to a specified volume. Default. NO The file will be written to a "scratch" tape.
VOLCNT	Maximum number of tape volumes to output dataset will require; used in conjunction with the PRIVATE=YES parameter.
VOL1	First volume serial number.
VOL2	Second volume serial number. Optional.
LABEL	Label type (SL, NL, NSL, AL, AUL, SUL, or BLP) for tape specification (the default is SL).
SPACETY	Allocation type - CYL or TRK (the default is TRK).
PRIMARY	Primary space allocation (the default is 20).
SECOND	Secondary space allocation (the default is 0).
RLSE	Unused space release specification.
DEFER	Deferred mounting specification.
DEN	Tape density- 1600/6250 (the default is 6250).
LRECL	Maximum logical record length (the default is 203).
BLKSIZE	Maximum block size (the default is 4060).
RECFM	Record format (the default is FB).

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

DUMMY	NO	The exit will be used to create a sequential file. Default.
	YES	The exit will not create a sequential file. It can be used for other special processing.
FOB	NO	FOB printers (see “Siemens Printer Support” in the IOA Installation guide) are not supported.
	YES	FOB printers (see “Siemens Printer Support”) are supported.
RULER	NO	Write the report to a sequential file without the ruler.
	YES	Write the report to a sequential file with the ruler.
EXPDT	yyddd	File expiration date.
TRTCH	COMP	Use compression mode when writing to cartridges.
	NOCOMP	Write without compression.

Backup Mission Management

Overview

Aspects of managing backup missions are explained under the following topics:

- Advanced Scheduling Issues
 - Backup According to Scheduling Dates
 - Report Decollating and Backup Mission Dependency
- Backup mission Workflow
 - Exception Handling
- Changing Backup Mission Retention Period

Advanced Scheduling Issues

Backup According to Scheduling Dates

It is possible to organize backup missions in such a way that each day's backup management (i.e., the backup workflow) is the same. The simplest solution is to specify `DAYS=ALL` in the Basic Scheduling parameters of all the backup missions. Once a day, when the backup mission Daily procedure is invoked, all backup missions scheduled for the coming 24 hours are placed on the Active Missions file.

This backup organization does not maximize the capabilities of CONTROL-D. You can organize the backup order in different ways for different days (e.g., a regular day, the end of the week, the end of the month, holidays). Via CONTROL-D it is possible to define several copies of the same backup mission. Each copy is an independent definition of the backup mission and should have different scheduling criteria, different runtime criteria, etc.

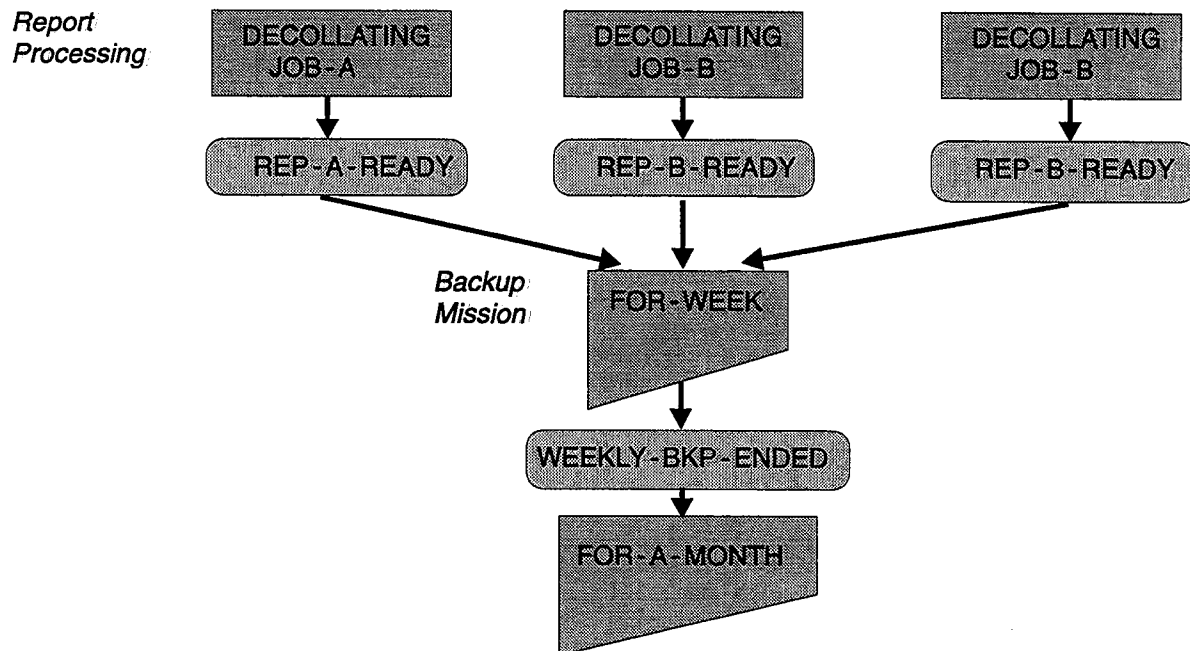
Example

Backup mission BKP0031D will start not later than 06:00 on every workday, will start not later than 05:00 at the end of the month, and will not be scheduled for holidays. Since there is no limit to the number of backup missions which can be defined, the user can organize the backup process to optimally suit every production day of the year.

Report Decollating and Backup Mission Dependency

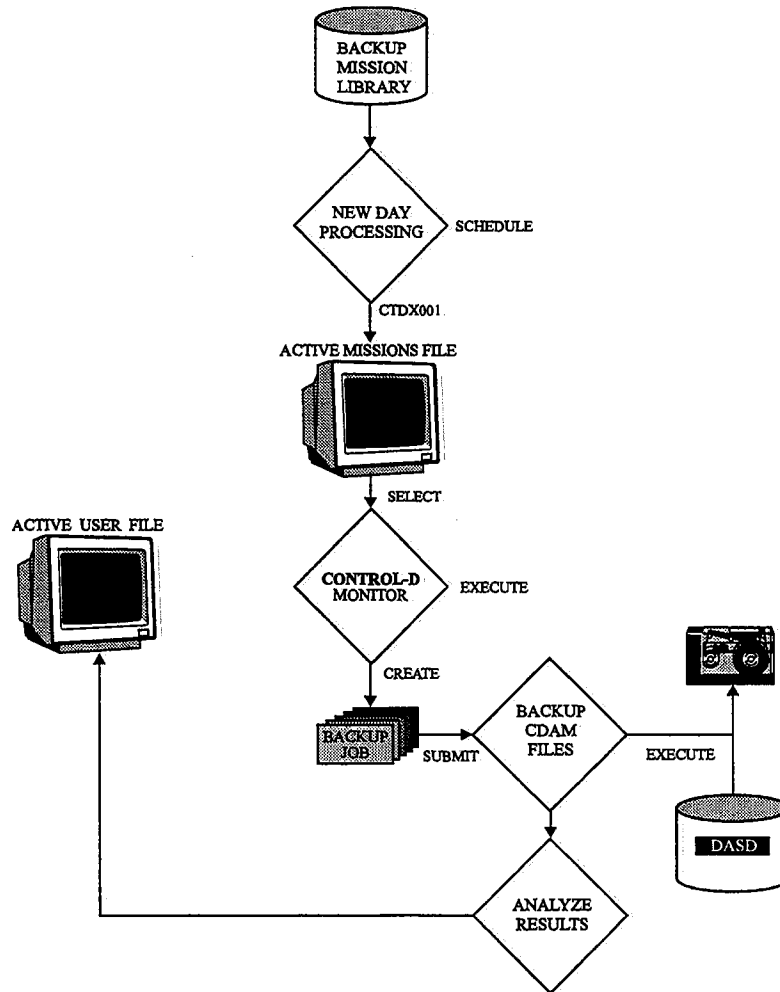
It is recommended to establish dependency between the successful decollating of reports and the backup process (by CONTROL-D backup missions). The dependency can be established using the prerequisite condition concept. Parameter OUT of the report decollating mission should specify a prerequisite condition that will be referenced by parameter IN of the backup mission. Using this method, backup occurs only when all the reports which should be backed up together have been decollated.

It is possible to define different report-backup dependencies for different scheduling dates (e.g., regular dates vs. end of the month). This can be done, as described above, by using multiple copies of the same backup mission, each one distinguished by different basic scheduling criteria and different runtime scheduling criteria (prerequisite conditions, time, priority, etc.).



Backup Mission Workflow

The backup mission workflow of CONTROL-D is shown below. It illustrates the stages involved from initially scheduling the backup mission through backing up the CDAM datasets.



The workflow of a backup mission is:

- The Active User Report List file is scanned by the CONTROL-D monitor for all reports requiring backup by the specified backup mission name. (This is a fast scan, not a scan of the entire file.) Reports which have been previously backed up, or have been restored from a backup, are not backed up a second time. The result of this process is a list of dataset names to be backed up.
- The CONTROL-D monitor reads a skeleton job from the library referenced by DD statement DADSKL of the CONTROL-D monitor. The member name must be the same as the backup mission name (e.g., BKP0031D).

Note Since the same library is used for migration, backup and restore mission skeletons, the name of each mission must be unique.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

When you define a new backup mission, you must create a new skeleton job with the same name.

The member must contain certain parameters which are interpreted by the CONTROL-D monitor. The parameters are:

<code>%DSNS%</code>	Resolves into a list of all the dataset names that have to be backed up. This list is in DF/DSS, DF/HSM, FDR, DMS/OS, ASM2 or ARCS format depending on installation parameters. Mandatory.
<code>%TIMESTAMP%</code>	Resolves into a timestamp, which is later used by CONTROL-D for internal purposes. Mandatory.
<code>%MISSNAME%</code>	Resolves into the backup mission name. Mandatory.
<code>%COND% cond-name</code>	When the preparation of the backup job is completed, the specified condition name is added to the IOA Conditions/Resources file. The ODAT of the condition is identical to the original scheduling date of the backup mission. This can be used for triggering the backup job under CONTROL-M. Optional.

- When CONTROL-D finishes preparing the member for submission, control is passed to exit CTDX010. This exit can modify the contents of the submitted job (by adding, deleting or modifying the JCL). For example, it is possible to use a format different from the DF/DSS, DF/HSM, FDR, DMS/OS, ASM2 or ARCS format for the dataset names list. The exit can return the following return codes:
 - 0 Submit the job.
 - 4 Do not submit the job (normally used when the submission is to be executed by a production control system).
- The member is written to the library referenced by DD statement DADJOB of the CONTROL-D monitor with a name identical to the name of the backup mission. Therefore, this library must be different from the library referenced by DD statement DADSKL. Depending on the return codes from the user exit, the job is, or is not, submitted. If CONTROL-M is installed, submission of the job can be triggered by the `%COND%` statement. (The advantage of this method is that overall production tape workload and tape usage priority are taken into consideration when submitting the backup job.)
- The original member supplied contains two steps:
 - The DF/DSS, DF/HSM, FDR, DMS/OS, ASM2 or ARCS backup utility.
 - A program which analyzes the output of the backup process and updates CONTROL-D files with the results (e.g., the tape numbers used).
- When the job finishes executing, the CONTROL-D monitor is notified (by the second step) and the backup mission's post-processing parameters (OUT and SHOUT) are processed.

Exception Handling

If the backup job abends in the first step (BACKUP), the status of the backup mission is changed to ENDED-NOTOK. When this happens, check why the abend occurred and correct the problem. After the problem has been corrected, rerun the backup mission from the Active Missions file.

If the backup job abends during the second step (ANALYZE), the status of the backup mission remains BACKUP IN PROCESS. When this happens reset the backup mission status to ENDED via job BKPRESSET in the CONTROL-D JCL library.

If the backup mission ended NOTOK because one or more CDAM files were not backed up successfully, reorder the mission to back up only the CDAM files which were not backup up successfully during the previous run. Rerunning the same mission (without reordering it) reprocesses all CDAM files of the previous run (including files which were successfully backed up).

Changing Backup Mission Retention Period

Backup missions use the following parameters to specify the retention period of a backed up CDAM file:

- | | |
|--------------------------|---|
| # OF DAYS TO KEEP | This specifies how many days should pass from the backup date to the expiration date. |
| # OF GENERATIONS TO KEEP | Specifies how many generations (executions) of the backup mission should pass until the generation expires. |

These parameters are specified in the backup mission definition screen. If there is a need to change from # OF GENERATIONS TO KEEP method to # OF DAYS TO KEEP method, a new backup mission should be used (e.g., the DO BACKUP statement in the Decollation definition mission should refer to a new backup mission name). If the same backup mission is used, an error message will appear and the backup mission will end NOTOK.

Migration Mission Management (CONTROL-V)

Overview

Various aspects of managing migration missions are explained under the following topics:

- Advanced Scheduling Issues
 - Distribution According to Scheduling Dates
 - Report Decollating and Migration Mission Dependency
- Migration Mission Workflow
 - Exception Handling

Advanced Scheduling Issues

Distribution According to Scheduling Dates

It is possible to organize migration missions in such a way that each day's migration management (i.e., the migration workflow) is the same. The simplest solution is to specify DAYS=ALL in the Basic Scheduling parameters of all migration missions. Once a day, when the migration missions Daily procedure is invoked, all migration missions scheduled for the coming 24 hours are placed on the Active Missions file.

This migration organization does not maximize the capabilities of CONTROL-V. You can organize the migration order in different ways for different days (e.g., a regular day, the end of the week, the end of the month, holidays). Using CONTROL-V it is possible to define several copies of the same migration mission. Each copy is an independent definition of the migration mission and should have different scheduling criteria, (e.g., different runtime criteria).

Since there is no limit to the number of migration missions which can be defined, the user can organize the migration process to optimally suit every production day of the year.

Report Decollating and Migration Mission Dependency

It is recommended to establish dependency between the successful decollation of reports and the migration process by CONTROL-V migration missions. The dependency can be established using the prerequisite condition concept. Parameter OUT of the report decollating mission should specify a prerequisite condition that will be referenced by parameter IN of the migration mission. Using this method, migration occurs only when all the reports which should migrate together have been decollated.

It is possible to define different decollation-migration dependencies for different scheduling dates (e.g., regular dates vs. end-of-month). This can be done, as described above, by using multiple copies of the same migration mission, each one distinguished by different basic scheduling criteria and different runtime scheduling criteria (prerequisite conditions, time, priority, etc.).

Migration Mission Workflow

The workflow of a migration mission is:

- The Active User Report List file is scanned by the CONTROL-D/V monitor for all reports requiring migration by the specified migration mission name. (This is a fast scan, not a scan of the entire file.) Reports which have previously migrated are not migrated a second time. The result of this process is a list of datasets to be migrated.
- The CONTROL-D/V monitor reads a skeleton job from the library referenced by DD statement DADSKL of the CONTROL-D/V monitor. The member name must be the same as the migration mission name (e.g., MIG0007Y).

Note Since the same library is used for migration, backup and restore mission skeletons, the names of the missions must be unique.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

When you define a new migration mission, you must create a new skeleton job (member) with the same name.

The skeleton job must contain certain parameters which are interpreted by the CONTROL-D/V monitor. The parameters are:

%DSNS% Resolves into a list of all the datasets that have to migrate. Mandatory. Used as input to the MIGRAT step.

%TIMESTAMP% Resolves into a timestamp which is later used by CONTROL-V for internal purposes. Mandatory. Used as input to the ANALYZE step.

%MISSNAME% Resolves into the migration mission name. Used as input to the ANALYZE step. Mandatory.

%COND% cond-name When the preparation of the migration job is completed, the specified condition is added to the IOA Conditions/Resources file. The ODAT of the condition is identical to the original scheduling date of the migration mission. This can be used for triggering the migration job under CONTROL-M. Optional.

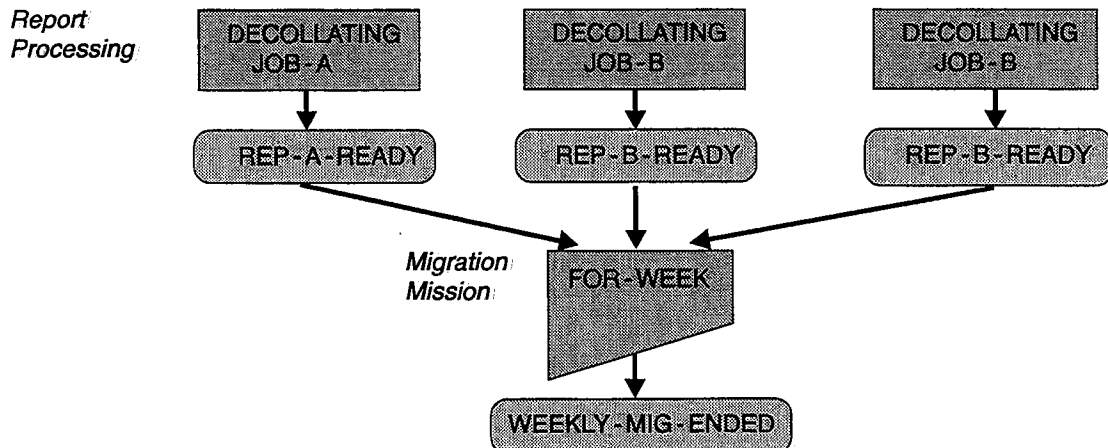
- When the CONTROL-D/V monitor finishes preparing the member for submission, control is passed to user exit CTDX010. This exit can modify the contents of the submitted job by adding, deleting or modifying the JCL. This exit can return the following return codes:

0 Submit the job.

4 Do not submit the job (normally used when the submission is to be executed by a production control system).

- The member is written to the library referenced by DD statement DADJOB of the CONTROL-D/V monitor with a name identical to the name of the migration mission. Therefore, this library must be different from the library referenced by DD statement DADSKL. Depending on the return codes from the user exit, the job is, or is not submitted. If CONTROL-M is installed, submission of the job can be triggered by the %COND% statement.

- When the job finishes executing, the CONTROL-D/V monitor is notified by the ANALYZE step and the migration mission's post-processing parameters (OUT and SHOUT) are processed.



Dual Migration

If the yes/no field in migration mission parameter SECONDARY is set to Y, the migration mission submits two migration jobs:

- The primary migration job uses the migration skeleton whose name is identical to the migration mission name. Parameter %LEVEL% is set to PRIMARY in the migration step and analyze step of this job. All migrated files created by this job have the prefix specified in media definition parameter DSNPREF.
- The secondary migration job uses the migration skeleton whose name is specified in name field of parameter SECONDARY. Parameter %LEVEL% is set to SECONDARY in the migration step and analyze step of this job. All migrated files created by this job have the prefix specified in media definition parameter SECPREF. See “Media Specific Parameters” in Section 6 of the IOA Installation Guide for more information about parameters DSNPREF and SECPREF.

See Section 4 of the CONTROL-V User Manual for more information about parameter SECONDARY.

By default, the primary migrated files accessed for viewing. Utility CTDUPBKP can be used to switch primary and secondary media information. After running the utility, the secondary migrated files are accessed for viewing. This utility can be run again to switch back to viewing the primary migrated files.

Information about the current migration media can be displayed in \$SYSDATA long format by using option A (Additional Info) in the Migrated Report list.

If a dual migration mission is rerun, both migration jobs will be resubmitted. However, if a migration job ended OK, the files migrated by that job will not migrate again. To re-migrate such files, the user must UNCATALOG the migrated reports before rerunning the migration mission.

Both the primary and secondary CDAM and index files are kept for the retention period defined in the migration mission definition. Retention information can be displayed in the \$SYSDATA long format.

Migration Media Definition

Each site may utilize any or all of the supported types of migration target media. Media-specific parameters (NAME, TYPE, DSNPREF, UNITNAME, DEVADDR, etc.) must be defined in member IOASPARM in the IOA INSTCTV library for each migration target media type (including DASD) that will be used. See “Installing CONTROL-V” in the IOA Installation Guide for details about these parameters.

The skeleton job for each migration mission specifies the migration target media. See “Migration Mission Skeletons”.

Cartridge (CART) Media Management

When migrating to media type CART, the migration process starts a new scratch volume for every run. The migration process places as many files as possible on the same volume.

Before writing each file, the migration process determines if the file will fit on the volume (based on the CART media capacity parameter CARTLEN specified in member IOASPARM). If the file will not fit, a new scratch volume is used. If the file is larger than the specified volume capacity, the migration process uses a new volume for that file and then uses as many volumes as needed (a maximum of five volumes).

Miscalculating the remaining space (e.g., if the specific volume was shorter or the compression factor was lower than expected), may cause the file to span more than one volume. This can result in slower retrieval response time.

Migration Mission Skeletons

In the CONTROL-D SKL library there is a skeleton member for each of the following migration media types:

CART	3480/3490/3490E Tape Cartridge.
DASD	Disks (e.g., 3380, 3390, fat-DASD 3390-9 or IBM 3995 Optical Library Dataserver model 153).
OSS	Kodak OPTISTAR Storage System (or compatible, e.g., DataWare OSS).
FTK	FileTek Storage Machine (SM).

All migration skeletons contain the following steps:

MIGRAT	Performs the migration.
ANALYZE	Runs a program which analyzes the output of the migration step and updates CONTROL-D/V files with the results.

The OSS migration skeleton contains the following additional steps which precede the steps describe above:

SRVD	Shuts down the OSS device in the IOA Online monitor to coordinate with the migration job.
OSSDBEXT	Creates the OSS Extract Dataset with information from the OSS Database.

Each migration mission must have a unique name and its own skeleton job. When you define a new migration mission, copy an appropriate existing skeleton and give it the same name as the new migration mission.

Migration to OSS

The EXEC statement in the OSS migration skeleton's SRVD and MIGRAT steps contains PARM='OSS'. Change OSS to the logical name of the target media specified in member IOASPARM in the IOA INSTCTV library.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

In the OSSDBEXT step, specify the tape name (VOLSER) of the OSS Database in parameter OSSDBVL.

The EXEC statement in the OSS migration skeleton's MIGRAT step contains PARM='OSS,NOCHECK'. It is recommended that you change NOCHECK to the prefix of a set of platters to which the reports should migrate. Meaningful prefixes should be used to create sets of platters containing reports with similar characteristics (e.g., viewed together, viewed with similar frequencies, viewed by same group of users). If the default NOCHECK is not changed, the report migrates to the first available platter.

DD statement SORTIN of the OSS migration skeleton's MIGRAT step must point to the OSS Extract Database for the media to which the reports should migrate. Use the name specified in member IOASPARM in the IOA INSTCTV library for the appropriate media. Change this parameter as required for migrating to various media.

Migration to Cartridge

The EXEC statement in the cartridge migration skeleton's MIGRAT step contains PARM='CART,NOCHECK'. Change CART to the logical name of the target media specified in member IOASPARM in the IOA INSTCTV library. Do not change the NOCHECK in this skeleton.

Migration to Disk, Fat-DASD, IBM 3995 Optical Library Dataserver

The EXEC statement in the DISK migration skeleton's MIGRAT step contains PARM='DISK,NOCHECK'. It is recommended that you change NOCHECK to the prefix of a set of VOLSERS to which the reports should migrate. Meaningful prefixes should be used to create sets of volumes containing reports with similar characteristics (e.g., viewed together, viewed with similar frequencies, viewed by same group of users). If the default NOCHECK is not changed, the report migrates to the first available VOLSER that belongs to the MVS UNIT specified in the media definition.

Migration to FileTek Storage Machine

The EXEC statement in the FTK skeleton's migrate step contains PARM='FTK1,NOCHECK'. Change FTK1 to the logical name of the target media specified in member IOASPARM in IOA INSTCTV library. Do not change the NOCHECK in this skeleton.

The following parameters must be specified in the FTKIN sysin file:

VSET Volume Set of the Storage Machine to be used for the migration mission.

FSET File Set of the Storage Machine to be used for the migration mission.

Both parameters must begin in column 1. No blanks are allowed.

Refer to the FileTek Storage Machine Documentation for information on how to determine the Volume Set and File Set.

If separate VSET and FSET values need to be specified for different groups of users, define a separate migration mission for each group.

Exception Handling

If the migration job abends in the MIGRAT step, rerun the job (either enter the JOB library and submit it, or if scheduled under CONTROL-M, use the CONTROL-M rerun option without restart).

If the migration job abends in the ANALYZE step, rerun the job from the first step (i.e., submit it again without restart).

If the ANALYZE step abends, the status of the migration mission remains MIGRATE IN PROCESS. To reset the migration mission status to ENDED, run the migration job from the ANALYZE step but change DD statement SYSIN to DUMMY. Sample job MIGRESET (which resets the migration mission) is in the JCL library.

If the migration mission ended NOTOK because one or more CDAM files did not migrate successfully, rerunning the same mission (without reordering it) reprocesses all the CDAM files of the previous run (including files which migrated successfully). Reordering the mission migrates only the CDAM files which did not migrate successfully during the previous run.

Rerunning a partially run migration job does not create duplicate copies of files on the target media. The migration job verifies that each file is cataloged (with the appropriate migration prefix). When running a previously ran cartridge migration job, data is not available for relative to absolute address conversion. To properly regenerate the data, uncatalog the file on the target media (cataloged by the previous run), and completely rerun the migration job.

Restore Mission Management

Overview

Various aspects of managing restore missions are explained under the following topics:

- Advanced Scheduling Issues:
 - Restoring According to Scheduling Dates
- Restore Mission Workflow
- Exception Handling
- Restoring with the Original Backup Utility

Advanced Scheduling Issues

Restoring According to Scheduling Dates

It is possible to organize restore missions in such a way that each day's restore management (i.e., the restore workflow) is the same. The simplest method is to specify DAYS=ALL in the Basic Scheduling parameters of all the restore missions. Once a day, when the restore mission Daily procedure is invoked, all restore missions scheduled for the coming 24 hours are placed on the Active Missions file.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

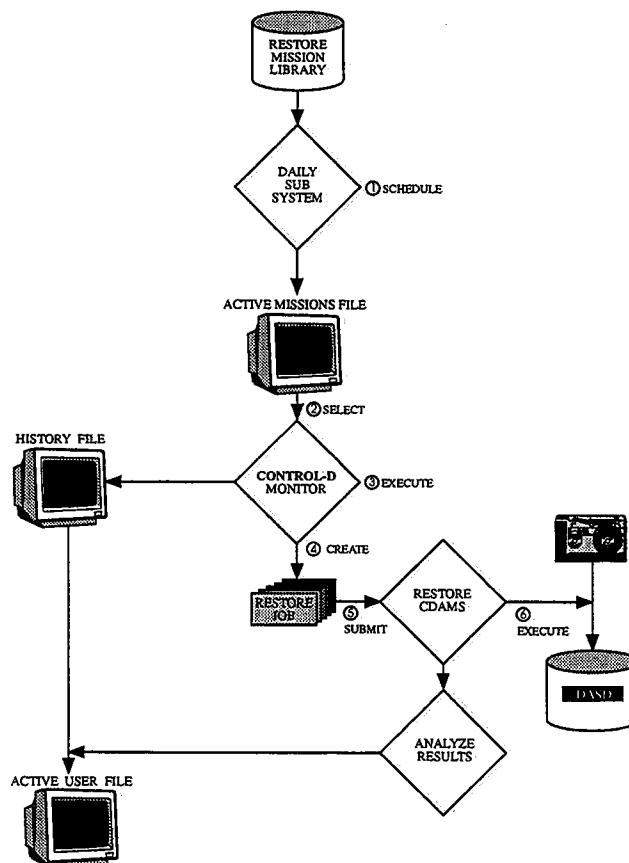
This restore organization does not maximize the capabilities of CONTROL-D. You can organize the restore order in different ways for different days (e.g., a regular day, the end of the week, the end of the month, holidays). Using CONTROL-D, it is possible to define several copies of the same restore mission. Each copy is an independent definition of the restore mission and should have different scheduling criteria, different runtime criteria, etc.

Example

The RSTNOON restore mission will start not later than 12:00 on every workday, will start not later than 11:00 at the end of the week, and will not be scheduled for holidays. Since there is no limit to the number of restore missions which can be defined, the user can organize the restore process to optimally suit every production day of the year.

Restore Mission Workflow

The restore mission workflow of CONTROL-D is shown below. It shows the stages involved from initially scheduling the restore mission through restoring the CDAM datasets.



The workflow of a restore mission is:

- The History User Report List file is scanned by the CONTROL-D monitor for all reports which need to be restored by the specified restore mission name. (This is a fast scan, not a scan of the entire file.) Restore is requested by online users via option R of the History Report List screen. The result of this process is a list of dataset names to be restored.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

- The CONTROL-D monitor reads a skeleton job from the library referenced by DD statement DADSKL of the CONTROL-D monitor. The member name must be the same as the restore mission name (e.g., RST0060M).

Note *Since the same library is used for migration, backup and restore mission skeletons, the names of the missions must be unique.*

When you define a new restore mission, you must create a new skeleton job with the same name.

- The member must contain certain parameters which are interpreted by the CONTROL-D monitor. The parameters are:

<code>%REPEAT%</code>	Beginning/end of a repeating step. Mandatory. Since outputs can be backed up on many volumes, it is sometimes necessary (with certain archiving products) to perform the restore by more than one step. The definition of this repeating step is found between the <code>%REPEAT%</code> and <code>%ENDREPEAT%</code> statements (not applicable to DF/HSM, DMS/OS, ASM2 or ARCS formats).
<code>%ENDREPEAT%</code>	
<code>%DSNS%</code>	Resolves into a list of all the dataset names that have to be restored. This list is in DF/DSS, DF/HSM, FDR, DMS/OS, ASM2, or ARCS format, depending on installation parameters. Mandatory.
<code>%VOLUMES%</code>	Resolves into the volume serial numbers of the tapes taking part in the restore. Mandatory (applies to DF/DSS or FDR formats).
<code>%TIMESTAMP%</code>	Resolves into a timestamp which is later used by CONTROL-D for internal purposes. Mandatory.
<code>%MISSNAME%</code>	Resolves into the restore mission name. Mandatory.
<code>%COND% cond-name</code>	When the preparation of the restore job has been completed, the specified condition name is added to the IOA Conditions/Resources file. The ODAT of the condition is identical to the original scheduling date of the restore mission. This can be used for triggering the restore job under CONTROL-M. Optional.
<code>%BKPUTIL%</code>	Resolves into the backup utility name. Optional.

The value of this parameter is determined in the following manner:

- The skeleton member is searched for the BKPUTIL assignment (e.g., BKPUTIL=HSM). The value assigned for this parameter is referred to as the backup utility name.
- If BKPUTIL is not set in the skeleton member, the value is taken from installation parameter BKPUTIL (in member CTDPARM in the IOA INSTCTD library).

Note *By default, all the sample skeleton members provided in the CONTROL-D SKL library already have the BKPUTIL parameter assigned with the appropriate backup utility.*

See “Restoring with the Original Backup Utility” later for details on how to use this parameter.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

- When CONTROL-D finishes preparing the member for submission, control is passed to user exit CTDX011. This exit can modify the contents of the submitted job (by adding, deleting or modifying the JCL). For example, it is possible to use a format different from the DF/DSS, DF/HSM, FDR, DMS/OS, ASM2 or ARCS format for the dataset names list. The exit can return the following return codes:
 - 0 Submit the job.
 - 4 Do not submit the job (normally used when the submission is to be executed by a production control system).
- The member is written to the library referenced by DD statement DADJOB of the CONTROL-D monitor, with a name identical to the name of the restore mission. Therefore, this library must be different from the library referenced by DD statement DADSKL. Depending on the return codes from the user exit, the job is, or is not, submitted. If CONTROL-M is installed, submission of the job can be triggered by the %COND% statement. (The advantage is that overall production tape workload and tape usage priority are taken into consideration when submitting the restore job.)

The original member supplied is divided into:

- The DF/DSS, DF/HSM, FDR, DMS/OS, ASM2 or ARCS restore utility. A step is created for each group of tapes (VOLSERs) containing reports to be restored (not with all products).
- A program which analyzes the output of the restore process and updates CONTROL-D with the results.

This step copies the restored user/report entries from the History User Report List file to the Active User Report List file and indicates that a restore has been performed.

- When the job finishes executing, the CONTROL-D monitor is notified (by the second step) and the restore mission's post-processing parameters (OUT and SHOUT) are processed.

Exception Handling

If the restore job abends in one of the first steps (such as the RESTORE step), rerun the job (enter the CONTROL-D JCL library and submit it or if scheduled under CONTROL-M, use the CONTROL-M rerun option without restart).

If the restore job abends in the last step (ANALYZE), rerun the job from the last step (i.e., submit it again without restart).

If the last step abends, the status of the restore mission remains RESTORE IN PROCESS. To reset the restore mission status to ENDED, run job RSTRESET (which resets the restore mission) from the CONTROL-D JOB library. In addition, all the restore requests should be performed again (using option R in screen U, in the History User Report List file). The restore mission finishes with ENDED NOTOK status.

Restoring With the Original Backup Utility

Sites which are moving from one backup utility to another (e.g., from FDR to DF/HSM) might end up having CDAM files which are backed up by different backup utilities. Each of the backed up CDAM files should be restored using the original backup utility.

To restore the CDAM file with the appropriate backup utility, the restore mission name should point to a skeleton member of the original backup utility. Parameter BKPUTIL in the skeleton member should be set to the original backup utility name.

Example

```
/**      BKPUTIL=FDRCAT
```

This statement should appear in the skeleton member to use the FDR backup utility regardless of the value specified for BKPUTIL in member CTDPARM in the IOA INSTCTD library.

Note This overrides the value specified to BKPUTIL in CONTROL-D installation parameter member CTDPARM in the IOA INSTCTD library.

IOA Archive Server

General

The IOA Archive server enables online viewing and immediate printing of reports and indexes which have migrated to non-disk storage media. The server provides access to the following media types:

CART	3480/3490/3490E Tape Cartridge.
DASD	Disks (e.g., 3380, 3390, fat-DASD 3390-9 or IBM 3995 Optical Library Dataserver model 153).
OSS	Kodak OPTISTAR Storage System (or compatible, e.g., DataWare OSS).
FTK	FileTek Storage Machine (SM).

There is no connection or dependency between the CONTROL-D/V monitor and the IOA Archive server. Each can be brought up and down independently.

Device Definition and Usage

Cartridge (CART) Media Definition

The cartridge devices defined in member IOASPARM in the INSTCTV library for media type CART can be used in the following two modes:

- Dedicated
- Dynamic

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Dedicated Device Mode

Dedicated devices are allocated to the server during the entire time the device is connected to the server. Devices can be defined explicitly by parameter DEVADDR, or implicitly by parameter DEVQTY. Devices can be connected (STARTed) or disconnected (STOPed) at any time using operator commands.

When using this mode, the number of devices available for the server is known. The response time for user requests can be predicted based on the server's workload.

Dynamic Device Mode

Dynamic devices are allocated to the server when needed, and freed when no requests are pending. Dynamic devices are defined implicitly by parameter DEVQTY. This parameter defines the maximum number of devices that are allocated when the server is started. The allocation is done upon request. No device is allocated until the first access is issued by an online viewing user. Parameter DYNDEVRLSE defines when the device will be freed after requests have been processed:

DYNDEVRLSE=(method, idle time)

Where:

method	SWITCH or NOSWITCH. Indicates whether to free the device when a request for another volume is to be processed (SWITCH), or to process requests for all volumes before releasing the device (NOSWITCH). Setting this parameter to SWITCH may cause a device to be released and allocated to another job when there are still requests pending.
idle time	Number of minutes to hold the device after the last request is processed. This parameter ensures that if a request is issued within the specified time interval, the device will still be available to the server. This parameter applies to both SWITCH and NOSWITCH methods.

KODAK/DataWare OSS Usage

Updating the OSS Database

Before updating the OSS database by adding/deleting virtual tapes or importing/exporting platters, stop the IOA Archive Server or the appropriate media in the IOA Archive Server.

When the update is complete, run job LODOSSDB from the IOA INSTCTV library to create the OSS Extract Dataset file. After this job completes, reactivate the IOA Archive Server or, if only the media was stopped, restart the media.

FileTek Storage Machine Media Definition

CONTROL-V communicates with the FileTek Storage machine via an application interface.

The Resources (devices) defined in IOASPARM for media type FTK are logical devices. A logical device task is activated whenever a request is made to access a dataset that is not already being handled by another logical device. Each device task handles only one dataset at a time.

FTK Media Definition

A special set of parameters is used to define FTK media during the CONTROL-V installation. Valid parameters are:

MAXCONN Maximum number of logical devices that can be active at any one time for the media. Each logical device opens a session with the Storage Machine (SM). See the documentation supplied with the FileTek Storage machine for information on how to determine an appropriate number of maximum sessions for the media.

Note If more than one media has been defined for the same SM, you must verify that sum of the values specified for parameter **MAXCONN** in all media definitions for the SM does not exceed the maximum number of sessions that the SM can support.

SMNAME Name of the FileTek Media being defined.

SUBSYS Name of the FileTek Host Storage Machine subsystem.

ACCOUNT Storage Machine Account ID used by CONTROL-V to open a session with the FileTek SM.

PASSWORD Password for the specified ACCOUNT.

GROUP An Access Group name for files migrated to this media.

GROUPPWD Used to specify passwords for access to files in the specified GROUP. Separate passwords can be specified for READ, WRITE, and DELETE access of the files in the GROUP.

Using the above parameters you can create several different definitions for FTK media. Some possible reasons for creating more than one definition for FTK media include:

- You have more than one FileTek Storage Machine at your site.
- You want to maintain several different groups of datasets on the same SM (e.g., with different passwords). To accomplish this, define the different FTK media with the same subsystem (SUBSYS) and storage-machine (SMNAME) name, but with different group names (GROUP) and group passwords (GROUPPWD).

The parameters used for FTK media definition are described in detail in the CONTROL-V section of the IOA Installation Guide.

Media and Resource Control

The use of each media and resource defined to the IOA Archive Server can be controlled using operator commands:

- Each media with all its resources can be stopped and restarted.
- Resources (devices, sessions, etc.) can be connected to or disconnected from the IOA Archive Server.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Operator commands for media and resource control are described as follows:

Media Control

Each media defined to the IOA Archive Server can be stopped and restarted separately.

To stop a media with all its resources, issue the operator command:

```
F IOASMON,STOP MEDIA=media-name
```

When the media is stopped, an appropriate message is displayed on the operator console.

To start a media, issue the following operator command:

```
F IOASMON,START MEDIA=media-name
```

When the media is ready to receive requests, an appropriate message is displayed on the operator console.

Resource Control

Resources and their usage differ by media type.

OSS Media - KODAK/DataWare OPTISTAR Storage Subsystem

The OSS works in 3480 emulation, and its resources are "3480" devices. The devices are referred to by device number. Each device defined to the IOA Archive Server can be stopped and restarted separately.

To stop an OSS device, issue the operator command:

```
F IOASMON,STOP MEDIA=media-name,DEVICE=device-number
```

When a device stops, an appropriate message is displayed on the operator console.

To start an OSS device, issue the operator command:

```
F IOASMON,START MEDIA=media-name,DEVICE=device-number
```

When a device is ready to process requests, an appropriate message will be displayed on the operator console.

Note Before starting an OSS device, make sure that the device is online and ready in the system.

CART Media - 3480/3490/3490E Tape Cartridge Subsystem

The resources are actual 3480 devices. Devices are referred to by device number, or by specifying the quantity of devices and letting the IOA Archive Server decide which to use or free. Each device defined to the IOA Archive Server can be stopped and restarted separately.

To stop a specific device, issue the following operator command:

```
F IOASMON,STOP MEDIA=media-name,DEVICE=device-number
```

When the device stops, an appropriate message is displayed on the operator console.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

To start a specific device, issue the following operator command:

```
F IOASMON, START MEDIA=media-name, DEVICE=device-number
```

Note Before starting a CART device, make sure that the device is online and ready in the system.

This command can be issued only when the devices are dedicated to the IOA Archive Server. If the devices are used dynamically, start a device by specifying device quantity. When the device is ready to process requests, an appropriate message is displayed on the operator console. To stop one/some/all devices of a specific media, issue the operator command:

```
F IOASMON, STOP MEDIA=media-name, DEVQTY=number/ALL
```

where number can be any integer from 1 through 99.

As each device is stopped, an appropriate message is displayed on the operator console.

To start one/some/all devices of a specific media, issue the operator command:

```
F IOASMON, START MEDIA=media-name, DEVQTY=number/ALL
```

where number can be any integer from 1 through 99.

As each device becomes ready to process requests, an appropriate message is displayed on the operator console.

Note Before starting the devices, make sure enough devices are online and ready in the system.

FTK Media - FileTek Storage Machine (SM)

The resources (devices) defined in IOASPARM for media type FTK are logical devices. A logical device task is activated whenever a request is made to access a dataset that is not already being handled by another logical device. Each logical device handles only one dataset at a time.

Up to 100 logical devices can be activated at one time depending on the value specified for CONTROL-V installation parameter MAXCONN, and optionally, the maximum set via a START or STOP operator command. If more than MAXCONN requests were made concurrently, they will wait in a queue for an available logical device.

Logical Device Status

Logical devices are each assigned one of the following statuses:

ACTIVE/BUSY	The device is now processing a request for file access.
IDLE	The device processed a request recently and is now waiting for another request.
INACTIVE	The device is not processing any requests.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

When a logical device finishes processing a request, it checks the DSN queue for the next request. If no request is found, the logical device enters the IDLE status. If no request is detected within a limited time period, the device is assigned a status of INACTIVE.

Note *If no requests are being processed and the number of usable devices is one or more, device number one retains a status of IDLE (i.e., its status will not be set to INACTIVE).*

Determining the Number of Usable Devices

As mentioned above, the maximum number of devices is defined via installation parameter MAXCONN.

You can, however, limit the number of devices to be used (usable devices) via the operator commands.

To reduce the number of usable devices, issue the following operator command:

```
F IOASMON, STOP MEDIA=media-name, DEVQTY=num/ALL
```

Where num is the number of logical devices to be brought down. num can be any integer from 1 through 100.

The STOP command determines which devices should be stopped, according to their statuses.

- The specified number of INACTIVE devices are stopped.
- If the number of devices to be stopped is greater than the number of devices that are presently INACTIVE, IDLE devices are stopped as well.
- If the number of devices to be stopped is greater than the number of devices that are either INACTIVE or IDLE, BUSY/ACTIVE devices are stopped as well.

BUSY devices to be stopped complete the request they are processing, before they accept a status of INACTIVE.

To increase the maximum number of devices, use the following operator command:

```
F IOASMON, START MEDIA=media-name, DEVQTY=num/ALL
```

Where num is the number of logical devices to be brought up. num can be any integer from 1 through 100.

Note *The maximum set by the START command cannot be greater than the number specified for CONTROL-V installation parameter MAXCONN.*

Starting/Stopping a Specific Logical Device

To stop a specific ACTIVE (or IDLE) device, issue the following operator command:

```
F IOASMON, STOP MEDIA=media-name, DEVICE=device-number
```

When the device is stopped, the maximum number of usable devices is reduced by 1, and an appropriate message is displayed on the operator console.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

To start a specific INACTIVE device, issue the following operator command:

```
F IOASMON, START MEDIA=media-name, DEVICE=device-number
```

When the device is started, the maximum number of usable devices is increased by 1, and an appropriate message is displayed on the operator console.

Note The maximum set by the START command cannot be greater than the number specified for CONTROL-V installation parameter MAXCONN.

Since the devices defined for the FTK type media are logical devices (i.e., not physical devices), starting or stopping any specific device has the same effect as using the DEVQTY parameter to increase/decrease the maximum number of usable devices by one.

Displaying Media Information

Information about a media can be displayed using the command:

```
F IOASMON, DISPLAY MEDIA=media-name
```

The information displayed as a result of this command includes media and resource information. The media information includes details such as:

- Media type
- System unit name/type to be used for this media
- Number of pending requests
- Type and number of resources

Resource information varies by media type but includes details such as:

- Device number
- Device use type (DEDICATED/DYNAMIC)
- Device status (ACTIVE/BUSY/INACTIVE/IDLE)
- Details about the current process, if there is one (e.g., user ID, volser, or dsn)

FTK Media

Resource information for FTK media includes:

- Maximum number of usable devices
- Number of devices that are active (with a status of BUSY, ACTIVE, or IDLE)
- Status (displayed for each logical device)
- User-ID and the name of the dataset being processed (displayed for each logical device)

Devices that are no longer usable, or are currently INACTIVE, are displayed on the console with a status of INACTIVE.

Problem Determination

The IOA Archive Server is supplied with internal debugging facilities. Under normal circumstances, these facilities are dormant. However, if required (e.g., your IOA representative requests debugging information), it is possible to activate the debugging facility as follows:

1. The debugging level can be set when starting the IOA Archive Server or during normal operation. To start the debugging facility while starting the IOA Archive Server, use the following command:

```
S IOASMON,DEBUG=level
```

To start the debugging facility during operation, use the following command:

```
F IOASMON,SET DEBUG=level
```

The debugging level can be any value from 0 (no debugging) to 255. The required debugging level, and instructions to set it with a START or MODIFY command, will be supplied by your IOA representative.

When the IOA Archive Server debugging facility is successfully activated, the following message is displayed on the operator console:

```
IOA128I IOASMON - DEBUG LEVEL IS SET TO level
```

Note It is strongly recommended that you do not activate the IOA Archive Server with parameter *DEBUG* on a regular basis for the following reasons:

- Spool utilization is greatly increased if a large amount of debugging information is generated.
- In case of a JES problem, the IOA Archive Server may become suspended while waiting for JES.

-
2. Debugging information is printed to DD statements PRTDBG and DADUMP of the IOA Archive Server procedure.
 3. When problem determination procedures are completed, turn off the debugging facility with the operator command:

```
F IOASMON,SET DEBUG=0
```

When the IOA Archive Server debugging facility is successfully deactivated, the following message is displayed on the operator console:

```
IOA128I IOASOM - DEBUG LEVEL IS SET TO 0
```

User Report List File Management

Overview

The following User Report List files are contained in the CONTROL-D repository:

- Permanent User Report List file
- Active User Report List file

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

- History User Report List file
- Migrated User Report List file (CONTROL-V only)

The contents of each file are described in detail.

The files are managed by the IOA Access Method. Each file consists of two separate sequential datasets:

- A Data component, where the actual record information resides.
- An Index component, which provides keyed access to records in the Data component.

Both Data and Index components may extend beyond one physical dataset (depending on whether or not automatic extension was specified for the file during allocation). Initially, a component is allocated as a single physical dataset called the primary extent.

See the explanation on IOA Access Method in Section 2 of this guide for more information.

The file naming convention is:

prefix.dbname.Ennn

where:

prefix Set to CONTROL-D installation parameter DBPREFD.

dbname Set to file identifier name, as follows:

HST CONTROL-D History User Report list Data component.

HSTI CONTROL-D History User Report list Index component.

ACT CONTROL-D Active User Report list Data component.

ACTI CONTROL-D Active User Report list Index component.

PRM CONTROL-D Permanent User Report list Data component.

PRMI CONTROL-D Permanent User Report list Index component.

MIG CONTROL-V Migrated User Report list Data component.

MIGI CONTROL-V Migrated User Report list Index component.

Ennn Extent sequence number suffix. The suffix consists of the letter E followed by nnn, a number from 000 to 255, indicating the extent sequence number.

Example

If CONTROL-D installation parameter DBPREFD contains the value CTD, the following file names are assigned for the Active User Report file:

CTD.ACT.E000 For the Data component.

CTD.ACTI.E000 For the Index component.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Each component uses its own file definition member which resides in the IOA INSTCTD library. The member includes the file's physical and logical attributes. The member name is DEFxxx for a Data component, or DEFxxxI for an Index component, where xxx is the dbname as described previously. The member is used by utility IOADBF. Additional utilities are provided to maintain these IOA Access Method files. See the Utilities section later in this guide for more details about these utilities.

The files can be activated with dual (mirror image) file support. The dual file name is similar to the main file name, except for the suffix (Dnnn instead of Ennn). The Dual file support can be defined for both the Data component and Index component. It is recommended that this feature be activated only for the Data component to reduce overhead, since the Index component can be built from the Data component.

See "CONTROL-D Repository Maintenance" later in this section for details about maintaining these files.

Permanent User Report List File

The Permanent User Report List file can be referred to as a report catalog. It contains, for each user, the list of reports the user (recipient) may receive at any time, their copy count (for the user), the print destination (if it is not one of the main computer printers such as LOCAL), and other information.

The Permanent User Report list file is used by end users (recipients) to permanently define the number of (printed) copies they want to receive from each report (in a given category), to request that the report is printed on a local printer, to assign a copy of the report to another user, and to permanently define other report list defaults.

Once a day, the report list in the Permanent User Report List file can be copied to the Active User Report List file. The user, for example, may change the copy count of his/her reports on the Active User Report List file, but that change remains in effect only for that day. On the next day, the reports of each user are copied again from the Permanent User Report List file to the Active User Report List file with the original (permanent) copy count. Using this simple method, the user can either set the copy count of a report permanently by modifying it in the Permanent User Report List file (using the Online Permanent User Report list screen), or temporarily modify it for a specific day in the Active User Report List file (using the Online Active User Report List screen). All other updatable fields (parameters) work the same way.

There are two ways to access the data stored in this file:

- Using utility CTDCP2A
- Using the decollation mission definition SEARCH parameter

Utility CTDCP2A

This utility is used to copy the list of reports from the Permanent User Report List file to the Active User Report List file. Newly-made changes to the Permanent User Report List file are then reflected in the Active User Report list file. Each report copied to the Active User Report List file has an original scheduling date associated with it. The original scheduling date is passed as a parameter to the utility. If the parameter is omitted, the date in the computer is used. (For more details, refer to Utility CTDCP2A in Section 9 of this guide.)

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

Running this utility more than once a day (e.g., after a computer crash) does not create duplicate entries. You can run this utility before or after the New Day procedure.

SEARCH Parameter

This is the recommended method for accessing the Permanent User Report List file. By setting this parameter to P (SEARCH=P) in the report decollation mission definition, the relevant Permanent User Report List file entries are accessed during the decollation process to retrieve the values of the report parameters. See Section 3 of the CONTROL-D User Manual for a description of parameter SEARCH.

Utility CTDCA2P

Utility CTDCA2P copies newly generated report/user entries from the Active User Report List file to the Permanent User Report List file (which maintains the list of all reports that can be created at any specific time).

New report/user entries are created on the Active User Report List file when a new recipient is detected in a certain report (e.g., when a new bank branch is opened). This entry has the default copy count of the report as defined in the report decollating mission. The utility copies the entries to the Permanent User Report List file, thereby automatically updating the User Report list on the Permanent User Report List file. It does not update User/Report entries that already exist in the Permanent User Report List file.

Utility CTDCA2P can be run more than once a day (e.g., after a computer crash). Duplicate entries are not created.

It is recommended that the utility be run once a day, preferably before the New Day procedure is invoked.

Active User Report List File

The Active User Report List file contains entries with the following report status:

- Reports due to be decollated (in WAIT DECOLLATION status). These entries are created by utility CTDCP2A.
 - The user can change the copy count and print destination.
 - The user can add additional report recipients.
- Reports which are decollated but are waiting to be bundle printed (in WAIT PRINT status):
 - The user can view the reports.
 - The user can change the copy count, print destination and any other report characteristics.
- Reports which are decollated but will not be bundle printed (in DECOLLATED status):
 - The user can view the reports.
 - The user can request an immediate print request.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

- Reports which have been printed (in PRINTED or PRINTED-WAIT BKP status):
 - The user can view the reports.
 - The user can request an immediate print request.
- CONTROL-V reports which are decollated but have not yet migrated away from DASD storage, or reports that have been migrated after the last run of utility CTDDLRP (i.e., reports in WAIT MIGRATE status):
 - The user can view the reports.
 - The user can request an immediate print request.

The Active User Report list files may contain reports from more than one day. Therefore, these files can contain more than one entry for each report/recipient/category. Each entry is distinguished by its original scheduling date and its decollation date.

Migrated User Report List File

The CONTROL-V Migrated User Report List file contains a list of all Migrated reports that have been deleted from DASD. These reports are in MIGRATED status. The user can:

- View the reports.
- Request an immediate print request.

The user can request that the Active User Report List screen display reports from the Active User file only, from the Migrated User file only, or from both User Report List files.

The Migrated User Report List files may contain reports from more than one day. Therefore, these files can contain more than one entry for each report/recipient/category. Each entry is distinguished by its original scheduling date and its decollation date.

Utility CTVCLMIG

This CONTROL-V utility erases from the Migrated User Report List file all report entries for which the retention period has expired (i.e., the number of days that the reports should be retained has passed). For relevant migrated media, the utility produces a list of tapes which can be freed for general use (i.e., returned to the free tape pool).

Each report copied to the Migrated User Report List file has an original scheduling date associated with it.

This utility can be run before or after the New Day procedure and it can be run more than once a day (e.g., after a computer crash).

This utility should be run at least once a week.

For more details, refer to the description of this utility in Section 9 of this guide.

History User Report List File

The History User Report List file contains a list of all backed up reports that have been deleted from the disk. The installation can control the time that the report will be retained in the History User Report List file. The file also contains details about the storage media of the report data (compressed dataset name on disk, or volsers of tape or cartridge).

To view a report, the user should issue a report restore request from the History User Report List file. When the report restore request is performed, the user report entry is copied from the History User Report List file to the Active User Report List file by the restore mission as soon as the report is restored from tape to disk.

Utility CTDDLRP

CONTROL-D utility CTDDLRP scans the Active User Report List file for reports whose Active User Entry has expired. For reports meeting the specified selection criteria, the following actions take place:

- Reports for which a backup or migration has been requested but not performed are flagged as WAITING FOR BACKUP (or WAITING FOR MIGRATION).
- The entry of each report for which backup and migration have been performed is moved from the Active User Report List file to the Migrated and the History User Report list files.

The entry of each report for which only migration has been performed is moved from the Active User Report List file to the Migrated User Report List file.

The entry of each report for which only backup has been performed is moved from the Active to the History User Report List file.

- A report entry that is not defined to be backed up or migrated is erased from the Active User Report List file.
- When the Active User Report List file has no entries for a report the compressed dataset containing that report is erased from DASD storage (if it does not contain data of other reports not yet erased).

Run this utility at least once a day, preferably after all backups have finished executing. See Section 9 of this guide for more details regarding utility CTDDLRP.

There is no harm in running utility CTDDLRP more than once a day (e.g., after a computer crash). You may run this utility before or after the New Day procedure.

Utility CTDCLEHIS

This utility erases from the History User Report List file all entries belonging to reports for which the backup period has expired (i.e., the number of days that the reports should be retained has passed). The utility produces a list of tapes which can be freed for general use (i.e., returned to the free tape pool).

This utility can be run before or after the New Day procedure and it can be run more than once a day (e.g., after a computer crash).

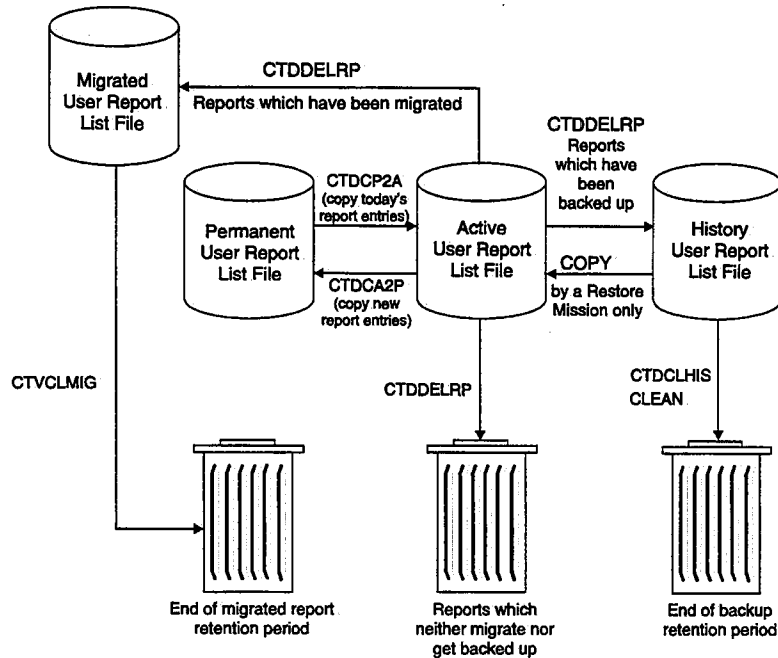
This utility should be run at least once a week.

For more details, refer to Section 9 of this guide.

Summary

The following steps should be applied when maintaining various user files, usually on a daily basis:

- Copy report definitions from the Permanent User Report List file to the Active User Report List file (utility CTDCP2A).
- Copy the new user/report entries from the Active User Report List file to the Permanent User Report List file (utility CTDCA2P).
- Erase old user/report entries from the Active User Report List file and erase their compressed datasets. Move/copy report entries from the Active User Report List file to the History User Report List file and/or the Migrated User Report List file (utility CTDELRP).
- Erase report information from the History User Report List file (utility CTDLHIS).
- Erase report information from the Migrated User Report List file (CONTROL-V utility CTVCLMIG).



Repository Maintenance

General

The CONTROL-D Repository consists of the following files:

- Active Missions file (AMF)
- Active Transfer file (ATF)
- User Report List files:
 - Active User Report List file.
 - Permanent User Report List file.
 - History User Report List file.
 - Migrated User Report List file for CONTROL-V.

All the files are self-maintained by the New Day procedure, or by certain utilities as described earlier (e.g., CTDDLRP, CTDCLHIS).

This maintenance is usually performed on a daily basis. Additional utilities are used to perform such maintenance as expanding file sizes, physical reorganization, etc. An explanation of how to use these utilities follows.

Expanding the Active Missions File

1. Change CONTROL-D installation parameter AMFSIZE in member CTDPARM to the required new Active Missions file size. Assemble the CONTROL-D installation parameters. (Alternatively, perform these steps using the CUSTOMIZE activity in ICE.)
2. Change parameter AMFSIZE in (job FORMAMF in the IOA INSTCTD library) to the new size of the Active Missions file.
3. Run member FORMAMF of the IOA INSTCTD library. Change the dataset names so you do not override the existing Active Missions file and its backup file (e.g., parameter TYPE should be changed to NEWAMF).
4. Stop all related CONTROL-D activities (e.g., CONTROL-D monitor, online activities).
5. Run utility CTDCAMF with the following parameters:

```
//EXEC CTDCAMF,  
//  OLDAMF='CTD.PROD.AMF',      old Active Missions file  
//  NEWAMF='CTD.PROD.NEWAMF'    new Active Mission file  
//      COPY  
//
```

Utility CTDCAMF copies the current Active Missions file to the new Active Missions file. The utility should end with a condition code of 0. See sample job CTDCAMF in the CONTROL-D JCL library for more details.

6. Rename the old Active Missions file (and its backup file using suffix AMB).
7. Rename the new Active Missions file (and its backup file using suffix AMB) to the former names of the old file.
8. Start the CONTROL-D monitor and resume operation.

Expanding the Active Transfer File

1. Change CONTROL-D installation parameter ATFBLK in member CTDPARM to the required new Active Transfer file size. Assemble the CONTROL-D Installation parameters. (Alternatively, perform these steps using the CUSTOMIZE activity in ICE.)
2. Stop all CONTROL-D related activities (e.g., CONTROL-D monitor, online activities).
3. Rename the old Active Transfer file (and its backup file - suffix ATB).
4. Change parameter ATFSIZE in job FORMATF in the IOA INSTCTD library to the new size of the Active Transfer file.
5. Run member FORMATF of the IOA INSTCTD library.
6. Run utility CTDCATF with the following parameters:

```
//EXEC CTDCATF,P=COPY,  
// INATF='CTD.PROD.OLDATF', old Active Transfer file  
// OUTATF='CTD.PROD.ATF' new Active Transfer file  
//
```

Utility CTDCATF copies the current Active Transfer file to the new Active Transfer file. The utility should be ended with a return code of 0.

7. Start the CONTROL-D activities and resume operation.

User Reports List File Housekeeping

Perform the following housekeeping activities on the User Report List file as required:

- Reorganize the files
- Rebuild the Index component
- Recover a damaged Data component

Reorganizing the Files

The Data and Index components of an IOA Access Method file may extend beyond one physical dataset. When an IOA Access Method file extends to many physical datasets, processing may slow down due to the fragmentation overhead. In this case, it is recommended to reorganize the IOA Access Method file so that the primary and secondary extent datasets are larger. Perform the following steps:

1. Stop all CONTROL-D related activities (CONTROL-D monitor, online activities, batch utilities).
2. Run the IOADUL utility to unload the contents of the Data component to a sequential file. See member CTDUFDUL in the CONTROL-D JCL library for an example.
3. Increase the value specified for parameter SPACE in the relevant DEFxxx and DEFxxxI definition members to specify larger primary and secondary extent sizes. This parameter should be changed in the definition members of both the Index and Data components.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

4. Delete all dataset extents of the Index and Data components. This step should be performed with caution. See member CTDUFDEL in the CONTROL-D JCL library for an example.
5. Run the IOADBF utility with parameter FUNC=INIT to allocate and format the new file's components using the new parameters. The utility should be activated twice, once for the Data component and a second time for the Index component. See member CTDUFDBF in the CONTROL-D JCL library for an example.
6. Run the IOADLD utility to reload contents of the Data component from the sequential dataset previously produced by utility IOADUL, and run the CTDDIB utility to rebuild the Index component. See member CTDUFRST in the CONTROL-D JCL library for an example.
7. Restart CONTROL-D and all other related activities.

Dynamic Sorting of the Active User Report List File

The Active User Report List file can be dynamically sorted. A detailed explanation follows.

The Index component of the IOA Access Method file functions as a balanced tree index. The CONTROL-D monitor can optionally sort the Data component of the Active User Report List file so that the physical order of the records is the same as the logical order of the Index component. This automatically provides increased I/O performance benefits for users of the CONTROL-D Online Viewing facility and utilities such as CTDDELRP.

Dynamic sorting of the Active User Report list file does not inhibit the simultaneous execution of other CONTROL-D activities, such as decollation, printing, backup and restore mission processing, or access to online user facilities.

Although dynamic sorting allows other processes to be performed concurrently on the same file components, IOA supplies a set of Sort Control parameters that can be specified to execute dynamic sort processing at a time of day and at intervals when access to CONTROL-D file components is at a minimum. A description of the Sort Control parameters can be found in the explanation of utility IOADBSR in Section 9 of this guide. Dynamic sort processing is disabled by default (see parameter ENABLE in the explanation of the Sort Control parameters).

Dynamic sort processing of the Active User Report list file is enabled and controlled by the parameters specified in member DBSRTPRM in the CONTROL-D PARM library (allocated to DD statement DASORTPR in the CONTROL-D monitor's procedure JCL).

IOA Access Method files can also be sorted by executing utility IOADBSR as a batch job. Utility IOADBSR is described in Section 9 of this guide.

Rebuilding the Index Component

To rebuild the Index component, perform the following steps:

1. Stop all CONTROL-D related activities (CONTROL-D monitor, online activities, batch utilities).
2. Run the CTDDIB utility to rebuild the Index component from the Data component. See member CTDUFDIB in the CONTROL-D JCL library for an example.
3. Restart CONTROL-D and all other related activities.

Recovering a Damaged File

If dual mirror image file processing is used, it is possible to immediately recover a damaged file. The dual file is an exact copy of the main file. The recovery procedure uses utility IOADCPY to copy the entire dual file to the main file or vice versa. Before starting the recovery procedure, make a backup copy of the main file and the Dual file.

If the Main file is damaged the recovery steps are as follows:

1. Stop all CONTROL-D related activities (CONTROL-D monitor, online activities, batch utilities).
2. Delete all extents of the Data component of the main copy only. For example, to delete all the extents of the Active User Report List file, list all the files with prefix CTD.ACT.E* and delete them.
3. Run the IOADCPY utility with parameter SUFFIX set to the Dual file (D000). See sample member CTDFUCPY in the CONTROL-D JCL library for an example.
4. Rebuild the Index component as described above.
5. Restart all CONTROL-D activities.

If the Dual file has been damaged the recovery steps are as follows:

1. Stop all CONTROL-D related activities (CONTROL-D monitor, online activities, batch utilities).
2. Delete all extents of the Data component of the Dual copy only. For example, to delete all the extents of the CONTROL-D Active User Report List file, list all the files with prefix CTD.ACT.D* and delete them.
3. Run the IOADCPY utility with parameter SUFFIX set to the Main file (E000). See sample member CTDFUCPY in the CONTROL-D JCL library for an example.
4. Rebuild the Index component as described above.
5. Restart all CONTROL-D activities.

See Section 9 of this guide for more details about IOA Access Method utilities for User Report List files.

SMF Accounting

CONTROL-D produces several types of SMF records which provide comprehensive accounting and chargeback information.

1. An SMF record for each printed report is produced during deferred printing. The Record number is specified in parameter SMF=nnn in member CTDPARM in the INSTCTD library. The record can be disabled by setting SMF to NO in member CTDPARM. The structure of the record is described in macro CTDSMF in the IOA MAC library. CONTROL-D exit 6 (CTDX006) receives each record before it is written to SMF files. The exit can change or add fields to the record.

Section 4: Customizing and Administering CONTROL-D and CONTROL-V

2. An SMF record can be produced for each printed report during immediate printing. This is triggered by optional wish number WD0892 in member IOADFLTC in the IOA SECUDATA library. The record number and structure are identical to the ones provided for deferred printing.
3. An SMF record describing activity of an online user is written during logoff operation performed under the IOA online monitor. The record contains CPU/EXCP consumed by the current user and also the number of operations performed by it (e.g., print, restore, delete requests). The record is produced by a special version of IOA exit 6 (IOAX006S). The desired record number should be set in the exit. The exit must be recompiled. The structure of the record is described by macro CTDSF1 in the IOA MAC library.
4. An SMF record created during a decollation process contains the number of pages/lines processed by a decollation mission, the CPU utilized by it, the number of report entries built, etc. The record is built by a special version of CONTROL-D exit 22 (CTDX022D). Desired record numbers should be set in the exit. The exit must be recompiled. The structure of the record is described in macro CTDSF2 in the IOA MAC library.



Section 5: Customizing and Administering CONTROL-O

General	5-1
Starting CONTROL-O	5-1
Shutting Down CONTROL-O.....	5-1
Replacing an Active CONTROL-O Monitor.....	5-2
Automatic Loading of Rules	5-2
Manual Loading of Rules	5-3
Manual Loading of CMEM Rules.....	5-4
Deleting (Deactivating) an Active Rule Table.....	5-5
Recommended Organization Method	5-5
Use of Two Rule List Members	5-7
Replacing the IPL CONTROL-O	5-8
Rule Scheduling	5-8
Displaying Active Rules.....	5-9
Controlling the Message Statistics Facility.....	5-9
Preventing Unnecessary Enlargement of Statistics Files.....	5-9
Handling Near-Full and Full Conditions for the Statistics File	5-10
Controlling the Automation Log Facility.....	5-10
Determining the Size of the Log	5-11
Preventing Logging of Unnecessary Messages	5-11
Controlling Rule Operation Mode.....	5-12
Global AutoEdit Variables.....	5-12
Defining a New Global Member.....	5-13
Automatic Compression of the Global AutoEdit Library	5-14
Modifying the CONTROL-O Sleeping Interval.....	5-15
Refreshing the CONTROL-O Security Cache	5-16
Problem Determination	5-16
Customization of Automation Options	5-18
AOP Overview	5-18
Menus.....	5-19
Menu Member Syntax	5-19
Parameters	5-19
Format Members.....	5-23

Client Programs.....	5-23
Program Input.....	5-24
Execution Process.....	5-24
Client Program Linkage Conventions.....	5-25
Considerations for When CONTROL-M and CONTROL-O Are Installed.....	5-25

General

Starting CONTROL-O

CONTROL-O is usually started during early stages of the IPL process. In this way CONTROL-O can automate the IPL process (as soon as CONTROL-O has been initialized). To start CONTROL-O as part of the IPL, member COMMNDnn in SYS1.PARMLIB (where nn is either the number specified in member IEASYS in SYS1.PARMLIB, or 00) should contain the command:

```
S CONTROLO, SUB=MSTR, OUTLIST=DUMMY[, ORDER=IPLRULES] [, TYPE=IPL]
```

Parameters SUB=MSTR and OUTLIST=DUMMY are necessary to start CONTROL-O before JES has been brought up. Parameters ORDER and TYPE are optional. For more details about these parameters and how/when CONTROL-O should be started, see "Recommended Organization Method" later in this section.

If you do not plan to use CONTROL-O to control the IPL process, CONTROL-O can still be started during IPL (after JES has been brought up) by adding the following command to member COMMNDnn in the SYS1.PARMLIB library:

```
S CONTROLO
```

It is also possible to manually issue the above mentioned operator commands.

Note *CMEM users:*

If CONTROL-M is installed, CONTROL-O assumes responsibility for the functions of the CMEM facility. Therefore it is important to verify that the CMEM monitor (CTMCMEM) is down before the CONTROL-O procedure is started.

For sites with CMEM rules that are to be implemented (triggered) in the early stages of the IPL process, command member COMMNDxx in SYS1.PARMLIB should contain the following command:

```
S CONTROLO, SUB=MSTR, OUTLIST=DUMMY[, ORDER=IPLRULES] [, CMORDER=IOAIPLCM] [, TYPE=IPL]
```

For more details about the parameters in the above statement, and how/when CONTROL-O should be started, see "Recommended Organization Method" later in this section.

Shutting Down CONTROL-O

Although it is possible to use an operator command (F CONTROLO,STOP or P CONTROLO) to shut down CONTROL-O, it is not recommended.

It is not usually necessary to shut down CONTROL-O. However, if a CONTROL-O shutdown becomes necessary, the active CONTROL-O monitor should be replaced by starting a new CONTROL-O monitor. See "Replacing an Active CONTROL-O Monitor" below.

Use an operator command to shut down CONTROL-O only if the replacement method does not resolve the problem. If operator command F CONTROLO,STOP or P CONTROLO is used, CONTROL-O shuts down after a few seconds.

Section 5: Customizing and Administering CONTROL-O

Note *CMEM users:*

When CONTROL-O is shut down, the CMEM facility is also shut down. To restart CONTROL-O CMEM support after CONTROL-O has been completely shut down, issue the following operator command:

S CONTROL-O,TYPE=CTOCMEM

Replacing an Active CONTROL-O Monitor

When a CONTROL-O monitor is started (via operator command S CONTROL-O) and a CONTROL-O monitor is already active in the computer, the current CONTROL-O monitor passes execution control to the new CONTROL-O monitor and then shuts down. It is not necessary to reload the rule tables; they are passed from the current monitor to the new one. To clean (erase) all CONTROL-O tables from memory, shut down the CONTROL-O monitor (usually this is done only in an emergency).

Automatic Loading of Rules

When CONTROL-O is started (and it is not replacing an active CONTROL-O monitor), it attempts to read a list of rule tables from the member referenced by DD statement DARULLST. These are the rule tables to be automatically loaded by CONTROL-O. The name of the member can be passed in the CONTROL-O procedure parameter ORDER. The supplied default member RULELIST is in the CONTROL-O PARM library. Each line in the list has the following format:

date library table {FORCE|NOFORCE|ORDER}

Where:

date Date of the rule. If a specific date is designated, it is used to analyze the Basic Scheduling parameters (see below). The date format is mmddyy, ddmmyy, or yymmdd (depending on the site standard).

An asterisk (*) can be specified to indicate the current CONTROL-O working date.

library Rule library name.

table Rule table name (or mask).

For every line in the list, CONTROL-O loads the specified rule table from the specified library.

If either NOFORCE or ORDER is specified, each rule's Basic Scheduling parameters are compared to the specified date (or the current CONTROL-O working date when * has been specified). If the rule should be scheduled on that date, the rule is loaded by CONTROL-O and activated.

If FORCE is specified (the default), each rule is loaded by CONTROL-O and activated. Use this option when you do not wish to use CONTROL-O scheduling options.

Note *CMEM users:*

When CONTROL-O is started (and it is not replacing an active CONTROL-O monitor), it attempts to read the member referenced by DD statement DACTMLST. This member is a PDS member which lists the CMEM rule tables to be automatically loaded by CONTROL-O.

The name of the member containing the list of CMEM rule tables can be passed in parameter CMORDER of the CONTROL-O procedure. The default member for the CMEM list is member IOACMEML in the IOA PARM library.

Manual Loading of Rules

Member RULELIST contains a list of basic rule tables to be activated by CONTROL-O as it is started. To load additional tables, or to replace a currently active table with a new (updated) copy of the rules in the table, use one of the following options:

Option 1

Enter the CONTROL-O Online facility and use the ORDER/FORCE option in the Table List screen (screen OR).

Option 2

Issue the operator command:

```
F CONTROL-O, O|F=library(table) [,D=date]
```

Where:

- | | |
|---------|--|
| O | Each rule is ordered. Each rule's Basic Scheduling parameters are compared to the specified date. If the rule should be scheduled on that date, the rule is loaded by CONTROL-O and activated. |
| F | Each rule is forced (loaded by CONTROL-O and activated). Use this option when you do not wish to use CONTROL-O scheduling options. |
| library | Rule library name. |
| table | Rule table name (or mask). |
| date | Scheduling date (optional). If no date is specified, the CONTROL-O working date is used as the default. Date format is mmddy, ddmmy, or yymmdd (depending on the site standard). |

Examples

```
F CONTROL-O, F=CTO.PROD.RULES(CICSPROD)
```

Loads table CICSPROD from CTO.PROD.RULES.

```
F CONTROL-O, F=CTO.PROD.RULES(*)
```

Loads all tables from CTO.PROD.RULES.

```
F CONTROL-O, F=CTO.PROD.RULES(PROD*)
```

Loads tables whose names start with PROD from CTO.PROD.RULES.

Section 5: Customizing and Administering CONTROL-O

```
F CONTROLLO,F=CTO.PROD.RULES(?CICS*)
```

Loads tables whose names start with any character followed by the string "CICS" from CTO.PROD.RULES.

If the table has already been loaded by CONTROL-O, the new copy of the table replaces all the rules of the table active under the CONTROL-O monitor.

If you wish to replace all the tables under CONTROL-O with the list of tables specified in DD statement DARULLST, use the operator command:

```
F CONTROLLO, O|F=ALL [,REBUILD]
```

Notes When *ALL* is specified, each table is ordered or forced according to its *FORCE/NOFORCE* specification in the rule list as defined in DD statement *DARULLST*.

When the *REBUILD* option is specified, all previously loaded rules are deleted. This option should be specified when using calendar dependent rules.

If the *REBUILD* option is not specified, previously loaded rules are either replaced by new copies of the rules or left unchanged.

The *CMEM* table is reloaded automatically when *ALL* is specified.

Manual Loading of CMEM Rules

The member referenced by DD statement DACTMLST during startup contains a list of CMEM rule tables to be automatically ordered by CONTROL-O when it is started.

To load additional tables, or to replace a currently active table with a new (updated) copy, use one of the following options:

Option 1

Enter the CMEM Online facility (=C), and use the *FORCE* option in the Table List screen.

Option 2

Issue the following operator command:

```
F CONTROLLO,C=library(table)
```

Where:

C Indication that CMEM rules are to be loaded. Each CMEM rule in the specified table(s) is ordered by CONTROL-O and activated.

library Rule library name.

table Rule table name (or mask).

Examples

```
F CONTROLLO,C=CTM.CMEM.RULES(DATASET)
```

Loads table DATASET from CTM.CMEM.RULES.

```
F CONTROLLO,C=CTM.CMEM.RULES(*)
```

Section 5: Customizing and Administering CONTROL-O

Loads all tables from CTM.CMEM.RULES.

```
F CONTROL-O,C=CTM.CMEM.RULES(BKP*)
```

Loads tables whose names start with BKP from CTM.CMEM.RULES.

```
F CONTROL-O,C=CTM.CMEM.RULES(?CICS*)
```

Loads tables whose names start with any character followed by the string "CICS" from CTM.CMEM.RULES.

If a specified CMEM table has already been loaded by CONTROL-O, the new copy of the CMEM table replaces all the rules of the table active under the CONTROL-O monitor.

The following operator command affects CMEM rules as well as CONTROL-O rules:

```
F CONTROL-O,O|F=ALL[,REBUILD]
```

This command affects CMEM rules in the following way:

- When ALL is specified, all CMEM rule tables defined in the CMEM list member (referenced by DD DACTMLST) are reloaded.
- When REBUILD is specified, all previously loaded CMEM rules are deleted as well as the CONTROL-O rules.
- If REBUILD is not specified, previously loaded CMEM rule tables are either replaced by new copies or left unchanged.

Deleting (Deactivating) an Active Rule Table

To deactivate an active rule table, use an operator command. For example:

```
F CONTROL-O,D=CTO.PROD.RULES(CICSPROD)
```

A single rule can also be deleted/held using appropriate line commands in the Rule Status screen.

Recommended Organization Method

CONTROL-O's primary purpose is to automate console operation. An especially important use of CONTROL-O is the automation of system startup (IPL). Following is the recommended CONTROL-O organization method for automating the IPL process. If you do not wish to automate the IPL process, CONTROL-O organization is simplified since certain MVS-related restrictions do not need to be addressed - in this case you may skip directly to "Rule Scheduling" later in this section.

System Definitions

- The IOA subsystem name (as defined in IOA Installation parameter SUBSYS) should appear as the first subsystem name after the primary subsystem in the subsystem list (member IEFSSNxx in SYS1.PARMLIB). This permits CONTROL-O to control commands directed to other MVS subsystems before they are executed by these subsystems. However, to control JES commands before they are executed by JES, CONTROL-O uses JES exits or an additional subsystem name, defined optionally in the CONTROL-O installation parameters. See parameter JCMDSSN in the IOA Installation Guide for more information.

Section 5: Customizing and Administering CONTROL-O

Note MVS 5.2 users:

The subsystem name can be defined using the method described above (IEFSSNnn - positional parameter form), or using the keyword parameter form of the IEFSSNnn PARMLIB member.

When using the keyword parameter form, add a record to the IEFSSNnn member in the format:

```
SUBSYS SUBNAME(XXXX)
```

where xxxx is the name of the subsystem as specified in the SSNAME parameter in the IOA installation parameters.

See the MVS/ESA SP 5.2 Initialization and Tuning Reference Manual for further details.

- The CONTROL-O procedure must reside in the SYS1.PROCLIB library.
- The CONTROL-O monitor should be started at an early stage of IPL. To do this, member COMMNDnn in the SYS1.PARMLIB library (where nn is either the number specified in member IEASYS in the same library, or 00) should contain the following command as one of the first commands in the member:

```
COM='S CONTROL0,SUB=MSTR,OUTLIST=DUMMY'
```

Note CMEM users:

The COMMNDnn member in SYS1.PARMLIB should contain the following command:

```
COM='S CONTROL0,SUB=MSTR,OUTLIST=DUMMY[,ORDER=IPLRULES]  
[,CMORDER=IOAIPLCM][,TYPE=IPL]'
```

The following CONTROL-O and IOA files are allocated to the CONTROL-O monitor:

- IOA LOAD library
- IOA Conditions/Resources file and its Synchronization file
- Calendar library
- PARM library
- Message Statistics file (dynamically allocated)
- Rules library (dynamically allocated)
- Automation Log file (dynamically allocated)
- Global Variables library (dynamically allocated)
- Because CONTROL-O is started as part of IPL, you should either catalog all these files in the MVS Master Catalog, or specify the VOLSER and UNIT parameters for each of the files. The dynamically allocated files (e.g., the Message Statistics file) must be in the Master catalog.

If you specified parameters VOLSER and UNIT for these files and you move the files to another disk, you must modify the CONTROL-O procedure as well.

Section 5: Customizing and Administering CONTROL-O

- It is recommended that CONTROL-O start and control the initialization of other system components (JES,VTAM, TSO, etc.).
- The on-site enqueue manager must be up before any attempt is made by CONTROL-O to update an IOA file. See the description of parameter QNAME in the IOA Installation Guide for more information.

Use of Two Rule List Members

Parameter ORDER of the CONTROL-O monitor specifies the name of a member containing the list of rules to be loaded by CONTROL-O upon startup. The easiest organization method is using one list member containing all CONTROL-O tables (e.g., the supplied member RULELIST). This is the preferred method when CONTROL-O will not control IPL. However, if CONTROL-O will control IPL, it should be organized somewhat differently.

As your site's use of CONTROL-O expands, it is possible that hundreds of operations rules will be defined. Loading all of these operation rules may take time. CONTROL-O does not start to analyze messages until it has finished loading all rules in the initialization rule list. Consequently, if the list is quite large, CONTROL-O may not detect some of the messages and commands issued during the beginning of the IPL process.

To ensure analysis of all messages, it is recommended to place all rules which control the IPL process in a separate list. This list is loaded when CONTROL-O is started by overriding the default list member name using parameter ORDER to designate member IPLRULES. Member IPLRULES should contain only the rule tables to be used during IPL. One of the rules in this member should load the remaining of the CONTROL-O rule tables.

Multiple Rule Table Lists for CMEM Rules

Parameter CMORDER specifies the name of the member which contains the list of CMEM rule tables (the CMEM list) to be loaded by CONTROL-O upon startup. It is normally recommended to list all your CMEM rule tables in one list (e.g., default member IOACMEML). However, as described below, it is sometimes recommended to list certain rule tables separately.

CONTROL-O does not start to analyze CMEM events until it has finished loading all rule tables in the CMEM list specified via parameter CMORDER. If a large number of rules have been defined at your site, CONTROL-O may not detect some CMEM events during the beginning of the IPL process.

To ensure detection of all CMEM events, it is recommended that you place CMEM rules to be triggered during the IPL process in a separate list (in member IOAIPLCM), and to specify this list via parameter CMORDER. Rule tables specified in this way are loaded and activated during the startup process.

Replacing the IPL CONTROL-O

Operation of CONTROL-O involves processing phases which are implemented by starting separate CONTROL-O monitors, each one replacing the previous one.

- Phase 1: When CONTROL-O is activated during IPL (with SUB=MSTR), certain functions are inactive (particularly, the debugging facilities). This is because JES is not yet up, and it is impossible to allocate a sysout file. For this reason, CONTROL-O is started with parameter OUTLIST=DUMMY.
- Phase 2: A simple CONTROL-O rule can detect the JES initialization message and then restart CONTROL-O (this time without parameters). A sample rule can be found in the table STARTSYS. The new CONTROL-O automatically takes control from the previous CONTROL-O used during IPL. You do not need to reload the rules.
- Phase 3: During shutdown of the system, if CONTROL-O controls the termination of JES, a third CONTROL-O monitor is required which does not allocate sysout files. This CONTROL-O monitor should be started (by a rule) with SUB=MSTR and OUTLIST=DUMMY.

Parameter TYPE of operator command START CONTROL-O enables you to specify the CONTROL-O processing phase for a specific monitor. Valid values for this parameter are:

- IPL The CONTROL-O monitor controls the IPL process.
- REGULAR The CONTROL-O monitor is a regular CONTROL-O monitor.
- SHUTDOWN The CONTROL-O monitor controls the shutdown of the system.

The CONTROL-O TYPE appears in the following initialization message:

```
CTO147I CONTROL-O INITIALIZATION COMPLETE,TYPE=type,SUB=job-entry-subsystem
```

This message can be used by CONTROL-O rules whenever phase-dependent processes are handled.

Note When CONTROL-O is brought up to replace an existing CONTROL-O monitor, it does not load rules from DD statement DARULLST. Instead, it takes control of the rules which were active under the previous CONTROL-O monitor. Only a new operator command F CONTROL-O,O=ALL loads the rules from DARULLST.

CMEM Users:

The above note is also true for CMEM rules to be loaded from DD statement DACTMLST.

Rule Scheduling

CONTROL-O is usually started at an early stage of IPL and remains active until shutdown. Most rules behave the same on any business date. However, it is sometimes necessary to schedule different rules for different business dates. For example, you may want to trigger a response to a certain message on a regular business date and a different response to the same message on the first day of a month.

Section 5: Customizing and Administering CONTROL-O

Under CONTROL-O you can define scheduling parameters for rules. For example, a rule can be scheduled for a specific date only. The scheduling criteria of rules are checked when a rule is ordered (either automatically or manually). Therefore, if a certain rule is sensitive to scheduling criteria, its table is loaded (ordered) on a daily basis. The rule scheduling criteria are then checked on a daily basis - the rules are loaded and activated only if they should be scheduled according to the scheduling criteria.

The recommended method to reload the rules on a daily basis is using a time-initiated command (a rule) which issues an order command for all CONTROL-O tables at a specified time of the day:

```
F CONTROL-O, O=ALL, REBUILD
```

Displaying Active Rules

The Rule Status screen displays a list of all rules which have been ordered and their statuses. The displayed rules can be held/deleted/freed. Alternatively, enter operator command F CONTROL-O, DISPLAY. The list is sent to the console.

Controlling the Message Statistics Facility

The Message Statistics facility allows accumulation and display of statistics for the messages/commands issued by the system on which CONTROL-O is active.

The Message Statistics facility counts messages and commands by message/command ID. Each time a message or command appears, the counters for the corresponding entry in the Statistics file are incremented. When a new message ID or %%STATID value is detected, a new entry is opened in the Statistics file.

Statistics accumulation is normally started automatically when CONTROL-O is brought up. However, when necessary, accumulation of statistics can be controlled by the following operator commands:

1. Stop statistics accumulation:

```
F CONTROL-O, STOPSTAT
```

This command can be used if statistical accumulation is no longer required or if it becomes necessary to reformat the Statistics file without bringing down CONTROL-O.

2. Start statistics accumulation:

```
F CONTROL-O, STARTSTAT
```

3. Reset statistics for all messages and commands:

```
F CONTROL-O, RESETSTAT
```

Preventing Unnecessary Enlargement of Statistics Files

The Message Statistics facility counts messages and commands by message/command ID. The first word of the message/command is the message/command ID.

Section 5: Customizing and Administering CONTROL-O

To prevent the Statistics files from becoming too large and from containing duplicate information, it might be desirable to group messages or commands according to other criteria. It would be useful, for example, to group messages/commands under one message/command ID when:

- Messages start with free text, and each new message is normally accumulated under a new message ID, thus unnecessarily enlarging the size of the Statistics file.
- Concatenation of JES2 commands with different operands normally results in separate statistics accumulation for each operand. For example, statistics for \$PJ05555 and \$PJ01234 are normally accumulated under different IDs, even though both IDs refer to the same command.
- Accumulating separate statistics for messages which are not of interest to the user can be wasteful.

AutoEdit reserved variable %%\$STATID can be used to specify the ID under which the messages/commands are accumulated. For example:

```
ON COMMAND $P*  
DO SET %%$STATID=$P
```

Rule table STATS from the RULES library has been provided to perform the necessary grouping by %%\$STATID. This table can be adapted/used according to site requirements.

Handling Near-Full and Full Conditions for the Statistics File

When the Statistics file becomes more than 90% full, CONTROL-O issues unrollable message CTO240W as a warning.

When CONTROL-O detects that the Statistics file has become full, it issues unrollable message CTO241E. CONTROL-O then stops tracking statistics for new messages but continues to update the counters of messages already listed in the Statistics file.

If message CTO240W and/or CTO241E is issued, it is recommended that the operator enlarge the Statistics file (via utility CTOCSF) and check if messages can be grouped by %%\$STATID (described above).

Controlling the Automation Log Facility

The Automation Log facility allows accumulation and display of automation information from all inputs available to CONTROL-O.

The Automation Log facility is started automatically when parameter AUTOMLOG in member CTOPARM is set to V or D. (Refer to the CONTROL-O section in the IOA Installation Guide.)

While CONTROL-O is active, the Automation Log facility is controlled by the following operator commands:

1. Stop Automation Log accumulation:

```
F CONTROL0,AUTOLOG=NO
```

Section 5: Customizing and Administering CONTROL-O

This command can be used when the Automation Log is no longer required or when it becomes necessary to reformat or rebuild the Automation Log file without bringing down CONTROL-O. Message CTO297I indicates that Automation Log accumulation has stopped.

The Automation Log can be viewed via screen OL, even when CONTROL-O is inactive or Automation Log accumulation has been stopped.

2. Start Automation Log accumulation:

```
F CONTROL0,AUTOLOG=YES
```

Message CTO297I indicates that Automation Log accumulation has started.

Note When the Automation Log is not active, CONTROL-O trace messages are written to the SYSOUT specified by DD statement DAACTLOG of the CONTROL-O monitor.

Determining the Size of the Log

The Automation Log is a wrap-around dataset, which means that the number of records in the file is constant and, when the file is full, a new record overwrites the oldest record. Old Automation Log entries may be archived/backed up for subsequent retrieval using utility CTOALOC (described in Section 9 of this guide).

If backup is performed on a regular basis, the size of the Automation Log should be such that no record is overwritten before backup has taken place. The appropriate size for an Automation Log is specified in installation parameter ALREC#. The backup frequency depends on site requirements.

Preventing Logging of Unnecessary Messages

Some messages may be of little interest. These messages should not be written to the Automation Log, so that the amount of unnecessary information on the Automation Log is minimized.

System AutoEdit variable %\$AUTOLOG can be used to control which messages are entered into the Automation Log. Valid values are:

Y (Yes) Record the message in the Automation Log.

N (No) Do not record the message in the Automation Log.

Example

```
ON MESSAGE TST*  
DO SET %$AUTOLOG=N
```

Prevents messages whose ID starts with TST from being written to the Automation Log.

Controlling Rule Operation Mode

The mode of operation (i.e., the trace mode) for a rule is determined by parameter `MODE` in its rule definition. Sometimes it is useful to override the operation mode of all active rules and verify that events and actions are recorded in a particular way. For example:

- Ensure a full trace of all rules (i.e., all events and actions are recorded) to facilitate analysis of the interaction between rules.
- Record (trace) only the triggering of every rule.

Global trace operations are requested via operator commands, as follows:

1. Activate a full trace:

```
F CONTROL0,LOG=ALL
```

All rules are fully traced as if they were defined with mode `LOG`. This operator command should only be used temporarily for specific tests because extended use of full trace mode can adversely affect CONTROL-O performance.

2. Trace rule triggering only:

```
F CONTROL0,LOG=TRIGGER
```

Only rule triggering is traced for all rules. However, rules defined with mode `LOG` are fully recorded.

3. Restore the default operation mode (as defined in the rule definition) for each rule:

```
F CONTROL0,LOG=DEFAULT
```

Global AutoEdit Variables

When CONTROL-O is started, it reads the members of the Global AutoEdit library and loads into memory the Global variables that reside in those members. The list of members loaded is specified in the `DD DAGLBLST` statement of the CONTROL-O monitor. A different Global library (in which the Global members reside) is used for each computer (SMF ID) on which CONTROL-O operates. The Global library name is composed of the prefix defined in the `GLBPREF` parameter of the CONTROL-O installation parameters, the Global identifier: `.GLB.`, then the string `CPU` followed by the SMF ID of the specified computer. Example: `CTO.Vxxx.GLB.CPUSYSA`.

Member `$GLOBAL` is usually the default Global member and it contains the CONTROL-O Global AutoEdit variables that are not assigned to specific Global members. The member format is identical to that of a CONTROL-M AutoEdit variables member format. Therefore CONTROL-O Global members can be referenced by CONTROL-M jobs via the `MEMSYM` and `LIBSYM` parameters.

Any updates or additions to Global variables (by rules or KOA scripts) are kept in memory by CONTROL-O and are available to all other rules. While CONTROL-O is active, Global variables can be reloaded from the PDS (partitioned dataset) Global library or written back to the PDS Global library, in totality or by member name, using the `LOADGLOBAL` and `WRITEGLOBAL` operator commands. When CONTROL-O is stopped, the Global variables are written back to their respective PDS members.

Section 5: Customizing and Administering CONTROL-O

To update the Global members with the current status of the Global variables, use the following operator command:

```
F CONTROL0,WRITEGLOBAL [(-memname|=ALL)]
```

When ALL is specified, all members listed in the DD DAGLBLST statement are updated.

When memname is specified, only the specified member is updated, and only on the condition that its name appear in the DD DAGLBLST statement.

When no specification follows the WRITEGLOBAL parameter, only the default \$GLOBAL member is updated.

To load Global members from memory, use the following operator command:

```
F CONTROL0,LOADGLOBAL [(-memname|=ALL)]
```

When ALL is specified, all members listed in the DD DAGLBLST statement are loaded.

When memname is specified, only the specified member is loaded, on the condition that its name appear in the DD DAGLBLST statement.

When no specification follows the LOADGLOBAL parameter, only the default \$GLOBAL member is loaded.

Defining a New Global Member

Rules related to a specific application can use their own Global member to isolate their variables from the rules of other applications, and enable the use of checkpoints or to perform initializations on an application basis.

To define a new Global member, it is necessary to create a member in the GLB library, and to add its name to the Global Member list. This list is defined by the DD DAGLBLST statement of the CONTROL-O monitor, and it usually resides in the DAGLBLST member of the CONTROL-O PARM library.

Each line describes a Global member. Valid format is:

member attribute

Where:

member Global member name.

attribute Type of Global member. Values are:

INOUT Load and update of the member are performed using commands **LOADGLOBAL** and **WRITEGLOBAL** as described above.

INPUT The member can be loaded but never written back. This option may be used for storing Global AutoEdit variables whose initial values are specified in the member and do not require checkpointing. The value of each Global AutoEdit variable may be changed and new Global AutoEdit variables can be added to the member during the CONTROL-O session, but these new values and new variables are not saved when CONTROL-O is stopped.

Section 5: Customizing and Administering CONTROL-O

PROTECTED Global AutoEdit variables which reside on a PROTECTED member cannot be updated during the CONTROL-O session and no new Global AutoEdit variables can be assigned to the member. This option can be used for storing Global AutoEdit variables that are "constants" (i.e., the messages of an application).

TEMPORARY The Global member associated with this entry does not reside on the disk; therefore, it is not loaded and it is not written back. This option is useful for Global AutoEdit variables that do not need to be saved after CONTROL-O is stopped.

Once the new Global member has been defined and listed as described above, it is necessary to activate it with the following operator command:

```
F CONTROL-O, LOADGLOBAL=member
```

Where:

member Global member name.

Automatic Compression of the Global AutoEdit Library

Because of its PDS (Partitioned dataset) organization, the Global AutoEdit library is prone to D37 ABENDs (due to insufficient space to write on disk either a new member or an updated version of an existing member). If, at your site, there is no product to keep the library compressed, CONTROL-O enables you to automatically compress the Global AutoEdit variables library, whenever it becomes full, as explained below.

The Automatic Compression facility is normally activated when the Global AutoEdit library is full (i.e., when a D37 abend is encountered during a WRITEGLOBAL operation).

Before attempting to compress the Global AutoEdit library, CONTROL-O backs the library up to a sequential file via utility IEBCOPY.

The Automatic Compression facility uses member \$\$COMPST of the Global AutoEdit library to track the progress of the compression operation. \$\$COMPST is a single record member that is created automatically when the Global library is defined.

For each step in the backup/compression process variable %%STATUS in member \$\$COMPST is updated to reflect the status of the compression process. Possible values for this variable are:

%%STATUS=S1 Backup of the Global AutoEdit library has been started but not completed.

%%STATUS=S2 Backup of the library was successful. Compression has been started but not completed (i.e., is either still in process or it has failed).

%%STATUS=OK Compression has been successfully completed.

Whenever the Global AutoEdit library is accessed (via a LOADGLOBAL or WRITEGLOBAL operation), the value of %%STATUS is checked to ensure that the library was not corrupted during the last automatic compress operation.

Section 5: Customizing and Administering CONTROL-O

- If the value of %%STATUS is OK, the requested LOADGLOBAL or WRITEGLOBAL operation is performed.
- If the value of %%STATUS is S2, the Automatic Compression facility automatically restores the Global AutoEdit library from the backup file and compresses it before execution of the LOADGLOBAL or WRITEGLOBAL operation.
- If the value of %%STATUS is S1, the Global AutoEdit library and backup may be corrupt and the operation cannot be performed. An error message is issued.

Enabling Automatic Compression

If automatic compression of the AutoEdit variables library has not been enabled at time of installation, it can be enabled by performing the following steps:

1. Bring down CONTROL-O.
2. Specify Y for parameter GLBCOMP in member CTOPARM of the IOA INSTCTO library, and submit job CTOPARMJ.
3. Submit job NEWGLOB in the IOA INSTCTO library. This job does the following:
 - Renames the old Global AutoEdit library.
 - Defines a new Global AutoEdit library (using the old library name) and a sequential backup file (used by the Automatic Compression facility).
 - Copies the contents of the renamed Global AutoEdit library to the new Global AutoEdit library.
4. Restart CONTROL-O.

Modifying the CONTROL-O Sleeping Interval

CONTROL-O wakes up every few seconds and checks on time-related events. This time interval is defined in the CONTROL-O installation parameters and can be changed by the system administrator. In addition, the interval can be modified by the operator command:

```
F CONTROL-O, INTERVAL=XX
```

Where XX represents the interval in seconds.

If CONTROL-M is active at your site, it is recommended that you specify an interval shorter than the CONTROL-M interval.

It is recommended that the interval be modified by automatic commands which will be invoked by the CONTROL-O monitor itself, according to set conditions and time ranges and not manually by the operator.

The optimal sleeping interval depends on the processing power of the machine. Depending on the processing power of the machine indicated below, the sleeping interval should usually not be less than the number of seconds indicated:

- For machines with less than 20 MIPS 10 seconds
- For machines with 20-50 MIPS 5-6 seconds
- For machines with over 50 MIPS 4 seconds

Section 5: Customizing and Administering CONTROL-O

There is no practical benefit in setting the interval to less than the above minimum. Moreover, it could actually slow down CONTROL-O's operation.

When the modification is accepted by CONTROL-O, the following message appears on the operator console:

```
CTO123I CONTROL-O INTERVAL IS SET TO ## SECONDS
```

Refreshing the CONTROL-O Security Cache

CONTROL-O security modules use a security block to identify each user for which an authority check is performed. The first time a user's security authorization is checked CONTROL-O creates a security block for that user. The security block can then optionally be saved for the next time the user's security authorization is checked.

Security blocks saved for subsequent checks are kept in the CONTROL-O Security Cache. The size of the security cache is defined in parameter RUNTCACH in member CTOPARM. If 0 is specified for parameter RUNTCACH, no security blocks are saved. (See "Performing Post-Installation Tasks" in the IOA Installation Guide for information about how to alter the specification for this parameter.)

The CONTROL-O Security Cache holds security blocks for only the last x users to have their security authorization checked (where x is the value specified for parameter RUNTCACH). For example, if parameter RUNTCACH contains a value of 10, the security cache holds a security block for each of the last ten users that were checked.

Changes made to a user's security authorization (since the last time that user's security block was created) are not automatically included in the information in the user's security block in the CONTROL-O Security Cache. However if a user's security authorization has been changed, and there is no security block in the CONTROL-O Security Cache for that user, changes made to the user's security authorization are put into effect the next time that user's security authorization is checked.

To immediately include new user authorization information in the CONTROL-O security cache, refresh the security cache via the following operator command:

```
F CONTROL-O,NEWSECDEF
```

This command refreshes all user authorization information in the CONTROL-O security cache.

When the modification is accepted, the following message appears on the operator console:

```
CTO251I RUNTIME SECURITY REFRESH ENDED OK
```

Problem Determination

CONTROL-O is supplied with internal debugging facilities: The ability to print an internal debugging trace and the ability to print the contents of the CONTROL-O internal data areas. Under normal circumstances, the debugging facilities are dormant. However, if required (i.e., your IOA representative has requested debugging information), it is possible to activate the debugging facilities as follows:

Activate the CONTROL-O internal trace:

1. Perform either step A or B.

Section 5: Customizing and Administering CONTROL-O

- A. Start a new CONTROL-O monitor with the following operator command:

```
S CONTROLO,DEBUG=level
```

The CONTROL-O monitor passes control to the new CONTROL-O monitor and shuts down.

or

- B. Issue the following operator command:

```
F CONTROLO,DEBUG=level
```

The required debugging level will be supplied by your IOA representative. It can be any value from 00 to 255 (00 specifies no debugging whatsoever).

Note *It is highly recommended that you not activate CONTROL-O with the DEBUG parameter on a regular basis because in case of a JES problem, CONTROL-O may become suspended while waiting for JES.*

2. The debugging information will be printed to DD statements PRTDBG and DADUMP of the CONTROL-O procedure.
3. When you have finished your problem determination procedures, start a new CONTROL-O using the following operator command:

```
S CONTROLO
```

or specify the operator command:

```
F CONTROLO,DEBUG=00
```

Print CONTROL-O Internal Data Areas

To print CONTROL-O internal data areas, issue the following operator command:

```
F CONTROLO,SNAP [=name1,name2 .....namen]
```

Where:

name1, name2,.... namen are the names of the CONTROL-O internal data areas.

When no name is specified, all data areas are printed. Your IOA representative will provide you with the list of the data area names and will specify which areas should be printed depending on the problem encountered.

Valid values are:

ALL	LINK	MTOPLB	OMT	RQCDLY	RQCSTO	SSVT
ALO	MAIN	MTOPND	OPR	RQCEXO	RQH	STO
CAS	MCT	MTOPNX	PARM	RQCFREE	RULES	SWT
CONLIST	MTO	MTOSRV	PND	RQCMTO	SEC	UCM
CONS	MTOINX	MTOSRVA	RFR	RQCRFR	SLO	VAR
CONSOLE	MTOLNK	MTOWSC	RQC	RQCSLO	SRV	WISHES
DLY	MTOMIX	MVS	RQCALO	RQCSRV	SSCT	WSC
EXO	MTOMPT					

Section 5: Customizing and Administering CONTROL-O

When the snap is completed, the following message appears on the console:

```
CTO150I SNAP COMMAND WAS PERFORMED SNAPID=XXXXX
```

Where:

SNAPID is the snap identifying number which appears on the lower right of the screen after the snap is completed.

Displaying Internal Resource Utilization

To obtain information on internal resource utilization, issue the following operator command:

```
F CONTROL0,WATERMARKS
```

The response to this command is displayed as message CTO15DI. See the IOA Messages and Codes manual for an explanation of this message.

Customization of Automation Options

Automation Options (AOP) screens can be modified and new screens defined to meet site requirements. (Refer to sections 1 and 2 in the CONTROL-O User Manual for a description of the Automation Options facility.)

AOP Overview

The Automation Options facility is comprised of the following components:

1. Menu Definition Members

Each menu definition is written in a member containing parameters (keywords and values) that describe the options in the menu. The following are defined for each option:

- Name of the option.
- Parameters that define the format of the display line.
- Name of the program to be executed, or (for a submenu) name of the member containing the submenu definition.

Menu definition members for each Automation Option should be assigned the same name as the menu, prefixed by ##, in the IOA MSG library. For example, the definition for menu OPER should reside in member ##OPER.

The main Automation Options menu is defined in member ##AOP.

To update or add an Automation Option, it is necessary to access the appropriate menu definition member and modify it as described later in this section.

2. Format Members

Formats describe how lines provided by a client program are to be displayed on the screen. The format of the primary Automation Options screen provided by CONTROL-O is specified in member \$\$AOP in the IOA MSG library. The user can write new formats and save them as members in the IOA MSG library.

Note Member `$$AOP` should not be modified to avoid problems in the execution of the feature.

3. Client programs

Client programs can be any independent load module, TSO CLIST, or REXX EXEC, which the Automation Options facility invokes either to perform functions or to provide lines to be displayed.

Menus

Update the appropriate member (in the IOA MSG library) with the parameters of the new options. Each of the parameters (described below) defines a line in the specification of the new Automation Option.

Menu Member Syntax

The following syntax rules should be considered when defining menus of new options.

- A menu statement consists of one or more lines, each with of a maximum of 72 columns.
- Each menu statement is comprised of a parameter (keyword) and its value, separated by one or several blank spaces. Parameters can be specified in uppercase or lowercase.
- Data can be specified anywhere between columns 1 and 71.
- A non-blank character in column 72 indicates that the statement is continued on the next line.
- Data on continuation lines must start at column 16.
- An asterisk (*) in column 1 specifies that the line is a comment.

Parameters

OPTION	Name of the option (1-8 characters) as it appears in the menu screen. OPTION must be unique within the menu member. Mandatory. Parameter OPTION heads a group of statements. The group ends when a new OPTION parameter or the end of file is encountered. Example: OPTION USERINFO
DESC	Description of the option as it appears in the menu screen. DESC is a string of 1-70 characters, enclosed in single quotes. Mandatory. Example: DESC 'Query USER database'
LINECMD	Line command character. This character defines the line command used to select the option to be executed from the Automation Options entry panel.

Section 5: Customizing and Administering CONTROL-O

The line command character can be any character, with the exception of ? which does not invoke the client program. When selected, ? displays the prompt window.

LINECMD heads a subgroup of statements that define actions to be performed when the specified line command is specified. The subgroup ends when a new LINECMD, a new OPTION, or end of file is encountered.

At least one LINECMD must be specified within an OPTION group of statements. The following parameters help define the LINECMD subgroup: PROGRAM, PROMPT, PARM, RETURNS, FORMAT, OPTLIST. They are described below.

PROGRAM

Program name (1 to 8 characters). Mandatory within a subgroup of statements headed by a LINECMD statement. PROGRAM names a client program which is invoked when the preceding line command LINECMD is entered. The program can be any regular load module, a TSO CLIST, or a REXX EXEC. The PROGRAM statement can include subparameters to define special program attributes. When no subparameters are specified, the client program performs according to Automation Options standard linkage conventions (described in member DOCOAACP in the IOA SECUDATA library) and is called an AOP program.

Optional subparameters are:

IMMEDIATE Prompt window is not displayed. If parameters are specified, they are passed directly to the program.

NOSCREEN Display lines are not returned. (The client program may perform its own terminal I/O, without using the Automation Options facility.) Programs specifying NOSCREEN can be invoked only under TSO.

ISPFENV Valid ISPF environment is required. Programs specifying ISPFENV can be invoked only under ISPF.

TSOCP Client program is invoked as a TSO command processor.

EXECPGM Parameters are passed to the program in the format of the PARM= statement of a JCL EXEC program.

REFRKEY Definition of function key(s) for REFRESH action. Valid values are ENTER and PF4. Default value is PF4.

Examples

PROGRAM AOPGUINF

Client program retrieves user information which is displayed later on the screen.

PROGRAM SDSF IMMEDIATE NOSCREEN TSOCP

TSO command processor does not receive any prompts and performs its own terminal I/O.

PROGRAM CLIST1 IMMEDIATE NOSCREEN TSOCP

Section 5: Customizing and Administering CONTROL-O

TSO CLIST does not receive any prompts and performs its own terminal I/O.

```
PROGRAM CTOTAMN
OPTLIST ## SAMPLE
```

Invoke supplied program CTOTAMN to display the submenu defined in member ##SAMPLE.

PROMPT

Request for data to be passed to the client program. Optional. A maximum of 14 PROMPT statements may be specified. When the LINECMD character is specified and at least one PROMPT parameter has been specified, a prompt window is opened with as many prompt lines as PROMPT parameters in the menu. The user fills in the required input, then presses <ENTER>, thus calling the client program. The answers to the prompts are passed to the client program as an array unless parameter PARM (described below) is specified. Valid format is:

```
PROMPT data1,length,data2
```

Where:

data1 String of 1-20 characters between quotes. It is considered a comment and should be used to describe the required input. Mandatory.

Length Decimal number (0-44) specifying the length of the prompt input field. Mandatory.

Data2

String in quotes specifying the prompt's initial value. Optional. If data2 is specified, its length must be less than or equal to length. Default value is blanks.

Examples

```
PROMPT 'User name =====>',8
```

The user is prompted to enter an 8-byte input field. The field is initialized to blanks.

```
PROMPT 'Department =====>',20,'TECH-SUPPORT'
```

The user is prompted to enter a 20-byte input field. The field is initialized to the value of 'TECH-SUPPORT' padded to the right with blanks.

PARM

String delimited by two identical characters (e.g., both PARM /xxx/ and PARM 'xxx' are valid; PARM /xxx' is invalid since the delimiters of the string are not identical).

The string contains input parameter(s) that is required by the client program. Data between the delimiters can consist of any characters (except the character chosen as delimiter), including special AutoEdit variables: %%Pn and %%. , where:

- %%Pn is the value assigned to PROMPT number n.
- %% . is a special variable used to concatenate two AutoEdit variables.

Functions such as %\$\$SUBSTR can also be specified.

Section 5: Customizing and Administering CONTROL-O

After AutoEdit symbols have been resolved, the string's length cannot exceed 255 characters.

Example 1

```
PARM #00' '77#
```

Passes to the client program the value: 00' '77

Example 2

```
PROMPT 'User name =====>', 8  
PROMPT 'Department =====', 20, 'TECH-SUPPORT'  
PARM /QUERY, USERID=%P1, DEPARTMENT=%P2/
```

The first parameter is blank and the second parameter is TECH-SUPPORT.

If the user enters UserA on the first PROMPT line, leaves the second PROMPT line unchanged, and presses <ENTER>, the following string is passed to the program:

```
QUERY, USERID=UserA, DEPARTMENT=TECH-SUPPORT
```

Note When both PROMPT and PARM statements are specified, the order of their appearance within the menu is not important; however, the data is passed to the program as specified in the PARM statement.

RETURNS

String, delimited by quotes (maximum 200 characters), which defines the layout of a line of display. Optional. Default is: 1 LINE +77. If display type A exists in the format member of the option, it must contain the name of all the variables defined here.

The string is a simple CONTROL-O template which names the format variables and their positions in the displayed line. Variable names must not exceed eight characters and their positions must be specified numerically only. (Refer to the description of the %\$PARSE function in Section 4 of the CONTROL-O User Manual for detailed information on templates.)

For each variable on the template, the following fields should be specified:

starting position Starting position of the variable on the template string (character).

variable name Name of the variable (in uppercase).

length Length of the variable (character).

Example

```
RETURNS '1 FIRSTNME +8 20 LASTNME +12 60 ADDRESS +35'
```

Divides the returned lines into the following displayable variables:

FIRSTNME Begins in column 1 and its length is 8 characters.

LASTNME Begins in column 20 and its length is 12 characters.

ADDRESS Begins in column 60 and its length is 35 characters.

Section 5: Customizing and Administering CONTROL-O

These variables must be defined in the corresponding format member.

Note RETURNS cannot be specified for programs specifying NOSCREEN.

FORMAT Name of the format member (1-8 characters) used for displaying the lines returned from the client program. Format member name must begin with \$\$, in accordance with Automation Options conventions. Optional. FORMAT is ignored if specified in a menu together with subparameters NOSCREEN, TSOCP, or EXECPGM.

Example

FORMAT \$\$UINFO

\$\$UINFO is the format member for use in displaying the return lines.

OPTLIST Name of a lower level menu member (1-8 characters). Optional. To process a lower level menu, invoke the supplied program CTOTAMN and specify the \$\$AOP format member (or a similar format member) to display your customized Automation Options screen. OPTLIST must begin with ##, in accordance with Automation Options conventions.

Example of Menu

```
OPTION USERINFO
DESC 'Query USER database'
LINECMD S
PROGRAM CTOTAMN
OPTLIST ##UINFO FORMAT $$AOP
RETURNS '1 OPTION +8 9 DESC +60'
```

When Option USERINFO is selected (by specifying S in the select option field of the Automation Options screen), a new screen is displayed, showing the user-defined Automation Options specified in the ##UINFO menu member.

Format Members

Format members are used to format the lines displayed on the screen. Refer to the "Customizing IOA Display Format Members" in Section 2 of this guide for details on how to define and customize format members.

Note The first @FIELD in each @LINE must be SAOPLCMD. The other @FIELDS in each @LINE must contain variables that match those in the RETURNS statement of the menu.

Client Programs

A client program is a load module, a TSO CLIST, or a REXX EXEC that is invoked by the Automation Options facility when selected by a line command character (other than ?).

Section 5: Customizing and Administering CONTROL-O

Programs may be called that do or do not require parameters to perform a certain function and then return without displaying lines. Sophisticated AOP programs receive parameters via PROMPTs or PARM statements and return lines for display.

The program attributes in the menu define the program characteristics. Before the program is called, the Automation Options facility reviews those attributes and takes special actions if necessary.

Examples

If ISPFENV is specified, the Automation Options facility ensures that a valid ISPF environment exists before invoking the client program.

If IMMEDIATE is specified, the Automation Options facility checks the PROMPTs and the PARM specification in the menu, if any, and passes the data to the client program without displaying it on the screen.

General registers 1 – 15 contain the following information at time of entry to the client program:

R15	Entry point of the called program
R14	Return address
R13	Address of save area
R1	Address of input data
R0	N/A
R2-R12	N/A

Program Input

The program input varies according to the type of the program: EXECPGM, TSOCP, or AOP (client program as described above). A full description of the input data is included in member DOCOAOC in the SECUDATA library.

Execution Process

Client programs of the TSOCP or EXECPGM type must follow the rules specified below:

1. The program is to be considered an extension of IOA. Since it runs under IOA, an abend of the client program triggers an abend of the IOA session. Storage overlays within the program may produce unpredictable results.
2. Registers must be saved and restored according to standard IBM conventions.
3. Areas acquired by the program during execution must be released before returning to the Automation Options facility. Failure to follow this rule may result in storage buildup, a situation that may lead to abends.
4. CLISTs and REXX EXECs can be used under MVS/XA or higher level operating systems.

Section 5: Customizing and Administering CONTROL-O

Client programs that are not of type TSOCP or EXECPGM are called AOP programs. A full description of the execution process of an AOP program is included in member DOCOAACP in the SECUDATA library.

Client Program Linkage Conventions

Client programs which do not specify attributes NOSCREEN, TSOCP, or EXECPGM, return a set of lines to the Automation Options facility. These lines are displayed according to the parameters specified in the format member referenced by the FORMAT statement in the menu definition member.

A full description of the Client Program Linkage Conventions for an AOP program is included in member DOCOAACP in the IOA SECUDATA library.

Considerations for When CONTROL-M and CONTROL-O Are Installed

When CONTROL-O and CONTROL-M are installed, the CONTROL-O monitor assumes the responsibilities of both the CMEM facility and CONTROL-O. The way in which the CONTROL-O monitor functions depends on how organizational requirements, scheduling, and automation have, until now, been addressed at your site.

The following DD statements in the CONTROL-O startup procedure are relevant to this issue:

DARULLST	References a member containing the list of CONTROL-O rule tables to be loaded by the CONTROL-O monitor. The default name for this member is RULELIST.
DACTMLST	References a member containing the list of CMEM rule tables to be loaded by the CONTROL-O monitor. The default name for this member is IOACMEML.

Below is a description of the different ways in which the CONTROL-O monitor can operate:

- If all automation was implemented only via the Online CONTROL-O Rule Definition facility (and the CMEM facility was not used):

Start CONTROL-O via command S CONTROL0. CONTROL-O loads the rule tables listed in the member referenced by DD statement DARULLST. The member referenced by DD statement DACTMLST must be empty.

- If automation was implemented via both the CONTROL-O Rule Definition facility and the CMEM facility:

Start CONTROL-O via command S CONTROL0. CONTROL-O first loads the CMEM rule tables listed in the member referenced by DD statement DACTMLST. CONTROL-O then loads the CONTROL-O rule tables listed in the member referenced by DD statement DARULLST.

Section 5: Customizing and Administering CONTROL-O

- If automation was implemented via both the CONTROL-O Rule Definition facility and the CMEM facility but, temporarily, only CMEM functionality is desired (e.g., for a trial period):

Start CONTROL-O via the following operator command:

```
S CONTROL-O,TYPE=CTOCMEM
```

This command causes the CONTROL-O monitor to load only CMEM rule tables. CONTROL-O loads the CMEM rule tables listed in the member referenced by DD statement DACTMLST. Loading of CONTROL-O tables is skipped. This mode of operation should not be used on a long-term basis.

More information on the management of the CMEM facility by the CONTROL-O monitor is provided at the beginning of this section.

Section 6: Customizing and Administering CONTROL-B

General	6-1
CONTROL-B New Day Procedure	6-1
Reformatting the Active Balancing File - Utility CTBFRM.....	6-2
Scheduling Balancing Missions.....	6-2
Date Control Record.....	6-2
Format of the Date Control Record.....	6-3
Use of the Date Control Record by the New Day Procedure	6-3
Invoking CONTROL-B (Runtime Environment)	6-4
Passing Arguments While Invoking CONTROL-B	6-5
Invoking CONTROL-B Via a Direct Call.....	6-6
Invoking CONTROL-B From a Job Step or Application Program.....	6-7
Invoking CONTROL-B by CONTROL-M	6-8
Invoking CONTROL-B by CONTROL-D	6-8
Invoking CONTROL-B With Balancing Missions.....	6-9



General

This section describes the initialization and customization features that are available within CONTROL-B.

CONTROL-B New Day Procedure

The CONTROL-B New Day procedure (CTBNDAY¹) performs certain automatic functions which start a new day under CONTROL-B. It performs daily clean-up operations on the CONTROL-B Active Balancing file and, optionally, other files. It also orders (activates) CONTROL-B missions.

The New Day procedure can be activated automatically by a scheduler (e.g., CONTROL-M) at a predefined time each day. It can also be activated manually by submitting a job or starting a started-task (STC).

The supplied New Day procedure activates program CTMILY. Program CTMILY reads the sequential dataset (or member) referenced by DD statement DAPROG. This dataset contains a list of programs to be activated.

The normal sequence of programs and utilities activated by the New Day procedure is:

Program	Purpose
CTBCHK	Checks the current date and its relation to the General Date Control record (see below). The program may communicate with the computer operator to verify that CONTROL-B is activated on the right date.
CTBFRM	Reformats the Active Balancing file. Unnecessary missions are erased and the file is compressed.
CTBBAO	Places balancing missions in the Active Balancing file according to the date in the Date Control record and the scheduling specifications.
CTBPDA	Records the end of the daily run.

¹ This name is defined during the installation process and may be different at your site.

Reformatting the Active Balancing File - Utility CTBFRM

As explained above, the New Day procedure activates utility CTBFRM to clean up the Active Balancing file. Utility CTBFRM reformats the Active Balancing file. Missions and rules that have already executed and ended OK, and missions whose MAXWAIT parameter has been exceeded, are erased from the file. Unscheduled rules are automatically assigned a MAXWAIT value of 0. After these mission entries are erased, the Active Balancing file is compressed.

Note If utility CTBFRM abends, information is recovered automatically after startup using an automatically-generated backup copy of the Active Balancing file named ABFBKP. For more information see utility CTBFRM in Section 9 of this guide.

Scheduling Balancing Missions

The second function performed by the New Day procedure is the determination of which missions may potentially be activated on a specific date (i.e., which missions will be placed in the Active Balancing file).

Which missions will be scheduled depends on the working date. See "Date Control Record" below for additional information.

Date Control Record

The CONTROL-B New Day procedure places mission entries in the Active Balancing file. The New Day procedure usually runs once a day on one or more balancing mission definitions. Mission orders are selected according to the working date specified to the New Day procedure. The New Day procedure is date-dependent and certain special situations must be handled. For example: the computer hasn't been working for a day or more due to holidays, hardware failure, etc.

To maintain control on the last running date of the New Day procedure, CONTROL-B uses a special Date Control record. The Date Control record is analyzed by the New Day procedure to determine the current running date, the last running date, and possible error situations.

In certain cases, it is possible to manually update the Date Control record using a regular editor. This affects scheduling accordingly. The Date Control record is referenced by DD statement DACHK in the CTBNDA procedure.

Format of the Date Control Record

Each Date Control record is 80 characters long, and contains the following fields:

Columns	Item	Purpose
01-06	date-1	Current (or last) original scheduling date.
18-23	date-2	Current (or last) original scheduling date of balancing missions using parameters DATES, DAYS, DCAL, and/or WCAL.
25-30	date-3	Current (or last) original scheduling date of balancing missions using the DATES, DAYS, DCAL and/or WCAL finish indicator.
43-48	date-4	Current (or last) original scheduling date of jobs using parameter WDAY5.
50-55	date-5	Current (or last) original scheduling date of jobs using the WDAY5 "finish indicator."
60-65	date-6	Last date the Active Balancing file was formatted by utility CTBFRM. This field prevents formatting from being performed twice on the same day. If there are any problems concerning the date, the program presents the operator with a series of questions.
67-72	date-7	"Finish indicator" date of the CONTROL-B New Day procedure.

The format of the dates is determined by the date standard used at your site.

Use of the Date Control Record by the New Day Procedure

The workflow of the New Day procedure is affected by the Date Control record and the contents of the Date Control record are updated by the New Day procedure. The main steps of the New Day procedure are:

- Checking the last running date of the New Day procedure (via internal program CTBCHK).

The first date in the Date Control record (column 1-6) is compared to the current installation working date (at the time of the run). If they are identical, a message indicates that the New Day procedure has already run today and the condition code is set to 0004.

If the current installation working date is earlier than the first date of the Date Control record, the New Day procedure stops executing and notifies the user that an attempt was made to run a New Day procedure before its time.

If the current installation working date is later than the first date of the Date Control record (the normal situation) the first date of the Date Control record (columns 1-6) is updated automatically to the current installation working date which is then used by all components of the New Day procedure as the current scheduling date.

If the New Day procedure did not run for more than one day, a warning message is issued and the New Day procedure attempts to schedule balancing missions for all of the days which have passed since the last scheduling date (according to the production parameters).

Section 6: Customizing and Administering CONTROL-B

The program “asks” the operator a series of questions regarding the computer's current date. This ensures that an incorrect date was not inadvertently entered during IPL.

- Placing balancing missions in the Active Balancing file according to the current scheduling date and the last running date (via internal program CTBBAO).

Program CTBBAO acts on a user-defined balancing mission table referenced by DD statement DABAL. The program checks whether each mission in the table should be scheduled on one or all the days that have passed since the last original scheduling date (date-3 or date-5) until the current installation working date (date-1). If a mission should be scheduled, it is placed in the Active Balancing file.

For example, if a computer did not operate from the 20th to the 23rd, a mission which was originally scheduled to run on the 20th did not run. Program CTBBAO decides whether or not the mission should be retroactively scheduled to run on the logical date of the 20th. See parameter RETRO in Section 6 of the CONTROL-B User Manual.

When the program finishes processing the mission definitions, the “finish indicator” dates (date-3 and date-5) are updated to the installation working date calculated by program CTBCHK (date-1).

Before program CTBBAO starts operating, it compares date-2 with date-3 and date-4 with date-5. If they are not equal it probably means that a previous run of program CTBBAO has abended. The user is notified and the program terminates. To correct the error, you must alter the Date Control record to the correct date values (using a standard editor).

Note When manually modifying the Date Control record, verify that the same missions are not scheduled to run twice on the same day.

- Recording the end of the Daily run (via program CTBPDA).

Program CTBPDA updates the “finish indicator” date (date-7) by placing in it the value of the current installation working date (date-1). This is used to indicate that the New Day procedure finished successfully.

It is highly recommended that utility CTBJAFDL be run as the last step of the New Day procedure. See Section 9 in this guide for more information about this utility.

Invoking CONTROL-B (Runtime Environment)

CONTROL-B rules are invoked by the CONTROL-B Runtime environment. Therefore, the Runtime environment must be invoked in order to invoke a CONTROL-B rule.

The CONTROL-B Runtime environment can be invoked by:

- A direct call

When the CONTROL-B Runtime environment is invoked by a direct call, the name of the rule to be invoked is one of the parameters passed in the call. The direct call to the Runtime environment can come from an application program, a job step, or another IOA product. The specified rule name is searched in the libraries referenced by DD statement DABRULE. No scheduling criteria are specified; the same rule is always activated.

Section 6: Customizing and Administering CONTROL-B

- A balancing mission

A search is performed for the specified balancing mission name in the Active Balancing file. After locating the appropriate mission, the rule specified by the mission is activated. When CONTROL-B is invoked using this method, scheduling criteria can be specified for date dependent balancing activities. For example, the same mission may activate different rules based on the day of the week using different categories of the mission.

These methods of rule activation are discussed on the following pages.

Passing Arguments While Invoking CONTROL-B

Arguments can be passed when directly invoking CONTROL-B via procedure CONTROLB by specifying the following in the JCL statements of the job step:

```
// EXEC CONTROLB,RULE=rule,GROUP=group,MISSION=mission,ARG=arglist
```

Where:

rule	Name of the rule to be invoked.
group	Group name of the rule to be invoked.
mission	Mission name of the rule to be invoked.
arglist	Arguments, separated by commas, to be passed to the rule.

The arguments specified in arglist are accessed via System variable RARGnn, where nn is a one- or two-digit number which represents the position of the argument in arglist. To access the value of the first argument in the list, specify RARG01 within a rule.

Example

The following sample JCL and rule definition demonstrate how an argument can be passed when invoking CONTROL-B. In this case, the value PROD1.COPY.FILE1 is passed to CONTROL-B as an argument. In CONTROL-B, this value is checked to indicate whether or not the dataset exists at the site. The JCL continues to run according to the result of the CONTROL-B activation.

```
//CHRTAPE JOB 0,ACCT,CLASS=A,MSGCLASS=X,COND=(0,NE)
//CTB EXEC CONTROLB,RULE=RULDSN,ARG='PROD1.COPY.FILE1'
//COPYTAPE EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//IN DD DSN=PROD.COPY.FILE1,DISP=SHR
//OUT DD DISP=(NEW,PASS),DSN=COPY.FILE1,
// VOL=SER=TAPE18,UNIT=TAPE,
// LABEL=(1,SL)
//SYSIN DD *
C O=OUT,I=IN
//
```


Section 6: Customizing and Administering CONTROL-B

```
LIBRARY : CTB.PROD.RULES                             RULE : RULDSN
COMMAND ===>                                      SCROLL===> CRSR
+-----+
OWNER   M42A                                   GROUP PROD1
UPDATED 08/08/96 - 12:19:37 BY M42A
DESC
OPTIONS
=====
EXECUTE BLOCK1  UPON                                 C
ON DATA
IF   ISDSN('%RARG01')                                 C
  DO TERMINAT = OK      COD 0000
ELSE
  DO PRINT = FILE %RARG01 DOES NOT EXIST             F  C
  DO TERMINAT = NOTOK    COD 0008
=====
EXECUTE          UPON                                 C
ON
===== >>>>>>>>>>>>>>>>>> END OF RULE DEFINITION PARAMETERS <<<<<<<<<<<<<<<<<< =====
PLEASE FILL IN RULE DEFINITION                                13.25.00
```

Invoking CONTROL-B Via a Direct Call

The CONTROL-B Runtime environment can be invoked by a direct call from any of the following:

Application program: The name of the rule to be invoked is specified by the application program's call to CONTROL-B. See Appendix C of the CONTROL-B User Manual for additional information.

Format

```
CALL CONTROLB(<rule name>)
```

Job step: The rule to be invoked is specified by the JCL statement of the job step (in the balancing job). This is the most frequently used method.

Format

```
// EXEC CONTROLB,RULE=rulename,GROUP=group,ARG=(arg1,...argn)
```

Note The GROUP specification is mandatory only if parameter GROUP in the rule definition was omitted. Otherwise (i.e., parameter GROUP was specified in the rule definition), the GROUP specification is optional.

The ARG specification allows up to 50 arguments. See "Passing Arguments while invoking CONTROL-B" below.

Another IOA product: The rule to be invoked is specified by parameter DO CTBRULE in a CONTROL-M job scheduling definition or a CONTROL-D decollating mission definition.

Section 6: Customizing and Administering CONTROL-B

When CONTROL-B starts processing, it searches for the specified rule in the libraries referenced by DD statement DABRULE.

If the rule is found, it is invoked. If the rule is not found, a runtime error occurs.

This flexibility enables the CONTROL-B user to select one or more appropriate points in the life cycle of a particular data source during which balancing activities should be performed. For example:

1. CONTROL-B can be invoked before the creation of a report to check the validity of the report job input.
2. CONTROL-B can be invoked before updating a database (e.g., a DB2 table) or an important file in the system. CONTROL-B checks the validity of the input used for the update.
3. After a report is created, it can be checked during the current job - before executing other steps in the job and before the report is distributed.
4. CONTROL-D can invoke a CONTROL-B rule to validate a decollation process.
5. During out-processing in CONTROL-M, CONTROL-B can be invoked to control a job flow, or to perform balancing actions which verify the results of a job run.

Invoking CONTROL-B From a Job Step or Application Program

The following parameters can be used to invoke CONTROL-B from a job step or from an application program:

- RULE
- MISSION
- GROUP
- ARG

Parameters RULE and MISSION cannot be specified together to invoke balancing operations (i.e., the user must decide whether CONTROL-B should be invoked directly or indirectly). Either the mission name or rule name should be specified but not both.

The group name is determined as follows:

1. If GROUP is specified in the invocation, that group name overrides any other group specification.
2. If GROUP is not specified in the invocation, the group name of the mission is used (if balancing operations are invoked by a mission).
3. Otherwise, the group name specified in the rule definition is used as the current group.

Note *The specified group must already be defined in the CONTROL-B repository.*

Section 6: Customizing and Administering CONTROL-B

User parameters can be passed to a CONTROL-B rule and accessed by the rule via system variable RARGnn, where nn is a one- or two-digit number which represents the position of the argument in the argument list. To access the value of the first argument in the list, specify RARG01 within a rule.

Examples of Job Step Invocation

Direct invocation of a CONTROL-B rule:

```
// EXEC CONTROLB,RULE=ACCTCHK,GROUP=ACCT,ARG='12/06/96'
```

CONTROL-B invocation via a mission:

```
// EXEC CONTROLB,MISSION=ACTMISS
```

Application program invocation:

```
CALL CONTROLB, (rulename,mission,groupname,result,argument-count,argument-list)
```

See the beginning of this section for more information about CONTROL-B invocation.

Invoking CONTROL-B by CONTROL-M

When CONTROL-B is invoked by CONTROL-M, it is not necessary to specify scheduling criteria for the rule (i.e., there is no need to define a balancing mission) because the scheduling of the rule is determined by the scheduling specified in the CONTROL-M job scheduling definition.

Parameter GROUP is determined as follows:

- If a group name is specified in the CONTROL-M job scheduling definition, the CONTROL-M group specification overrides any other group specification.
- If a group name is not specified in the CONTROL-M job scheduling definition, the group specified in the CONTROL-B rule definition is used as the current group.

A group must be specified in the CONTROL-M job scheduling definition so that the CONTROL-M job and the CONTROL-B rule are logically associated with each other. The group in the CONTROL-B repository is determined outside of the rule by the group specified in the calling CONTROL-M job scheduling definition.

Invoking CONTROL-B by CONTROL-D

When CONTROL-B is invoked by CONTROL-D, it is not necessary to specify scheduling criteria for the rule (i.e., there is no need to define a balancing mission) because the scheduling of the rule is determined by the scheduling specified in the CONTROL-D decollating mission definition.

Parameter GROUP is determined as follows:

- If a group name is specified in the CONTROL-D decollating mission definition, the CONTROL-D group specification overrides any other group specification.
- If a group name is not specified in the CONTROL-D decollating mission definition, the group specified in the CONTROL-B rule definition is used as the current group.

Section 6: Customizing and Administering CONTROL-B

It is highly recommended that a group be specified in the CONTROL-D decollating mission definition (using the SCHED command in the Decollating Mission Definition screen) so that the CONTROL-D decollating mission definition and the CONTROL-B rule will be logically associated with each other. The group in the CONTROL-B repository is determined outside of the rule by the group specified in the calling CONTROL-D decollating mission definition.

Invoking CONTROL-B With Balancing Missions

Balancing missions can be used to invoke CONTROL-B from one of the following:

Application program Name of the mission to invoke is specified by a program call to CONTROL-B in the application program. See Appendix C in the CONTROL-B User Manual for additional information.

Example

```
CALL CONTROLB(<mission>, ...);
```

Job step Name of the mission to be invoked is specified in the JCL statements of the job step (which invokes the CONTROL-B Runtime environment).

Example

```
// EXEC CONTROLB,  
  
// MISSION=DAILYBAL, GROUP=group, ARG=(arg1, ..., argn)
```

Note *The GROUP specification is mandatory only if parameter GROUP in the rule definition was omitted. Otherwise (i.e., parameter GROUP was specified in the rule definition), the GROUP specification is optional.*

The ARG specification allows up to 50 arguments. See "Passing Arguments While Invoking CONTROL-B" above.

When the call/step which invokes CONTROL-B is invoked, CONTROL-B first checks each entry in the Active Balancing file to locate the appropriate balancing mission. The Active Balancing file search method consists of the following phases:

Phase 1 All mission entries with corresponding mission names are checked to see if their runtime scheduling criteria are met. Missions whose runtime scheduling criteria are met advance to the second phase for further checking.

Phase 2 Missions which have passed the first phase are checked to see if the values specified by parameters JOB and STEP match the job name and (optional) step name of the current call/step.

Note *The * mask character can be specified at the end of JOB and STEP names.*

Section 6: Customizing and Administering CONTROL-B

Parameter SCOPE of each eligible mission entry in the Active Balancing file is checked to determine a JOB/STEP match. Parameter SCOPE specifies one of the following options:

- SINGLE** If JOB and STEP values match, the mission entry is eligible for further checking only if it has not been invoked before. Option SINGLE guarantees that this mission entry is only invoked once - by a single CONTROL-B invocation.
- STEP** If JOB and STEP values match, the mission entry is eligible for further checking if it has not been invoked before, or it was previously invoked by the current STEP in the current JOB. Option STEP enables the mission entry to be invoked repeatedly, provided that the mission entry is invoked from the same job step each time (e.g., from an application program that invokes CONTROL-B from within a loop).
- JOB** If the JOB value matches, the mission entry is eligible for further checking if it has not been invoked before, or it was previously invoked by the current JOB. Option JOB enables the mission entry to be invoked by several CONTROL-B invocations, provided that the invocations all originated from the same job.
- ALL** The mission entry is always eligible for further checking. Option ALL enables the mission entry to be invoked regardless of the JOB and STEP that previously invoked the mission entry.

The first mission that matches the above criteria is selected for execution. If no matching mission is found, a runtime error occurs.

Section 7: Customizing and Administering CONTROL-T

General	7-1
The CONTROL-T Real-Time Environment	7-1
Initialization.....	7-2
Procedure CTTINIT.....	7-2
Loading of Rules.....	7-6
Termination.....	7-6
New Day Procedure	7-7
New Day Functions.....	7-7
Repository Structure	7-9
Media Database Structure.....	7-9
Data File Contents.....	7-10
Fields in the Data File.....	7-11
Index File Contents.....	7-12
Structure of the Trace File.....	7-17
Structure of the Stacking Statistics File.....	7-17
Repository Maintenance	7-19
Verifying Data Integrity of Media Databases.....	7-19
Manual Update of the Media Database.....	7-21
Enlarging the Media Database.....	7-21
Enlarging the Trace File.....	7-22
Repository Backup and Recovery	7-23
General.....	7-23
Backup and Recovery.....	7-23
Disaster Recovery.....	7-24
Selective Recovery.....	7-25
General Considerations.....	7-25
Robotic Library Support	7-25
Robotic Tape Libraries Using System Managed Tape Functions.....	7-25
IBM Tape Library Dataserver Support.....	7-26
StorageTek Silo Support.....	7-28
MEMOREX TELEX Automatic Tape Library (ATL) Support.....	7-32
EMASS Automatic Tape Library (ATL) Support.....	7-32
Cartridge Message Display Customization	7-33
External Data Manager Support	7-35
HSM Support.....	7-35

DMS/OS Support	7-35
CA-ASM2 Support	7-36
DF/SMS Support	7-37
Adjusting ACS Routines	7-38
Dynamic Dataset Stacking Facility	7-39
General Considerations.....	7-40
Activating the Dynamic Dataset Stacking Facility.....	7-40
Activating the Dynamic Dataset Stacking Facility in Global Test Mode	7-41
Testing the Dynamic Dataset Stacking Facility.....	7-41
Activating the Dynamic Dataset Stacking Facility for the Entire System.....	7-42
CONTROL-R Support to CONTROL-T	7-43
CONTROL-R Driver Exit CTRX001G.....	7-43
Installing the CONTROL-R Interface	7-43
CONTROL-T Application Programming Interface	7-44
Background Information	7-45
CONTROL-T Control Table (TCT)	7-45
Media Database Structure	7-45
Base Level vs. High Level API	7-46
Base Level API.....	7-47
Record Access by the Base Level API.....	7-47
Macro CTTIOS.....	7-47
Base Level API Access of the Media Database	7-51
Examples for Base Level API	7-53
High Level API.....	7-68
Macro CTTACCDB	7-68
Macro CTTCHKDB	7-70
Media Database Error Handling Routine.....	7-70
Example for High Level API.....	7-71

General

This section describes how the different components and files of CONTROL-T are activated, controlled, and maintained on a regular daily basis. The following topics are discussed in this section:

- The CONTROL-T Real-Time Environment
- New Day Procedure
- Repository Structure
- Repository Maintenance
- Repository Backup and Recovery
- Robotic Library Support
- Cartridge Message Display Customization
- External Data Manager Support
- DF/SMS Support
- Dynamic Dataset Stacking Facility
- CONTROL-T Application Programming Interface

Some administrative tasks are performed via CONTROL-T utilities. The available utilities are described in detail in Section 9 of this guide.

Additional customization and some security can be implemented via CONTROL-T exits. The available CONTROL-T exits are described in Section 10 of this guide.

The CONTROL-T Real-Time Environment

The Real-Time Environment of CONTROL-T is established during the Initialization process.

Once the Real-Time Environment is established, CONTROL-T takes control of tape volumes processed in the system. This is accomplished by the MVS interfaces of CONTROL-T. CONTROL-T performs extensive validity checks to ensure the data integrity of the media library at the site. In addition, stacking can be performed for non-specific mount requests.

As a dataset moves through the system, CONTROL-T gathers relevant information, searches rules to determine the required actions (such as retention and vaulting information), and updates the Media Database.

These topics are described on the following pages.

Section 7: Customizing and Administering CONTROL-T

Initialization

The first step in using CONTROL-T is to initialize CONTROL-T in each CPU. This is done by starting procedure CTTINIT, and is usually performed as part of the IPL process. The initialization process establishes the CONTROL-T Real-Time Environment, according to the following steps:

- Creates the main CONTROL-T Control Table.
- Loads rule tables, pool definitions, and vault definitions into common storage.
- Opens the files necessary for the Real-Time Environment: The Media Database, Trace and Stacking Statistics files.
- If requested by the user, loads and defines the CONTROL-T SVC (the SVC number is specified as an installation parameter).
- If requested by the user, establishes interfaces between CONTROL-T and MVS by setting intervention points in MVS where CONTROL-T obtains control. These interfaces remain until a termination process is run.
- Establishes the WTO (Write to Operator) Intercept so that mount messages can be modified.
- Initializes the subsystem for handling stacking and other functions.

The block of information in common storage containing key control data and pointers is referred to as the CONTROL-T Control Table (TCT). Its common storage address is used by all real-time components (e.g., the CONTROL-T SVC and the subsystem) and certain other components (e.g., EDM controlled datasets). All activity starts from this block, and the block contains pointers to all CONTROL-T tables and modules in storage. The control block, along with its related routines, remains in common storage until CONTROL-T is shut down.

Upon completion of the initialization process, CONTROL-T's Real-Time Environment is established and CONTROL-T takes control of removable media processing.

After initialization, no address space is needed for the CONTROL-T Environment.

Procedure CTTINIT

The CTTINIT procedure controls the initialization, termination, and operation of CONTROL-T. The following operator command invokes the CTTINIT procedure:

```
S CTTINIT, PARM= 'MODE=xxxxx [, DBGLEVEL=nnn, DBGJOB=xxxxxxxx ] '
```

The parameters of this command are described in detail below. The PARM field can contain one of the following values:

Note The *MODE* keyword can be omitted. For example, *PARM='MODE=INIT'* and *PARM=INIT* are equivalent.

Section 7: Customizing and Administering CONTROL-T

Parameters Relating to Initialization/Termination/Refreshing

MODE=INIT Initializes CONTROL-T. For details on how CONTROL-T is initialized, refer to "Initialization" in Section 1 of the CONTROL-T User Manual. This command must be issued on each MVS system in which CONTROL-T is to be activated.

MODE=TERM Terminates CONTROL-T. This command must be issued on each MVS system in which CONTROL-T was activated.

MODE=RELOAD,EXIT=exit Loads a new copy of a CONTROL-T exit without the need to restart CONTROL-T. Specify the full name of the exit module, e.g., CTTX004.

Note Reloadable exits are: CTTX002, CTTX003, CTTSE03, CTTX004, CTTSE04, CTTX005, CTTX006, CTTSE06, CTTX009, CTTSE09, CTTX010, and IOAX035.

MODE=RELOAD,TBLT=RULE When CONTROL-T is initialized, it reads a list of rule tables from the member referenced by DD statement DARULLST in procedure CTTINIT. The supplied default member RULLIST is in the CONTROL-T PARM Library. For every line in the list, CONTROL-T loads the specified rule table from the specified library.

To replace all currently active rules with a new (updated) copy of the rules, issue the following operator command:

S CTTINIT, PARM='MODE=RELOAD, TBLT=RULE'

MODE=RELOAD,TBLT=POOL When CONTROL-T is initialized, it reads pool definitions from the member referenced by DD statement DAPOOLS in procedure CTTINIT. These pools are defined via the CONTROL-T Pool Definition screen.

If you modified or added pools and you want to replace the currently active pools with a new (updated) copy of the pools, issue the following operator command:

S CTTINIT, PARM='MODE=RELOAD, TBLT=POOL'

MODE=RELOAD,TBLT=VAULT When CONTROL-T is initialized, it reads vault definitions from the member referenced by DD statement DAVLTS in procedure CTTINIT. These vaults are defined via the CONTROL-T Vault Definition screen.

If you modified or added vaults and you want to replace the currently active vaults with a new (updated) copy of the vaults, issue the following operator command:

S CTTINIT, PARM='MODE=RELOAD, TBLT=VAULT'

Section 7: Customizing and Administering CONTROL-T

MODE=RELOAD,TBLT=PARM

The following CTTPARM options can be modified while CONTROL-T is active. There is no need to shut down CONTROL-T.

ABENDACT	EXPDTDDN	RECREATE
BLPDEF	EXPDTYPE	RTNUPD
CNGMSGID	EXTRNVOL	SCRPROT
DAYTIME	LBLROUTC	SMSINTR
DEFABEND	MODE	STKDEFFT
DEFEXPDT	MSGFMT	STKSRCHL
DSNMCHK	NLASKOP	STKTEST
DYNDS	OVERJCL	VLBYDS1
DYNSTYP	RBTTYPE	X98ASKOP
DYNVOL		

To modify these options, run job CTTPARMJ to create a new CTTPARM module. Then issue the following operator command:

```
S CTTINIT, PARM='RELOAD, TBLT=PARM'
```

Parameters Relating to MVS Interfaces

MODE=VERIFY

Verifies that CONTROL-T MVS interfaces can be applied and the SVC can be installed.

This mode can be used during CONTROL-T installation, or when bringing up a new MVS version, to determine if CONTROL-T can be initialized.

MODE=CHECK

Checks if CONTROL-T MVS interfaces are already applied and the SVC already installed.

Because CONTROL-T has no active address space (job or started task) in the system, this mode is used to determine if CONTROL-T is up.

VERIFY can be executed only if the CONTROL-T real-time environment is not active. CHECK can be executed whether or not CONTROL-T is active.

Parameters Relating to Operating Status

MODE=DORM

Makes CONTROL-T temporarily dormant. After this command is issued, CONTROL-T does not intervene in any job, does not check expiration dates, etc.

The CONTROL-T environment (CONTROL-T tables, rules, I/O control blocks, etc.) is saved.

Section 7: Customizing and Administering CONTROL-T

MODE=SUSPEND	Temporarily suspends removable media activity. After this command is issued, all jobs which access removable media wait for an operator reply specifying how to proceed. The operator can specify that processing should continue without CONTROL-T, should wait for a RESUME command, or should abend. The CONTROL-T environment (CONTROL-T tables, rules, I/O control blocks, etc.) is saved.
MODE=RESUME	Causes normal CONTROL-T operations to resume after MODE=DORM or MODE=SUSPEND commands are issued.
MODE=STOPSTK	Stops the Dynamic Stacking facility.
MODE=STARTSTK	Starts/restarts the Dynamic Stacking facility.

Problem Resolution Parameters

Note The following parameters should only be used if requested by your IOA representative or Technical Support.

DBGLEVEL	Debug level. A numeric value from 0 to 255 can be specified. 0 No debug messages are printed. 1-255 Debug messages are printed.
DBGJOB	Name of the job for which debugging should be enabled. If DBGJOB is not specified, or DBGJOB=ALL is specified, debugging is enabled for all jobs.

Sample JCL for procedure CTTINIT can be found in the CONTROL-T Procedure Library. The following DD statements are important for managing and maintaining CONTROL-T:

DARULLST	References a member which contains a list of the rules that should be loaded upon initialization/reload.
----------	--

Example

```
* CTT.V500.RULES    BKPRULE    ORDER
* CTT.V500.RULES    SAVRULE    ORDER
```

For additional information see "Loading of Rules" below.

DAPOOLS	References a member that contains pool definitions. By default, this member is \$\$POOL in the PARM Library. Several pool definition members can be concatenated and specified with this DD statement.
DAVLTS	References a member that contains vault definitions. By default, this member is \$\$VAULT in the PARM Library. Several vault definition members can be concatenated and specified with this DD statement.

Loading of Rules

When CONTROL-T is started (or rules need to be re-loaded), CONTROL-T attempts to read a list of rule tables from the member referenced by DD statement DARULLST. These rule tables are automatically loaded by CONTROL-T. The supplied default member RULLIST is in the CONTROL-T PARM library. Each line in the list has the following format:

date library table operation

Where:

date	The date of the rule table. If a specific date is designated, it is used to analyze the Basic Scheduling parameters (see below). The date format is mmddy, ddmyy, or yymmdd (depending on the site standard). Specifying an asterisk (*) indicates the current CONTROL-T Installation Working Date.
library	Name of the library in which the rule table is located.
table	Name of the rule table.
operation	Operation to be performed on the rules in the specified table. Valid values are: ORDER Each rule's Basic Scheduling parameters are compared to the specified date (or the current CONTROL-T Installation Working Date when * is specified). If the rule should be scheduled on that date, the rule is loaded by CONTROL-T and activated. Default. FORCE Each rule is loaded by CONTROL-T and activated. Use the FORCE option when you do not want to use CONTROL-T scheduling options.

For every line in the list, CONTROL-T loads the specified rule table from the specified library.

Termination

CONTROL-T should be in operation from computer startup to shutdown (i.e., all the time) so that it can intercept all removable media management activities. Information about jobs which access removable media while CONTROL-T is inoperative, is not recorded.

When CONTROL-T is terminated, a check is triggered for active jobs which access removable media in the system. If such jobs exist, the operator is prompted to cancel the termination, retry the termination, or continue (force) the termination.

When CONTROL-T is terminated, all traces of the CONTROL-T Real-Time Environment are removed from the system, including the following:

- Interfaces established between CONTROL-T and MVS (i.e., intervention points in MVS) are removed.
- The CONTROL-T SVC is removed from the system.
- The WTO Intercept is disabled; mount messages are no longer modified.

New Day Procedure

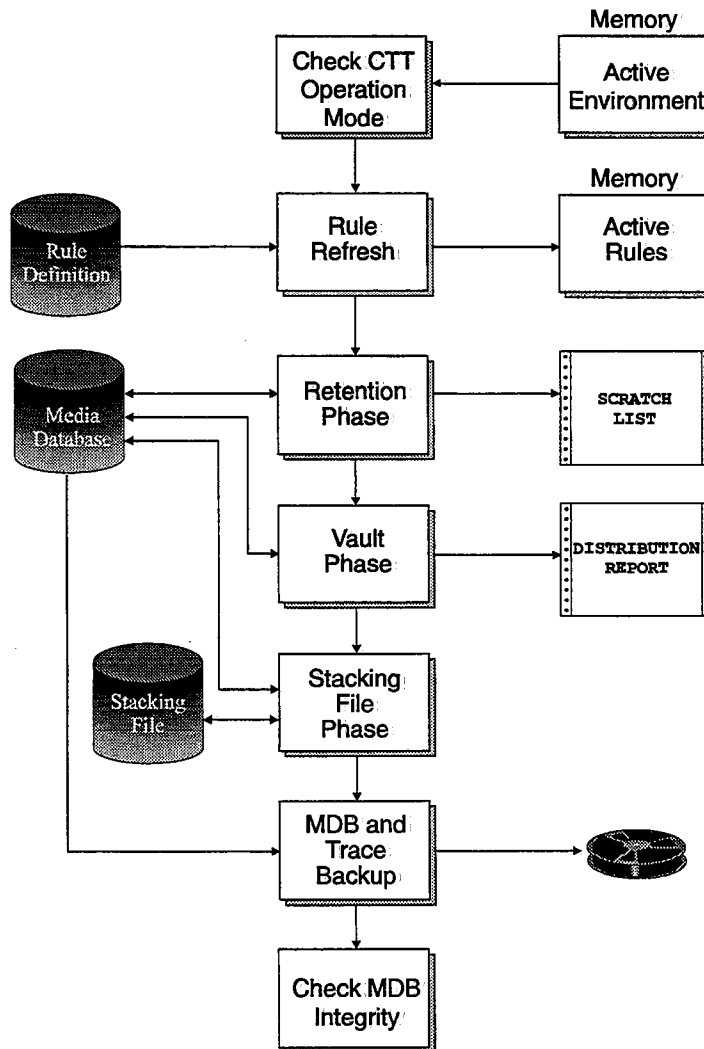
The New Day procedure is the primary mechanism used to perform daily maintenance on the CONTROL-T Media Database and related files.

The New Day procedure, called CTTDAY, can be activated automatically by a scheduler (e.g., CONTROL-M) at a predefined time each day. The New Day procedure can also be activated manually by submitting a job.

New Day Functions

The maintenance operations illustrated and described below are performed by the New Day procedure:

New Day Processing¹



¹ The order of processing may be changed in certain situations.

Section 7: Customizing and Administering CONTROL-T

Check CONTROL-T Operation Mode

Procedure CTTINIT (with MODE=CHECK specified) checks whether or not CONTROL-T is active, and whether or not Dynamic Dataset Stacking is operational. This step also checks the expiration date of the specified password.

Rule Refresh

Procedure CTTINIT can be invoked with parameter MODE=RELOAD to refresh CONTROL-T's active rules. The user should consider whether or not this step should be included in the New Day procedure. The following are some considerations that should be taken into account:

1. Rules are automatically refreshed when CONTROL-T is initialized (usually when an IPL is performed). If a rule definition is changed (in the Definition Library), the modified definition takes effect only when the next IPL is performed (monthly, weekly, etc.).
2. The retention management and vault management utilities optionally load rules from the library as their first step. In this case, all subsequent retention and vault decisions are based on refreshed rules.

While these utilities may operate with new, refreshed rules, the CONTROL-T Real-Time environment continues to use the older, original rules in memory.

To avoid confusion over which rule definitions are being accessed, the user can implement the Rule Refresh option at any time. Because of this situation, a rule refresh step is usually performed as the first step of the New Day Procedure.

3. CONTROL-T's Rule Scheduling facility (enabled by specifying Scheduling parameters while defining rules) allows you to define when rules should be activated based on calendars and scheduling criteria. For example, this allows you to specify a different retention period for the same dataset when that dataset is created on weekdays and weekends.

Retention Management

Utility CTTRTM scans the Media Database and determines which volumes/datasets have expired. These volumes are marked as scratch volumes in the CONTROL-T Media Database.

This utility can produce various reports concerning the current status of scratch volumes in the Media Database.

Vault Management

Utility CTTVTM determines which volumes should be moved and the locations to which they should be moved. This utility can also produce a distribution report.

Stacking Statistics File

Utility CTTSTK updates the Stacking file with new statistics collected on each dataset. This utility uses data from the Trace file.

Section 7: Customizing and Administering CONTROL-T

- Media Database Backup** This backup consists of the following steps:
1. Writing a Started Media Database Backup record to the Trace file.
 2. Backing up the Media Database data file and the Trace file using any standard backup utility. In the supplied job, utility IEBGENER is used for the backup.
 3. Writing an Ended Media Database Backup record to the Trace file.
- This backup can be used if a Media Database restore is required. See "CONTROL-T Database Backup And Recovery" later in this section.
- Check Media Database Integrity** Utility CTTIDB is run to check the Media Database for integrity errors. See utility CTTIDB in Section 9 for more information.

Repository Structure

The CONTROL-T Repository is composed of the following files:

- Media Database
- Trace file
- Stacking Statistics file

Media Database Structure

The CONTROL-T Media Database is composed of the following primary physical files:

- Data file Contains all required data about volumes and datasets in the Media Database.
- Index file Provides keyed access to data in the Data file.

Note Another file, the Trace file, is used to track updates and facilitate recovery of the Media Database. (See "Structure of the Trace File" later in this section for more information.)

The Data and Index files are allocated via the following DD statements:

- Data file DD statement DAMDB
- Index file DD statement DAMDI

Note Although these files are sequential and may be read using standard MVS QSAM services, the structure or format of these files may change in future releases. Therefore, it is highly recommended that the Media Database be accessed only via the standard macros described in "CONTROL-T Application Programming Interface" later in this section.

Section 7: Customizing and Administering CONTROL-T

Data File Contents

The Data file contains all the data about any volume or dataset. In CONTROL-T, unlike other tape management systems, volume information and dataset information are stored in the same file, but with different record types and formats.

The two most common record types in the CONTROL-T Data file are:

Volume record Contains information regarding a volume.

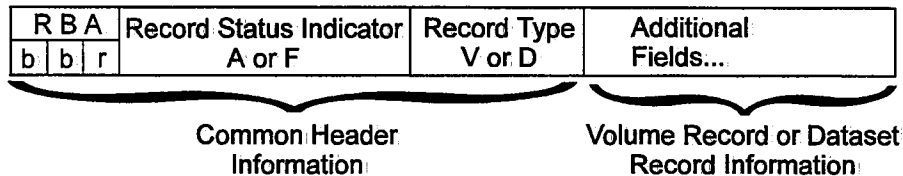
Dataset record Contains information regarding a dataset.

Note *Additional record types and formats exist in the database but are not described in this section because they are not relevant for most users.*

Volume and dataset records are mapped via Assembler macros CTTDVL and CTTDDS, respectively.

All record types in the Data file have a common header at the beginning of the record. This header contains the following information about the record:

Record in the Data File



RBA Specifies the internal record location identifier (explained later). Each record in the Data file is assigned an RBA that indicates the location of the record in the Data file. Access to a specific Data file record can be performed via its RBA if known (or via the index, as explained later in this section). Access of a data record via its RBA is known as "non-keyed access."

Record Status Indicator Indicates whether the record is active or free.

Record Type Indicates whether the record is a volume or dataset record.

Note *The term RBA as used in this section should not be confused with the term Relative Byte Address as referred to in IBM literature. The format of the CONTROL-T RBA is bbr. The first two bytes identify the block number in the Data file and the last byte identifies the record number in that block.*

Fields in the Data File

Some of the more important fields in the volume and dataset records in the Data file are listed below. Each field is assigned an internal name (a symbol in a DSEC which describes the record), and an external name (a logical name used by the user in INCLUDE/EXCLUDE statements). A comprehensive list of these fields can be found in "Logical Field Names for the CONTROL-T Repository" in Appendix D and in the IOA MAC library.

Commonly Referenced Volume Record Fields

Internal Name	External Name	Description
DVLRBA	DVLRBA	RBA of the record.
DVLRID	DVLRID	Record status (A-Active, F-Free).
DVLRTYPE	DVLRTYPE	Record type (V-Volume record).
DVLSTAT DVLSTA2	VOLSTAT VOLIND	Various status and flag bytes.
DVLFLG1	VOLFLAGS	Various status and flag bytes.
DVLVOLSR	VOLSER	Volume serial number.
DVLLTYPE	LBLTYP	Label type of the volume.
DVLACTD#	ACTIVEDS	Number of active (non-expired) datasets on the volume.
DVLLBLNM	LBLNUM	Label number of the last active dataset on the volume.
DVLVSEQ	VOLSEQ	Volume sequence number for a volume that is part of a multi-volume chain.
DVLNEXT	NEXTVOL	Next volume (of the multi-volume chain).
DVLPREV	PREVVOL	Previous volume (of the multi-volume chain).
DVLFIRST	FIRSTVOL	First volume (of the multi-volume chain).

Section 7: Customizing and Administering CONTROL-T

Commonly Referenced Record Fields

Internal Name	External Name	Description
DDSRBA	DDSRBA	RBA of the record.
DDSRID	DDSRID	Record status (A-Active, F-Free).
DDSRTYPE	DDSRTYPE	Record type (D-Dataset record).
DDSSTAT	DSSTAT	Various status bytes
DDSFLG1	DSFLAGS	Various flag bytes.
DDSDSN	DSNAME	Dataset name.
DDSLBLNM	DSLABEL	Label number (dataset sequence number on the tape).
DDSVOLSR	DSVOLSER	Volser of the dataset (or the first volser of a multi-volume dataset).
DDSVOLS#	VOLSNUM	Number of volumes in the dataset (if the file spans multiple volumes).

Index File Contents

The Index file makes keyed access of records in the Data file possible. The Index file contains records which, in turn, contain index elements. Each index element contains the following information:

Index Element

Key Type	Key	Additional Information	R B A		
			b	b	r

Key Type D,V, or L (see "Key Types" below).

Key Information which used to identify a record in the data file). Each index element contains one key only.

RBA Internal record location identifier of the relevant record in the Data file.

Additional info Bytes used for repository handling mechanisms.

Key Types

Each index element contains a particular type of key which is used for accessing records in the data file.

Full and partial keys of each type, are described below:

V Used to index volume records in the Data file. One index element containing one V-type key is created for each volume record in the Data file. The full V-type key for each volume record is the full volume serial number. This is a unique key.

Section 7: Customizing and Administering CONTROL-T

- D** Index to dataset records in the Data file. One index element containing one D-type key is created for each dataset record in the Data file. The full D-type key for each dataset record is the full dataset name. Because duplicate dataset names may exist in the media library, this is a non-unique key (i.e., more than one index element may exist with the same D-type key).
- L** Index to dataset records in the Data file. An element with an L-type key is created for each volume on which a specific dataset resides. For example, if a dataset records indicates that the dataset spans three volumes, three L-type keys (one for each volume) are created to reference that dataset record. The full L-type key is comprised of the full volume serial number and the full file sequence number (label) of the dataset on the volume. This is a unique key.

Depending on the function being performed when accessing the Index file (described later in this section), specification of the full key or a partial key may be specified.

A partial key is a key which is shorter than the full length of the key in the index file. The most common usage of a partial key is the specification of a prefix (e.g., dataset name prefix instead of a full dataset name). Partial keys can also be used when all necessary information is not available.

All index elements have the same fixed length, which is determined by the index elements with D-type keys because the D-type key is the longest full key (i.e., the dataset name).

It should be noted that index element length exceeds the length of the longest key because index elements contain additional bytes which are used by the CONTROL-T Repository handling mechanisms.

Index elements with V-type keys are mapped via assembler macro CTTDVX. Index elements with D-type keys are mapped via assembler macro CTTDDX. Index elements with L-type keys are mapped via assembler macro CTTDLX.

Volume/Dataset Relations

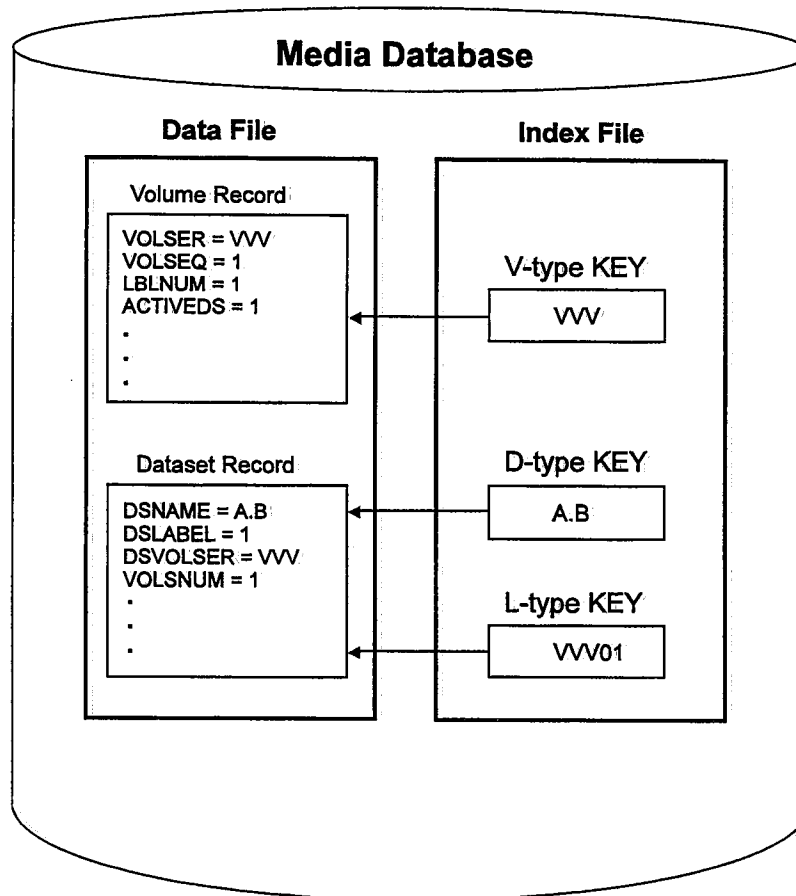
The Media Database contains a record for each volume and for each dataset (in the Data file). Fields in these records, and indexes of various types, are used to describe the relationships between a volume and its respective datasets. These fields and indexes also describe the relationship between volumes in a multi-volume chain.

The examples below show the relationship between the relevant fields in the volume records, dataset records, and the index elements that reference these records.

Section 7: Customizing and Administering CONTROL-T

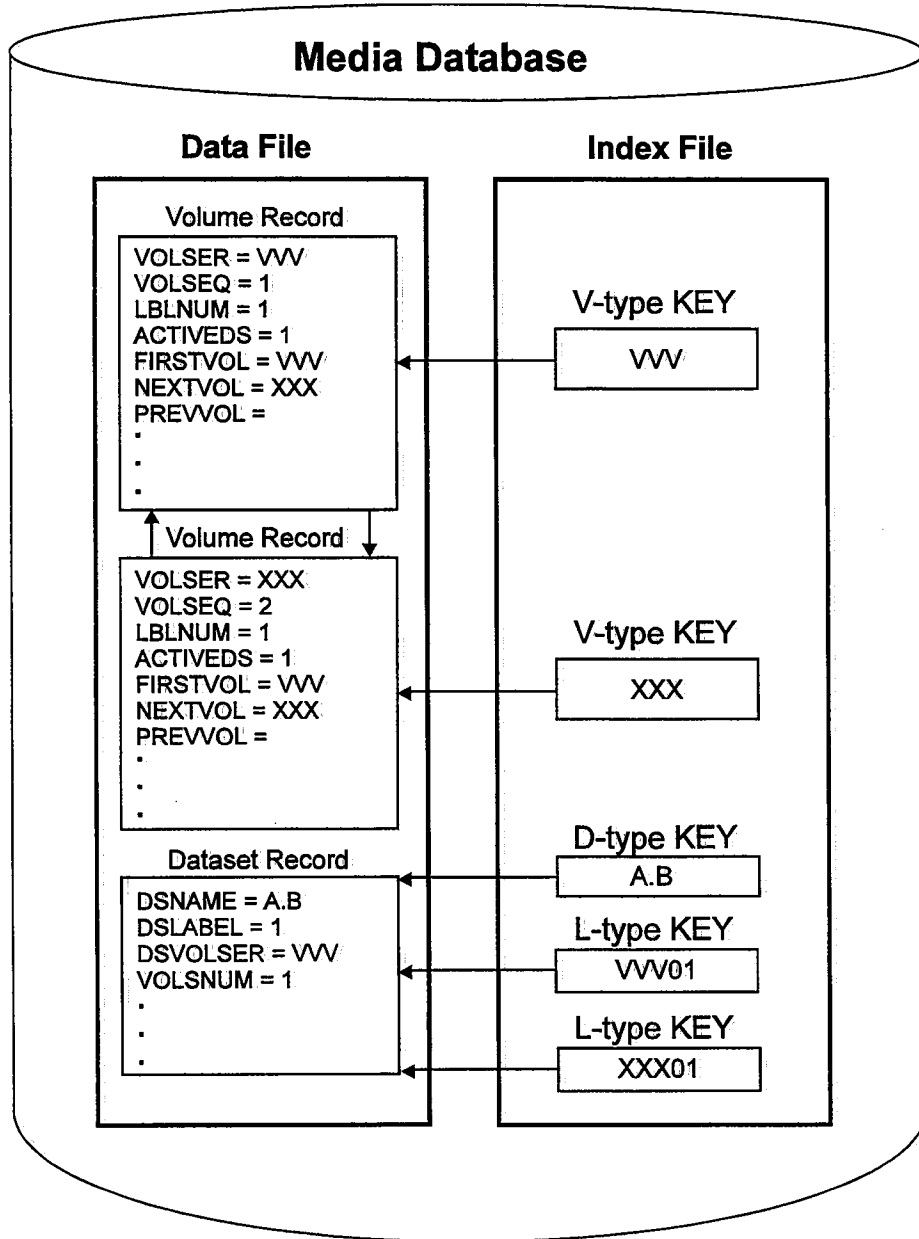
Example 1

This example shows the record and index configuration for a single volume (VVV) containing only one dataset (called A.B).



Example 2

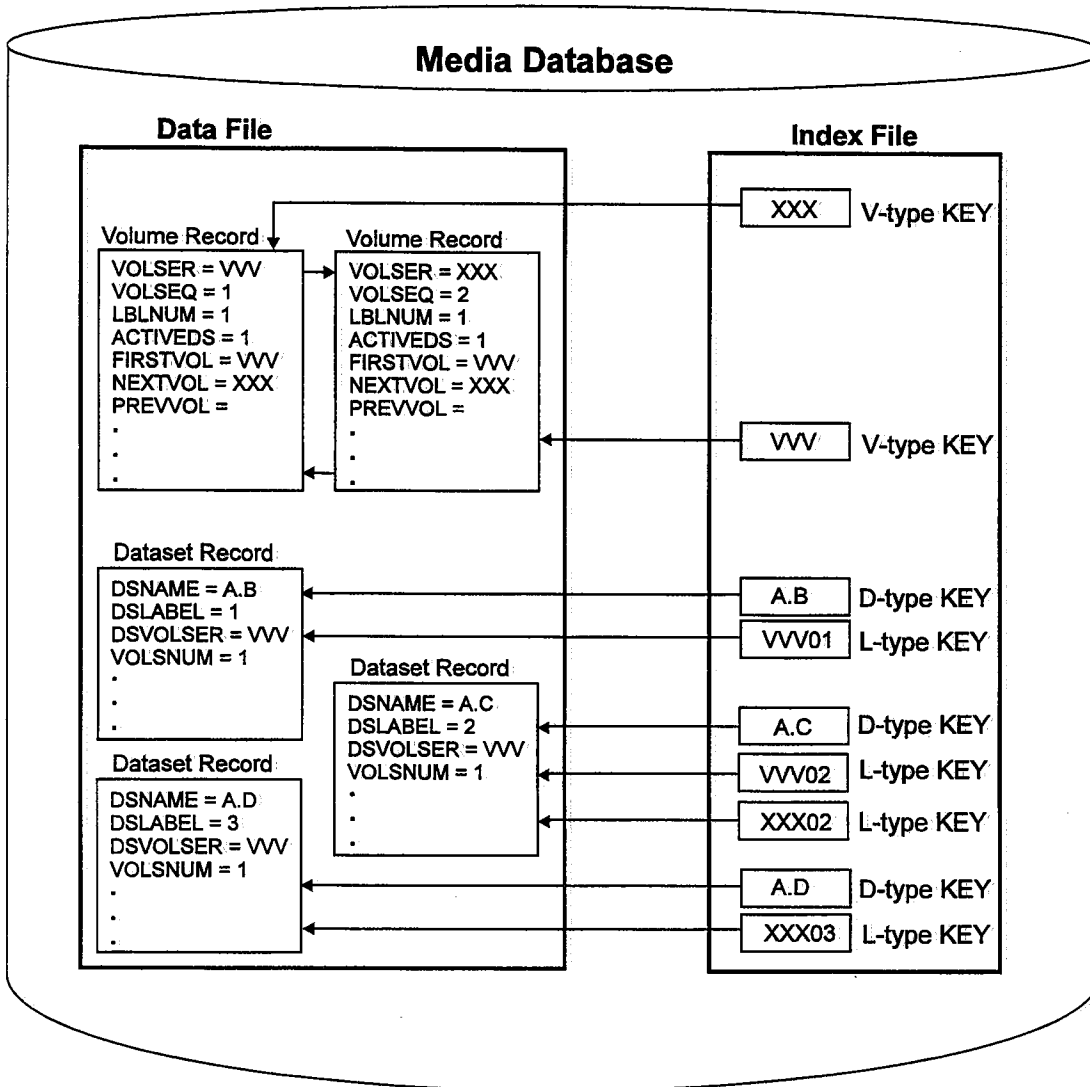
This example shows the record and index configuration for a single dataset (A.B) that spans two volumes (VVV and XXX).



Section 7: Customizing and Administering CONTROL-T

Example 3

This example shows the record and index configuration for a group of datasets (A.B, A.C, and A.D) contained in a multi-volume chain (volumes VVV and XXX). Dataset A.B resides on volume VVV. Dataset A.C starts on volume VVV and continues on volume XXX. Dataset A.D resides on volume XXX.



Structure of the Trace File

The Trace file is used for the following primary purposes:

- To track updates of the Media Database. This information is used to facilitate recovery of the Media Database when necessary.
- To serve as a queue for requests to be performed by the IOA Functional monitor.

The Trace file is automatically maintained by CONTROL-T mechanisms that handle the Media Database and the requests to the IOA Functional monitor.

The length of logical records in the Trace file may vary, depending on the type of information described in the record.

Logical records in the Trace file are composed of a header (mapped by macro CTTARC) followed by the record data (mapped by the macro relevant for the type of traced action).

The Trace file is processed as a cyclic file. Records are written on the oldest logical record. When the physical end-of-file is reached, the next record is written on the first physical record.

Fields in the Trace File

Some of the more important fields in the Trace file are listed below. A comprehensive list of these fields can be found in "Logical Field Names for the CONTROL-T Repository" in Appendix D, and in the IOA MAC Library.

ARCLL	Record Length.
ARCTYPE	Record type.
ARCIDENT	Name of the component that issued this record.
ARCSTAMP	Record time stamp (i.e., when the record was created).
ARCDATA	Actual information that is recorded.

Note The Trace file can be copied to a sequential variable-blocked file, enabling the output to be formatted in a number of different ways. (See the description of utility CTTACP in the utility section of this manual for more information.)

Structure of the Stacking Statistics File

The Stacking Statistics file contains statistical information about datasets processed by CONTROL-T. This information is gathered from various locations (e.g., the Media Database, or the Trace file) by utility CTTSTK. The information in this file is used by the Dynamic Dataset Stacking facility in the real-time environment.

Section 7: Customizing and Administering CONTROL-T

The Stacking Statistics file is composed of the following physical files:

- Data file Contains statistical information about datasets that were processed by CONTROL-T.
- Index file Provides keyed access to data in the Data file.

Records in the Data and Index files are mapped by Assembler macros CTTSTK and CTTSTX, respectively.

Data File Contents

The Data file contains statistical information about datasets that have been processed by CONTROL-T. A dataset is identified by its name and the name of the job which created the dataset.

All record types in the Data file have a common header at the beginning of the record. This header contains the following information about the record:

RBA	Specifies the internal record location identifier (explained later).
Record Status Indicator	Indicates whether the record is active or free.
Record Type	Indicates whether the record is a volume or dataset record.

Each record in the Data file is assigned an RBA that indicates the location of the record in the Data file.

Access to a specific data file record can be performed via its RBA if known, or via the index. Access of a data record via its RBA is known as "non-keyed access."

Note *The term RBA as used in this section should not be confused with the term Relative Byte Address referred to in IBM literature. The format of the CONTROL-T RBA is bbr. The first two bytes identify the block number in the Data file and the last byte identifies the record number in that block.*

Fields in the Data File

Some of the more important fields in the Data file are listed below. A comprehensive list of these fields, can be found in "Logical Field Names for the CONTROL-T Repository" in Appendix D, and in the IOA MAC Library.

STKKEY	Record key (the dataset name and job name).
STKOB	Accumulated number of observations.
STKPRED	Predicted length of the dataset in feet.
STKSTAMP	Record time stamp.

Index File Contents

The Index file contains index records. These index records enable keyed access of records in the Data file.

Repository Maintenance

Verifying Data Integrity of Media Databases

In the CONTROL-T Media Database, data of different types are stored in separate records and are linked via the following mechanisms:

- Data fields in data records, which reference other data records (e.g., field DDSVOLSR in a dataset record indicates the record of the volume containing the beginning of the dataset).
- Index records which link data records (e.g., the L-index record).
- Counters within data records (e.g., field DVLACTD# in a volume record indicates the number of datasets in the volume).

Media Database integrity is the degree to which the links between data/index records are free of errors. A high level of integrity is necessary to ensure smooth processing by CONTROL-T. Use utility CTTIDB to check Media Database integrity on a daily basis (e.g., as a part of the CONTROL-T New Day Procedure).

Causes of Media Database Integrity Errors

As a rule, normal tape processing by CONTROL-T should not cause integrity errors. However, such errors can be caused by external events that interrupt CONTROL-T's processing, resulting, for example, in partially updated records. Some common causes of integrity errors are:

- System-crash.
- Cancellation of a job that accesses tapes in the middle of its processing.
- Interruption of a CONTROL-T utility that updates records in the Media Database.

Examples of Errors Due to Interruption of CONTROL-T Processing

- A volume record points to the next record in a multi-volume chain (via the NEXTVOL field), but that volume does not point back (field PREVVOL).
- A dataset spans a number of volumes different from what is indicated in the record for that dataset.
- An L-index points to a scratch volume.

Integrity errors should not occur often. If utility CTTIDB detects an unusually large number of integrity errors, it is recommended that you immediately investigate the cause of these errors and fix the problem.

Section 7: Customizing and Administering CONTROL-T

Recovering From Integrity Errors

Some integrity errors are automatically resolved by CONTROL-T:

- L-indexes that point to a scratch volume are removed as soon as the scratch volume is mounted in response to a scratch tape request.
- Some utilities (i.e., CTRTM and CTTVTM) have restart capabilities which allow them to continue from the point at which the utility was interrupted.

Integrity errors that are not automatically fixed by CONTROL-T should be handled by first studying the nature of the error, and then determining what Media Database updates should be performed to correct the error.

Analyzing an Integrity Error

Check the output of utility CTTIDB every day. If an integrity error is detected, you should be able to determine the type of error, and which data/index records are involved, from the output of the utility. Below is a list of the most common integrity errors detected via utility CTTIDB:

- Information in a data record and corresponding information in an index record do not match.
- A counter field within a data record has an incorrect value.
- A group of volumes is chained incorrectly or some volumes in the chain have conflicting attributes (e.g., vaulting criteria).
- Data fields linking data records point incorrectly.
- A scratch volume points to, or is pointed to, by an active volume record.

The method used to correct the error should be chosen, based on the information found in the output of utility CTTIDB and, optionally, in the record data viewed via the Inquire/Update Media Database Screen (screen TI).

Correcting Integrity Errors

Integrity errors are corrected by updating the data records and/or index records in error. Errors involving data records should be corrected, one at a time, via utility CTTMUP. After fixing each error it is recommended to rerun utility CTTIDB, because fixing one error often eliminates other errors. This can be done automatically by specifying CHECK=YES in the TYPERUN statement of utility CTTMUP. The output of utility CTTIDB after it is rerun will accurately reflect the integrity errors that remain.

Utility CTTMUP can also be used to fix most index related problems by rebuilding the indexes for a volume, a dataset, or a group of volumes and datasets.

When a large number of index errors are found in the Media Database, consider fixing these errors by rebuilding the index component via utility CTTBIX.

Note *If you choose to rebuild the index of the Media Database, it is necessary to first halt all tape processing.*

For a detailed description of utilities CTTIDB, CTTMUP and CTTBIX see Section 9 of this guide.

Manual Update of the Media Database

Manual update of the Media Database is accomplished via utility CTTMUP. This utility can be used to add new volume/dataset records to the Media Database, or to fix integrity errors detected via utility CTTIDB.

Before using utility CTTMUP, it is important to familiarize yourself with the structure of the Media Database (described earlier in this section). Attempting to fix integrity errors without properly understanding the Media Database structure can lead to even more integrity errors.

When updating a volume record, utility CTTMUP does not always automatically update the corresponding dataset records, and vice versa. When updating data fields that are used to link records, be sure to update all relevant records so that no new integrity errors are created. For example, when you add a dataset to a volume, verify that the new number of datasets is recorded in the volume record.

When updating groups of volumes (multi-volume chains), use the group statements (prefixed "GRP") to implement the desired change in the records of all the volumes in the group.

After performing the necessary updates, it is highly recommended that you run utility CTTIDB to verify the results. This can be accomplished by running the utility manually, or by including CHECK=YES in the TYPERUN statement of utility CTTMUP.

See the description of utility CTTMUP in Section 9 of this guide for more information on manual update of the Media Database.

Enlarging the Media Database

At times, it may become necessary to increase the size of the Media Database (MDB). This situation arises if one of the Media Database components (index or data) is almost at maximum capacity, or if extensive additions to the Media Database need to be made in the near future.

To enlarge the Media Database:

1. Calculate the space needed for your Media Database.

To calculate the amount of space needed, see "Media Database Calculations" in the CONTROL-T Installation Considerations and the appendix called "IOA Dataset: Disk Storage and Security Requirement Table" in the IOA Installation Guide. The Media Database can be enlarged by increasing the number of blocks, by enlarging the block size, or via a combination of the two methods. (By following the instructions listed here, any of these extension methods are possible.)

2. Shut down CONTROL-T via the following operator command:

```
S: CTTINIT, PARM='MODE=TERM'
```

3. Back up the active Media Database by running CONTROL-T utility CTTDBDUL, IBM's utility IEBGENER, or any other similar tool.
4. Create and format new Media Database Data Components by running the same job which is run at installation time to allocate Media Database components (see step "CONTROL-T Media Database" in the CONTROL-T Installation Procedure in the IOA Installation Guide), but with a larger allocation size.

Section 7: Customizing and Administering CONTROL-T

Note Before running this job, verify that at least one of the procedure parameters which compose the Media Database component names (*DBPREFT*, and *DBFILET*) is modified. By doing so, the active Media Database will not be affected.

5. Unload the Media Database data component via CONTROL-T utility CTTDBDUL.
6. Rename the production Media Database data component and index component.
7. Rename the newly-created components.

Rename the newly-created Media Database data component to the name of the production data component. Rename the newly-created Media Database index component to the name of the production index component.

8. Reload the Media Database data component into the new (and larger) data component via CONTROL-T utility CTTDBDL.
9. Run utility CTTBIX.
10. Restart CONTROL-T via the following operator command:

```
S CTTINIT, PARM='MODE=INIT'
```

Enlarging the Trace File

At times, it may be necessary to increase the size of the CONTROL-T Trace file. This situation arises if the Trace file often reaches maximum capacity, or if an increase in tape activity at your site is imminent.

To enlarge the Trace file:

1. Calculate the space needed for the Trace file.

To calculate the space needed, review the appendix called "IOA Dataset Storage Requirements Table" in the IOA Installation Guide. At this stage, you should also determine the correct value of installation parameter TRCSIZE in member CTTPARM in the IOA INSTCTT library. The Trace file can be enlarged either by increasing the number of blocks, by enlarging the block size, or by a combination of these methods. By following the instructions listed here, any of these extension methods are possible. (For more information on the TRCSIZE parameter see the IOA Installation Guide.)

2. Shut down CONTROL-T via the following operator command:

```
S CTTINIT, PARM='MODE=TERM'
```
3. Back up the active Trace file via CONTROL-T utility CTTACP, IBM utility IEBGENER, or any other similar tool.
4. Create and format a new Trace file by running the same job used at time of installation to allocate the Trace file, but with a larger allocation size.

Note Before running this job, verify that at least one of the procedure parameters which compose the Trace file name (*DBPREFT* and *DBFILET*) is modified. By doing so, the active Trace file is not affected.

Section 7: Customizing and Administering CONTROL-T

5. Update installation parameter TRCSIZE in member CTTPARM of the IOA INSTCTT library according to the size of the new Trace file.
6. Run job CTTPARMJ. Running this job (located in the INSTCTT library) makes the new value of TRCSIZE available.
7. Rename the production Trace file.
8. Rename the newly-created Trace file to the former name of the production Trace file.
9. Copy the old Trace file to the new (and larger) Trace file via CONTROL-T utility CTTACP. An example of CTTACP activation is provided below:

```
//name JOB ,IOA,CLASS=A,MSGCLASS=X
//*
//* COPY ONE CONTROL-T TRC DATASET TO ANOTHER TRC
//*
//TT2ACP EXEC CTTACP,
//      TRCIN='CTT.V122.TRC.OLD',
//      TRCOUT='CTT.V122.TRC'
//SYSIN DD *
      COPY FROM=TRACE,TO=TRACE
//
```

10. Restart CONTROL-T via the following operator command:

```
S CTTINIT, PARM='MODE=INIT'
```

Repository Backup and Recovery

General

CONTROL-T has a comprehensive set of utilities which ensure data integrity and provide full recovery in situations caused by system/disk crashes or user error. The following pages describe the procedures that the IOA administrator should perform in various situations.

See Section 9 in this Guide for a detailed description of the utilities mentioned below.

Backup and Recovery

CONTROL-T provides several utilities to facilitate the backup and recovery process. Backups can be performed at any time using batch utilities. At most sites, backups are initiated by the New Day procedure.

Normally, the Media Database is backed up first. It updates the Trace file with the date and time that backup started and completed. The Trace file is backed up as soon as the Media Database backup completes successfully. Therefore, each Media Database backup has a corresponding Trace file backup. All removable media processing may continue during this process.

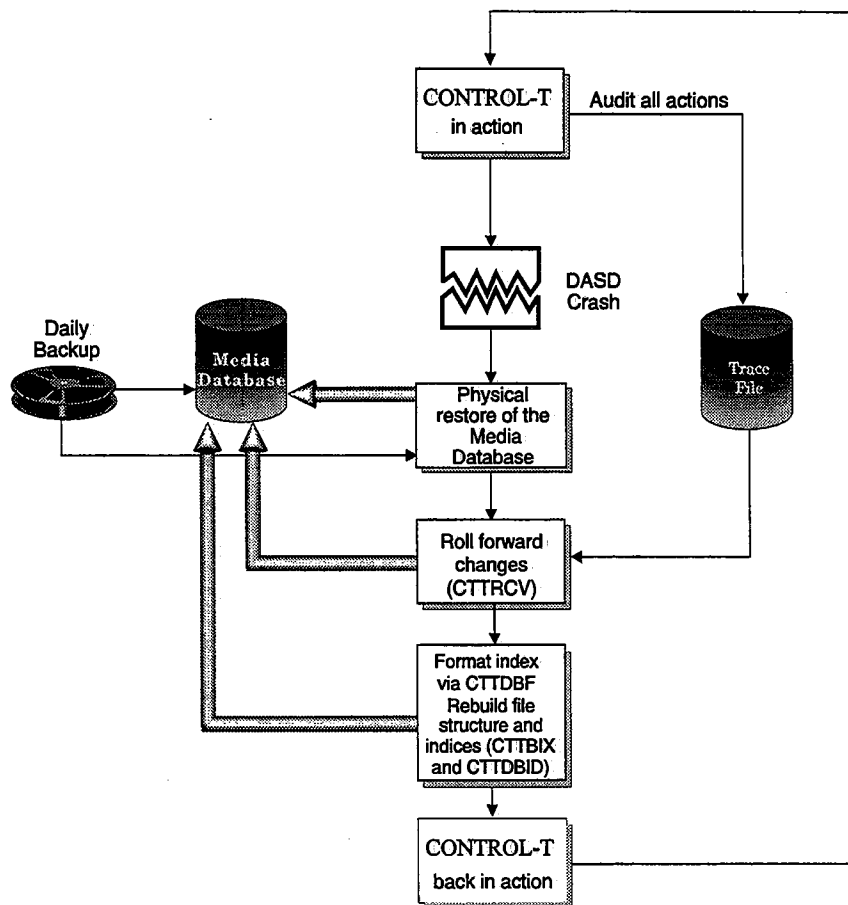
Section 7: Customizing and Administering CONTROL-T

If the Media Database or some portion of it must be recovered:

- Restore the latest Media Database backup using the standard site backup utility. Run utility CTTRCV to apply all updates since that backup from the active Trace file (a Roll-Forward process).
- Based on the current Media Database, utility CTTRCV can withdraw all updates until a certain point (a Roll-Back process).
- Utility CTTRCV can also remove all updates in a specified time range or generated by a specific job.
- Once the restore is complete, all indexes are rebuilt by utility CTTBIX and the free space chain is rebuilt by utility CTTDBID.

Disaster Recovery

CONTROL-T recovery is performed using the Trace file from the last Media Database backup via several steps. First, a backup/restore utility (e.g., IEBGENER) is run to restore the Media Database. Then, utility CTTRCV is run to roll forward all Trace file Updates since the last backup. Utilities CTTBIX/CTTDBID are then run to rebuild the file structure and indexes.



Note Before rebuilding the index file, the index format utility (CTTDBF) should be run.

Selective Recovery

This method recovers only a selection of changes which satisfy specified selection criteria. For example, to recover the actions of a bad maintenance job (e.g., a retention management job to which incorrect parameters were specified) all decisions/actions made by this job should be reversed. Only selected portions of the Media Database are recovered. This is a logical, backward recovery in two steps via utilities CTTRCV and CTTBIX/CTTDBID.

Note After completing the rollback updates, the index formatting utility (CTTDBF) should be run, followed by utility CTTBIX which is used to rebuild the index file.

General Considerations

Refer to the following chart to determine when to use each recovery method.

Comparison of Physical vs. Logical Recovery Methods

	Physical Recovery	Logical Recovery
When to use:	After a physical disaster	After user errors
What will be recovered:	Entire Media Database	Selected portions of the Media Database
How the data is recovered:	Via restore and roll-forward	Via roll-back

Robotic Library Support

CONTROL-T supports the various Robotic Tape Library systems available on the market. The installation procedure is described below.

For a general description of CONTROL-T support for Robotic Tape Library systems, refer to "Robotic Tape Library Support" in Section 4 of the CONTROL-T User Manual.

Robotic Tape Libraries Using System Managed Tape Functions

The CONTROL-T interface to robotic tape libraries using System Managed Tape Functions (e.g., HDS 7450/7430, and COMPAREX 6394/6395 robots) is activated in the same way that the CONTROL-T interface to IBM SMS Managed tape libraries is activated. Therefore, to activate the CONTROL-T interface to robotic tape libraries that use System Managed Tape functions, perform the steps in "IBM Tape Library Dataservers Support" (below), and then continue with the steps in "IBM SMS Managed Tape Library Support" (which follows).

If you have already activated the CONTROL-T interface to IBM Tape Library Dataservers with SMS managed tape library support, there is no need to repeat this process.

IBM Tape Library Dataserver Support

To activate the IBM robot interface of CONTROL-T, perform the following steps:

- A. Specify IBMT as one of the values for parameter RBTTYPE (via CONTROL-T step 2.10 in ICE).
- B. Save the existing procedure CTTRTM under the name CTTRTM2. Rename procedure CTTRTMB in the procedure library to CTTRTM.

The new CTTRTM procedure contains a step which activates the interface with the IBM robot.

- C. Save the existing procedure CTTVTM under the name CTTVTM2. Rename procedure CTTVTMB in the Procedure Library to CTTVTM.

The new CTTVTM procedure contains a step which activates the interface with the IBM 3495 robot.

CONTROL-T can interface with either IBM SMS Managed Tape Library Support, or IBM Basic Tape Library Support (BTLS). Continue with the procedure appropriate for your site.

IBM SMS Managed Tape Library Support

- A. Edit member CTT3495 in the IOA SAMPLE library. Verify that N is indicated for variable BTLS in the following statement:

```
BTLS      EQU      C'N'          <=== BTLS SUPPORT (Y/N)
```

- B. Edit member ASM3495 in the IOA SAMPLE library and submit the job. This member compiles and links routine CTT3495 into the IOA LOAD library. CTT3495 contains the code necessary for interfacing with the IBM 3495 software.

To manage this module using SMP/E, use member IOAEXSMP in the IOA JCL library instead of member ASM3495 (described above).

User Exits

The following sample user exits (located in the IOA SAMPLE library) can optionally be installed as part of the CONTROL-T to IBM robot interface:

CBRUXCUA Change Use Attribute

Allows a change in the Use attribute of a volume only if this change has already been recorded in the Media Database. Implementation of this exit helps maintain Media Database integrity.

The compile and link job for this user exit is in member ASMUXCUA in the IOA SAMPLE library.

CBRUXEJC Tape Eject

Turns off the IN THE ROBOT status bit in the volume record of the ejected volume in the Media Database.

The compile and link job for this user exit is in member ASMUXEJC in the IOA SAMPLE library.

Section 7: Customizing and Administering CONTROL-T

CBRUXENT Tape Entry

This exit does the following:

- Turns on the IN THE ROBOT status bit in the volume record of the inserted volume.
- Updates the internal location (library in the robot) in the volume record of the relevant volume in the Media Database.
- Sets the INITIAL USE attribute of the volume in the robot to the status in the relevant volume record in the Media Database.

The compile and link job for this user exit is in member ASMUXENT in the IOA SAMPLE library.

CBRUXVNL Volume Not In Library

Issues a message to the operator with the volume location (as recorded in the Media Database). On receipt of this message the operator should insert the volume into the robot and reply R (retry) to continue.

The compile and link job for this user exit is in member ASMUXVNL in the IOA SAMPLE library.

IBM Basic Tape Library Support (BTLS)

1. Edit member CTT3495 in the IOA SAMPLE library and make the following changes:

- Indicate that BTLS should be supported via the following statement:

```
BTLS    EQU    C'Y'        <=== BTLS SUPPORT (Y/N)
```

Assign one of the BTLS library units to the BTLSUNIT parameter in the line which follows this line.

- If you are using 4-digit device numbers (MVS ESA Release 5 and above) change the value specified for parameter BTLSDIGS from 3 to 4 via the following statement:

```
BTLSDIGS EQU    4          <=== NUM OF DIGITS IN UNIT ADDRESS (3/4)
```

Example

If 123 is the address of one of your robot units, and you are using a release prior to MVS ESA 5, the parameters section should include the following statements:

```
BTLS    EQU    C'Y'        <=== BTLS SUPPORT (Y/N)
BTLSUNIT EQU    C'123'     <=== DEFINE ROBOT UNIT ADDRESS
BTLSDIGS EQU    3         <=== NUM OF DIGITS IN UNIT ADDRESS (3/4)
```

2. Edit member ASM3495 in the IOA SAMPLE library and submit the job. This member compiles and links member CTT3495 into the IOA LOAD library. Member CTT3495 contains the code necessary for interfacing with the IBM 3495 software.

To manage this module using SMP/E, use member IOAEXSMP in the IOA JCL library instead of member ASM3495 (described above).

Section 7: Customizing and Administering CONTROL-T

3. Add DD statements BTLIN and BTLISOUT to step CTTRBM in the procedures for utilities CTTVTM and CTTRTM.
 - DD statement BTLIN must point to a temporary file.
 - DD statement BTLISOUT must point to a SYSOUT file.

Example

If SYSDA is used as a temporary DASD unit, and BTLIS messages should be printed to the default MSGCLASS SYSOUT class, the above mentioned DD statements should be as follows:

```
//BTLIN DD UNIT=SYSDA,SPACE=(TRK,1)
//BTLISOUT DD SYSOUT=*
```

4. Add DD statements BTLIN and BTLISOUT to your Online environment (i.e., to the user's logon procedure) to support options EXPIRE and VAULT in the Inquire/Update Media Database screen (Screen TI).

StorageTek Silo Support

To activate the StorageTek Silo interface of CONTROL-T, the following steps should be performed:

Basic Support

1. Specify STK as one of the values for parameter RBTTYPE in member CTTPARM.
2. Save this member.
3. Edit member CTTPARMJ in the IOA INSTCTT library and submit the job. This job creates load modules CTTPARM and IOAPARM in the IOA LOAD library.

CONTROL-T can interface with either StorageTek's Host Software Component (HSC) or the Client Server Component (CSC). Follow the procedure appropriate for your site.

StorageTek Host Software Component (HSC) Support

1. Edit member CTTRSTK in the IOA SAMPLE library. Verify that 'HSC' is indicated in the following statement:

```
STKTYPE DC CL4 'HSC'
```

2. Edit member ASMRSTK in the IOA SAMPLE library and submit the job. This member compiles and links member CTTRSTK into the IOA LOAD library. CTTRSTK contains the specific code for interfacing with the StorageTek Host Software Component (HSC).

If you wish to manage this module using SMP/E, refer to member IOAEXSMP in the IOA JCL Library.

3. Save the old procedure CTTRTM under the name CTTRTM2. Rename procedure CTTRTM in the procedure library to CTTRTM. The new CTTRTM contains an additional step which interfaces with the StorageTek Silo.

Section 7: Customizing and Administering CONTROL-T

4. Save the old procedure CTTVTM under the name CTTVTM2. Rename procedure CTTVTMB in the procedure library to CTTVTM. The new CTTVTM contains an additional step which interfaces with the StorageTek Silo.
5. When using CONTROL-T, the HSC is immediately notified about expired tapes, and their status is updated in the HSC's Control Dataset(s). CONTROL-T uses the HSC API to do this. If the HSC load library (SLS.SLSLINK) is not in the MVS linklist, it should be added to DD statement STEPLIB of all environments that may expire a volume. This includes:
 - The CTTRBM step in utilities CTTRTM and CTTVTM.
 - The Online Environment. Depending on whether you operate CONTROL-T Online facility under TSO or under the IOA Online monitor, the HSC load library should be added to the TSO logon procedure STEPLIB and/or to the IOA Online monitor STEPLIB.
 - If you have implemented EDM support (e.g., using exit ARCTVEXT for HSM or exit TMSCTLEX for DMS/OS), the HSC load library should be added to the HSM, or DMS/OS procedure STEPLIB.
6. User exit SLSUX06 (Volume Insert/Delete) can be optionally installed to turn status bit IN THE ROBOT on and off, and to update the internal location of the volume (the LSM address) in the relevant volume record in the Media Database.

The sample source for this user exit is located in member SLSUX06 in the IOA SAMPLE library.

The compile and link job for this user exit is in member ASMRUX6 in the IOA SAMPLE library.

If you choose to implement user exit SLSUX06, you must first initialize the IN THE ROBOT status bit and the LSM address. This is performed via program CTTSLS.

- The sample source for this program is located in member CTTSLS in the IOA SAMPLE library.
- The compile and link job for this program is in member ASMSLS in the IOA SAMPLE library.
- A sample job which executes this program is in member CTTSLSJ in the IOA SAMPLE library.

StorageTek Client System Component (CSC) Support

1. Edit member CTTRSTK in the IOA SAMPLE library. Replace 'HSC' with CSC in the following statement:

```
STKTYPE DC CL4 'HSC'
```

2. Edit member ASMRSTK in the IOA SAMPLE library and submit the job. This member compiles and links member CTTRSTK into the IOA LOAD library. CTTRSTK contains specific code for interfacing with the StorageTek Client System Component (CSC).

If you wish to manage this module via SMP/E, refer to member IOAEXSMP in the IOA JCL library.

Section 7: Customizing and Administering CONTROL-T

3. Save the old procedure CTTRTM under the name CTTRTM2. Rename procedure CTTRTMC in the procedure library to CTTRTM. The new CTTRTM contains an additional step which interfaces with the StorageTek Silo.
4. Allocate a sequential file as a fixed block with a record length of 80 bytes. Use the values specified for parameters %OLPREFT% and %OLVERT% to name this dataset as follows:

prefix.version.DASRCMD

where

prefix Value specified for parameter %OLPREFT% in member DEFPARMT.

version Value specified for parameter %OLVERT% in member DEFPARMT.

This sequential file should be large enough to hold the maximum number of volumes that might be expired by CTTRTM in one day (one record per volume).

Notes *StorageTek CSC is not able to perform automatic ejects. After running utility CTTVTM, all necessary ejects must be performed manually.*

No application Program Interface (API) is supplied with StorageTek CSC. For this reason CONTROL-T uses CSC utility SCUADMIN to notify CSC about expired tapes. As a result of this method of operation, note the following:

- *CSC is not notified when volumes are expired manually from the CONTROL-T Online environment or when volumes are expired via utility CTTMUP.*
- *When an External Data Manager (EDM) is set to scratch volumes immediately, CSC is not notified of the scratched tapes. It is therefore recommended that the EDM exit (e.g., ARCTVEXT for HSM) scratch volumes in deferred mode. When deferred mode is used, EDM volumes are given the status PENDING SCRATCH by the EDM exit, and are scratched by utility CTTRTM the next time it is run, at which time CSC is notified.*

CTTTPI Interface to StorageTek Robots

The initialization program of the StorageTek Host Software Component invokes a site-specific media initialization program. To indicate that you wish to use utility CTTTPI for volume initialization, follow the instructions for the cartridge initialization utility in StorageTek's reference manual with the following adaptation:

1. Specify PROG(ctttpiname) to StorageTek's utility

where:

ctttpiname is either CTTTPI or (if you renamed utility CTTTPI as described in the final adjustments to CONTROL-T installation) IEHINITT.

2. Pass the CTTTPI input control statements to StorageTek's Host Software Component via DD statement SLSINIT.
3. Specify ***** (six asterisks) for CTTTPI parameter SER.

Section 7: Customizing and Administering CONTROL-T

HSC/CSC Scratch Pool Support

If you are not using scratch pools under CONTROL-T, specify N for parameter DYNWTO in member CTT Parm so that the mount message will not be changed by CONTROL-T. (See "Performing Post-Installation Tasks" in the IOA Installation Guide for more information on how to change the value for this parameter.)

If you intend to use scratch pools under CONTROL-T, specify M or Y for parameter DYNWTO in member CTT Parm. CONTROL-T will modify the mount messages, and the scratch pool identification will be added to them.

To interpret the CONTROL-T mount message, HSC/CSC job processing user exit (SLSUX01) should be used, and the HSC/CSC scratch pool definitions must be synchronized with the pool definitions in CONTROL-T. HSC/CSC scratch pools are defined via the SCRPOOL HSC/CSC initialization statements.

To implement scratch pooling in your StorageTek Silo, the following procedure should be followed:

1. Edit member ASMRUX1 in the IOA SAMPLE Library and submit the job. This member compiles and links exit SCSUX01 into the HSC LOAD library (SLSLINK). SCSUX01 contains the specific code for interpreting CONTROL-T mount messages.

If you wish to manage this module using SMP/E, refer to member IOAEXSMP in the IOA JCL library.

2. Follow these guidelines to keep CONTROL-T pool definitions and HSC scratch pool definitions synchronized:

- Since HSC SCRPOOL statements are less flexible than CONTROL-T pool definitions, only CONTROL-T pools that can also be defined to HSC should be defined (e.g., do not use wildcard characters in pool definitions).
- Since HSC supports only 13 characters as a valid scratch pool name, use pool names of 13 characters or less in CONTROL-T.
- HSC does not enable dynamic (i.e., during operation) changes to SCRPOOL statements. Therefore, when dynamically changing CONTROL-T pools, care should be taken to not interfere with HSC processing.

ExLM Support

If you are not using Expert Library Manager (ExLM) software from StorageTek, ignore this topic.

The interface between CONTROL-T and ExLM is performed by utilizing the "CUSTOM Tape Management System Support" for release 1.2.0 of ExLM. To activate the interface you need release 1.2.0 of ExLM with PTF L1L001U or L1L002C (or any other PTF supplied by StorageTek technical support). If you do not have the above release or PTF level of ExLM, contact your StorageTek representative.

To access the CONTROL-T Media Database from ExLM, the following procedure should be followed:

- Define CONTROL-T as the TMS used in the ExLM control statements, as follows:

```
TMS CUSTOM NAME(CONTROLT) DDNAME(DAMDB) MODNAME(CTTXLM);
```

Section 7: Customizing and Administering CONTROL-T

- Add the IOA LOAD library to DD statement STEPLIB of the ExLM JCL. This is not required if you have added the IOA LOAD library to the MVS Linklist.
- Add a DD statement for the Media Database data component to the ExLM JCL, as follows:

```
//DAMDB DD DSN=ctt.v500.MDEB,DISP=SHR
```

Note *The name of the Media Database data component should be adjusted to the prefix and version of the CONTROL-T Repository used at your site.*

Samples for the ExLM JCL and ExLM control statements can be found in members EXLMJOB and EXLMSAMP in the IOA SAMPLE library.

MEMOREX TELEX Automatic Tape Library (ATL) Support

To activate the MEMOREX TELEX Automatic Tape Library (ATL) interface of CONTROL-T, perform the following steps:

1. Specify MEMOREX as one of the values for parameter RBTTYPE in member CTTPARM.
2. Save member CTTPARM.
3. Edit member CTTPARMJ in the IOA INSTCTT library and submit the job. This job creates load modules CTTPARM and IOAPARM in the IOA LOAD library. All steps of this job must end with a condition code of 0.
4. Edit member ASMMMRX in the IOA SAMPLE library and submit the job. This member compiles and links member CTTMMRX from the IOA SAMPLE library into the IOA LOAD library. Member CTTMMRX contains the code necessary for interfacing with the MEMOREX TELEX ATL software (LMS) during immediate SCRATCH/EJECT operations (in the Online environment). The robot database is synchronized with the Media Database via a batch procedure which is supplied by MEMOREX. (Refer to the MEMOREX TELEX documentation for more information.)

To manage this module using SMP/E, use routine IOAEXSMP in the IOA JCL library instead of member ASMMMRX (described above).

EMASS Automatic Tape Library (ATL) Support

To activate the EMASS Automatic Tape Library (ATL) interface of CONTROL-T, perform the following steps:

Note *The method used to activate EMASS tape library support can also be used for interfacing with GRAU and Bosch tape libraries.*

- A. Specify EMASS as one of the values for parameter RBTTYPE in member CTTPARM.
- B. Save member CTTPARM.
- C. Edit member CTTPARMJ in the IOA INSTCTT library and submit the job. This job creates load modules CTTPARM and IOAPARM in the IOA LOAD library.

Section 7: Customizing and Administering CONTROL-T

- D. Edit member ASMHACC in the IOA SAMPLE library and submit the job. This member compiles and links member CTTHACC from the IOA SAMPLE library into the IOA LOAD library. Member CTTHACC contains the code required for interfacing with the EMASS ATL software (HCC) during vault management operations and immediate SCRATCH/EJECT operations (in the On-Line environment). The robot database is synchronized with the Media Database via a batch procedure which is supplied by EMASS. (For more information, refer to the EMASS documentation.)

To manage this module using SMP/E, use routine IOAEXSMP in the IOA JCL library instead of member ASMHACC (described above).

- E. Save the old CTTVTM procedure under the name CTTVTM2. Rename procedure CTTVTMB in the procedure library to CTTVTM. The new CTTVTM contains an additional step which interfaces with the EMASS robot.

Depending on the ABBASEND interface (which is used by the CONTROL-T interface) requirements, you may need to add the following DD statement to the CTTRBM step:

```
HACCOUT DD SYSOUT=*
```

Cartridge Message Display Customization

The CONTROL-T interface for displaying messages on cartridge magnetic tape subsystems allows you to:

- Display the first seven characters of a pool name on the cartridge drive LCD display when a scratch tape is requested from a specific scratch pool.
- Stop automatic cartridge load when two consecutive tape mounts result in the rejection of the mounted tapes. To enable this feature, the Cartridge Stack Loader (CSL) must be set to SYSTEM mode.

Message display installation exit IGXMSGEX is provided as a module that you can use to customize messages to be displayed on an IBM 3480 tape drive. This exit is optional, and is invoked only if it is link-edited with the message display execution module (IGC00030).

Installing the Message Display Interface

Installation of the message display interface is performed via the application of an SMP/E USERMOD to your MVS SMP/E environment. This USERMOD introduces two CONTROL-T CSECTs (IGXMSGEX, and CTTMSGD) which are linked to module IGC00030 in the LPA library.

Note *If you already have another Message Display interface installed, make sure that it is removed before installing the CONTROL-T interface.*

The steps described below are applied to your MVS SYS1.LPALIB library. You can apply these changes to your MLPA library instead, by overriding the LPALIB DD statement (either through DDDEF, or through your MVS SMP/E JCL procedure).

Section 7: Customizing and Administering CONTROL-T

To install the CONTROL-T interface to the Message Display, perform the following steps:

1. Edit member IGXMSGEX in the IOA SAMPLE library. Member IGXMSGEX contains a sample message display installation exit which allows control to be passed to several modules for customization of MVS messages before they are displayed on an IBM 3480 tape drive. All modules to be passed control are listed under label MODLIST in this sample member. By default, this list contains only one module (CTTMSGD), which is the CONTROL-T message customization module.

You can let other modules process tape drive messages in parallel to the CONTROL-T interface (e.g., for parallel processing by CA-1) by adding an entry for each of these modules under label MODLIST in sample member IGXMSGEX. (See comments in this sample member for the valid format of module specifications.)

It is recommended that you save a backup copy of sample member IGXMSGEX before modifying it. All changes to this sample member must be saved before proceeding to the next step (Step 2).

2. Use SMP/E online panels to find the Owning FMID of "MOD IGX00030" and write it down for later reference (Step 4a below).
3. Save a backup copy of module IGC00030 before continuing with the next step. This module is located in your LPA library.
4. Edit member INSTMSGD in the IOA INSTCTT library. This member contains a job that uses SMP/E to compile member IGXMSGEX and link-edit it with module IGC00030.

Tailor the job as follows:

- A. Replace the FMID with the FMID obtained in Step 2.
- B. Replace the SMP JCL procedure name with your installation's MVS SMP/E JCL procedure.
- C. Update the target zone name to reflect your MVS target environment name.
- D. Add the IOA SMPPTS and IOA MAC libraries to your procedure's SYSLIB specification.

Submit the job.

The job should complete with a return code of 0.

To activate the CONTROL-T message display interface, either IPL your system, or use a system tool (e.g., RESOLVE) to dynamically add the new IGC00030 module to your MLPA.

Note *If you have already implemented the CONTROL-T Message Display Interface (described above) and you wish to update member IGXMSGEX, repeat only Step 4 (above) using job REDOMSGD instead of job INSTMSGD.*

External Data Manager Support

CONTROL-T supports various External Data Managers, such as HSM and DMS/OS and CA-ASM2. The installation procedures are described below.

For a general description of CONTROL-T support for External Data Managers (EDM), refer to External Data Manager Volumes in Section 4 of the CONTROL-T User Manual.

Note *Verify that there is an active CONTROL-T environment (established via CTTINIT) on each CPU in which an External Data Manager (EDM) is being invoked. If no CONTROL-T environment exists, the EDM interface fails to update the CONTROL-T Media Database and a synchronization problem between CONTROL-T and the EDM is created.*

HSM Support

To activate the HSM interface of CONTROL-T, perform the following steps:

1. Edit member ASMTVEXT in the IOA SAMPLE library. This member compiles and links the HSM Tape Volume exit, ARCTVEXT. ARCTVEXT is called by HSM whenever a tape volume no longer contains valid data. The exit marks the volume as scratch in the Media Database and, optionally, interfaces with a Robotic Tape Library at the site.

If you are currently not using exit ARCTVEXT, submit the job. If you are using this exit, member ARCTVEXT contains instructions for integrating your current ARCTVEXT exit with CONTROL-T's exit in Global Test and Global Phased modes. You should stop using your other product's ARCTVEXT exit when you move CONTROL-T to Global Production mode.

2. Verify that the IOA LOAD library is defined as one of the STEPLIB or LINKLIST libraries.
3. Verify that you have created rules that identify datasets created by HSM as EDM-controlled (DO RETENTION=EDM). The job name, dataset name, program name, or volser can be used as selection criteria. You may want to exclude the CDS backups made by HSM from HSM control. This can be done by defining special rules for these datasets.

Note *ADSM (ADSTAR Distributed Storage Manager) invokes deletion exit ARCTVEXT when it releases a tape. This exit, which is invoked in the ADSM address space, is identical to the ARCTVEXT exit used by HSM.*

DMS/OS Support

To activate the DMS/OS interface of CONTROL-T, perform the following steps:

1. Edit member ASMCTLEX in the IOA Sample library and submit the job. This member compiles and links the DMS Tape Volume exit, TMSCTLEX. Exit TMSCTLEX is called by DMS whenever a tape volume no longer contains any valid data. The exit marks the volume as scratch in the Media Database and, optionally, interfaces with the Robotic Tape Library at the site.

Section 7: Customizing and Administering CONTROL-T

If you are currently not using exit TMSCTLEX, submit the job. If you are using this exit, member TMSCTLEX contains instructions for integrating your current TMSCTLEX exit with CONTROL-T's exit in Global Test and Global Phased modes. Stop using your other product's TMSCTLEX exit when you move CONTROL-T to Global Production mode.

2. Verify that the IOA LOAD library is defined as one of the STEPLIB or LINKLIST libraries.
3. Make sure CONTROL-T contains rules that identify datasets created by DMS as EDM-controlled (using parameter DO RETENTION=EDM). The job name, dataset name or program name can be used as selection criteria.

CA-ASM2 Support

CONTROL-T and CA-ASM2 interface via CA-ASM2 exit \$FTEXT (the Free Tape Exit). CA-ASM2 passes control to this exit before freeing a tape. In the sample for this exit supplied with CONTROL-T (member AS2FTEXT in the IOA SAMPLE library), the CONTROL-T expiration utility (CTTEXP) is called to expire the volume being freed (i.e., to mark it SCRATCH in the CONTROL-T Media Database).

To activate the CA-ASM2 interface of CONTROL-T, perform the following steps:

1. Define all CA-ASM2 tapes as members in tape pools of CONTROL-T.

Definition of tape pools is not related to the value of CA-ASM2 installation parameter \$TAPPOOL. Tape pools must be defined for CA-ASM2 tapes regardless of whether or not CA-ASM2 tape pools are enabled. Pool definitions under CONTROL-T ensure that only CA-ASM2 SCRATCH tapes are used by CA-ASM2 jobs.

2. If parameter \$TAPPOOL is equal to X'01' (i.e., CA-ASM2 tape pools are enabled), skip to Step 3 (below). Otherwise, define rules that instruct CONTROL-T to use only tapes from these pools for datasets created by CA-ASM2.

The dataset names in the rules you define must be identical to those which appear in MOUNT messages for the relevant volumes. If you changed the Dummy dataset name (written as the first file on each CA-ASM2 tape), use the new name in the rules also.

3. Edit member AS2FTEXT in the IOA SAMPLE library. This member contains the \$FTEXT supplied with CONTROL-T, and it includes complete instructions on how to edit the member.

Note *When the CONTROL-T copy of \$FTEXT finishes processing, it can optionally give control to another copy of \$FTEXT. This can be useful when working with CONTROL-T in PHASED or TEST mode in parallel with another tape management product.*

4. Edit member ASMFTEXT in the IOA SAMPLE library. This job compiles and links exit \$FTEXT. Submit the job.
5. No updates to the CONTROL-T Media Database are required from CA-ASM2 exit \$NTEXT (New Tape).

If you choose to compile the original copy of exit \$NTEXT supplied with CA-ASM2, and CONTROL-T is the only tape management product installed, verify that NONE is specified for parameter &TAPEMGT before this exit is compiled.

6. Verify that the IOA LOAD library is defined as one of the STEPLIB or LINKLIST libraries.

DF/SMS Support

The CONTROL-T to DF/SMS Interface enables you to:

- Determine the expiration date of a dataset on the tape according to DF/SMS Management Class definitions.
- Use Management Class as a selection parameter in rule definitions (see ON MGMTCLAS in Section 3 of the CONTROL-T User Manual). See “CONTROL-T to DF/SMS Interface” in Section 4 of the CONTROL-T User Manual for information regarding the use of this interface.

To activate the DF/SMS interface of CONTROL-T, perform the following steps:

1. Edit member CTTPARM in the IOA INSTCTT library:
 - Specify Y for parameter SMSINTR.
 - Review the description of the OVERJCL installation parameter, and determine if its value should be changed.
 - Save this member.
2. Edit member CTTPARMJ in the IOA INSTCTT library, and submit the job. This job creates load modules CTTPARM and IOAPARM in the IOA LOAD library. All steps in this job must end with a condition code of 0.
3. If CONTROL-T is active, refresh the installation parameters in memory. To do this, issue operator command `S CTTINIT,PARM='MODE=LOAD,TBLT=PARM'`.
4. Define special Storage Classes and Management Classes for tape datasets. Few of the Management Class attributes are relevant for CONTROL-T and Storage Class attributes are not at all relevant for CONTROL-T. It is necessary to define special Storage Classes because DF/SMS activates the Management Class ACS routine only if a Storage Class is assigned to the dataset.
5. Adjust Installation ACS routines to support the new values for parameter &ACSENVIR set by CONTROL-T. (See “Automatic Class Selection” in Section 4 of the CONTROL-T User Manual for more information.)

See “Adjusting ACS Routines” below for some examples of DF/SMS ACS routines adjusted for CONTROL-T activation.

6. Translate and validate the newly adjusted ACS routines (via ISMF).
7. Validate the modified SCDS (via ISMF).
8. If validations in Steps 6 and 7 were performed successfully, activate the new SCDS.

Note At this stage, the CONTROL-T to DF/SMS interface is active. If the DF/SMS ACS routines assign a Management Class for a newly created dataset, the Management Class is recorded in the CONTROL-T Media Database and used by utility CTTRTM.

9. Optionally, you can define rules that use Management Class as a selection criterion (i.e., ON MGMTCLAS). If you define such rules, you must refresh the CONTROL-T rules in memory. To do this, issue the following operator command:

```
S CTTINIT,PARM='MODE=RELOAD,TBLT=RULE'
```

Section 7: Customizing and Administering CONTROL-T

Adjusting ACS Routines

Below are sample adjustments to ACS routines that allow for support of new values (supplied by CONTROL-T) for parameter &ACSENVIR.

Example 1

In this example, Management Class SHORTERM is assigned to all tape datasets that end with .TEMP.

Note that the same Storage Class (CTTSC) is assigned to all tape datasets. It is assigned only to enable DF/SMS to invoke a Management Class ACS routine.

Storage Class Adjustments

```
/* ***** */
/* DEFINE TAPE UNIT NAMES          */
/* ***** */
FILTLIST TAPES INCLUDE(TAPE*,CART*,T3420*,T3480*,'3420','3480')
/* ***** */
/* SET A STORAGE CLASS FOR TAPE DATASETS */
/* ***** */
WHEN (&ACSENVIR='CTTOPEN' && &UNIT=&TAPES)
SET &STORCLAS = 'CTTSC'
```

Management Class Adjustments

```
/* ***** */
/* SET A MANAGEMENT CLASS FOR TAPE  '*.TEMP' DATASETS */
/* ***** */
WHEN (&ACSENVIR='CTTOPEN' && &LLQ='TEMP')
SET &MGMTCLAS = 'SHORTERM'
```

Example 2

In this example, tape dataset DSN.WEEKLY is assigned to pool WEEKBKP, and Management Class KEEP MNTH is assigned to this dataset in the CONTROL-T Media Database (MDB).

Note that the same Storage Class (CTTSC) is assigned to all tape datasets. It is assigned only to enable DF/SMS to invoke a Management Class ACS routine.

A different Management Class is assigned for each value of parameter &ACSENVIR.

During mount processing for dataset DSN.WEEKLY, Management Class WEEK is assigned to this dataset. Rule 1 assigns pool WEEKBKP to this Management Class.

During open processing for this dataset, Management Class KEEP MNTH is assigned to this dataset and this information is recorded in the Media Database. Rule 2 causes the volume of this dataset to be vaulted in vault SAFE for 10 days.

Storage Class Adjustments

```

/*****/
/* DEFINE TAPE UNIT NAMES      */
/*****/
FILTLIST TAPES INCLUDE(TAPE*,T3420*,T3480*,'3420','3480')
/*****/
/* SET A STORAGE CLASS FOR TAPE DATASETS */
/*****/
WHEN (&ACSENVIR='CTTOPEN' && UNIT=&TAPES)
SET &STORCLAS = 'CTTSC'
WHEN (&ACSENVIR='CTTMNTV' && UNIT=&TAPES)
SET &STORCLAS = 'CTTSC'

```

Management Class Adjustments

```

/*****/
/* SET A MANAGEMENT CLASS FOR TAPE 'DSN.WEEKLY' DATASETS */
/*****/
WHEN (&ACSENVIR='CTTOPEN' && &DSN='DSN.WEEKLY')
SET &MGMTCLAS = 'KEEPMNTH'
WHEN (&ACSENVIR='CTTMNTV' && &DSN='DSN.WEEKLY')
SET &MGMTCLAS = 'WEEK'

```

CONTROL-T Rule 1

```

ON MGMTCLAS = WEEK      And/Or/Not A
ON DATASET  = *
=====
DO POOL     = WEEKBKP

```

CONTROL-T Rule 2

```

ON MGMTCLAS = KEEPMNTH And/Or/Not A
ON DATASET  = *
=====
DO VAULT    = SAFE
UNTILDAYSO010

```

Dynamic Dataset Stacking Facility

The Dynamic Dataset Stacking facility is a unique CONTROL-T feature that is significantly different from conventional and batch dataset stacking.

If you intend to implement dataset stacking, read this description of the Dynamic Dataset Stacking facility even if you are already familiar with conventional and batch dataset stacking. The information provided in this text will help you activate and test CONTROL-T's Dynamic Dataset Stacking facility.

For a general description of the Dynamic Dataset Stacking facility, refer to "Stacking" in Section 4 of the CONTROL-T User Manual. The following discussion assumes an understanding of CONTROL-T's stacking algorithms and operation.

General Considerations

Before activating the Dynamic Dataset Stacking facility, verify that the parameters that relate to the Dynamic Dataset Stacking facility are set during installation.

In general, the CONTROL-T Dynamic Dataset Stacking facility selects a volume for stacking only if the following conditions are met:

- Parameter DYNSTK=Y is specified in member CTTPARM and a DO STACK=Y statement is specified in a matching rule.
- The job name and dataset name of the creating job and created dataset match the job name and dataset name that appear in the stacking statistics file. If you want to change this criteria (e.g., consider all datasets with a specific prefix as the same dataset for stacking purposes), you can use the Dynamic Dataset Stacking facility user exit CTTX002 to override the default operation of CONTROL-T.
- The average length in feet of the specified dataset is read from the appropriate entry in the Stacking Statistics file. If this information cannot be found, stacking cannot continue. To avoid this problem, specify a default length for datasets via parameter STKDEFFT in member CTTPARM.
- An appropriate volume is found in the Media Database. This volume must be eligible for stacking, have sufficient free space, and belong to the same pool as the dataset to be stacked. Depending on the specification for CONTROL-T parameter DYNSTYP, the volume used for stacking may also need to have the same vaulting pattern as the dataset to be stacked.

The search for a suitable volume can be controlled by Exit 10 (Find Stackable Volume) which controls the stacking algorithm. For more information about Exit 10, refer to member DOCTX010 in the IOA SECUDATA library.

The search can be limited to a specific number of volumes via parameter STKSRCHL in member CTTPARM.

Note The above list is not a complete list of conditions, considerations and parameters. See "Stacking" in Section 4 of the CONTROL-T User Manual.

Non-scratch volumes converted to CONTROL-T from another Tape Management System are considered not eligible for stacking. Stacking is only performed for volumes that were moved from scratch status to non-scratch status (i.e., the first dataset was created on the scratch volume) while CONTROL-T was operational.

Activating the Dynamic Dataset Stacking Facility

Before activating the Dynamic Dataset Stacking facility, set and create all the required definitions in accordance with the specifications outlined in "Stacking" in Section 4 of the CONTROL-T User Manual.

To activate the Dynamic Dataset Stacking facility, the Stacking Statistics file must contain stacking information for created datasets. This file was created during the conversion process from your previous Tape Management System and is updated on a daily basis as part of the CONTROL-T New Day procedure.

Activating the Dynamic Dataset Stacking Facility in Global Test Mode

When CONTROL-T operates in Global Test mode, the Dynamic Dataset Stacking facility is not activated (since CONTROL-T does not intervene with the environment). However, it is sometimes desired to check the Dynamic Dataset Stacking facility before moving CONTROL-T to Global Production mode.

To activate the Dynamic Dataset Stacking facility while CONTROL-T is operating in Global Test mode, specify Y (Yes) for parameter STKTEST in member CTTPARM.

Testing the Dynamic Dataset Stacking Facility

For ease of testing, these instructions assume that your Stacking Statistics file does not contain sufficient information, but you would like to test the Dynamic Dataset Stacking facility anyway.

Testing a single job with a single dataset is usually the first step in activating the Dynamic Dataset Stacking facility. This can be accomplished by following the guidelines for one of the options listed below:

Option 1 Testing the Stacking Facility Without Updating Information in the Stacking Statistics File

A. Specify S for parameter DYNSTYP in the member CTTPARM.

Ensure that a value other than 0 is specified for parameter STKDEFFT in member CTTPARM.

B. Ensure that a number of volumes contain sufficient free space (but are not empty). These must be volumes that were "created" (i.e., their status was changed from SCRATCH to ACTIVE) while CONTROL-T was operational.

C. Create a dataset that matches the selection criteria (ON statements) in a rule containing the statement DO STACK=Y.

Option 2 Testing the Stacking Facility With Updating Information in the Stacking Statistic File

A. Create test datasets for stacking using a specific dataset name and job name. Perform this step while CONTROL-T is active so that information about these datasets will be stored in the CONTROL-T Trace file.

B. Run the CONTROL-T stacking Statistic utility (CTTSTK) with parameter TYPERUN MODE=TRC. The utility will collect dataset information from the Trace file and update the Stacking Statistics file. Refer to Section 6 of the CONTROL-T User Manual for additional information about utility CTTSTK.

C. The Dynamic Dataset Stacking facility is only activated for datasets that are to be cataloged. Leaving them cataloged prevents cataloging new datasets with the same name.

Section 7: Customizing and Administering CONTROL-T

- D. Make sure that you have a number of volumes with sufficient free space on them (but not empty). These must be volumes that were "created" (i.e., their status changed from scratch to active) while CONTROL-T was operational. These volumes may be required, according to CONTROL-T parameters, to belong to the same pool and have the same vaulting pattern as the test datasets.
- E. Recreate another version of the datasets created in step A using the same dataset name and job name. The datasets must be cataloged for the Dynamic Dataset Stacking facility.

Activating the Dynamic Dataset Stacking Facility for the Entire System

To activate the Dynamic Dataset Stacking facility for the entire system, set the parameters described above and in "Stacking" in Section 4 of the CONTROL-T User Manual. After activating and testing the Dynamic Dataset Stacking facility in Global Test mode, activating it in Global Production mode should not present any problem.

One point to consider is the specification of similar dataset names and job names as stackable. By default, CONTROL-T keeps statistics by specific dataset names and job names. As a result, a dataset is considered eligible for stacking only if it is created with the same dataset name and by the same job name, with the exception of Generation Data Groups (GDGs) that are treated as the same dataset name.

If you need to stack datasets whose names may vary (e.g., they contain the creation date as part of the dataset name) or that are created by a different job each time, you can implement user exit CTTX002 that overrides the Dynamic Dataset Stacking facility default operations.

For additional information on user exit CTTX002, refer to member DOCTX002 in the IOA SECUDATA library.

The following aspects of the search procedure should be considered when setting values for the parameters used by the Dynamic Dataset Stacking facility:

- CONTROL-T searches for a stackable volume for each tape dataset in the JCL. Datasets to be stacked must conform to the criteria described in Section 4 of the CONTROL-T User Manual. The datasets are then stacked according to the information in matching rules.
- The search process can be limited to a specific number of volumes via parameter STKSRCHL, so that only the specified number of volumes are scanned to find a suitable volume for stacking.
- The search process has function calls to user exit CTTX010 (Find Stackable Volume) which can be used to change search algorithm decisions (i.e., to Accept a volume, Reject a volume, or Supply a new volume to be used for stacking). For further information about Exit 10, refer to member DOCTX010 in the IOA SECUDATA library.

CONTROL-R Support to CONTROL-T

CTR001 (CONTROL-R exit 1) must be defined as an interface between CONTROL-R and CONTROL-T. This exit must be defined in addition to (not as a replacement for) any previously defined CONTROL-R exit 1 (e.g., an interface between CONTROL-R and another product such as HSM or CA-1).

The CONTROL-R Driver exit (described below) is used to invoke the CONTROL-T interface exit 1 and any previously defined exit 1(s).

Exit 1 used for the CONTROL-T interface receives dataset DELETE requests and changes the CONTROL-T controlled dataset status to Pending-Scratch. This enables expiration of the specified dataset by the next run of Retention Management utility CTTRTM. Via customization of this exit, you can cause CONTROL-T controlled datasets to be scratched immediately (i.e., without running utility CTTRTM).

CONTROL-R Driver Exit CTR001G

The CONTROL-R Driver exit invokes multiple user exits optionally used to provide programming interfaces to various dataset management software packages (e.g., HSM or CONTROL-T). The CONTROL-R Driver exit sequentially invokes all user exits listed in its source text (see step 4 below). Any number of single purpose CONTROL-R user exits may be listed.

Installing the CONTROL-R Interface

Perform the following steps to install the CONTROL-R to CONTROL-T interface.

1. Verify that YES is specified for CONTROL-R installation parameter TAPEMS in member CTRPARM.
2. Compile and link the existing CONTROL-R exit 1 via member ASMEXIT in the CONTROL-T JCL library. Include the following linkage-editor statements in the JCL to verify that the load module created by the link-edit step is placed in the IOA LOAD library and assigned the name CTRX001A:

```
ENTRY          CTRX001
NAME           CTRX001A(R)
```

3. Use member ASMEXIT to compile and link the sample exit used to implement the CONTROL-T to CONTROL-R interface. Member ASMEXIT is located in the CONTROL-T JCL library. The sample exit is located in member CTRX001Z in the IOA SECUDATA library.

Include the following linkage-editor statements in the JCL to verify that the load module created by the link-edit step is placed in the IOA LOAD library and assigned the name CTRXCTT:

```
ENTRY          CTRX001
NAME           CTRXCTT(R)
```

Section 7: Customizing and Administering CONTROL-T

4. Compile and link Driver exit CTRX001G via member ASMEXIT in the CONTROL-T JCL library. Include the following linkage-editor statements in the JCL to ensure that the load module created by the link-edit step is placed in the IOA LOAD library and assigned the name CTRX001:

```
CHANGE      CTRX001(CTR001A)
INCLUDE     SYSLIB(CTR001A)
CHANGE      CTRX001(CTR001CTT)
INCLUDE     SYSLIB(CTR001Z)
ENTRY       CTRX001
NAME        CTRX001(R)
```

The following exit names are provided with the CTRX001G sample Driver exit:

```
CTR001CTT   Name of the CONTROL-T to CONTROL-R interface exit 1.
CTR001A     Name of an additional CONTROL-R exit 1 that might be in use at
            the site.
```

See member CTRX001G in the IOA SECUDATA library for more information on the logic implemented by this exit.

CONTROL-T Application Programming Interface

The CONTROL-T Application Programming Interface provides an interface between CONTROL-T and user application programs. CONTROL-T API requests are performed using Assembler macros. All macros mentioned in this section, including mapping macros, are located in the IOA MAC library.

Calling programs can interface with CONTROL-T via either a Base Level or a High Level API. The differences between these types of API are discussed later in this section.

Note *Examples of macro code usage and JCL are provided later in this section.*

This section contains the following information:

Background Information

- CONTROL-T Control Table (TCT)
- Media Database structure and access
- Base Level vs. High Level API

Base Level API

- Record access by the Base Level API
- Macro CTTIOS
- Base Level API access of the Media Database

- Examples of macro access and Base Level API functions

High Level API

- Macro CTTACCDB
- Macro CTTCHKDB
- Media Database Handling Routines
- Example of macro usage and JCL for High Level API

Background Information

Users should become familiar with the following background information before using the CONTROL-T Application Programming Interface.

CONTROL-T Control Table (TCT)

CONTROL-T utilizes a control table across all environments under which CONTROL-T operates. This table, called the CONTROL-T Control Table (TCT), is mapped by macro CTTTCT.

The TCT contains, among other values, the various parameters specified in member CTTPARM during installation. The CONTROL-T API uses various indicators from the TCT in its decision-making process.

Note *The TCT, and all other major CONTROL-T blocks, are allocated above the 16MB line. Therefore, all programs using the CONTROL-T API must run in extended addressing mode (AMODE 31).*

The TCT is not a load module and therefore cannot be loaded using the LOAD macro. To load the TCT, use routine CTTTLD.

Sample instructions for loading the TCT are provided in the examples included in this section.

Media Database Structure

The API consists of Assembler macros that facilitate interaction with the CONTROL-T Media Database. Therefore, familiarity with the Media Database format is a prerequisite to using the API. For a detailed description of the Media Database see "Media Database Structure" earlier in this section.

Base Level vs. High Level API

A calling program can interface with CONTROL-T via either the Base Level or High Level API (Application Programming Interface). The Base Level API offers versatility with few fixed requirements. The High Level API provides more automation but requires an active CONTROL-T Real-Time Environment and APF authorization. Below is a table which details the differences between these two types of API access methods.

Note The term "Real-Time Environment" as used in this section refers to the CONTROL-T environment initialized by operator command: `S CTTINIT,PARM='MODE=INIT'`

Function	Base Level API	High Level API
Available Functions	Reads any Media Database record.	Performs Read, Update, Add, pr Delete functions on volume records only.
Sequential Read	Supported by the READNEXT function.	Not supported.
Authorization Required from the Calling Program	None.	APF authorization.
Keyed / Non-keyed Access	Both keyed and non-keyed (RBA based) access are supported.	Only keyed access supported.
Media Database Allocation	The calling program must allocate DAMDB, DAMDI and DATRC files.	The calling program does not need to allocate Database components. The Real-Time Environment Data file, Index file, and Trace file are allocated automatically for the calling program.
Load of CONTROL-T Control Table (TCT)	The calling program must load a CONTROL-T Control Table (TCT).	The calling program need not load a CONTROL-T Control Table (TCT). The Real-Time Environment TCT is accessed for the calling program.
CONTROL-T Real-Time Environment	Base Level API can be used whether or not the CONTROL-T Real-Time Environment is active.	High Level API can be used only if the CONTROL-T Real-Time Environment is active.

Base Level API

When accessing the Media Database, the Base Level API can perform the following functions:

OPEN	Access of the Media Database must begin with a request to open the database.
READ	This request reads a record. This function is generally used for reading specific records.
READNEXT	Reads the record(s) following the specified record, i.e., the next record(s). This function is generally used for reading a series of records.
CLOSE	Access of the Media Database must end with a request to close the database.

Record Access by the Base Level API

As mentioned earlier, records in the database can be accessed by non-keyed or keyed methods:

- To perform non-keyed access of database records, the full RBA must be specified.
- To perform keyed access of database records, the value specified depends on the function performed:
 - To perform a READNEXT operation, the entire index record must be specified.
 - To perform a READ operation, it is sufficient to use a partial key. Of course, the full key or even the full index record can be used.

Both READ and READNEXT requests return the full index record to the area pointed to by parameter KEY=.

A READ request is usually used for accessing a specific record. Since a READ request can use a partial keys as input, it is also useful when you have only partial data available (e.g., a volser without a label number when using the L-type index).

A READNEXT request is usually used for accessing series of records. Since, however, a READNEXT request requires specification of a full input value (whereas a READ request can use a partial key as an input value), a sequence of READNEXT requests must be preceded by a READ request.

Macro CTTIOS

CONTROL-T APIs use macro CTTIOS to access the Media Database. However, before the Base Level API can use macro CTTIOS, the CONTROL-T Control Table (TCT) must be loaded.

Instructions for loading the TCT are included in the examples for Base Level API later in this section.

Section 7: Customizing and Administering CONTROL-T

Format

Instructions for macro CTTIOS are formatted as follows:

```
Label    CTTIOS    request-type,
          function-code,
          RC=reason-code-field,
          [ENV=envir-id,]
          [TCT=tst-address,]
          [DBGLVL=debug-level,]
          [REC=buffer-address,]
          [RBA=field,]
          [DATA=Y/N,]
          [KEY=key-field,]
          [KEYTYPE=key-type,]
          [KEYLEN=key-length,]
          [MF={L, (E, list-address)}]
```

Note Macro CTTIOS may contain other parameters and fields, but only the parameters described in this section should be used. Specification of other parameters can lead to unpredictable results, and these additional parameters may not be supported in future releases of CONTROL-T.

Parameters

request-type Type of the request from macro CTTIOS. Mandatory. This parameter must always be included as the first parameter in the macro instructions. Valid values are:

VOL Volume data requests.
DS Dataset data requests.
ANY Both volume and dataset data requests.

function-code Function requested from macro CTTIOS. Mandatory. This parameter must always be included as the second parameter in the macro instructions. Valid values are:

OPEN Open the Media Database (to establish access to the Media Database).
CLOSE Close the Media Database (to terminate access to the Media Database).
READ Read data from the Media Database.
READNEXT Read the next data item from the Media Database.

These functions are described in more detail in "Base Level API Access of the Media Database" later in this section.

Section 7: Customizing and Administering CONTROL-T

RC Points to a four-byte area into which macro CTTIOS places the reason code. Mandatory for all request types and function codes.

Note The reason code should not be confused with the return code located in register 15 (described later).

Reason codes are extremely useful when analyzing the reasons a request failed, and should be kept for debugging and error handling purposes. Reason code explanations can be found in the description of message CTT200S in the IOA Messages and Codes Manual.

ENV Points to a four-byte field which identifies the environment under which the Media Database is opened. Mandatory for Media Database OPEN requests.

The value specified in the ENV= parameter is passed to user exits and is recorded in the Trace file when Media Database updates are performed.

TCT Address of the TCT returned by macro CTTTLD. Mandatory for Media Database OPEN requests.

DBGLVL One-byte field specifying the tracing level to be used for further executions of macro CTTIOS. Mandatory for Media Database OPEN requests. It should contain value X'00'.

REC The buffer into which macro CTTIOS puts data retrieved from the Media Database. Mandatory for READ requests. REC should point to an area large enough to hold the data record retrieved from the database by the macro.

RBA Record location identifier used to determine which record should be retrieved from the Data file.

For READ requests, the field should specify the RBA of the record to be read (or X'000100' for the first record). For READNEXT requests, the field should specify the RBA of the record preceding the record to be read (i.e., the RBA returned by the previous READ/READNEXT request).

DATA Whether or not to read the Data record of the specified dataset/volume. Valid only when the READ or READNEXT function code (described above) is specified. Valid values are:

YES Read the Data record of the specified dataset/volume.

NO Do not read the Data record of the specified dataset/volume.

KEY Points to a valid index record (mapped by macro CTTDVX, CTTDDX or CTTDLX, described earlier in this section) that serves as the key (or partial key) by which macro CTTIOS retrieves the data record from the Media Database. Valid for READ requests. This parameter is related to parameters KEYTYPE and KEYLEN.

For READ requests, this field should specify the key (or partial key) of the record to be read. For READNEXT requests, this field should specify the key of the record preceding the record to be read (i.e., the key returned by the previous READ/READNEXT request).

Section 7: Customizing and Administering CONTROL-T

KEYTYPE Type of index being used. Valid when accessing data records that may be pointed to by more than one index type (i.e., dataset records).

Because volume records are indexed by only one key type (the V-type index), this parameter should not be specified when performing keyed access of volume records.

When performing keyed access to the dataset records, valid values are:

D When using the D-type key (mapped by DSECT CTTDDX). Default.

L When using the L-type key (mapped by DSECT CTTDLX).

KEYLEN Points to a four-byte field which specifies the length of the key to be searched. The record length of the Index file is identical for all index types; therefore, this parameter is useful when using an index value shorter than the maximum index length of the file, or when using a partial key.

Parameter KEYLEN should be used with both full keys and partial keys. Set all necessary fields in the key area (pointed to by parameter KEY) and specify a KEYLEN parameter that includes only the desired fields. When the full key is specified, set the parameter KEYLEN to the full length of the key field. This is demonstrated in the examples provided in this section.

Macro CTTIOS reads the entire index record into the area specified by parameter KEY. Therefore, when using this parameter, be sure to allocate sufficient storage for the entire index record, or storage overlays will occur.

MF Macro invocation form. Valid values are:

L List form Generate all required data areas.

E Execute form Perform the request specified by the macro parameters using the data areas generated by the List form of the macro. When the Execute form of the macro is used it must be accompanied by a pointer to the List form of the macro.

This same list form address must be used for all subsequent accesses of the Media Database via macro CTTIOS until the Media Database is closed.

All invocations of macro CTTIOS should access the same parameter area. It is therefore preferable to specify the combined Execute and List forms of the macro.

Section 7: Customizing and Administering CONTROL-T

Macro CTTIOS must be coded in accordance with all standard macro Assembler coding restrictions.

The values in general purpose registers 0, 1 and 14 are modified by macro CTTIOS.

Macro CTTIOS accesses a set of variables generated by macro CTTDBTP. Therefore, macro CTTDBTP should be included in the source code of the calling program.

Return Codes

Macro CTTIOS returns a return code in general purpose register 15. This should not be confused with the reason code returned in parameter RC described above.

0	Successful execution of the request
4	Requested record not found (READ requests) / End-of-File (READNEXT requests)
Greater than 4	Unsuccessful execution of the request; the reason-code indicates additional information regarding the reason for failure of the request.

For more details on the return codes and reason codes of macro CTTIOS, see the description of message CTT200S in the IOA Messages and Codes Manual.

Base Level API Access of the Media Database

Note Sample macro code and examples are provided later in this section.

Opening/Closing Files

Access of the Media Database must begin with an OPEN request. The OPEN request automatically opens the Data, Index and Trace files. A Media Database OPEN request should be accompanied by parameters ENV, TCT and DBGLVL (described earlier).

Access of the Media Database must end with a CLOSE request. The CLOSE request automatically closes all the files and performs all other required activities.

Reading Files

As mentioned earlier, Media Database READ requests can be made using either of the following function codes: READ and READNEXT.

The differences between these requests should be understood if they are to be used effectively. These function codes, the correlation between them and the various key and RBA related parameters are described below.

READ Requests

The READ request is used to read a specific data record from the Media Database. The READ request locates the requested data record and retrieves it into the area specified by the REC= parameter.

Either a keyed or non-keyed (i.e., RBA) access method can be used to specify the record to be read.

Section 7: Customizing and Administering CONTROL-T

When using keyed access to perform a READ from the Media Database, either a full or partial key can be specified. The READ request returns the whole index record (of the data record being read) into the area specified by parameter KEY=. Therefore, even when using a partial key, parameter KEY= must specify an area large enough to contain the whole index record (not just the partial key).

If the requested record is not found, either because there is no record at the specified RBA or because no record exists with the specified full or partial key, macro CTTIOS returns a return code of 4.

Using the READ request in conjunction with partial keys is useful for reading from the Media Database when you have partial data only.

For example, assume you know a volser and want to find the name of the first dataset on the volser, but you do not know the label number for this dataset (e.g., the first label may belong to a file that is already expired). In this case, you can use partial keys as follows:

Select request type VOL, specify KEYTYPE=L and set the VOLSER field in the L-type index to the known volser. Instead of specifying the label number, read using a KEYLEN= parameter that includes the volser number but does not include the label number. Macro CTTIOS then retrieves the first record that matches the volser number, which is the dataset record of the first label number that exists for this volser. The area specified by the key field will be updated with the full-key value of the retrieved record.

READNEXT Requests

The READNEXT request is used to read the next data record(s) from the Media Database. A READNEXT request can be used with either an RBA or a full index record. READNEXT finds and reads the record (matching the request type specified in macro CTTIOS) that follows the specified RBA or index (i.e., the next record), and then updates the KEY= field or RBA= field with the corresponding value of the record being read.

Because READNEXT requests require specification of an RBA value or a full index record, and because READ requests always store the RBA value or the full index record of the record read, a READ request should always precede a READNEXT request or set of READNEXT requests. The READ request can be specified with either the RBA= or KEY= parameters and partial keys can be used. The READNEXT request should then be specified without explicitly modifying the value in the KEY or RBA fields, and without specifying parameter KEYLEN.

Because both READ and READNEXT update the key or RBA value with each invocation, it is possible to follow the READ request with more than one READNEXT request. Each READNEXT request will, in its turn, read the next record and update the index record (pointed to by the KEY parameter) or the RBA (pointed to by the RBA parameter) to the value corresponding to the record being read.

The READNEXT function is generally used for reading a series of records in the Media Database, such as all records matching a prefix, or all records until the end-of-file.

For example, to read all records that match a dataset name prefix, first specify a READ with the partial key (the dataset prefix) followed by one or more READNEXT requests. The first record matching the dataset prefix will be read (if one exists), and the READNEXT requests will read all subsequent records (datasets) following the first match. If no matching record exists, CTTIOS positions the program for subsequent READNEXT requests according to ordinary key sorting rules.

Note: A READNEXT operation does not automatically stop after all records matching the prefix (or any other partial key) are read. The READNEXT request stops reading, and returns a return code of 4, only when the end-of-file is reached. Therefore, your application program must check when the READNEXT request results in a record that does not match the partial key, and stop issuing more READNEXT requests.

Sequential Reading of the Entire Media Database – Keyed vs. Non-Keyed Access

The entire Media Database can be read either by using indexes (e.g., reading all volumes by using the V-type index) or by using RBAs to read through the entire Media Database. Every program that requires sequential access to the entire Media Database must select the appropriate access method. When selecting the access method, the following points should be considered:

- When reading the Media Database using RBAs, one I/O operation is performed for each block (one block contains a number of records). Thus, the number of I/O operations required for reading the entire Media Database is relatively small. When reading the Media Database sequentially via indexes, each record access results in one access to the index, and one access to the data. This may result in a high I/O count and less than satisfying performance for the program.
- When reading the Media Database via indexes, records are read in ascending sort order. When reading the Media Database using RBAs, records are read in the order they appear in the data file (not necessarily sorted).

It is much faster to read the Media Database using RBAs rather than using indexes. If data items from the Media Database do not have to be sorted, it is highly recommended to use RBA-based reading of the Media Database.

Even if the data items retrieved from the Media Database should be sorted, it might be more efficient (depending on the environment) to read the Media Database using RBAs, and sort it after you have read it. You can also use sort utilities to merge volume and dataset records retrieved from the Media Database, instead of performing keyed accesses for every volume (or for every dataset).

Examples for Base Level API

Below are several examples of how to perform the various functions described in this section. Note, however, that these are examples and should not be used without appropriate modification. Most of the examples provided do not check the return code from macro CTTIOS. Some examples refer to Assembler symbols that are not included in the example.

Loading the TCT into Storage

The TCT (CONTROL-T Control Table) is not a load module and therefore cannot be loaded using the LOAD macro. To load the TCT, use macro CTTTLD. Below is a sample program which uses macro CTTTLD to load the TCT.

Section 7: Customizing and Administering CONTROL-T

Sample TCT Load Request

```
CALL    CTTTLD, (LOAD, TCTADDR, 0, 0)    LOAD THE TCT
LTR     R15, R15                          SUCCESSFUL?
BNZ     TCTFAIL                            NO - TERMINATE
.
.
TCTFAIL WTO    'TCT LOAD FAILED, PROGRAM TERMINATED'
ABEND   8, DUMP ABEND PROGRAM WITH U0008
.
.
LOAD    DC     CL8 'LOAD'                  CONSTANT FOR CTTTLD
TCTADDR DS     A                          ADDRESS OF TCT
```

Sample Media Database OPEN Request

```
CALL    CTTTLD, (LOAD, TCTADDR, 0, 0)    Load the TCT
.
.
L       R12, TCTADDR
OPENMDB CTTIOS ANY, OPEN, ENV=$ENV, DBGLVL=$DBG, TCT=(R12),    Open the MDB *
        RC=RESCODE, MF=(E, IOSLIST)
.
.
IOSLIST CTTIOS MF=L
$ENV    DC     CL4 'USER'                  Environment specification
$DBG    DC     X'00'                       Debug level
TCTADDR DS     A                          Address of TCT
RESCODE DS     F
LOAD    DC     CL8 'LOAD'                  Constant for CTTTLD
```

Note The return code from macro CTTTLD should also be checked.

Sample CLOSE Request

```
CLOSEMDB CTTIOS ANY, CLOSE, MF=(E, IOSLIST)
.
.
IOSLIST CTTIOS MF=L
```

Keyed Reading From the Media Database

CONTROL-T Base Level API provides keyed access to data residing in the Media Database. Keyed access is achieved using the indexes, as explained above.

Section 7: Customizing and Administering CONTROL-T

Following are samples for keyed access requests from the Media Database:

Sample "Read by Volser" Request

	L	R2,=A(VOLKEY)	Volume key field
	USING	DVX,R2	
	XC	VOLKEY,VOLKEY	
	MVC	DVXVOLSR,=C'123456'	Prepare vol. key
	LA	R15,=A(L'DVXVOLSR)	Length of volser
READVOL	CTTIOS	VOL,READ,KEY=VOLKEY,RC=RESCODE,REC=VOLREC, * KEYLEN=(R15),MF=(E,IOSLIST)	Read from MDB *
	LTR	R15,R15	Was read OK?
	BNZ	NOTFOUND	No - treat accordingly
	.		
	.		
VOLKEY	DS	XL(DVKLEN)	Vol. index key
VOLREC	DS	XL(DVLLEN)	Vol. record buffer
RESCODE	DS	F	CTTIOS reason code
IOSLIST	CTTIOS	MF=L	CTTIOS list form
	.		
	.		
	* CONTROL-T MAPPING MACROS		
	CTTDVL		Vol. data record
	CTTDVX		Vol. index record
	CTTDBTP		CTTIOS parameters block

Section 7: Customizing and Administering CONTROL-T

Sample "Read by Dataset Name" Request

Since D-type indexes are non-unique (i.e., more than one dataset may have the same name) the sample shown below will read only the record of the first dataset in the Media Database with the specified name.

LA	R2, DSNKEY	D.S. key field
USING	DDX, R2	
XC	DSNKEY, DSNKEY	
MVC	DDXDSN, =CL44 'CTT.SAMPLE.FILE'	Prepare D.S. key
LA	R15, =A(L'DDXDSN)	
CTTIOS	DS, READ, KEY=DSNKEY, REC=DSNREC,	Read from MDB *
	KEYLEN=(R15), ENQ=Y, MF=(E, IOSLIST)	
LTR	R15, R15	Was read OK?
BNZ	NOTFOUND	No - treat accordingly
.		
.		
DSNKEY	DS XL(DDXLEN)	D.S. index key
DSNREC	DS XL(DDSLEN)	D.S. record buffer
RESCODE	DS F	CTTIOS reason code
IOSLIST	CTTIOS MF=L	CTTIOS list form
.		
.		
* CONTROL-T MAPPING MACROS:		
	CTTDDS	D.S. data record
	CTTDDX	D.S. index record
	CTTDBTP	CTTIOS parameters block

Note In the above example, the *KEYTYPE* parameter is omitted. It defaults to *KEYTYPE=D*.

Section 7: Customizing and Administering CONTROL-T

Sample "Read by Volser and File Sequence Number" Request

LA	R2, DLSKEY	D.S. key field
USING	DLX, R2	
XC	DLSKEY, DLSKEY	
MVC	DLXVOLSR, =C'123456'	Copy volser
LA	R3, 1	First D.S. on volser
STCM	R3, B'0011', DLXLBLNM	Store label number
LA	R15, =A(L'DLXVOLSR+L'DLXLBLNM)	
CTTIOS	DS, READ, KEY=DLSKEY, KEYLEN=(R15), KEYTYPE=L, REC=DSNREC, MF=(E, IOSLIST)	Read from MDB *
LTR	R15, R15	Was read OK?
BNZ	NOTFOUND	No - treat accordingly
.		
.		
DLSKEY	DS XL(DLXLEN)	D.S. L-type key
DSNREC	DS XL(DDSLEN)	D.S. record buffer
RESCODE	DS F	CTTIOS reason code
IOSLIST	CTTIOS MF=L	CTTIOS list form
.		
.		
* CONTROL-T MAPPING MACROS		
	CTTDDS	D.S.data record
	CTTDLX	D.S.L-type index
	CTTDBTP	CTTIOS parameters block

Section 7: Customizing and Administering CONTROL-T

Sample "Read All Datasets With Specified Prefix" Using the READNEXT Request

LA	R2, DSNKEY	D.S. key field	
USING	DDX, R2		
L	R3, =A(DSNREC)	Record addressability	
USING	DDS, R3		
XC	DSNKEY, DSNKEY	Clear index	
MVC	DDXDSN(L'DSPRFX), DSPRFX	Dataset's prefix	
LA	R15, =A(L'DSPRFX)	Length of prefix	
CTTIOS	DS, READ, KEY=DSNKEY, REC=DSNREC, KEYLEN=(R15), MF=(E, IOSLIST)	Read from MDB	*
.			
.			
READNEXT	CTTIOS DS, READNEXT, KEY=DSNKEY, REC=DSNREC, MF=(E, IOSLIST)	Read next dataset from MDB	*
LTR	R15, R15	Successful ?	
BNZ	NOMORE	No - skip further READNEXTs	
CLC	DDSDSN(L'DSPRFX), DSPRFX	Is it still the same prefix	
BE	READNEXT	Yes - loop again	
NOMORE	EQU		*
.			
.			
DSPRFX	DC CL15 'DATASET.PREFIX.'	Dataset's prefix	
DSNKEY	DS XL(DDXLEN)	D.S. index key	
DSNREC	DS XL(DDSLEN)	D.S. record buffer	
RESCODE	DS F	CTTIOS reason code	
IOSLIST	CTTIOS MF=L	CTTIOS list form	
.			
.			
* CONTROL-T MAPPING MACROS			
	CTTDDS	D.S. data record	
	CTTDDX	D.S. index record	
	CTTDBTP	CTTIOS parameters block	

Non-Keyed (RBA-Based) Reading of the Media Database

CONTROL-T Base Level API also provides non-keyed access to data residing in the Media Database. Non-keyed access is achieved by using the RBA as the pointer to the desired record in the data file, as explained above.

Following are a few samples for RBA-based access requests from the Media Database:

Section 7: Customizing and Administering CONTROL-T

Sample "Read a Record" Request

MVC	RECRBA,=X'000203'	Set the RBA to block 2, record 3
CTTIOS	ANY,READ,RC=RESCODE,REC=AREA,RBA=RECRBA, MF=(E,IOSLIST)	Read from MDB *
LTR	R15,R15	Does record exists ?
BNZ	NOTFOUND	No - treat accordingly
L	R3,=A(AREA)	R3 points to the record read
USING	DVL,R3	
CLI	DVLRTYPE,DVLRVL	Is it a volume record ?
BE	TREATVOL	Yes - treat accordingly
CLI	DVLRTYPE,DDSRDS	Is it a dataset record ?
BE	TREATDS	Yes - treat accordingly
B	TREATOTH	Treat other record types
.		
.		
AREA	DS CL(DVLLLEN)	Record buffer (both vol and D.S.)
RECRBA	DS XL3	RBA of record
.		
.		
* CONTROL-T MAPPING MACROS		
CTTDVL		Vol. data record
CTTDDS		D.S. data record
CTTDBTP		CTTIOS parameters block

In the above example:

- You do not have to check whether or not the record being read is free or active, since macro CTTIOS does that for you. If the record is free, the macro returns a return code of 4 (not found).
- Since the header of all record types is the same, it is possible to compare the record type FIELD from the volume DSECT against the record type EQUATE from the dataset DSECT.

Sample "Read Next Record" Request

MVC	RECRBA,DVLRBA	Set RBA of current record
CTTIOS	ANY,READNEXT,REC=AREA, RBA=RECRBA,RC=RESCODE,MF=(E,IOSLIST)	Read the next record from MDB *
LTR	R15,R15	
.		
.		

In the above example:

- The rest of the sample program may be the same as the prior example.
- When using the READNEXT function code, macro CTTIOS skips free records and looks for active records to read. A return code of 4 (not found) signals that the end-of-file was reached.

Section 7: Customizing and Administering CONTROL-T

Sample Media Database Access Program

The sample access program sequentially reads all the Media Database using RBAs and produces the following reports:

- All volumes with all the datasets for each volume.
- All datasets with all the volumes for each dataset.

This program provides a working example for a wide variety of requests available through macro CTTIOS.

```
*****
*
*           CTTSAM1
*
* FUNCTION: SAMPLE PROGRAM TO DEMONSTRATE ACCESS TO THE CONTROL-T
*           MEDIA DATABASE (MDB) USING THE BASE LEVEL API.
*           THE PROGRAM READS THE MDB AND PRODUCES TWO REPORTS:
*           1. A VOLUME REPORT CONTAINING ALL VOLUMES IN THE MDB
*              AND ALL DATASETS EXISTING ON EACH VOLUME
*           2. A DATASET REPORT CONTAINING ALL DATASETS IN THE MDB
*              AND ALL VOLUMES THAT ARE PART OF EACH DATASET
*
* RETURN CODES: 00 - OK
*               04 - MDB EMPTY
*               08 - ERROR DURING PROCESSING
*
* LOGIC: THE PROGRAM READS THE MDB SEQUENTIALLY USING RBAS. VOLUME
*        AND DATASET RECORDS ARE PROCESSED, AND OTHER RECORDS ARE
*        DISREGARDED.
*        FOR VOLUME RECORDS, THE DATASETS OF THE VOLUME ARE READ,
*        USING THE L-TYPE INDEX. A READ REQUEST WITH NO LABEL
*        NUMBER (VOLSER ONLY) IS PERFORMED FIRST. THE READNEXT
*        REQUEST IS THEN USED REPEATEDLY UNTIL ALL DATASETS ON THE
*        VOLUME ARE PROCESSED.
*        FOR DATASET RECORDS, THE VOLUMES OF THE DATASET ARE READ
*        THE FIRST VOLSER IS TAKEN FROM THE DATASET RECORD, AND
*        AFTERWARDS, THE VOLUME RECORDS ARE BEING READ, AND THE
*        NEXT VOLUME RETRIEVED FROM THEM, UNTIL ALL VOLUMES OF THE
*        DATASET HAVE BEEN READ.
*
* DD CARDS:  DAMDB - MDB DATA FILE
*            DAMDI - MDB INDEX FILE
*            DATRC - TRACE FILE
*            DARPTVOL - VOLUME REPORT FILE
*            DARPTDSN - DATASET NAMES REPORT FILE
*
```

Section 7: Customizing and Administering CONTROL-T

```

*                                                                    *
*  DISCLAIMER:  THIS SAMPLE PROGRAM IS PROVIDED ON AN AS IS BASIS,  *
*                WITHOUT ANY WARRANTY,  EITHER EXPRESS OR IMPLIED  *
*                                                                    *
*  REGISTERS:  R13      -  BASE                                       *
*                R2      -  MDB RECORD KEY  (WHERE APPLICABLE)      *
*                R3      -  MDB RECORD BUFFER                       *
*                                                                    *
*  ATTRIBUTES:  AMODE 31                                           *
*                RMODE 24                                           *
*                                                                    *
*  TO COMPILE AND LINK THE PROGRAM :                                  *
*                                                                    *
*  //CTTSAM1 JOB ...                                               *
*  //ASM      EXEC CTTASM                                           *
*  //C.SYSIN DD DSN=CTT.V210.SAMPLE(CTTSAM1)                        *
*  //L.SYSLMOD DD DISP=SHR,DSN=YOUR.LOAD.LIBRARY                   *
*  //L.SYSIN  DD *                                                *
*    INCLUDE SYSLIB(CTTTLD)                                         *
*    MODE RMODE(24),AMODE(31)                                       *
*    NAME CTTSAM1(R)                                               *
*  /**                                                              *
*  //RUN EXEC PGM=CTTSAM1,COND=(0,NE),REGION=60M                  *
*  //STEPLIB DD DISP=SHR,DSN=YOUR.LOAD.LIBRARY                     *
*  //DAMDB DD DISP=SHR,DSN=CTT.V210.MDBD                             *
*  //DAMDI DD DISP=SHR,DSN=CTT.V210.MDBI                             *
*  //DATRC DD DISP=SHR,DSN=CTT.V210.TRC                             *
*  //DARPTVOL DD SYSOUT=*                                           *
*  //DARPTDSN DD SYSOUT=*                                           *
*  //                                                                    *
*  /**                                                                    *
*                                                                    *
*****

```

```

*****
EJECT
CTTDDS      D.S.  DATA RECORD
CTTDVL      VOL.  DATA RECORD
CTTDVX      VOL.  INDEX RECORD
CTTDDX      D.S.  INDEX RECORD
CTTDLX      D.S.  L-TYPE INDEX
CTTDBTP     CTTIOS PARAMETERS BLOCK

EJECT
MACRO
&NAME      MYPUT &DCB,&REC
&NAME      LA    R15,++6              ADDRESS FOLLOWING BASSM
          BASSM R14,R15              CHANGE AMODE

```

Section 7: Customizing and Administering CONTROL-T

```

PUT   &DCB,&REC           PUT THE DESIRED RECORD
LA    R15,*,+10          ENDING ADDRESS
O     R15,=X'80000000'
BSM   0,R15
MEND

*****
EJECT
CTTSAM1 CSECT
BEGIN *,EQR=YES
CTTLEVEL
SPACE 3

***
* INITIALIZE PROGRAM - LOAD THE TCT AND OPEN FILES
***

CALL CTTTLD,(LOAD,TCTADDR,0,0)  LOAD THE TCT
LTR   R15,R15                SUCCESSFUL ?
BNZ   LTCTFAIL              NO - TERMINATE
SPACE 3
OPEN  (RPTVOL,OUTPUT)       OPEN VOLUME REPORT OUTPUT FILE
LTR   R15,R15                SUCCESSFUL ?
BNZ   OUTERROR              NO - TERMINATE
OPEN  (RPTDSN,OUTPUT)       OPEN DATASET REPORT OUTPUT FILE
LTR   R15,R15                SUCCESSFUL ?
BNZ   OUTERROR              NO - TERMINATE
MYPUT RPTVOL,VOLMSG         PRINT VOLUME REPORT HEADER
MYPUT RPTDSN,DSNMSG        PRINT DATASET REPORT HEADER
SPACE 3
L     R12,TCTADDR
MVC   MSGOP,=CL8'OPEN'
CTTIOS ANY,OPEN,ENV=$ENV,DBGLVL=$DBG, OPEN MDB      *
      TCT=(R12),RC=RESCODE,MF=(E,IOSLIST)
LTR   R15,R15                SUCCESSFUL ?
BNZ   MDBERROR              NO - TERMINATE
SPACE 3

***
* SEQUENTIAL READING OF THE MDB, BY RBA
***

MVC   RECRBA,RBA0          START WITH THE FIRST RECORD
MVC   MSGOP,=CL8'READ'
CTTIOS ANY,READ,RC=RESCODE,REC=AREA,RBA=RECRBA, *
      MF=(E,IOSLIST)  READ THE FIRST RECORD
LTR   R15,R15                SUCCESSFUL ?
BZ    READOK                 YES - PROCESS THE NEXT RECORD
CH    R15,=H'4'             WAS IT BECAUSE FIRST RECORD IS FREE?

```

Section 7: Customizing and Administering CONTROL-T

```

BH      MDBERROR          NO - TERMINATE          *
                                     OTHERWISE TRY TO READ THE NEXT ONE
MVC     MSGOP,=CL8'READNEXT'
CTTIOS ANY,READNEXT,RC=RESCODE,REC=AREA, READ NEXT RECORD  *
      RBA=RECRBA,MF=(E,IOSLIST)
LTR     R15,R15           WAS READNEXT SUCCESSFUL ?
BZ      READOK            YES - PROCESS THE RECORD
CH      R15,=H'4'        E.O.F ?
BE      MDBEMPTY         YES - MDB IS EMPTY
B       MDBERROR         OTHERWISE - TERMINATE
SPACE 3

***
* LOOP ON ALL RECORDS
***
READOK  EQU  *
      L      R3,=A(AREA) ESTABLISH ADDRESSABILITY
      USING DVL,R3      TO THE RECORD BUFFER
      CLI    DVLRTYPE,DVLRVL IS THIS A VOLUME RECORD ?
      BE     TREATVOL    YES - TREAT ACCORDINGLY
      CLI    DVLRTYPE,DDSRDS IS THIS A DATASET RECORD ?
      BE     TREATDS     YES - TREAT ACCORDINGLY
READNEXT EQU  *
      MVC     MSGOP,=CL8'READNEXT'
      CTTIOS ANY,READNEXT,RC=RESCODE,REC=AREA, READ NEXT RECORD  *
      RBA=RECRBA,MF=(E,IOSLIST)
      LTR     R15,R15     WAS READ SUCCESSFUL ?
      BZ      READOK     YES - PROCESS THIS RECORD
      CH      R15,=H'4'  E.O.F ?
      BE      EODAD      YES - ISSUE MESSAGE
      B       MDBERROR   OTHERWISE TERMINATE
      DROP   R3
      SPACE 3

***
* TREAT ONE VOLUME RECORD - PRINT ALL ITS DATASETS
***
TREATVOL EQU  *
      USING DVL,R3      ESTABLISH ADDRESSABILITY
      MVC     VOLUME,DVLVOLSR MOVE VOLSER TO OUTPUT RECORD
      MYPUT  RPTVOL,OUTVOL
      TM      DVLSTAT,DVLSSCR IS IT SCRATCH ?
      BNO    GETDSNS     NO - GET DATASETS ON THE VOLUME
      MVC     DSNAME,SCRIND ELSE - ISSUE SCRATCH MESSAGE
      MYPUT  RPTVOL,OUTDSN PRINT THE SCRATCH MESSAGE
      B       ENDEVOL    END VOLUME PROCESSING

```

Section 7: Customizing and Administering CONTROL-T

```

GETDSNS EQU *
LA R2,DLSKEY ESTABLISH ADDRESSABILITY
USING DLX,R2 TO D.S. L-TYPE INDEX
XC DLSKEY,DLSKEY CLEAR INDEX FIELD
MVC DLKVOLSR,DVLVOLSR MOVE VOLSER TO KEY FIELD
DROP R3 VOLUME RECORD NOT NEEDED ANY MORE
LA R15,=A(L'DLKVOLSR)
MVC MSGOP,=CL8'READ'
CTTIOS DS,READ,KEY=DLSKEY, READ FROM MDB *
RC=RESCODE, *RC *
KEYTYPE=L,KEYLEN=(R15),REC=AREA,MF=(E,IOSLIST)
CH R15,=H'4' *NFD
BE ENDVOL *NFD1
BH MDBERROR *NFD

PRINTDSN EQU *
USING DDS,R3 ADDRESSABILITY (R3->>RECORD BUFFER)
MVC DSNAME,DDSASN MOVE DSNAME TO OUTPUT RECORD
MYPUT RPTVOL,OUTDSN PRINT IT
MVC MSGOP,=CL8'READNEXT'
CTTIOS DS,READNEXT,KEY=DLSKEY, READ NEXT RECORD FROM MDB *
RC=RESCODE, *RC *
KEYTYPE=L,REC=AREA,MF=(E,IOSLIST)
LTR R15,R15 SUCCESSFUL ?
BZ PROCDSN YES - PROCESS DSN
CH R15,=H'4' NO - WAS IT BECAUSE THE END OF MDB ?
BH MDBERROR NO - ISSUE AN ERROR MESSAGE
B ENDVOL YES - PROCESS NEXT VOLUME

PROCDSN EQU *
CLC DLKVOLSR,VOLUME ARE WE STILL IN THE SAME VOLUME ?
BE PRINTDSN YES - PRINT IT AND PROCESS NEXT DS

ENDVOL EQU *
B READNEXT GO TO GET NEXT RECORD
SPACE 3

***
* TREAT ONE DATASET RECORD - PRINT ALL ITS VOLUMES
***

TREATDS EQU *
USING DDS,R3 ESTABLISH ADDRESSABILITY
MVC DSNAME,DDSASN MOVE DSNAME TO OUTPUT RECORD
MYPUT RPTDSN,OUTDSN PRINT IT
MVC VOLUME,DDSVOLSR MOVE FIRST VOLSER TO MESSAGE
MYPUT RPTDSN,OUTVOL PRINT MESSAGE
LH R4,DDSVOLSR# NUMBER OF VOLUMES IN DATASET
CH R4,=H'1' IS THERE ONLY ONE VOL IN DATASET ?

```

Section 7: Customizing and Administering CONTROL-T

```

BNH  ENDDSN          YES - END PROCESSING OF DATASET
DROP R3              ADDRESSABILITY NOT NEEDED ANY MORE
SPACE 3

***
* FIND ALL VOLUMES OF DATASET
***

LA   R2,DVXKEY       ESTABLISH ADDRESSABILITY
USING DVX,R2         TO VOLUME INDEX FIELD
USING DVL,R3         AND TO VOLUME RECORD
XC   DVXKEY,DVXKEY   CLEAR VOLUME KEY FIELD
BCTR R4,0            DECREMENT VOL COUNT (1 PRINTED)
NEXTVOL EQU *
MVC  DVXVOLSR,VOLUME
LA   R15,=A(L'DVXVOLSR)
MVC  MSGOP,=CL8'READ'
CTTIOS VOL,READ,KEY=DVXKEY, READ FROM MDB:
      RC=RESCODE,
      REC=AREA,KEYLEN=(R15),MF=(E,IOSLIST)
CH   R15,=H'4'
BE   ENDDSN
BH   MDBERROR
MVC  VOLUME,DVLNEXT  NEXT VOLUME (PREV ALREADY PRINTED)
MYPUT RPTDSN,OUTVOL  PRINT THE VOLUME SERIAL
BCT  R4,NEXTVOL      AND GO GET NEXT VOLUME (IF ANY)
ENDDSN EQU *         END OF DATASET PROCESSING
B    READNEXT        GO TO GET NEXT RECORD
DROP R3
SPACE 3

*
MDBERROR EQU *
LR   R5,R15
CVD  R5,DOUBLE
UNPK MSGRC(2),DOUBLE+6(2)
OI   MSGRC+1,X'F0'

*
L    R5,RESCODE
CVD  R5,DOUBLE
UNPK MSGRSN(5),DOUBLE+5(3)
OI   MSGRSN+4,X'F0'

*
IOAEDIT 'CTT200S _ FAILED FOR DAMDB DATASET.',(MSGOP,8),
      INTO=MDBERR1+8,REGSAVE=YES,MF=(E,WORKEDIT)
MDBERR1 WTO 'CTT200S OPER FAILED FOR DAMDB DATASET.'
*
IOAEDIT '          RC=_, REASON=_',(MSGRC,2,MSGRSN,5), ,

```


Section 7: Customizing and Administering CONTROL-T

```

                INTO=MDBERR2+8, REGSAVE=YES, MF=(E, WORKEDIT)
MDBERR2  WTO:   '          RC=XX,  REASON=YYYYY'
*
        B:      RET8
OUTERROR WTO:   'UNABLE TO OPEN OUTPUT FILE'
        B:      RET8
LTCTFAIL WTO:   'TCT LOAD FAILED'
        B:      RET8
MDBEMPTY WTO:   'MDB IS EMPTY'
        B:      RET4
EODAD    EQU:   *
        MYPUT RPTVOL, EOFMSG
        MYPUT RPTDSN, EOFMSG
        CLOSE (RPTVOL,,RPTDSN)  CLOSE OUTPUT FILES
        CTTIOS ANY,CLOSE,
*
        RC=RESCODE,
*
        MF=(E, IOSLIST)
CALL     CTTTLD, (DELETE, TCTADDR, 0, 0)  FREE TCT AREA
        B:      RET0
RET8     LA:    R15, 8
        B:      ENDPGM
RET4     LA:    R15, 4
        B:      ENDPGM
RET0     SLR:   R15, R15
        B:      ENDPGM
ENDPGM   EQU:   *
        SPACE 3
        BRTRN (15)
        EJECT
***
* CONSTANTS
***
RBA0     DC:    XL3'000100'          RBA OF FIRST RECORD IN DATABASE
$ENV     DC:    CL4'SAM1'
$DBG     DC:    X'00'
LOAD     DC:    CL8'LOAD'
DELETE   DC:    CL8'DELETE'
PATTERN  DC:    X'402120202020'
        SPACE 3
***
* DATA AREAS
***
AREA     DS:    CL(DVLLN)  RECORD BUFFER (BOTH VOL AND D.S.)
RECRBA   DS:    XL3          RBA OF RECORD

```

Section 7: Customizing and Administering CONTROL-T

```

RESCODE DS F REASON CODE OF PROGRAM
DLSKEY DS XL(DLXLEN) D.S. L-TYPE INDEX KEY
DVXKEY DS XL(DVXLEN) VOL. INDEX RECORD
TCTADDR DS A ADDRESS OF THE TCT
DOUBLE DS D
MSGRC DS CL6 RC FROM CTTIOS FOR CTT200S
MSGRSN DS CL6 REASON CODE FOR CTT200S
MSGOP DS CL8 OPERATION FOR CTT200S
DS 0F
WORKEDIT DS CL256 OPERATION FOR CTT200S
SPACE 3
***
* OUTPUT MESSAGES
***
OUTDSN DS CL80 DATASET OUTPUT RECORD
ORG OUTDSN
DC CL6' DSN= ' RECORD PREFIX
DSNAME DS CL(L'DSDSN) DATASET NAME
DC CL(80+OUTDSN-*)' ' FILLER
ORG
OUTVOL DS CL80 VOLUME OUTPUT RECORD
ORG OUTVOL
DC CL6' VOL= ' RECORD PREFIX
VOLUME DS CL(L'DVLVOLSR) VOLUME SERIAL
DC CL(80+OUTVOL-*)' ' FILLER
ORG
VOLMSG DC CL80' *** VOLUME REPORT ***'
DSNMSG DC CL80' *** DATASET REPORT ***'
EOFMSG DC CL80' *** END OF MDB WAS REACHED *** '
SCRIND DC CL44' *** THIS IS A SCRATCH VOLUME ***'
SPACE 3
***
* CTTIOS IN LIST FORM
***
IOSLIST CTTIOS MF=L
SPACE 3
***
* DATA CONTROL BLOCKS
***
RPTVOL DCB DDNAME=DARPTVOL,MACRF=PM,LRECL=81,RECFM=FBA, *
DSORG=PS
RPTDSN DCB DDNAME=DARPTDSN,MACRF=PM,LRECL=81,RECFM=FBA, *
DSORG=PS
END

```

High Level API

Calling programs that access the CONTROL-T Media Database via Basic Level API must load a CONTROL-T Control Table and allocate the Media Database components. High Level API performs these actions automatically.

High Level API uses the Real-Time Environment TCT and the Real-Time Environment's Media Database components.

As a result:

- High Level API cannot be used when the CONTROL-T Real-Time Environment is not active.
- Calling programs intended for use with High Level API have to be APF authorized.

CONTROL-T High Level API is implemented by two Assembler macros:

CTTACCDB Accesses the Media Database.
CTTCHKDB Checks the last access of the Media Database (via macro CTTACCDB).

The following different functions can be performed by High Level API via macro CTTACCDB:

START Initializes the High Level API. Access of the Media Database must begin with this function.
READVOL Reads a requested volume record from the Media Database.
ADDVOL Adds a volume record to the Media Database.
DELVOL Deletes a volume record from the Media Database.
UPDTVOL Updates a volume record in the Media Database.
END Terminates the High Level API functions. Access of the Media Database must end with this function.

After each run of macro CTTACCDB, it is highly recommended to check the completion code of the access to the Media Database via macro CTTCHKDB. This macro tests the return code from the API and transfers control to an error handling routine which is written by the user as part of the calling program.

Macro CTTACCDB

As explained previously, High Level API access to the Media Database is performed via macro CTTACCDB.

Format

Instructions for macro CTTACCDB are formatted as follows:

```
label       CTTACCDB       function-code,  
                          R EC=buffer-address,  
                          VOL=volser,  
                          PARMS=parms-work-area
```

Section 7: Customizing and Administering CONTROL-T

Parameters

function-code	Function requested from macro CTTACCDB. Mandatory. This parameter must always be included as the first parameter. Valid values are: START Perform initialization functions required to enable access to the Media Database of CONTROL-T (e.g., gain access to the Real-Time Environment TCT, and open the Real-Time Environment Media Database). END Perform cleanup functions after using High Level API (e.g., close the Media Database). READVOL Read a volume record from the Media Database. A buffer-address, a volser, and a parms-work-area must be specified with this function code. ADDVOL Add a volume record to the Media Database. A buffer-address and a parms-work-area must be specified with this function code. DELVOL Delete a volume record from the Media Database. A volser and a parms-work-area must be specified with this function code. UPDTVOL Update a volume record in the Media Database. A buffer-address and a parms-work-area must be specified with this function code.
buffer-address	Buffer used by macro CTTACCDB for either storing data retrieved from the Media Database, or retrieving data to be inserted to the Media Database. This variable should point to an area defined as follows: REC DS (DVLEN)X This parameter is mandatory when function-code READVOL, ADDVOL or UPDTVOL is specified.
volser	Volume serial number. This variable should point to an area six characters in length. This variable is mandatory when function-code READVOL or DELVOL is specified.
parms work-area	Macro invocation work area. Mandatory. This variable should point to an area defined as follows: ADBPARMS DS (ADBPLEN)X

All invocations of CTTACCDB must access the same parameter area. Therefore, you must use the same parms-work-area in all calls to macro CTTACCDB.

Macro CTTACCDB must be coded in accordance with all standard macro Assembler coding restrictions.

Values in general purpose registers 0, 1, 14, and 15 are modified by macro CTTACCDB.

Macro CTTACCDB accesses a set of variables generated by macro CTTADBP and by macro CTTDBTP. Therefore, the source of the calling program should include both these macros.

Section 7: Customizing and Administering CONTROL-T

Macro CTTCHKDB

This macro is used to check return codes resulting from access of the Media Database via macro CTTACCDB.

Format

Instructions for macro CTTCHKDB are formatted as follows:

```
label      CTTCHKDB  ERR=error-routine,  
                    PARS=work-area
```

Where:

error-routine Points to a subroutine in the source code of the calling program that handles errors in accessing the Media Database. For more information see "Media Database Error Handling Routine" below. Mandatory.

work-area Invocation work area of the last CTTACCDB macro. Mandatory.

The value for this field must be the same work area specified in the parameters for CTTACCDB.

Macro CTTCHKDB must be coded in accordance with all standard macro Assembler coding restrictions.

Values in general purpose registers 0, 1 and 15 are modified by macro CTTCHKDB.

Media Database Error Handling Routine

The calling program must supply an error handling routine. This error handling routine is given control whenever macro CTTCHKDB determines that the last access to the Media Database was unsuccessful (e.g., the requested volume was not found).

High Level API accesses a set of variables generated by macro CTTADBP. Some of these variables contain relevant information on the Media Database error handling routine. The variables are located in the area specified via parameter PARS to macro CTTACCDB, and are mapped by macro CTTADBP. Below is a list of these relevant variables:

The High Level API invokes macro CTTIOS to access the Media Database. For more information see the description of macro CTTIOS earlier in this section.

ADBPRC	Return code from the last High Level API function. Full word. Possible values are:
0	OK
8	Invalid function requested
12	CONTROL-T TCT not found (i.e., CONTROL-T not active)
16	A CTTIOS command failed (see parameters ADBPIRC and ADBPIRSN described below)
20	Internal error (CTTIOS handle not supplied)
24	Media Database record not supplied

Section 7: Customizing and Administering CONTROL-T

	28	Calling program is not APF authorized.
	32	Key (volser) not supplied.
	36	Function code is not allowed (denied by Exit CTTX006).
ADBPFUNC		Last CTTACCDB function. The field for this variable is eight characters long.
ADBPIRC		Return code from the last run of macro CTTIOS which was invoked by the High Level API. Full word. (For more details on the return codes of macro CTTIOS see the description of message CTT200S in the IOA Messages and Codes Manual.)
ADBPIRSN		Reason code returned from the last run of macro CTTIOS invoked by the High Level API. Full word. (For more details on the reason codes of the macro CTTIOS see the description of message CTT200S in the IOA Messages and Codes Manual.)
ADBPIOPR		Last CTTIOS function issued by the High Level interface. This variable is contained in an 8-character field.

Example for High Level API

This sample access program reads the volume record of volume V00001 from the Media Database using High Level API, and issues a message about the status of this volume (i.e., SCRATCH, ACTIVE, or NOT FOUND IN MDB). This sample also includes an example for the Media Database error handling routine.

Note The code shown below is just an example and should not be used without appropriate modification.

```
*****
*
*                               CTTSAM2
*
*
* FUNCTION:  SAMPLE PROGRAM TO DEMONSTRATE ACCESS TO THE CONTROL-T
*           MEDIA DATABASE (MDB) USING THE HIGH LEVEL API.
*           THIS PROGRAM READS THE MDB VOLUME RECORD OF A CERTAIN
*           VOLUME, (IN THIS SAMPLE: V00001) AND ISSUES A MESSAGE
*           ABOUT THE VOLUME'S STATUS: ACTIVE OR SCRATCH.
*           IN THIS CODE YOU WILL ALSO FIND AN EXAMPLE FOR AN
*           MDB ERROR HANDLING ROUTINE THAT SHOULD BE CODED BY
*           ANY CALLING PROGRAM THAT USES THE HIGH LEVEL API.
*
* DD CARDS:  NONE (THE HIGH LEVEL API USES THE CONTROL-T REAL
*               TIME ENVIRONMENT FILES).
*
```

Section 7: Customizing and Administering CONTROL-T

```
*  DISCLAIMER:  THIS SAMPLE PROGRAM IS PROVIDED ON AN AS IS BASIS,  *
*                WITHOUT ANY WARRANTY,  EITHER EXPRESS OR IMPLIED  *
*
*  REGISTERS:  R13    -  BASE  *
*                R4    -  MDB VOLUME RECORD  (MAPPED BY CTTDVL)  *
*                R9    -  CTTACDDB PARS  (MAPPED BY CTTADBP)  *
*
*  ATTRIBUTES:  AMODE 31  *
*                RMODE 24  *
*
*  TO COMPILE AND LINK THE PROGRAM :  *
*
*  //CTTSAM2 JOB ...  *
*  //ASM      EXEC CTTASM  *
*  //C.SYSIN DD DSN=IOA.V500.SAMPLE(CTTSAM2)  *
*  //L.SYSLMOD DD DISP=SHR,DSN=YOUR.LOAD.LIBRARY  *
*  //L.SYSIN  DD *  *
*  MODE RMODE(24),AMODE(31)  *
*  NAME CTTSAM2(R)  *
*  /**  *
*  //RUN EXEC PGM=CTTSAM2,COND=(0,NE),REGION=60M  *
*  //STEPLIB DD DISP=SHR,DSN=YOUR.LOAD.LIBRARY  *
*  //DAMDB DD DISP=SHR,DSN=CTT.V500.MDED  *
*  //DAMDI DD DISP=SHR,DSN=CTT.V500.MDBI  *
*  //DATRC DD DISP=SHR,DSN=CTT.V500.TRC  *
*  //DARPTVOL DD SYSOUT=*  *
*  //DARPTDSN DD SYSOUT=*  *
*  //  *
*
*****
```

Section 7: Customizing and Administering CONTROL-T

```

EJECT ,
CTTSAM2 CSECT                                HIGH LEVEL API SAMPLE
BEGIN *,EQR=YES
SPACE 1
LA     R9,ADBPAMS                            R9--> CTTADB PAMS
USING ADBP,R9
XC     FLAG,FLAG                             INIT INTERNAL FLAG
SPACE 1
*****
*  PERFORM INTIALIZATION REQUIRED VIA HIGH LEVEL API: START COMMAND *
*****
SPACE 1
CTTACCDB START,PARMS=ADBPAMS                 INITIALIZATION
CTTCHKDB ERR=ADBEROR,PARMS=ADBPAMS          CHECK START OPERATION
SPACE 1
OI     FLAG,$OPENED                          MARK:  MDB IS OPENED
SPACE 1
*****
*  READ MDB VOLUME RECORD OF VOLUME:  V00001 VIA READVOL COMMAND  *
*****
SPACE 1
MVC   VOL(6),=CL6'V00001'  VOLSER TO READ
SPACE 1
CTTACCDB READVOL,REC=VOLREC,VOL=VOL,PARMS=ADBPAMS
CTTCHKDB ERR=RVOLEERR,PARMS=ADBPAMS
SPACE 1
NI     FLAG,X'FF'-$OPENED                    MARK:  MDB IS NOT OPENED
SPACE 1
*****
*  PERFORM CLEANUP REQUIRED VIA HIGH LEVEL API:  END COMMAND      *
*****
SPACE 1
CTTACCDB END,PARMS=ADBPAMS
CTTCHKDB ERR=ADBEROR,PARMS=ADBPAMS
SPACE 1
*****
*  ISSUE A MESSAGE ACCORDING TO VOLUME'S STATUS:  ACTIVE OR SCRATCH *
*****
SPACE 1
LA     R4,VOLREC                             R4-->MDB VOL RECORD
USING DVL,R4
TM     DVLSTAT,DVLSACT                       AN ACTIVE VOLUME ?
BNO   WTOSCR ..N,  GO ISSUE:  SCRATCH
SPACE 1
WTO   'VOLUME V00001 IS ACTIVE'  VOLUME IS ACTIVE

```


Section 7: Customizing and Administering CONTROL-T

```

      B      EXIT
      SPACE 1
WTOSCR DS   0H                VOLUME IS SCRATCH
      WTO   'VOLUME V00001 IS SCRATCH'
      SPACE 1
      B      EXIT
      DROP  R4                WAS: MDB VOL RECORD
      SPACE 1
*****
*   MDB ERROR HANDLING ROUTINE - FOR READVOL FUNCTION   *
*****
      SPACE 1
RVOLERR DS   0H                READVOL ERROR
      L     R15,ADBPRC        R15 - CTTADB RC
      C     R15,=F'16'        A CTTIOS FAILURE ?
      BNE   ADBERROR ..N, GO ISSUE ERROR MSG
      SPACE 1
      L     R15,ADBPIRC        R15 - RC OF CTTIOS
      C     R15,=F'4'         VOLUME NOT FOUND ?
      BNE   ADBERROR ..N, GO ISSUE ERROR MSG
      SPACE 1
      WTO   'VOLUME V00001 NOT FOUND IN MEDIA DATABASE'
      SPACE 1
      NI    FLAG,X'FF'-$OPENED
      CTTACDDB END,PARMS=ADBPAMS  TRY TO PERFORM CLEANUP
      CTTCHKDB ERR=ADBERROR,PARMS=ADBPAMS
      SPACE 1
      B      EXIT                NO OTHER UPDATES
      SPACE 1
*****
*   MDB ERROR HANDLING ROUTINE - FOR START/END FUNCTIONS *
*****
ADBERROR DS   0H
      MVC   MSG1FUNC(8),ADBPFUNC  CTTACDDB FUNCTION FOR MESSAGE
      MVC   MSG1VOL(6),=CL6'V00001' VOLSER FOR MESSAGE
      L     R3,ADBPRC            CONVERT
      CVD   R3,DW ..CTTACDDB
      UNPK  MSG1RC(3),DW+6(2) ..RETURN-CODE
      OI    MSG1RC+2,X'F0' ..FOR MESSAGE
      SPACE 1
      MVC   WTO1+10(MSG1LEN),MSG1  MOVE MESSAGE TO BE WTO'ED
WTO1    WTO   '
                                           ',ROUTCDE=11
      MVC   MSG2FUNC(8),ADBPPIOPR LAST CTTIOS FUNCTION FOR MSG
      L     R3,ADBPIRC            CONVERT

```

Section 7: Customizing and Administering CONTROL-T

```

CVD   R3,DW ..LAST CTTIOS
UNPK  MSG2RC(3),DW+6(2) ..   RETURN-CODE:
OI    MSG2RC+2,X'F0' ..     FOR MESSAGE:
SPACE 1
L     R3,ADBPIRS          CONVERT
CVD   R3,DW ..LAST CTTIOS
UNPK  MSG2RSN(5),DW+5(3) ..  REASON-CODE:
OI    MSG2RSN+4,X'F0' ..    FOR MESSAGE:
SPACE 1
MVC   WTO2+10(MSG2LEN),MSG2  MOVE MESSAGE TO BE WTO'ED
WTO2  WTO   '
                                           +
                                           ',ROUTCDE=11
SPACE 1
B     EXIT
SPACE 1
EXIT  DS    0H
TM    FLAG,$OPENED          IS MDB STILL OPENED ?
BNO   SKIPEND ..N,         SKIP END
SPACE 1
CTTACDDB END,PARMS=ADBPAMS
SPACE 1
SKIPEND DS    0H
BRTRN 0
SPACE 1
*****
*   WORK AREAS
*****
DW     DS    D              DOUBLE WORD:
*
FLAG   DS    X              INTERNAL FLAG
$OPENED EQU  X'80'         MARK:  MDB IS OPENED
*
MSG1   DS    0C              1ST ERROR MESSAGE
      DC    C'CTTACDDB FUNCTION: '
MSG1FUNC DS    CL8
      DC    C' FOR VOLUME: 'MSG1VOL DS    CL6
      DC    C' FAILED.  RC: '
MSG1RC  DS    CL3
MSG1LEN EQU  *-MSG1
*
MSG2   DS    0C              2ND ERROR MESSAGE
      DC    C'LAST CTTIOS FUNCTION: '
MSG2FUNC DS    CL8
      DC    C' RC: '

```

Section 7: Customizing and Administering CONTROL-T

```
MSG2RC DS CL3
      DC C' REASON: '
MSG2RSN DS CL5
MSG2LEN EQU *-MSG2
*
VOL DS CL6 VOLUME SERIAL NUMEBR TO READ
VOLREC DS (DVLEN)X MDB RECORD BUFFER
*
ADBP ARMS DS (ADBPEN)X CTTACDDB WORK P ARMS
*****
* MAPPING DSECTS: *
*****
      CTTDBTP , CTTIOS P ARMS
      CTTADBP , CTTACDDB P ARMS
      CTTDVL , MDB VOLUME RECORD MAP
END
```

Section 8: Maintaining IOA Products

Introduction	8-1
IOA Release 5.0.0 Packaging	8-1
Product Packaging.....	8-1
Language Support.....	8-2
Applying Periodic Maintenance.....	8-3
Preliminary Steps.....	8-3
Installation Steps	8-3
Step 1: Apply Periodic Maintenance	8-3
Applying Ad Hoc Maintenance	8-4
Introduction	8-4
Installation Steps	8-4
Step 1: Run SMP/E to RECEIVE, APPLY CHECK and APPLY the Fix.....	8-4
Step 2: Customize and Propagate Fix Elements to Working Libraries	8-5
Step 3: Enable Fixes Applied to ICE.....	8-10
Step 4: Copy JCL Procedures and ISPF Elements to System Libraries (Optional).....	8-11



Introduction

In IOA Release 5.0.0, maintenance is performed through SMP/E only, via PTFs containing object replacement elements.

The following paragraphs present a technical overview of the IOA Release 5.0.0 installation in order to facilitate the understanding of the maintenance processes.

IOA Release 5.0.0 Packaging

IOA Release 5.0.0 is packaged as a pre-installed SMP/E environment containing three SMP/E zones: GLOBAL, IOA5TZN (Target zone), and IOA5DZN (Distribution zone).

IOA Release 5.0.0 supports use of any one of the following installation modes for the SMP/E CSI:

- A new CSI file loaded with the GLOBAL, IOA5TZN and IOA5DZN zones.
- An existing global zone for the IOA GLOBAL zone and two new CSI files, one for the IOA5TZN zone and one for the IOA5DZN zone.
- An existing CSI file for all three zones. The IOA GLOBAL zone is added to the existing global zone, while the IOA5TZN and IOA5DZN zones form separate entities in the file.

Product Packaging

The products in IOA Release 5.0.0 are packaged as different functional components as listed in the following table. IOA is the base component for all products.

Table 1: IOA Release 5.0.0 Functional Components

FMD	Product
IOA5000	IOA Release 5.0.0 Common Routines, Samples, etc.
IOA500C	IOA Release 5.0.0 C Language Modules
CTM5000	CONTROL-M Release 5.0.0 Modules and Samples
CTD5000	CONTROL-D Release 5.0.0 Modules and Samples
CTD500C	CONTROL-D Release 5.0.0 C Language Modules
CTR5000	CONTROL-R Release 5.0.0 Modules and Samples
CTO5000	CONTROL-O Release 5.0.0 Modules and Samples
CTB5000	CONTROL-B Release 5.0.0 Modules and Samples
CTB500C	CONTROL-B Release 5.0.0 C Language Modules
CTW5000	MVS Gateway for Enterprise Controlstation Release 5.0.0 Modules and Samples
CTW500C	MVS Gateway for Enterprise Controlstation Release 5.0.0 C Language Modules
CTV5000	CONTROL-V Release 5.0.0 Modules and Samples
CTT5000	CONTROL-T Release 5.0.0 Modules and Samples

Language Support

IOA Release 5.0.0 supports various languages. Program elements for each language are packaged as unique functions in IOA Release 5.0.0 as follows:

Table 2: Language Elements

FMID	Elements Contained
IOA500F	IOA Release 5.0.0 French Elements
IOA500G	IOA Release 5.0.0 German Elements
IOA500J	IOA Release 5.0.0 Japanese Elements

A different SMP/E environment exists for each language supported by IOA Release 5.0.0. Each environment contains the IOA Release 5.0.0 functions listed in Table 1 above, as well as the appropriate language function (applied and accepted). Each language function contains a LOAD library in which the load modules which support the language in question are stored. These load modules, as well as screen and message source members, are named according to the language they support (e.g., GEMMABC is the name of the message member for ABC).

This naming convention enables SMP/E to support the libraries of the basic English installation as well as the libraries of the language support installation. However, it prevents maintenance from identifying the modules that should be updated. The following example demonstrates aliases that are utilized to overcome this problem.

The example in the table below compares the linking of comparable English and German load modules (containing the same functionality). In the example, the English load module is linked into the IOA LOAD library while the German NLS load module is linked into the LOADGER library.

Table 3: Example

English	German	Comments
ORDER CTMABC	ORDER CTMABC	Non NLS MOD
ORDER CTMMABC	ORDER GEMMABC	Language sensitive MOD
INCLUDE (CTMABC)	INCLUDE (CTMABC)	
INCLUDE (CTMMABC)	INCLUDE (GEMMABC)	
NAME CTMABC(R)	ALIAS CTMABC	
	NAME GEMABC(R)	
LOAD	LOADGER	SYSLMOD name

Applying Periodic Maintenance

Periodic maintenance is applied via the Installation and Customization Engine (ICE).

Perform the following preliminary steps and installation steps to apply periodic maintenance.

Preliminary Steps

1. Load the maintenance tape files to the BASE.MAINT library.
2. Refer to the IOA Installation Guide for instructions on how to invoke and use ICE.
3. Invoke ICE and verify that "IOA" is specified in the "Product ID" field, and that "YES" is specified in the "Enforce Step Order" field.
4. Select option 4 (Maintenance) on the ICE menu. Major steps for installing maintenance are then displayed.
5. Apply the major steps listed below.

Installation Steps

Step 1 Apply Periodic Maintenance

Select major step 1 "Apply Periodic Maintenance." The Minor Steps Selection screen is displayed.

Step 1.1 Specify Maintenance Level

Select minor step 1 "Specify Maintenance Level".

In the Maintenance Level Specification screen, specify the maintenance to be applied in the format yynn, and specify YES in the "Do you want to reset the steps status?" field.

Exit the Maintenance Level Specification screen.

Mark this step as completed.

Step 1.2 View Maintenance Documentation (Optional)

If written documentation is available view it. This step is optional.

Follow the steps specified in the documentation to apply the maintenance.

Applying Ad Hoc Maintenance

Introduction

Fixes to IOA Release 5.0.0 are applied using SMP/E. (The Installation and Customization Engine is not used to apply Ad Hoc Maintenance.)

Once a fix is applied by SMP/E, the affected elements which reside in base libraries (e.g., source members, the JCL, samples) often contain unresolved installation variables and are likely to require further tailoring. These elements must be propagated (i.e., copied and tailored) to the working libraries. Other elements, such as load modules, are applied by SMP/E directly to the working libraries and do not require any further processing.

The following steps describe how to install the fix in the active environment.

Installation Steps

Step 1 Run SMP/E to RECEIVE, APPLY CHECK and APPLY the Fix

LOAD the fixes into your IOA MAINTLIB library. It is recommended that you use the same name for the member as the fix name.

Example

FPnnnnn

Where:

“p” Product ID

“nnnnn” Fix number (e.g., FI12345)

Run an SMP/E job using the supplied IOASMP procedure to:

- RECEIVE the fixes into your SMP/E CSI file which contains IOA Release 5.0.0.
- APPLY CHECK the fixes for detection of errors (e.g., missing prerequisite fixes).

Note Procedure *IOASMP* is used in examples throughout this section. However, if installation parameter *PROCPREFA* was changed in *ICE* from its supplied default, the procedure name should be modified accordingly. This also applies to other JCL procedures which are executed. Similarly, the target zone that is indicated in the sample job below is the default zone name *IOA5TZN*. If installation parameter *IOATZN* was modified in *ICE*, the modified value should be used instead.

The following is a sample JCL for an SMP/E RECEIVE and APPLY CHECK job:

```
//SMPJOB JOB  , 'your job-card', .....
//SMPSTEP  EXEC IOASMP
//SMPPTFIN DD  DISP=SHR, DSN=IOA.V500BASE.MAINTLIB(FI12345)
//          DD  DISP=SHR, DSN=IOA.V500BASE.MAINTLIB(FMnnnnn)
//SMPCTL   DD  *
          SET BDY(GLOBAL).
          RECEIVE SYSMODS LIST.
          SET BDY(IOA5TZN).
          APPLY S(FI12345, FMnnnnn, ..... ) CHECK.
/*
```

Before applying a fix, verify its SMP/E HOLD status. A fix may be in HOLD status due to an error or an action that must be performed. The output of the SMP/E RECEIVE SYSMODS LIST lists any HOLDDATA associated with the fixes that were received.

If the current fix being applied is in HOLD status, the HOLDDATA indicates the reason. If the reason is ACTION, a comment describes the steps to be taken prior to or following the APPLY of the fix. Ensure that these instructions are carefully followed.

If the fix is held, the APPLY CHECK fails and displays an informational message. If you have reviewed the HOLDDATA and know which actions are required, you can add the keyword BYPASS(HOLDSYS) to the APPLY CHECK statement and rerun it. If, however, the reason for the hold is ERROR, the fix should not be applied until a corrective fix is available.

The following is a sample SMP/E APPLY CHECK job with the bypass for the HOLD status:

```
//SMPJOB  JOB  , 'your job-card', .....
//SMPSTEP  EXEC IOASMP
//SMPCTL   DD  *
          SET BDY(IOA5TZN).
          APPLY S(FI12345) BYPASS(HOLDSYS) CHECK.
/*
```

When the APPLY CHECK completes successfully, the CHECK keyword should be removed and the APPEY should be run to install the fix in the IOA target environment.

Step 2 Customize and Propagate Fix Elements to Working Libraries

Perform this step only if a fix that was applied in Step 1 is on HOLD with an ACTION reason that requires tailoring and propagation.

Applicable fixes are those which modify elements (PDS members) in base libraries, which may require file tailoring and are likely to contain unresolved variables (e.g., parameter ILPREFA). Two IOA utilities, IOAFTR and IOAINS, are run for these fix elements. Note that fixes to load modules are always applied directly to the active environment.

Section 8: Maintaining IOA Products

During propagation, all elements applied by a single SMP/E APPLY command should be propagated at the same time to verify that all prerequisite and corequisite fixes are applied to the working environments. To determine which elements require propagation, and the appropriate action, review the Element Summary report in the SMPRPT dataset (produced by SMP/E APPLY) and follow the rules below:

Elements that meet any of the following criteria do not require propagation:

- Elements with an entry in the LMOD SYSLIB column, except user and security exits.
- Elements with one of the following listed in the SYSLIB library column: MAC, MSG, PANEL, ISMSG, CICSSAMP, SAMPREPS, BANNERS, CAL.
- Elements that are part of ICE (i.e., element whose names beginning with IOAI and that have a SYSLIB library specification of INSTALL).
- Elements with a SYSLIB library of GENERAL. For these elements, a different procedure, described later in this section, is required.

The following sample Element Summary report sample shows two elements that require propagation: CONTROLM (in the IOA PROCLIB library), and \$PROFFLD (in the IOA PARM library). Elements CTDSEC, CTMCUSI and IOAHLP do not require propagation as they all have entries listed in the LMOD SYSLIB column. \$\$FRM does not require propagation as its SYSLIB library is MSG, and IOAICHLP is part of ICE (i.e., starts with IOAI and has INSTALL listed in the SYSLIB library column).

ELEMENT SUMMARY REPORT FOR APPLY CHECK PROCESSING									
ELEMENT SYSMOD	ELEMENT SYSMOD	ELEMENT STATUS	CURRENT FMID	CURRENT RMID	DISTLIB LIBRARY	SYSLIB LIBRARY	ASSEM NAMES	LOAD MODULE	LMOD SYSLIB
MAC APPLIED	CONTROLM	APPLIED	IOA5000	FI00075	APROCLIB	PROCLIB			FI00075
SRC FI00071	CTDSDEC APPLIED	APPLIED	IOA5000	FI00071	AMSG	MSG	CTDSDEC	CTDTAMS	LOAD
MOD FI00072	CTMCUSI APPLIED	APPLIED	IOA5000	FI00072	AIOALOAD			CTMCINT	LOAD
MOD FI00030	IOAHLP APPLIED	APPLIED	IOA5000	FI00030	AIOALOAD			CTMCTSO IOAHLP	LOAD INSL
HELPEP APPLIED	\$\$FRM	APPLIED	IOA5000	FI00073	AMSG	MSG			FI00073
PARM APPLIED	\$PROFFLD	APPLIED	IOA5000	FI00083	APARM	PARM			FI00083
SAMP APPLIED	IOAICHLP	APPLIED	IOA5000	FI00030	AINSTALL	INSTALL			FI00030

To propagate the required elements, edit member IOAPROPJ in the IOA INSTALL library. This member copies and tailors the members that were updated or replaced by the fix from the base libraries into the environment libraries.

The default IOAPROPJ member is presented below:

```
//MYJOB JOB 'my job card',....
/*=====
/* PERFORM:
/* CHANGE '????????' 'member-name' ALL
/* CHANGE '???????' 'library suffix' ALL
/*=====
//PROPI EXEC IOAPROP,
//          BASEPRF=IOA500.BASE,          <===== Specify YOUR BASEPRF
//          ILPREFA=IOA500.TESTI         <===== Specify YOUR ILPREFA
//IOAFTR.SYSIN DD *
COPY INDD=IG??????,OUTDD=OG??????      <===== Input cards to IOAFTR
SELECT MEM=????????,REPLACE=YES <===== Input cards to IOAFTR
/*
//IOAINS.DALIBS DD *
*---+---1---+---2---+---3---+---4---+
???????? IOA500.TESTI.??????          <===== Input cards to IOAINS
/*
```

Change the member name that was changed or replaced and the last qualifier (suffix) of the library in accordance with the value listed in the SYSLIB library column for that element. For example, if member CONTROLD was replaced in the PROCLIB library, the JCL will appear as follows:

```
//MYJOB JOB 'my job card',....
/*=====
/* PERFORM:
/* CHANGE '????????' 'member-name' ALL
/* CHANGE '???????' 'library suffix' ALL
/*=====
//PROPI EXEC IOAPROP,
//          BASEPRF=IOA500.BASE,          <===== Specify YOUR BASEPRF
//          ILPREFA=IOA500.TESTI         <===== Specify YOUR ILPREFA
//IOAFTR.SYSIN DD *
COPY INDD=IPROCLIB,OUTDD=OPROCLIB      <===== Input cards to IOAFTR
SELECT MEM=CONTROLD,REPLACE=YES <===== Input cards to IOAFTR
/*
//IOAINS.DALIBS DD *
*---+---1---+---2---+---3---+---4---+
CONTROLD IOA500.TESTI.PROCLIB          <===== Input cards to IOAINS
/*
```

Section 8: Maintaining IOA Products

If more than one member or more than one library is changed by the fix, you can repeat the control cards of both utilities as necessary. For example, the following is a sample job that propagates members CONTROLM and CONTROLD in the PROCLIB library, and member \$PROFFLD in the PARM library:

```
//MYJOB JOB 'my job card',....
/*=====
/* PERFORM:
/*  CHANGE '????????' 'member-name' ALL
/*  CHANGE '#####' 'library suffix' ALL
/*=====
//PROP1 EXEC IOAPROP,
//          BASEPRF=IOA500.BASE,      <===== Specify YOUR BASEPRF
//          ILPREFA=IOA500.TESTI     <===== Specify YOUR ILPREFA
//IOAFTR.SYSIN DD *
COPY INDD=IPROCLIB,OUTDD=OPROCLIB  <===== Input cards to IOAFTR
SELECT MEM=CONTROLM,REPLACE=YES <===== Input cards to IOAFTR
SELECT MEM=CONTRULD,REPLACE=YES <===== Input cards to IOAFTR
COPY INDD=IPARM,OUTDD=OPARM        <===== Input cards to IOAFTR
SELECT MEM=$PROFFLD,REPLACE=YES <===== Input cards to IOAFTR
/*
//IOAINS.DALIBS DD *
*---+---1---+---2---+---3---+---4---+
CONTROLM IOA500.TESTI.PROCLIB      <===== Input cards to IOAINS
CONTRULD IOA500.TESTI.PROCLIB      <===== Input cards to IOAINS
$PROFFLD IOA500.TESTI.PARM         <===== Input cards to IOAINS
/*
```

After performing the necessary customization, submit the job. All steps must end with a condition code of 0.

Processing Elements with a SYSLIB Library of GENERAL

Elements that have a SYSLIB library of GENERAL should be processed according to the following rules:

- Elements belonging to non-installed products require no further processing.
- Elements belonging to installed products must be copied manually to the product library as directed by the HOLD action of SMP/E. The element may contain unresolved installation variables. Therefore, utility IOAINS should be run to resolve them. To run utility IOAINS, modify job IOAINSJ in the IOA INSTALL library (as described below).

The default IOAINSJ member is presented below:

```
//MYJOB JOB 'my job card',...
//*****
/** PLEASE MAKE THE FOLLOWING CHANGES: *
/** CHANGE '????????' 'MEMBER-NAME' ALL *
/** CHANGE '@@@@@@' 'FULLY-QUALIFIED-LIBRARY-NAME' *
/** CHANGE '&&&' '3-CHARACTER-PRODUCT-ID' ALL (E.G. 'CTM') *
/** CHANGE '#' '1-CHARACTER-PRODUCT-ID' (E.G. 'M') *
//*****
//CTMINST EXEC PGM=IOAINS,PARM='&&&', <=== Check Parm
// REGION=5120K,TIME=30
//STEPLIB DD DISP=SHR,DSN=%BASEPREF%.INSLOAD
//SYSPRINT DD SYSOUT=*
//PRTDBG DD DUMMY
//SYSUDUMP DD SYSOUT=*
//DAREP DD DISP=SHR,DSN=&ILPREF%.INSTALL(DEFPARMA)
// DD DISP=SHR,DSN=&ILPREF%.INST&&&(DEFPARM#) <=== Check library/member
//DALIBS DD
* IOAINS INPUT STATEMENTS
*---+---1---+---2---+---3---+---4---+
???????? @@@@@@ <=== IOAINS CONTROL CARDS
/*
```

Change the member as follows:

- Replace ??????? with the member name from the ELEMENT NAME column.
- Replace @@@@@@ with the fully-qualified dataset name. The SMP/E HOLD data will contain the library suffix.
- Replace &&& with the three-character product ID from the SMP/E HOLD data to specify the value in the PARM field and the name of the product installation library.
- Replace # with the one-character product ID from the SMP/E HOLD data to resolve the name of the DEFPARMx member in the product installation library.
- If the product is CONTROL-R, repeat the second DAREP line and modify it so that it points to member DEFPARMM in the IOA INSTCTM library.
- If the product is CONTROL-V, repeat the second DAREP line and modify it so that it points to member DEFPARMD in the IOA INSTCTD library.

IOAINS control cards may be repeated in the same IOAINS step for elements belonging to the same product.

Section 8: Maintaining IOA Products

The following sample shows IOAINSJ tailored to process this member:

```
//MYJOB JOB 'my job card',....
//*****
//* PLEASE MAKE THE FOLLOWING CHANGES: *
//* CHANGE '????????' 'MEMBER-NAME' ALL *
//* CHANGE 'GGGGGGGG' 'FULLY-QUALIFIED-LIBRARY-NAME' *
//* CHANGE '&&&' '3-CHARACTER-PRODUCT-ID' ALL (E.G. 'CTM') *
//* CHANGE '#' '1-CHARACTER-PRODUCT-ID' (E.G. 'M') *
//*****
//CTMINST EXEC PGM=IOAINS, PARM='CTR', <=== Check Parm
// REGION=5120K, TIME=30
//STEPLIB DD DISP=SHR, DSN=your.basepref.INSLOAD
//SYSPRINT DD SYSOUT=*
//PRTDBG DD DUMMY
//SYSUDUMP DD SYSOUT=*
//DAREP DD DISP=SHR, DSN=your.ilprefa.INSTALL(DEFPARMA)
// DD DISP=SHR, DSN=your.ilprefa.INSTCTR(DEFPARMR) <== for CONTROL-R
// DD DISP=SHR, DSN=your.ilprefa.INSTCTM(DEFPARMM) <== for CONTROL-M
//DALIBS DD
* IOAINS INPUT STATEMENTS
*---+---1---+---2---+---3---+---4---+
CTRCNPJ your.ilprefr.JCL <=== INPUT CARDS TO IOAINS
/*
```

Submit the job. All steps must end with a condition code of 0.

Step 3 Enable Fixes Applied to ICE

If a fix results in the replacement of ICE tables, the new copy of the table must be merged with all values previously entered during installation, customization, etc.

Perform this step only if the fix was held in SMP/E with reason ACTION, and the hold comment refers specifically to enabling ICE fixes.

After applying the fix, enter ICE and choose the Housekeeping activity from the ICE Main menu. Select "Refresh Tables" to refresh the ICE elements that were replaced by the fix.

Step 4 Copy JCL Procedures and ISPF Elements to System Libraries (Optional)

If a fix replaces JCL procedures, CLISTs or ISPF elements which were copied to system libraries during installation, these elements need to be recopied.

Perform this step only if the fix was held with reason ACTION, and the hold comment specifically refers to copying JCL procedures and ISPF elements to system libraries.

Skip this step if JCL procedures and ISPF elements are not copied during installation to your system libraries.

If you chose to copy JCL procedures to your system PROCLIB (i.e., installation parameter PROCLIB contains a value other than DONTCOPY), any updated procedures should be recopied. If the value of installation parameter PROCPREFA was modified and does not use the default value of IOA, the procedures should be copied with a new name that matches the value specified for PROCPREFA.

If you choose to copy CLISTs or ISPF messages, panels, skeletons or tables to your system libraries, (i.e., a value other than DONTCOPY was specified for installation parameters CLIST, ISPFMSG, ISPFPNL, ISPFSKL, or ISPFTBL), updated elements should be recopied.



Section 9: Utilities

General	9-1
Summary of Utilities	9-1
IOA Utilities	9-1
CONTROL-M Utilities.....	9-2
CONTROL-D/V Utilities.....	9-2
CONTROL-O Utilities	9-3
CONTROL-B Utilities.....	9-4
CONTROL-T Utilities	9-5
IOA Utilities	9-7
IOABLCAL - Build Calendars.....	9-7
IOACLRES - Clean IOA Conditions/Resources File	9-13
IOACND - Perform Condition/Resource Adjustments.....	9-16
IOACOPRS - Copy/Resize IOA Conditions/Resources File	9-20
IOACPLOG - Copy the IOA Log File	9-22
IOADBF - Allocate/Format an IOA Access Method File.....	9-24
DD Statement DAFILE	9-24
IOA Access Method File Definition Statements.....	9-25
IOADBSR - Sort Records in an IOA Access Method Data Component.....	9-31
IOADCPY - Recover an IOA Access Method File.....	9-33
IOADIG - Verify Integrity of an IOA Access Method File Data Component.....	9-35
IOADII - Verify Integrity of an IOA Access Method File Index Component.....	9-36
IOADLD - Load IOA Access Method File Contents	9-37
IOADPT - Print IOA Access Method File Content.....	9-38
IOADUL - Unload IOA Access Method File Contents	9-39
IOALDNRS - Load Manual Conditions File.....	9-41
IOANOTE - Write a Message to the IOA Log File.....	9-44
IOATEST - Simulate Production Environment Events.....	9-45
IOAVERFY - Verify IOA File Installation and Status	9-47
CONTROL-M Utilities	9-49
CTMBGRP - Convert Regular Scheduling Tables to Group Scheduling Tables	9-49
CTMCAJF - Maintain the Active Jobs File	9-51
CTMJSA - Job Statistics Accumulation.....	9-55
CONTROL-B Utilities	9-56
CTBABI - Initialize Active Balancing File	9-56
CTBCMP - Compile Rule Definitions	9-57
CTBDBIB - Rebuild a CONTROL-B Index File	9-58

CTBDEVCG - Change Number of Variable Generations	9-60
CTBDEVCP - Copy Variables or Groups to Another File	9-62
CTBDEVDL - Delete Variables/Groups from Variable/Group Files	9-65
CTBDEVIG - Check the Integrity of Variable/Group Files	9-67
CTBFRM - Compress the Active Balancing File	9-68
CTBJAFCP - Copies Entries in Rule Activity/Report Files	9-69
CTBJAFDL - Deletes Entries From Rule Activity/Report Files	9-71
CTBJAFIG - Check the Integrity of Rule Activity/Report Files	9-73
CONTROL-D/V Utilities	9-74
CTDCAMF - Compress Active Missions File	9-74
Compressing the Active Missions File	9-74
Changing the Size of the Active Missions File	9-74
CTDCATF - Compress Active Transfer File	9-76
Compressing the Active Transfer File	9-76
Changing the Size of the Active Transfer File	9-76
CTDCA2P - Copy Active User Report File to Permanent	9-78
CTDCCU - Delete Unneeded CDAM Files	9-80
CTDCLHIS - Clean History User Report List File	9-82
MODE Parameter	9-82
CTDCP2A - Copy Permanent User Report File to Active	9-83
CTDDELRP - Delete Reports Which Are Not Needed	9-85
Parameters	9-85
DAYS Statement	9-86
INCLUDE/EXCLUDE Statements	9-87
CTDDELRP Deletion Rules	9-88
Examples	9-89
CTDDIB - Rebuild Index Component of a User Report List File	9-91
CTDUPBKP - Update Backup Details	9-92
Selection Parameters	9-93
CHANGETO Parameters	9-94
Report Parameters	9-95
CTVCLMIG - Clean Migrated CDAM Files	9-98
CTVDELI - Delete Index Files from DASD	9-99
EXEC Statement	9-99
DAYS Statement	9-100
INCLUDE/EXCLUDE Statements	9-100

CONTROL-O Utilities	9-103
CTOCSF - Copying the Message Statistics File	9-103
CTORSTM - Statistics Report Utility	9-104
SORT Parameter	9-104
INCLUDE/EXCLUDE Statements	9-104
CTOALOCP - Backing Up the Automation Log	9-105
CONTROL-T Utilities	9-107
Record Selection Logic (INCLUDE/EXCLUDE Statements)	9-107
General	9-107
Expressions	9-107
INCLUDE Statements	9-107
EXCLUDE Statements	9-108
INCLUDE/EXCLUDE Block	9-108
Single INCLUDE/EXCLUDE Block Processing	9-108
Multiple INCLUDE/EXCLUDE Blocks	9-109
INCLUDE/EXCLUDE Examples	9-109
CTTACP - Copy Trace File	9-111
CTTAFR - Format and Initialize Trace File	9-113
CTTBIX - Rebuild Media Database Index	9-114
CTTCRDF - Allocate and Format Data/Index Files	9-116
CTTCTRC - Allocate and Format Trace File	9-119
CTTDBDLD - Load Media Database/Stacking Statistics File Data Components	9-120
CTTDBDUL - Unload Media Database/Stacking Statistics File Data Components	9-121
CTTDBF - Format Existing Data/Index Files	9-122
CTTDBIB - Rebuild Indexes of Stacking Statistics File	9-123
CTTDBID - Check Integrity of Data File	9-124
CTTDBII - Check Integrity of Index File	9-125
CTTDBPRT - Print Data Records	9-126
CTTDLD - Add/Delete/Convert Volumes	9-127
CTTGVL - User Interface to Stacking Facility	9-130
CTTIDB - Check Integrity of the Media Database	9-132
CTTMER - Merge Records Into the Media Database	9-134
CTTMUP - Manually Update the Media Database	9-136
CTTRCV - Recover the Media Database Using Trace File	9-144
CTTRPT - Database Extraction and Report Utility	9-148
CTTRTM - Perform Retention Management	9-167
CTTSPL - Split the Media Database	9-177
CTTSTK - Build and Update Stacking Statistics File	9-182

CTTTPI - Tape Initialization, Information Mapping, and Tape Erasure	9-184
Safe Tape Initialization - Function INITT	9-184
Media Information Mapping - Function TAPEMAP	9-185
Safe Tape Erasure - Function TAPERAS	9-186
CTTTPI and I/O Error Recovery	9-187
Input Parameters - SYSIN DD File	9-189
Invoking CTTTPI From Another Program	9-195
CTTTRB - Synchronize Trace File and Media Database Backup	9-197
CTTUDB - Update the Media Database From Transaction File	9-198
CTTVTM - Perform Vault Management	9-199

General

This section describes the administrative utilities available for each IOA product.

A summary of these utilities (by product) is provided below. This section contains a detailed description of each utility (in alphabetical order by product).

Summary of Utilities

The following list provides a short description of each utility used in the IOA environment. These utilities are fully described later in this section unless otherwise noted.

Note Some utilities prefixed by IOA replace corresponding utilities prefixed by CTM, CTD, and/or CTB in previous releases. In such cases, the old utilities are supported for your convenience; however, we recommend using the IOA utilities.

IOA Utilities

IOABLCAL	Builds calendars.
IOACLRES	Cleans the IOA Conditions/Resources file.
IOACND	Perform Condition/Resource Adjustments.
IOACOPRS	Copies, or changes the size of, the IOA Conditions/Resources file.
IOACPLOG	Copies contents of the IOA Log file to a sequential file or to another Log file.
IOAFRLOG	Allocates and formats an IOA Log file (used during IOA installation).
IOAFRNRS	Allocates and formats an IOA Manual Conditions file (used during IOA installation).
IOAFRRRES	Allocates and formats an IOA Conditions/Resources file (used during IOA installation).
IOALDNRS	Loads/creates the IOA Manual Conditions list.
IOATEST	Simulates various events in the IOA environment.
IOAVCND	For VM/CMS environments. Sets prerequisite conditions in a CONTROL-M monitor running on an NJE-connected MVS machine. (See the description of utility IOACND in this section of the manual.)
IOAVERIFY	Verifies installation and produces reports on the status of various IOA files.

CONTROL-M Utilities

Note Unless otherwise noted, all CONTROL-M utilities are described in the CONTROL-M User Manual.

CTMAESIM	Tests the syntax of AutoEdit control statements.
CTMBGRP	Converts regular scheduling tables to group scheduling tables (described later in this section).
CTMBLDAE	Builds an AutoEdit symbol member using a CONTROL-M calendar as input.
CTMBLT	Creates CONTROL-M Scheduling tables.
CTMCAJF	Compresses or changes the size of the CONTROL-M Active Jobs file (described later in this section).
CTMFRCKP	Allocates and formats a CONTROL-M Active Jobs file (used during CONTROL-M installation).
CTMJSA	Accumulates job statistical data in the CONTROL-M Job Statistics file (described later in this section).
CTMRCAL	Creates a calendar for each job defined in CONTROL-M showing the actual execution date(s) of the job.
CTMRELRS	Releases Quantitative resources from a job step.
CTMSIM	Activates the Simulation and Forecasting facility.

CONTROL-D/V Utilities

CTDAMUTI	Copies, moves and deletes CDAM files (described in Section 5 of the CONTROL-D User Manual).
CTDCAMF	Compresses or changes the size of the Active Missions file.
CTDCATF	Compresses or changes the size of the Active Transfer file.
CTDCA2P	Copies new report/user entries from the Active User Report List file to the Permanent User Report List file.
CTDCCU	Identifies CDAM files not referenced by the Active User Report List file and optionally deletes them.
CTDCLHIS	Erases backup tapes which have expired based on parameter # OF DAYS TO KEEP specified in the Backup Mission Definition screen.
CTDCP2A	Copies empty report/user entries from the Permanent User Report List file to the Active User Report List file.

CTDDELRP	Moves report entries from the Active to the Migrated and/or History Report List file and erases their compressed datasets from DASD storage.
CTDDIB	Rebuilds a CONTROL-D User Report List file's index component.
CTDFRCOM	Allocates and formats a Communication file.
CTDSMFRP	Produces a paper CONTROL-usage report (described in Section 6 of the CONTROL-D User Manual).
CTDUPBKP	Updates Backup information in the User Report List files (Active/Migrated/History/Permanent).
CTVCLMIG	Cleans entries from the Migrated User Report List file and erases or uncatalogs their CDAM datasets.
CTVDELI	Deletes unneeded index files from disk based on user-specified selection groups.

IOA Access Method Utilities (for User Report List Files)

IOADBF	Formats a CONTROL-D User Report file's index or data component.
IOADBSR	Sorts and reorganizes a CONTROL-D User Report file's data component.
IOADCPY	Restores a CONTROL-D User Report file from a dual mirror image copy.
IOADIG	Verifies the integrity of a CONTROL-D User Report file's data component.
IOADII	Verifies the integrity of a CONTROL-D User Report file's index component.
IOADLD	Loads a CONTROL-D User Report file's index or data component contents from a sequential dataset.
IOADPT	Prints a CONTROL-D User Report file's data and index component records in SNAP dump format.
IOADUL	Unloads a CONTROL-D User Report file's index or data component contents to a sequential dataset.

CONTROL-O Utilities

CTOALOCF	Backs up the Automation Log.
CTOCSF	Copies the Message Statistics file.
CTORSTM	Produces reports on message statistics.

CONTROL-B Utilities

CONTROL-B Group and Variable File Handling Utilities

- CTBDBVCP Copies a specified variable or group of variables to another set of variable/group files.
- CTBDBVDL Deletes a variable or a group of variables from variable/group files.
- CTBDBVIG Checks the integrity of variable/group data files.
- CTBDBVCG Changes the number of generations of a Database Variable.

CONTROL-B Report and Rule Activity File Handling Utilities

- CTBJAFCP Copies invocation and report entries to other Rule Activity and Report files.
- CTBJAFDL Deletes unnecessary entries from the Rule Activity and Report files.
- CTBJAFIG Checks the integrity of Rule Activity and Report data files.

CONTROL-B Active Balancing File Utilities

- CTBABI Initializes the Active Balancing file after creation.
- CTBFRM Compresses the Active Balancing file (usually on a daily basis as part of the New Day procedure).

IOA Access Method File Utilities

- CTBDBIB Rebuilds the indexes of specified CONTROL-B data files.
- IOADBF Allocates and Formats an IOA Access Method file's data or index component.
- IOADIG Verifies the integrity of an IOA Access Method file's data component.
- IOADII Verifies the integrity of an IOA Access Method file's index component.
- IOADPT Prints an IOA Access Method file's data and index component records in SNAP dump format.
- IOADCPY Restores an IOA Access Method file from a dual mirror image copy.

Miscellaneous Utilities

- CTBVXRF Produces a report indicating which variables are used by which rules, (described in Section 8 of the CONTROL-B User Manual).

CONTROL-T Utilities

Trace Utilities

CTTACP	Copies the Trace file.
CTTAFR	Formats and initializes the Trace file.
CTTCTRC	Allocates and formats the Trace file.
CTTTRB	Synchronizes the Trace file and the Media Database backup process.

Stacking Utilities

CTTGVL	Provides a user interface to the Stacking Facility.
CTTSTK	Builds and updates the Stacking Statistics file.

Media Database Utilities

CTTBIX	Rebuilds the index of the Media Database.
CTTDLD	Adds non-scratch volumes, adds and deletes scratch volumes, and converts from other tape management systems.
CTTIDB	Checks the integrity of the Media Database.
CTTMER	Merges records from an extract file (created by utility CTTSP) into the Media Database.
CTTMUP	Performs manual update of the Media Database.
CTTRCV	Recovers Media Database contents using the Trace file.
CTTSPL	Copies or moves (Splits) specified records from the Media Database.
CTTUDB	Updates Media Database records from a transaction file (which is created by utility CTTIDB).

Daily Management Utilities

CTTRTM	Performs retention management.
CTTVTM	Performs vault management.

Section 9: Utilities

Database Handling Utilities (for the Media Database and Stacking Statistics File)

CTTCRDF	Allocates and formats various data and/or index files.
CTTDBDL	Loads Media Database and Stacking Statistics file data components from a sequential dataset.
CTTDBDUL	Unloads Media Database and Stacking Statistics file components to a sequential dataset.
CTTDBF	Formats existing data and/or index files.
CTTDBIB	Rebuilds the indexes of the Stacking Statistics file.
CTTDBID	Checks the integrity of a data file.
CTTDBII	Checks the integrity of an index file.
CTTDBPRT	Prints (in dump format) records from a database file to a sysout.

Miscellaneous CONTROL-T Utilities

CTTRPT	Generates reports and from extracted Media Database information.
CTTTPI	Initializes tapes, erases tapes, and produces a detailed report of the logical and physical contents of a volume.

IOA Utilities

IOABLCAL - Build Calendars

The scheduling of jobs, missions and rules in the IOA environment is frequently based on calendars. A data center may use different calendars for different scheduling requirements. For example:

WORKDAYS	All working days in the year
HOLIDAYS	All non-standard holidays
ENDOFMON	Business end-of-month scheduling
NOTEOFMN	All working days except end-of-month days

In many cases, a calendar is related to, and can be derived from, a different calendar or from a combination of two different calendars. For example, in the calendars listed above, calendar NOTEOFMN can be derived from a combination of calendars WORKDAYS and ENDOFMON.

When calendars are related, a change to one calendar frequently affects other calendars. This can make manual maintenance of calendars time-consuming and error-prone. Utility IOABLCAL simplifies and automates calendar maintenance.

Through parameters supplied to the utility, the user specifies two existing calendars and the operation (or formula) which determines how the calendars are combined to build a new calendar.

The new calendar created by the utility is a regular calendar. It can be viewed using Option 8 (Calendars) of the IOA Primary Option Menu and it can be used as an input calendar in subsequent runs of the utility. Thus, using several basic calendars, numerous other calendars can be created.

It is recommended that all steps which calculate calendars be collected into one job, and that the job be run once a day. This will refresh the calendars in the event that changes are made to the basic calendars.

Parameters

Parameters are supplied to the utility either via DD statement DABCLIN (or SYSIN) or via PARM. Any number of parameter statements may be specified. Parameter statement syntax is:

```
calendar1 operator calendar2 = calendar3
```

Where:

calendar1, calendar2 Input calendars.

operator One of the following operators:
+, -, *, <W nn, >W nn, <> W, >< W

calendar3 Resulting calendar.

If the operation is +, calendar3 will contain all days marked Y in calendar1 and additionally all days marked Y in calendar2.

If the operation is -, calendar3 will contain all days marked Y in calendar1 excluding those days marked Y in calendar2.

If the operation is *, calendar3 will contain all days marked Y in calendar1 which are also marked Y in calendar2. The day has to be marked for execution in both calendars to be marked in calendar3.

If calendar2 is a relative calendar, additional logic is required to handle the * operation listed above. (A relative calendar is a calendar which contains days marked + or - instead of, or in addition to, days marked Y or N.) Each day in calendar2 which contains a + or - sign is checked against the same day in calendar1. If the day is marked for execution in calendar1, it will be marked for execution in calendar3. However, if the day is not marked for execution in calendar1, the result depends on how the day was marked in calendar2:

- If the day was marked + in calendar2, the closest next day marked for execution in calendar1 will be marked for execution in calendar3.
- If the day was marked - in calendar2, the closest previous day marked for execution in calendar1 will be marked for execution in calendar3.

Operations <W nn and >W nn create a new calendar (calendar3) by shifting the days in an old calendar (calendar1) a specified number of working days according to the specified workdays calendar (calendar2):

Symbol > indicates that days should be shifted to the right (i.e., to subsequent days).

The number of working days to shift is nn.

The logic for operations < W nn and >W nn is as follows:

Each day marked for execution in calendar1 is checked against calendar2.

- If the same day is a working day in calendar2, nn working days in calendar2 are counted in the direction of the shift and the resulting day is marked as a working day in calendar3.
- If the same day is not a working day in calendar2, the closest working day in the direction of the shift is identified in calendar2. From that working day, nn working days in calendar2 are counted in the direction of the shift and the resulting day is marked as a working day in calendar3.

- If operation <>W or ><W is specified, then for each working day (i.e., day marked for execution) in calendar1:
 - If the same day is a working day in calendar2, it is marked as a working day in calendar3.
 - If the same day is not a working day in calendar2, the closest working day (in either direction) is marked as a working day in calendar3.
 - If two working days are equally closest (i.e., the same number of days before and after), the day selected depends on the parameter specified:
- If <>W was specified, the earlier working day is selected as the working day for calendar3.
- If ><W was specified, the later working day is selected as the working day for calendar3.

Note If specifying operator <W nn, >W nn, <>W, or ><W, calendars must be defined in year order, and must contain the same years.

Activating the Utility

```
// EXEC IOABLCAL
calendar1 operator calendar2 = calendar3
```

Where:

operator is +, -, *, <W nn, >W nn, <>W or ><W

Return Codes

0	Operation performed successfully
8 and above	Operation failed

Example 1

The following parameter is passed to the utility:

calendar1 * calendar2 = calendar3

Assuming the following calendars:

```

S ----- S ----- S -----
1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9      calendar1
  Y Y Y Y Y   Y Y Y Y Y Y               Y
S ----- S ----- S -----

```

and

```

S ----- S ----- S -----
1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9      calendar2
  Y Y + -   +           +   -
S ----- S ----- S -----

```

The resulting new calendar is:

```

S ----- S ----- S -----
1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9      calendar3
  Y Y Y Y   Y           Y               Y
S ----- S ----- S -----

```

Note the following points in this example:

- Since the operation is *, days 2 and 3, which are marked for execution in both calendar1 and calendar2, will be marked for execution in calendar3.
- Day 4 is marked + in calendar2 and is marked for execution in calendar1. Therefore, it will be marked for execution in calendar3.
- Day 5 is marked - in calendar2 and is marked for execution in calendar1. Therefore, it will be marked for execution in calendar3.
- Day 6 is not marked in calendar2. Therefore, even though it is marked Y in calendar1, it will not be marked for execution in calendar3. This same reasoning applies to days 8, 9, 10, 11, 12, 13 and 19. However, days 8, 13 and 19 are marked for execution for other reasons described below.
- Day 7 is marked + in calendar2 and is not marked for execution in calendar1. The closest following day marked for execution in calendar1 is day 8. Therefore, day 8 will be marked for execution in calendar3.
- Day 14 is marked + in calendar2 and is not marked for execution in calendar1. The closest following day marked for execution in calendar1 is day 19. Therefore, day 19 will be marked for execution in calendar3.
- Day 16 is marked - in calendar2 and is not marked for execution in calendar1. The closest preceding day marked for execution in calendar1 is day 13. Therefore, day 13 will be marked for execution in calendar3.

Example 2

The following parameter is passed to the utility:

OLDCAL <W 2 WORKDAYS = NEWCAL

Assuming the following calendars:

```

S ----- S ----- S -----
1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9
                Y      Y Y Y      Y
                                                    OLDCAL
S ----- S ----- S -----

```

and

```

S ----- S ----- S -----
1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9
      Y Y Y Y      Y Y Y Y      Y Y Y
                                                    WORKDAYS
S ----- S ----- S -----

```

The resulting new calendar is:

```

S ----- S ----- S -----
1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9
      Y      Y Y      Y Y
                                                    NEWCAL
S ----- S ----- S -----

```

Note the following points in this example:

- Operation <W 2 specifies that for each working day in OLDCAL, IOABLCAL should find the corresponding working day in WORKDAYS, and then shift two workdays to the left.
- The first workday in OLDCAL is day 5. This is also a workday in WORKDAYS. The 2nd workday to the left of day 5 in WORKDAYS is day 3, which becomes a workday in NEWCAL.
- The next workday in OLDCAL is day 10. This is also a workday in WORKDAYS. The 2nd workday to the left of day 10 in WORKDAYS is day 6, which becomes a workday in NEWCAL.
- The next workday in OLDCAL is day 11. This is also a workday in WORKDAYS. The 2nd workday to the left of day 11 in WORKDAYS is day 7, which becomes a workday in NEWCAL.
- The next workday in OLDCAL is day 12. This is also a workday in WORKDAYS. The 2nd workday to the left of day 12 in WORKDAYS is day 10, which becomes a workday in NEWCAL.
- The next workday in OLDCAL is day 15. This is not a workday in WORKDAYS. The closest workday to the left, in WORKDAYS, is day 13. The 2nd workday to the left of day 13 in WORKDAYS is day 11, which becomes a workday in NEWCAL.

Example 3

The following parameter is passed to the utility:

OLDCAL ><W WORKDAYS = NEWCAL

Assuming the following calendars:

```

S ----- S ----- S -----
1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9
           Y   Y Y   Y   Y
                                           OLDCAL
S ----- S ----- S -----

```

and

```

S ----- S ----- S -----
1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9
      Y Y Y Y Y   Y Y Y Y   Y Y Y
                                           WORKDAYS
S ----- S ----- S -----

```

The resulting new calendar is:

```

S ----- S ----- S -----
1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9
           Y   Y   Y   Y   Y
                                           NEWCAL
S ----- S ----- S -----

```

Note the following points in this example:

- Operation ><W specifies that for each working day in OLDCAL, IOABLCAL should find the closest working day in WORKDAYS, and make that day a workday in NEWCAL.
- The first workday in OLDCAL is day 5. This is also a workday in WORKDAYS, so it becomes a workday in NEWCAL.
- The next workday in OLDCAL is day 8. This is not a workday in WORKDAYS. The next closest workday in WORKDAYS is day 7, so day 7 becomes a workday in NEWCAL.
- The next workday in OLDCAL is day 9. This is not a workday in WORKDAYS. The next closest workday in WORKDAYS is day 10, so day 10 becomes a workday in NEWCAL.
- The next workday in OLDCAL is day 12. This is also a workday in WORKDAYS, so it becomes a workday in NEWCAL.
- The next workday in OLDCAL is day 15. This is not a workday in WORKDAYS. The next closest workdays in WORKDAYS are days 13 and 17 (which are equally close to day 15). Since the parameter operation statement ><W began with symbol >, the value to the right (i.e., the subsequent day) is chosen. Therefore, day 17 becomes a workday in NEWCAL.

IOACLRES - Clean IOA Conditions/Resources File

This utility cleans the IOA Conditions/Resources file of conditions which are no longer required. It is recommended that this utility be activated as part of IOA maintenance.

Parameters

Unless the utility is instructed otherwise via parameters, it will delete all prerequisite conditions it finds in the Conditions/Resources file.

The utility outputs a list of all the deleted prerequisite conditions.

Prerequisite conditions which should not be deleted (e.g., prerequisite conditions needed throughout the year) are specified by means of control statements, which are specified via DD statement DACRSIN or SYSIN. The formats of these statements are displayed below, followed by explanations:

```
IGNORE { COND } { cond - prefix } [ FROM { date } ] [ TO { date } ]
        { CONDIN }      *                [ - n ] [ - n ]
```

```
CONTROLM { YES }
          { NO }
```

```
CONTROLD { YES }
          { NO }
```

IGNORE statements instruct the utility not to delete the indicated prerequisite condition(s), as follows:

- An IGNORE COND statement instructs the utility not to delete the indicated prerequisite condition under any circumstances.
- An IGNORE CONDIN statement instructs the utility not to delete the indicated prerequisite conditions if the utility determines that the prerequisite conditions are referenced conditions. A referenced condition is a prerequisite condition appearing in the IOA Conditions/Resources file at the same time that it appears as an IN condition in an active job, mission or rule (e.g., one which is currently present in the Active Jobs file or Active Missions file).
- cond-prefix specifies the prefix of prerequisite conditions which should not be deleted. An asterisk indicates all prerequisite conditions.
- If a FROM/TO combination is supplied, a specified condition will not be deleted if its date is in the date range. Conditions not in the date range will be deleted, unless they are referred to by another IGNORE statement.
- date can be either a specific date in the site-defined standard (mmddy, dmmyy or yymmdd) or a relative date. A relative date is specified in the format -n, where n is the number of days before the current date (e.g., -5 means five days before the current date). Relative and specific dates can be mixed (e.g., a specific FROM date and a relative TO date).

When a FROM and/or TO date is not specified, the current date is the default for the unspecified date.

The TO date (regardless of whether specific, relative or assumed) cannot precede the FROM date.

CONTROLM and CONTROLD statements indicate whether or not the utility should check prerequisite conditions in the Active Jobs file and/or the Active Missions file, respectively, if an IGNORE CONDIN statement is specified.

By default, each of these files is checked if its respective product is installed (according to the IOA Installation options). These statements can be used to override the default. You can specify either, both or none of these statements.

These statements can impact the result of an IGNORE CONDIN statement because the utility cannot determine that a prerequisite condition is referenced unless it can check the Active Jobs/Missions file for the job/mission containing the IN prerequisite condition.

Valid values for these statements are described below:

CONTROLM Determines whether or not the CONTROL-M Active Jobs file is checked during the cleanup process of the IOA Conditions/Resources file. Optional.

YES Check the Active Jobs file.

NO Do not check the Active Jobs file.

CONTROLD Determines whether or not the CONTROL-D Active Missions file is checked during the cleanup process of the IOA Conditions/Resources file. Optional.

YES Check the Active Missions file.

NO Do not check the Active Missions file.

The utility can be run simultaneously with the CONTROL-M and the CONTROL-D monitors.

If the job is submitted under CONTROL-M, AutoEdit variables are supported.

Activating the Utility

```
// EXEC IOACLRES
```

Return Codes

0	Operation performed successfully
4	Execution error, operation stopped
8 and above	Critical error, task not performed

Example 1

It is recommended that you run this job weekly. Since this job contains AutoEdit variables, it must be submitted directly via the CONTROL-M monitor. (The example assumes date standard mmddyy. For ddmmyy format, the AutoEdit variable should be %%ODAY.%%OMONTH.%%OYEAR. For yymmdd format, the AutoEdit variable should be %%OYEAR.%%OMONTH.%%ODAY.)

```
// EXEC IOACLRES:
  IGNORE COND * FROM -10 TO %%OMONTH.%%ODAY.%%OYEAR
  IGNORE COND CICS
  IGNORE COND DB-
```

Example 2

Clean the resources file of all prerequisite conditions not referenced by jobs in the Active Jobs file, except those prerequisite conditions beginning with CICS. (Prerequisite conditions beginning with CICS are not deleted.) Do not check the Active Missions file to see if a prerequisite condition is referenced.

```
// EXEC IOACLRES:
  IGNORE CONDIN * CONTROLD NO
  IGNORE COND CICS
```

IOACND - Perform Condition/Resource Adjustments

This utility adds, verifies the existence of, deletes, or modifies prerequisite conditions and resources in the IOA Conditions/Resources file. It can be activated by the operator as a started task (STC), from TSO, or from within a user program.

The addition, deletion, modification or verification is recorded three ways: in the job sysout, as a message on the operator console, and as an event in the IOA Log file.

Parameters can be received by the utility in two ways: via DD statement DACNDIN (or SYSIN), or via PARM. The parameters required depend on whether prerequisite conditions, control resources or quantitative resources are being processed.

If it is necessary to add, delete, verify or modify prerequisite conditions or resources during a job step, the utility can be called as a procedure from within the application program using procedure name CTMCND, as described later in "Activating the Utility (and Examples)."

Parameters for Processing Prerequisite Conditions

action COND condition condition-date

Where:

action	Action to be performed by the utility. Valid actions are:
	ADD Add the specified prerequisite condition.
	DELETE Delete the specified prerequisite condition.
	CHECK Verify the existence of the specified prerequisite condition.
condition	1-20 character prerequisite condition.
condition-date	Date associated with the prerequisite condition. Valid actions are:
	Any specific date (in format mmdd or ddmm, depending on the site standard)
	DATE Current Gregorian date
	WDATE IOA Installation working date
	Date character-masking for DELETE and CHECK functions is supported as follows:
	* Represents any value in the corresponding position of the date (format mmdd or ddmm depending on the site standard). Up to four asterisks can be specified. For example:
	**** Any date
	*2*2 Valid combinations of:
	Months: February and December
	Days: 02, 12, and 22

Parameters for Processing Control Resources

action CONTROL resource control

Where:

action Action to be performed by the utility. Valid actions are:

ADD Add the specified Control resource under the specified type of control.

DELETE Delete the specified resource which is under the specified type of control.

resource Resource name.

control Type of control. Valid values are:

E Exclusive control.

S Shared control.

Parameters for Processing Quantitative Resources

action RESOURCE resource quantity

Where:

action Action to be performed by the utility. Valid actions are:

ADD Add the specified Quantitative resource with the quantity specified. quantity is mandatory.

DELETE Delete the resource (and the entire existing quantity). quantity must not be specified.

CHANGE Adjust the existing quantity of the Quantitative resource to or by the value specified in quantity. quantity is mandatory.

resource Quantitative resource name.

quantity Quantity of the Quantitative resource.

For CHANGE actions, valid values can have the format n, +n, or -n where n is an integer. When n is specified, the quantity of the Quantitative resource is set to n. When +n or -n is specified, the quantity is increased or decreased by n.

For ADD or DELETE actions, valid values can only have the format n, where n is the quantity of the Quantitative resource to be added or deleted.

Activating the Utility (and Examples)

As a job step:

```
//CND1 EXEC IOACND
  ADD COND JOB-A-ENDED-OL 0505
  DELETE COND JOB-A-DID-NOT-RUN 0505
  CHANGE RESOURCE TAPE -1
  ADD RESOURCE INIT-CPUA 3
  DELETE RESOURCE INIT-CPUB
  ADD CONTROL PROD-DATA-BASE E
```

As a job, containing AutoEdit variables, submitted through the CONTROL-M monitor:

```
//CND1 EXEC IOACND
  ADD COND      AR-FILE-009-CREATED  %%OMONTH.%%ODAY
```

As an operator command:

```
S IOACND,PARM='ADD COND END-CICS 0102'
```

As a called procedure in an application program:

```
      CALL CTMCND, (PARMCOND), VL
      LTR  R15, R15
      BNZ  ERROR
      ...
PARMCOND DC  AL2(80)
DC  CL80'ADD COND CICS-F
```

Note Program IOACND works in 24-bit addressing mode. The program is reusable but not re-entrant. The program operates in AMODE 24.

```
ILE-CLOSED 0102'
```

Return Codes

ADD	0	Prerequisite condition or resource added
	8	Prerequisite condition or resource already exists
	12 and above	Function failed
DELETE	0	Prerequisite condition deleted
	8	Prerequisite condition to be deleted does not exist
	12 and above	Function failed

CHECK	0	Prerequisite condition exists
	8	Prerequisite condition does not exist
	12 and above	Function failed
CHANGE	0	Quantity of Quantitative resource adjusted successfully
	8	Quantitative resource does not exist
	12 and above	Function failed

Note *If you are working in a VM/CMS environment, utility IOAVCND can be used to set prerequisite conditions in a CONTROL-M monitor running on an NJE-connected MVS machine.*

The EXEC statement for utility IOAVCND utilizes standard IOACND input, and sends a job to the MVS node to set the conditions.

Example

```
EXEC IOAVCND 'ADD COND VM-READY STAT'
```

IOACOPRS - Copy/Resize IOA Conditions/Resources File

This utility copies the contents of the IOA Conditions/Resources file to another Conditions/Resources file which may or may not be the same or size or version.

Copying/Resizing the IOA Conditions/Resources File

Note *Utility IOACOPRS cannot run simultaneously with the IOA product monitors (e.g., CONTROL-M monitor). If the monitors are not shut down, the utility will terminate. Online users may remain logged on, but should not access the Online facility while the size of the Conditions/Resources file is being changed.*

To change the size of the IOA Conditions/Resources file, perform the following steps:

1. Change IOA installation parameters RESLREC and RESREC# in member IOAPARM to the required new IOA Conditions/Resources file size.
2. Reassemble member IOAPARM using job IOAPARMJ.
3. After doing this, every access to the IOA Conditions/Resources file (online or batch) will try to utilize the new size and may abend. Therefore, all access to the IOA Conditions/Resources file should be stopped until the process is complete.
4. Change parameter LRECL and RECNUM in procedure IOAFFRES to the new size of the IOA Conditions/Resources file.
5. Run member FORMRES in the IOA INSTALL library after changing the dataset names (so you will not override the existing IOA Conditions/Resources file and its backup file).

If you are working in dual checkpoint mode, run member FORMDRES.

Note *If you are working in dual checkpoint mode, the mirror file must also be allocated with the new size before starting the IOA product monitors (e.g., the CONTROL-M monitor). Otherwise, the monitors may abend.*

6. Run utility IOACOPRS.

The utility will copy the current IOA Conditions/Resources file to the new IOA Conditions/Resources file.

7. Rename the old IOA Conditions/Resources file (and its mirror file - suffix ALTRES).
8. Rename the new IOA Conditions/Resources file (and its mirror file - suffix ALTRES) to the former names of the old files.
9. Restart the IOA product monitors and resume operation.

The utility must terminate with a return code of 0.

Parameters

The utility receives parameters via the invocation of the procedure.

OLDRES Name of the old IOA Conditions/Resources file.

OLDSNC Name of the old IOA Conditions/Resources Synchronization file.
NEWRES Name of the new IOA Conditions/Resources file.
NEWSNC Name of the new IOA Conditions/Resources Synchronization file.

Return Codes

0 Operation performed successfully
8 and above Operation failed

Activating the Utility (and Example)

```
// EXEC IOACOPRS,  
//        OLDRES='IOA.old.RES',        OLD CONDITIONS/RESOURCES  
//        OLDSNC='IOA.old.SNC',        OLD CONDITIONS/RESOURCES  
//*                                        SYNCHRONIZATION FILE  
//        NEWRES='IOA.new.NEWRES',    NEW CONDITIONS/RESOURCES  
//*                                        FILE  
//        NEWSNC='IOA.new.NEWSNC'    NEW CONDITIONS/RESOURCES  
//*                                        SYNCHRONIZATION FILE
```

IOACPLOG - Copy the IOA Log File

Utility IOACPLOG copies the contents of the IOA Log file. The contents can be copied to a sequential file or to another Log file. Usage of the utility depends on the type of target file.

Copying to a Sequential File

The log is generally copied to a sequential file for archiving purposes. The sequential file can then be restored access under ISPF (e.g., browsing) and for use by report generators.

A date range for the log records to be copied can be specified. If a date range is not specified, the entire Log file is copied.

The target sequential file is specified in DD statement DALOGSEQ.

It is recommended that the utility be used on a daily basis to copy the contents of the Log file to a sequential file. Use of AutoEdit parameters can simplify the process.

Parameters

Parameters are supplied via DD statement DALCPPRM (or SYSIN). The utility procedure contains a reference to the existing Log file, in DD statement DALOGCUR.

COPYTOSEQ	Indicates that the utility is being used to copy records from the IOA Log file to a sequential file.
FROMDATE yymmdd	Earliest Log record date used to select records to be copied. AutoEdit variable are supported.
TODATE yymmdd	Latest log record date to used to select records to be copied. AutoEdit variable are supported.

Activating the Utility (and Example)

```

/** %%SET %%D = %%CALCDATE %%ODATE -1
// EXEC IOACPLOG
//DALOGSEQ DD DSN=prefix.seqfile,DISP= ... Indicates the target sequential file
//DALCPPRM DD *
    COPYTOSEQ
    FROMDATE %%D
    TODATE %%D
//

```

Return Codes

0	Operation performed successfully
Other	Operation failed

Copying to a Log File

This function is generally used to copy the Log file to another, larger Log file when it becomes necessary to enlarge the size of the Log file.

The target Log file is specified in DD statement DALOG.

Parameters

Parameters are supplied via DD statement DALCPPRM (or SYSIN). The utility procedure contains a reference to the existing Log file, in DD statement DALOGCUR.

COPYTOLOG Indicates that the utility is being used to copy the IOA Log file to another Log file.

Activating the Utility (and Example)

```
// EXEC IOACPLOG
//DALOG DD DSN=prefix.logfile,DISP= ...        Indicates the target Log file
//DALCPPRM DD *
      COPYTOSEQ
//
```

Return Codes

0 Operation performed successfully
Other Operation failed

IOADBF - Allocate/Format an IOA Access Method File

Utility IOADBF is used to allocate and format an IOA Access Method file's data and index components.

Parameters

The following parameters can be specified to the utility in the EXEC statement which invokes the IOADBF JCL procedure:

FUNC	Function to be performed:
INIT	Allocates and initializes (formats) a new IOA Access Method file component.
FORM	Reformats an existing IOA Access Method file component.
EXTEND	Allocates and formats a new secondary extent for an existing IOA Access Method file component.
CHANGE	Changes/Updates an IOA Access Method file component's internal attribute settings. This function should be used when it is necessary to change the settings of parameters in an IOA Access Method file definition after the file has been created.
DUALOK	Resets the flag that indicated a corrupted Dual Mirror Image file (DUALBAD). This function should be used after a corrupted Dual Mirror Image file has been restored via utility IOADCPY.
D	Dataset name suffix of the IOA library containing the IOA Access Method file Definition Statements member. The member name is specified via parameter M below.
M	Name of the IOA library member which contains the IOA Access Method file Definition Statements and whose dataset name suffix is specified via parameter D above.

DD Statement DAFILE

When executing utility IOADBF with parameter FUNC set to EXTEND or CHANGE, a DAFILE DD statement, specifying the name of the IOA Access Method file component's primary dataset extent, must be included in the JCL step executing the utility.

IOA Access Method File Definition Statements

The following IOA Access Method file definition statements can be specified as input to utility IOADBF in the IOA library and member referred to above by EXEC statement parameters D and M:

TYPE	IOA Access Method file type:
F	Data component, fixed record format.
I	Index component.
V	Data component, variable record format. This value is not valid for use with CONTROL-B files.

The TYPE statement is not applicable when executing utility IOADBF with the FORM or CHANGE function specified.

DSN	IOA Access Method file's dataset name, excluding the extent number suffix (.E000 through .E255) which is appended automatically by the utility when the IOA Access Method file is allocated.
------------	--

The DSN statement is not applicable when executing utility IOADBF with the INIT.FORM or CHANGE function specified.

UNIT	Name of the DASD unit to be used when allocating a new IOA Access Method file extent (e.g., 3390).
-------------	--

VOL	List of one to six DASD volume serial numbers, enclosed in parenthesis and separated by commas, on which the IOA Access Method file's extents can be allocated.
------------	---

All volumes must be defined to MVS as being associated with the unit name specified in parameter UNIT above.

If an out-of-space condition occurs due to lack of space in the specified volumes, additional volumes can be added. (A maximum of six volumes is permitted.) Utility IOADBF should be executed with EXEC parameter FUNC=CHANGE (this stores the new setting internally within the IOA Access Method file) and then the CONTROL-D monitor should be restarted.

If VOL is not specified, the IOA Access Method file will be allocated using only parameter UNIT.

UNITD	Name of the DASD unit to be used when allocating a new dual mirror image file extent (e.g., 3390).
--------------	--

VOLD	List of one to six DASD volume serial numbers, enclosed in parenthesis and separated by commas, on which the dual mirror image file's extents can be allocated.
-------------	---

Note *It is recommended that you specify different volumes for the primary and dual mirror copy of each file to ensure greater data integrity.*

SPACE Amount of space, in blocks, to be allocated for the primary and, optionally, for each secondary IOA Access Method file extent.

Note *Secondary extents cannot be specified for CONTROL-B files.*

It is recommended that a secondary space amount be specified to avoid termination of the CONTROL-D monitor when the primary extent becomes full and an out of space condition is detected.

If an out of space condition occurs, secondary extent allocations can be set by specifying a secondary amount in parameter SPACE, executing utility IOADBF with EXEC parameter FUNC=CHANGE (this stores the new setting internally within the IOA Access Method file), and restarting the CONTROL-D monitor.

Example 1

SPACE=(1500) A 1500-block IOA Access Method file is allocated. If the file becomes full, the CONTROL-B rule will terminate.

Example 2

SPACE=(1500,300) A 1500-block IOA Access Method file is allocated. If the file becomes full and a secondary extent is automatically or manually allocated (see IOA Access Method file Definition parameter EXTEND), each newly-allocated extent will be 300 blocks in size.

Parameter SPACE's primary space amount is not applicable when executing utility IOADBF with the FORM or CHANGE function specified.

DUAL	Y (Yes)	Allocate a dual mirror image copy of the specified file. The dual mirror image file's contents are kept synchronized with the contents of the primary file. Use of the dual mirror image file provides maximum data recovery capabilities when the primary file's disk or contents become damaged or inaccessible. Specifying DUAL=Y is recommended for the Data component file.
	N (No)	Do not allocate a dual mirror image copy of the specified file.
DUALM	Y (Yes)	If dual mirror image file processing is used, the execution terminates if a non-recoverable I/O error occurs while processing the dual mirror image copy of the file.
	N (No)	If dual mirror image file processing is used, execution continues if a non-recoverable I/O error occurs during processing of the dual mirror image copy of the file.

DUALST Y (Yes) Timestamp checkpoint processing is performed internally to ensure that the primary and dual mirror image copies of the file are fully synchronized with each other. Enabling this option insures greater data integrity at the expense of increasing I/O processing overhead.

 N (No) Timestamp checkpoint processing is not performed.

Refer to "CONTROL-D Installation Considerations" in the IOA Installation Guide for more information.

EXTEND Specifies secondary extent to be allocated. This parameter is not valid for use with CONTROL-B files.

Specifying A (the default) causes a secondary extent to be automatically allocated when an out of space condition is detected.

Specifying M requires a secondary extent to be manually allocated after an out of space condition is detected by executing the utility with EXEC parameter FUNC=EXTEND specified.

Parameter EXTEND is ignored if no secondary space amount is specified via parameter SPACE.

Parameter EXTEND is not applicable when executing utility IOADBF with the FORM or CHANGE function specified.

BLKSIZE Block size of the IOA Access Method file, in bytes. For all index components and for data components with variable length records, it is recommended that the block size be set to the maximum half track capacity of the device on which the IOA Access Method file is allocated, as follows:

BLKSIZE=19068 For 3350 DASD devices.

BLKSIZE=23476 For 3380 DASD devices.

BLKSIZE=27998 For 3390 DASD devices.

BLKSIZE=22928 For 9345 DASD devices.

For data components with fixed length records, where the block size must be a multiple of the record length, it is recommended that the block size be set as close to, without exceeding, the device's half track capacity.

Parameter BLKSIZE is not applicable when executing utility IOADBF with the FORM or CHANGE function specified.

BUFL Number of I/O buffers for which extended storage will be obtained to access the IOA Access Method file.

Note *This parameter is not valid for use with CONTROL-B files.*

Buffer length is determined by parameter BLKSIZE, described above. A larger number improves I/O throughput at the expense of increased storage requirements. A smaller number reduces storage requirements at the expense of degrading I/O throughput.

LRECL Logical record length for IOA Access Method file data components only. This parameter should not be specified for index components.

The setting for parameter LRECL is IOA application dependent and should not be modified unless instructed otherwise by Technical Support.

Parameter LRECL is not applicable when executing utility IOADBF with the FORM or CHANGE function specified.

KEYLEN Record key length for IOA Access Method file index components only. This parameter should not be specified for data components.

Parameter KEYLEN is not applicable when executing utility IOADBF with the CHANGE function specified.

The setting for KEYLEN is IOA application dependent and should not be modified unless instructed otherwise by Technical Support.

COMPRS Name of the IOA supplied routine to perform Data component record compression. This parameter should not be specified for index components.

Notes *Parameter COMPRS is not available when executing utility IOADBF or with the CHANGE function specified.*

Parameter COMPRS is not available when executing utility IOADBF on a CONTROL-D file.

The setting for COMPRS is IOA application dependent and should not be modified unless instructed otherwise by Technical Support.

COMPTB Name of the IOA supplied compression table to be used by the compression routine when performing data component record compression. This parameter should not be specified for index components.

Notes *Parameter COMPTB is not available when executing utility IOADBF with the FORM or CHANGE function specified.*

Parameter COMPTB is not available when executing utility IOADBF on a CONTROL-B file.

The setting for COMPTB is IOA application dependent and should not be modified unless instructed otherwise by Technical Support.

Example 1

```
//INIT EXEC IOADBF, FUNC=INIT, D=INSTCTD, M=DEFACT
```

Contents of member DEFACT in the IOA INSTCTD library:

```
TYPE=V
DSN=CTD.V500.ACT
BLKSIZE=27998
LRECL=27974
SPACE=(1000,200)
BUFL=100
UNIT=3390
VOL=(PRD021, PRD022, PRD023, PRD024, PRD025, PRD026)
COMPRS=CTDUFM
COMPTB=CTDUFMA
```

Example 2

```
//CHANGE EXEC IOADBF, FUNC=CHANGE, D=INSTCTD, M=DEFHSTI
//DAFILE DD DISP=SHR, DSN=CTD.V500.HSTI.E000
```

CONTROL-D Samples

Refer to sample jobs FORMUF1 and FORMUF2 in the IOA INSTCTD library for examples of how to allocate and format IOA Access Method files.

CONTROL-B Samples

Utility IOADBF is used by the following jobs in the INSTCTB library. Refer to these jobs for additional examples.

CREFABF	Allocates and formats the Active Balancing file.
CREFBKP	Allocates and formats the Active Balancing file backup file.
CREFDBM	Allocates and formats the Database Variable Basic file.
CREFDBV	Allocates and formats the Database Variable Generation file.
CREFGRP	Allocates and formats the Group file.
CREFJAF	Allocates and formats the Rule Activity file.
CREFREP	Allocates and formats the Report file.
CONVABF	Converts the Active Balancing file from CONTROL-B Release 1.2.0 to the format for Release 5.0.0.
CONVGRP	Converts the Group file from CONTROL-B Release 1.2.0 to the format for Release 5.0.0.
CONVJAF	Converts the Rule Activity file from CONTROL-B Release 1.2.0 to the format for Release 5.0.0.
CONVMOD	Converts the IOA Access Method Variable Basic file from CONTROL-B Release 1.2.0 to the format for Release 5.0.0.

CONVREP Converts the Report file from CONTROL-B Release 1.2.0 to the format for Release 5.0.0.

CONVVAR

The following jobs can be used as samples for formatting existing data and index files:

JOBFABF	Active Balancing file
JOBFGRP	Group file
JOBFMOD	IOA Access Method Variable Basic file
JOBFVAR	IOA Access Method Variable Generation file
JOBFBKP	Backup file
JOBFJAF	Rule Activity file
JOBFREP	Report file

The following jobs can be used as samples for formatting and rebuilding the index file for an existing data file:

JOBBGRP	Group file
JOBBBDM	Access Method Variable Basic file
JOBBDBV	Access Method Variable Generation file
JOBBJAF	Rule Activity file
JOBBREP	Report file

Example

If the index file of the Rule Activity file was damaged by a disk crash but the data file was not damaged (i.e., the actual information about invocations, etc., is accurate), use member JOBBJAF to reformat the index file and rebuild the index entries.

IOADBSR - Sort Records in an IOA Access Method Data Component

Utility IOADBSR sorts the records in an IOA Access Method file's data component in the same logical order as the records in the associated IOA Access Method file's index component.

Sorting an IOA Access Method file's data component improves the I/O performance.

Note The CONTROL-D Active User Report List file's data component can optionally be sorted automatically by the CONTROL-D monitor. See "Dynamic Sorting of the Active User Report List File" in Section 4 of this guide for more information.

Parameters

The following parameters can be specified to the utility in the EXEC statement which invokes the IOADBSR JCL procedure:

DBPREFA	High level dataset name qualifier (prefix) of the IOA Access Method file.
DBFILE	IOA Access Method file's dataset name suffix.
PARMLIB	The last qualifier of the IOA product library containing the input Sort Control statements.
M	Member name in the IOA product dataset containing the input Sort Control statements.

Sort Control Parameters

The following Sort Control parameters can be specified for dynamic sorting. Parameters that are not applicable to batch sorting with utility IOADBSR are indicated as such.

ENABLE	Whether or not dynamic sort processing should be disabled. This parameter is ignored by utility IOADBSR.
	Y (Yes) Enable dynamic sort processing.
	N (No) Disable dynamic sort processing.
TIMEFROM	Earliest time of day, in hhmm format, at which time dynamic sort processing may be performed. This parameter is ignored by utility IOADBSR.
TIMETO	Latest time of day, in hhmm format, at which time dynamic sort processing may be performed. This parameter is ignored by utility IOADBSR.
RBAOFS	Offset location within an index record of a four-byte field containing the address of the associated data record.
	This parameter is for internal use only and should not be modified unless instructed otherwise by Technical Support.

- INTERVAL** Interval, in hhmm format, at which dynamic sort processing will automatically start during the time range specified by parameters TIMEFROM and TIMETO. This parameter is ignored by utility IOADBSR.
- DELYBLK** Amount of time, as a four-digit value, in hundredths of seconds, that dynamic sort processing will wait before processing a new block. For example, a value of 0001, 0010 or 0100 will set the delay time to one hundredth, one tenth or one full second, respectively.
- START-WHEN-OVER** Limit of different Database file data component blocks referred to by all index records within any single index component block that will automatically trigger dynamic sort processing.
- This parameter is for internal use only and should not be modified unless instructed otherwise by Technical Support.
- STOP-WHEN-OVER** Maximum number of different data component blocks that can be referred to by all index records within any single index component block when dynamically sorting the index component block.
- The STOP-WHEN-OVER parameter is for internal use only and should not be modified unless instructed otherwise by Technical Support.

Example

```
// EXEC IOADBSR,DBPREFA=CTD,
//          DBFILE=ACT,
//          PARMLIB='CTD.PROD.PARM',
//          M=DBSRTPRM
```

Refer to sample job CTDUFSR in the CONTROL-D JCL library for an example of how to sort the data component of the CONTROL-D History User Report list.

IOADCPY - Recover an IOA Access Method File

Utility IOADCPY is used to recover an IOA Access Method file from the dual mirror image copy, and vice versa.

If an IOA Access Method file is used with dual file support, and either the working copy or the dual mirror image copy is corrupted or becomes unavailable, the damaged or missing file must be recovered. Utility IOADCPY restores the damaged or missing copy of the file from the copy that is still intact and available (including all file extents). If the dual copy is corrupted, the utility also resets the parameter in the control block of the IOA Access Method file that indicated a corrupted dual mirror image file (DUALBAD).

Activating the Utility

```
//COPY EXEC IOADCPY, DBFILE=ACT, SUFFIX=E000
```

When executing this utility, DD statement DAIN must be included in the JCL stream, and must specify the full name of the IOA Access Method file uncorrupted copy.

Note When using this utility, the activities of all the relevant applications should be stopped. See the relevant product section for more information on the recovery process.

Examples

If the dual mirror image copy of the CONTROL-D Active User Report List is corrupted, the DAIN DD statement should specify the name of the main copy:

```
//DAIN DD DISP=SHR, DSN=CTD.PROD.ACT.E000
```

If the main copy of the CONTROL-D Active User Report List is corrupted, the DAIN DD statement should specify the name of the dual mirror image copy:

```
//DAIN DD DISP=SHR, DSN=CTD.PROD.ACT.D000
```

Parameters

The following parameters can be specified to the utility in the EXEC statement which invokes the IOADCPY JCL procedure:

SUFFIX	Suffix of the dataset name of the uncorrupted file.
E000	If the dual mirror image copy is corrupted, the suffix of the main copy of the IOA Access Method file should be specified.
D000	If the main copy is corrupted, the suffix of the dual mirror image copy of the IOA Access Method file should be specified.
UNIT	Name of the DASD unit to be used when allocating new extents for the damaged or missing file.
VOL	List of one to six DASD volume serial numbers, enclosed in parenthesis and separated by commas, on which the new extents of the damaged or missing file can be allocated.

All volumes must be defined to MVS as being associated with the unit name specified in parameter UNIT above.

If VOL is not specified, the extents will be allocated using only parameter UNIT.

The above parameters are optional, and should only be used when it is necessary to override the UNIT and VOL (or UNITD and VOLD) parameters specified in the IOA Access Method file definition member when the IOA Access Method file was first created (member DEFxxx in the IOA INSTCTx library). After running the utility with these parameters, run utility IOADBF with FUNC=CHANGE specified to update the new values for these parameters in the IOA Access Method file control block.

Return Codes

0	Operation performed successfully
4	Missing DD statement or open error occurred for SYSPRINT file
8	Missing DD statement or open error occurred for input IOA Access Method file
12	Error in input parameters
16	Error in IOA Access Method file name
20	Error occurred in dynamic allocation
24	Internal error occurred
28	Insufficient storage
32	Missing DD statement or open error occurred for output IOA Access Method file
36	Error in input parameters - volume serial number was not supplied

See sample member CTDUFCPY in the CONTROL-D JCL library for more details on how to use this utility.

IOADIG - Verify Integrity of an IOA Access Method File Data Component

Utility IOADIG is used to verify the integrity of an IOA Access Method file's data component structure. If instructed, utility IOADIG will correct the errors that have been detected.

Note All file-related activity of the relevant product must be stopped before this utility is run:

- If the file being processed is a CONTROL-D file, all CONTROL-D monitors should be down when running this utility.
 - If the file being processed is a CONTROL-B file, all the CONTROL-B activities should be held, e.g., no CONTROL-B rules should be invoked while this utility is running.
-

Activating the Utility

```
//CHECK EXEC IOADIG,DBFILE=filename,FUNC=R|W
```

Parameters

The following parameters can be specified to the utility in the EXEC statement which invokes the IOADIG JCL procedure.

DBFILE	IOA Access Method file's identifier name (DBNAME) suffix.
FUNC	Function to be performed on the file.
	R Report errors detected.
	W Report and correct errors detected. As a precaution, it is recommended that a backup copy be made of the IOA Access Method file data component prior to executing utility IOADIG with parameter FUNC set to W.

Example

```
//CHECK EXEC IOADIG,DBFILE=HST,FUNC=R,
```

Refer to sample job CTDUFCHK in the CONTROL-D JCL library for an example of how to check the structural integrity of an IOA Access Method file data component.

IOADII - Verify Integrity of an IOA Access Method File Index Component

Utility IOADII is used to verify the integrity of an IOA Access Method file's index component structure. If errors are detected, the index component should be rebuilt using the appropriate index rebuilding utility (e.g., CTDDIB for CONTROL-D files or CTBDBIB for CONTROL-B files).

Activating the Utility

```
//CHECK EXEC IOADII,DBFILE=filename
```

Parameters

The following parameters can be specified to the utility in the EXEC statement which invokes the IOADII JCL procedure.

DBFILE IOA Access Method file's dataset name suffix.

The utility also provides the number of records in the file.

Example

```
//CHECK EXEC IOADII,DBFILE=GRP
```

Refer to sample job CTDUFCHK in the CONTROL-D JCL library for an example of how to check the structural integrity of an IOA Access Method file index component.

IOADLD - Load IOA Access Method File Contents

Utility IOADLD is used to load an IOA Access Method file's contents from a sequential file previously created by utility IOADUL.

Notes *All activity of the relevant product must be stopped before this utility is run. For example, if the file being processed is a CONTROL-D file, all CONTROL-D monitors should be down when running this utility.*

Utility IOADLD cannot be run on CONTROL-B files.

This utility is typically used when an IOA Access Method file is migrated to another DASD volume. In this case, the following are performed:

1. Utility IOADUL is run to unload the IOA Access Method file data component's records to a sequential dataset.
2. A new IOA Access Method file is allocated on a new DASD volume.
3. This new IOA Access Method file is formatted using utility IOADBF.
4. Utility IOADLD is executed to reload the IOA Access Method file data component records from the sequential dataset previously produced by IOADUL.
5. The appropriate index rebuilding utility (e.g., CTDDIB for CONTROL-D files) is executed to rebuild the IOA Access Method file index component based on the data component's record contents.

Activating the Utility

```
//LOAD EXEC IOADLD
```

Parameters

The following parameters can be specified to the utility in the EXEC statement which invokes the IOADLD JCL procedure:

SQFILE Input sequential file's dataset name suffix.

DBFILE Output IOA Access Method file's dataset name suffix.

Example

```
//LOAD EXEC IOADLD, SQFILE=ACTSEQ, DBFILE=ACT
```

Refer to sample job CTDUFRST in the CONTROL-D JCL library for an example of how to load an IOA Access Method file's data component records from a sequential dataset and then rebuild the IOA Access Method file's index component.

IOADPT - Print IOA Access Method File Content

Utility IOADPT prints IOA Access Method file data and index records in SNAP dump format.

Activating the Utility

```
//PRINT EXEC IOADPT
```

Parameters

The following parameters can be specified to the utility in the EXEC statement which invokes the IOADPT JCL procedure.

FUNC	Print function to be performed.
0	Print IOA Access Method file index and data components.
1	Print IOA Access Method file index component only.
DB0	Print IOA Access Method file index and data component control records.
DBFILE	IOA Access Method file's identifier name (DBNAME) suffix.
KEYFROM	Value of the record key from which to print. Optional.
KEYTO	Value of the record key until which to print. Optional.
RBAOFF	RBA offset in the index record key pointing to the data component record. The default setting for this parameter is 0.

This utility processes both the data and index components of a IOA Access Method file.

If KEYFROM and KEYTO parameters are not specified or are specified as blank, all records are printed.

Example

```
//PRINT EXEC IOADPT, FUNC=0, DBFILE=ACT
```

Refer to sample job CTUDFDPT in the CONTROL-D JCL library for an example of how to print IOA Access Method file record contents.

IOADUL - Unload IOA Access Method File Contents

Utility IOADUL is used to unload an IOA Access Method file data component's record contents to a sequential file.

Notes *All activity of the relevant product must be stopped before this utility is run. For example, if the file being processed is a CONTROL-D file, all CONTROL-D monitors should be down when running this utility.*

Utility IOADUL cannot be run on CONTROL-B files.

This utility is typically run when an IOA Access Method file is migrated to another DASD volume. In this case, the following are performed:

1. Utility IOADUL is run to unload the IOA Access Method file data component's records to a sequential dataset.
2. A new IOA Access Method file is allocated on a new DASD volume.
3. This new IOA Access Method file is formatted using utility IOADBF.
4. Utility IOADLD is executed to reload the IOA Access Method file data component records from the sequential dataset previously produced by IOADUL.
5. Utility the appropriate index rebuilding utility (e.g., CTDDIB for CONTROL-D files) is executed to rebuild the IOA Access Method file index component based on the data component's record contents.

Activating the Utility

```
//UNLOAD EXEC IOADUL, DBFILE=filename, SQFILE=seqfilename, RBAOFF=offset
[, FROMKEY=fkey] [TOKEY=tkey]
```

Parameters

The following parameters can be specified to the utility in the EXEC statement which invokes the IOADUL JCL procedure.

DBFILE	Input IOA Access Method file's dataset name suffix.
SQFILE	Output sequential file's dataset name suffix.
RBAOFF	RBA offset in the index record key pointing to the data component record. The default setting for this parameter is 0.
FROMKEY	Value of the record key from which to copy. Optional.
TOKEY	Value of the record key until which to copy. Optional.

If FROMKEY and TOKEY parameters are not specified or are specified as blank, all records are copied.

Example

```
//UNLOAD EXEC IOADUL,DBFILE=PRM,SQFILE=PRMSEQ
```

Refer to sample job CTDUFDUL in the CONTROL-D JCL library for an example of how to unload a IOA Access Method file data component to a sequential dataset.

Note *For a full list of the files this utility supports, see "The IOA Access Method" in Section 2 of this guide.*

IOALDNRS - Load Manual Conditions File

This utility creates and loads the Manual Conditions file which can be viewed via the IOA Manual Conditions screen described in Section 2 of the IOA product-specific user manuals. The utility is usually run every day after the New Day procedure finishes executing.

To identify conditions which should be placed in the Manual Conditions file, the utility first performs a search for all prerequisite IN conditions required for the submission of jobs/missions/reports on the particular day. The search for prerequisite IN conditions is performed by checking the job scheduling definitions of all jobs in the Active Jobs file and by checking the mission definitions of missions in the Active Missions file.

Once the prerequisite IN conditions have been identified, the utility eliminates non-manual conditions from the list to arrive at the list of manual conditions which need loading. A prerequisite IN condition is not included in the Manual Conditions file if it satisfies either of the following criteria:

- The prerequisite condition is added to the IOA Conditions/Resources file by an OUT or DO COND statement in a job/mission/report which is active that day.
- The prerequisite condition already exists in the IOA Conditions/Resources file.

All prerequisite IN conditions which do not meet the above criteria are assumed to be manual conditions and are placed in the Manual Conditions list.

Through parameters passed to the utility, it is possible to exclude the Active Jobs file and/or the Active Missions file from the search for manual conditions. Of course, if both files are excluded, no manual conditions will be found, even if they exist.

Every time the utility is activated, the Manual Conditions file is re-created and all previous conditions are automatically deleted from the file. It is possible to override this default by using parameter ADDMODE.

This utility can be used in conjunction with KeyStroke Language utility ADDMNCND. (Refer to "Maybe Jobs" in Section 6 of the CONTROL-M User Manual for details.)

The conditions to be included in the file can be limited by control statements supplied via DD statement DALNRIN (or SYSIN).

Parameters

IGNORE	Indicates conditions to be ignored during the building of the Manual Conditions file. Valid values are:
IN prefix	All conditions beginning with the specified prefix are ignored when the file is created.
OUT prefix	Any reference (i.e., additions) to the condition whose prefix is defined in the IGNORE statement appearing in production parameter OUT are ignored by the algorithm used to build the Manual Conditions file. An asterisk (*) signifies that all OUT parameters are ignored.

CODES prefix	Any reference (i.e., additions) to a condition whose prefix is defined in the IGNORE statement appearing in production parameter DO COND are ignored by the algorithm used to build the Manual Conditions file. An asterisk (*) signifies that all DO COND parameters are ignored.
WHEN prefix	Any reference (i.e., additions) to the condition whose prefix is defined in the IGNORE statement appearing in parameter WHEN DO COND is ignored by the algorithm which is used to build the Manual Conditions file. An asterisk (*) signifies that all WHEN DO COND parameters are ignored.
CTM STATUS ENDED	Manual conditions for jobs in the Active Jobs file which have finished executing (ended OK or NOTOK) are ignored by the algorithm used to build the Manual Conditions file. If this parameter is used, parameter CONTROLM must be set to YES.
CTD STATUS ENDED	Manual conditions for missions in the Active Missions file which have finished executing (ended OK or NOTOK) are ignored by the algorithm used to build the Manual Conditions file. If this parameter is used, parameter CONTROLD must be set to YES.
CONTROLM	Determines whether or not information in the CONTROL-M Active Jobs file should be used in the calculation which builds the Manual Conditions list. Optional. Default taken from installation parameters.
YES	Include CONTROL-M information in the calculation which builds the Manual Conditions file.
NO	Build the Manual Condition file based on CONTROL-D/V information only.
CONTROLD	Determines whether or not information in the CONTROL-M Active Missions file should be used in the calculation which builds the Manual Conditions file. Optional. Default taken from installation parameters.
YES	Include CONTROL-D information in the calculation which builds the Manual Conditions file.
NO	Build the Manual Conditions file based on CONTROL-M information only.
ADDMODE	If this parameter is specified, when the existing Manual Conditions file is deleted, only manual conditions from that file which were added (via the ADD command) to the IOA Conditions/Resources file are retained and appear in the new Manual Conditions file.

Note When CONTROL-M and CONTROL-D/V are both active, it is recommended that a shared control parameters member be used.

Activating the Utility (and Example)

```
// EXEC IOALDNRS
//DALNRIN DD *
IGNORE IN CICS
IGNORE IN IMS
IGNORE OUT DB-FILE
IGNORE CODES DB-FILE
CONTROLM YES
```

Return Codes

- | | |
|---|---|
| 0 | Operation performed successfully |
| 4 | File is full; no more space to add conditions |
| 8 | Critical error, list not created |

IOANOTE - Write a Message to the IOA Log File

This utility writes a message to the IOA Log file. It can be invoked as a started task (STC) or as a job step.

On terminals supporting color, messages written by IOANOTE appear on the Log screen in a different color than other messages, making them more readily identifiable.

The message to be written to the Log file is specified via parameter PARM.

Parameters

PARM Text of the message to be written.

Activating the Utility

As a job step:

```
// EXEC IOANOTE, PARM=text
```

As a started task:

```
S IOANOTE, PARM='text'
```

Return Codes

0	Operation performed successfully
4	No parameters passed to utility program
8	Operation failed

Example

```
// EXEC IOANOTE, PARM='PAYROLL APPLICATION MESSAGES BEGIN HERE'
```

IOATEST - Simulate Production Environment Events

This utility simulates various events in the production environment. For example, it can simulate step termination codes or cause the step to "wait" a specified interval.

Parameters

CMD	Issue an operator command. The command can be either an MVS/JES command or, when MVS is running under VM, a CP command. (CP commands must start with characters CP.) Maximum text length = 76 characters.
WTO	Write message to operator console. Maximum text length = 76 characters.
WTOR	Write message to operator console with reply. Maximum text length = 75 characters.
WTOM	Issue multi-line message. major is the primary message line. Subsequent minor lines are secondary message lines, delimited by a +++ line. A maximum of ten message lines are permitted.

Example

```
WTOM= major
      minor1
      minor2
      minor n
      +++
```

ROUT	Valid route code from 0 to 16 (0 = no route code). Default: all route codes.
DESC	Valid descriptor code from 0 to 16. Default: 0 (none).
CONID	Console ID to which the message is issued. Valid range: 0-999. Default: 0.
DEFDELAY	where n = number of 1/100 seconds. The program waits for a given time both before a WTO/WTOR/WTOM message or command is executed, and before the program ends. 1-4 digits. Default: 100 (one second).
WAIT	Cause the step to wait n seconds. Valid range: 0-999.
TERM=Cnnn	Condition code to be simulated. Valid range: 0-4095.
TERM=Unnnn	User abend code to be simulated. Valid range: 0-4095.
TERM=Snnn	System abend code to be simulated. Valid hexadecimal range: 000-FFF.

WTO, ROUT, DESC, CONID, CMD, WAIT, and TERM can also be specified in the PARM keyword of the EXEC job step. If this method is used, only the PARM parameters are executed in the following order: WTO, WAIT, TERM. When using PARM, DEFDELAY is always 0.

Examples

```
//EXEC IOATEST,  
// PARM='WTO=PROGRAM ERROR, WAIT=10, TERM=U0008'
```

Issues WTO message with the text "PROGRAM ERROR", waits 10 seconds, and terminates the step with user abend U0008.

```
// EXEC IOATEST  
//SYSIN DD *  
CMD=D T  
TERM=C0004  
//
```

Issues the operator command 'D T', and then terminates with condition code C0004.

IOAVERIFY - Verify IOA File Installation and Status

Utility IOAVERIFY verifies IOA product installation and produces reports on the utilization of various IOA files.

The actions performed by the utility depend on the input statements specified. The following input statements can be specified, each with its own parameters:

DISPLAY Controls the scope of display. Optional. Multiple DISPLAY statements can be specified. A DISPLAY statement is in effect until the next display statement. Valid parameters for this statement are:

ALL Both information and errors are displayed. Default.

ERROR Only error messages and warning messages are displayed.

VERIFY Displays and verifies information about the requested item(s). Optional, but at least one VERIFY or LIST statement should be specified or there will be no utility output. Multiple VERIFY statements can be specified. A VERIFY statement remains in effect until the next VERIFY statement is specified.

Valid formats for a VERIFY statement are:

VERIFY SYSTEM GENERAL

Displays general information about the system environment.

VERIFY PASSWORD

Displays CPU identification information.

VERIFY FILE filetype

Where:

filetype Indicates files to be displayed and verified in this run of utility IOAVERIFY. Valid values for filetype are:

ALL Includes all files listed below.

CKP/AJF CONTROL-M Active Jobs file.

AMF CONTROL-D Active Missions file.

RES IOA Conditions/Resources file.

NRS IOA Manual Conditions file.

LOG IOA Log file.

Notes CKP and AJF are two names for the same CONTROL-M file. Both names are supported.

If VERIFY FILE ALL is specified, the program determines which products are installed, and displays information on all relevant filetypes.

LIST Displays subsystem control blocks and IOA parameter tables. This statement helps IOA support personnel during problem determination. Optional, but at least one VERIFY or LIST statement should be specified or there will be no utility output. Multiple LIST statements can be specified. A LIST statement remains in effect until the next LIST statement is specified.

Valid formats for a LIST statement are:

LIST PARM parmname

or

LIST SUBSYS subsys-name

Where:

parmname	Names of parameter tables to be displayed.
ALL	Snapshot dumps of parameter tables for all IOA products installed
CTBPARAM	Snapshot dump of module CTBPARAM
CTDPARAM	Snapshot dump of module CTDPARAM
CTMPARAM	Snapshot dump of module CTMPARAM
CTOPARAM	Snapshot dump of module CTOPARAM
CTTPARAM	Snapshot dump of module CTTPARAM
CTVPARAM	Snapshot dump of module CTVPARAM
CTRPARAM	Snapshot dump of module CTRPARAM
ECSPARAM	Snapshot dump of module ECSPARAM
IOAPARAM	Snapshot dump of module IOAPARAM
IOAXPARAM	Snapshot dump of module IOAXPARAM
subsys-name	Name of subsystem whose control blocks are to be displayed. Optional. If omitted, the subsystem name specified in member IOAPARAM is used. A snapshot dump is taken of all the data areas relating to the subsystem.

END Terminates input to the utility. Optional. Only one END statement can be specified, at the end of the utility statements.

Activating the Utility

```
//VERIFY EXEC IOAVERIFY
//DAVRFIN DD *
```

Return Codes

0	Operation performed successfully
4	Warning message
8 and above	Severe error found

CONTROL-M Utilities

CTMBGRP - Convert Regular Scheduling Tables to Group Scheduling Tables

Utility CTMBGRP automatically converts regular scheduling tables into Group scheduling tables.

During conversion from a regular scheduling table to a Group table, the following actions are automatically performed by the utility:

- Entity is created for the table.
- A Group name is defined and added (in protected mode) to the GROUP field in each job scheduling definition.
- The format of the regular job scheduling definitions is modified to match the format of job scheduling definitions in Group scheduling tables (i.e., the SCHEDULE TAG field is added to, and supported in, the job scheduling definitions).

Note *Compress jobs within a table (i.e., job scheduling definitions with the MINIMUM and PDS fields defined) are converted, but these jobs are not assigned SCHEDULE TAG fields. Instead, their only scheduling criteria after the conversion are the MINIMUM and PDS criteria. However, a SCHEDULE TAG field can be manually added following conversion.*

Statements and Parameters

- All utility messages are written to the SYSPRINT file.
- The START/END conversion messages are also written to the IOA Log file.
- Parameters are passed to the utility via DD statement SYSIN.

Therefore, the datasets referenced by the following DD statements should be specified:

DALOG	References the IOA Log file.
SYSPRINT	References the utility message file. This file can be a SYSOUT file.
SYSIN	References the utility input file.

Keywords in the SYSIN statements must be specified beginning in column 1 and in the exact sequence indicated. Lines with an asterisk (*) in column 1 are treated as a comment line.

Parameters required for the conversion are:

IN-TABLE=old-table	Old (regular) scheduling table member
OUT-TABLE=new-table	New (Group) scheduling table member
GROUP=group-name	Group name (1 - 20 in length) to be assigned to the jobs in the scheduling table.

Multiple tables can be converted in a single run of the utility by passing multiple pairs of IN/OUT/GROUP lines. These pairs can be separated by a comment line.

Note The value assigned to OUT-TABLE cannot be identical to value of IN-TABLE.

Activating the Utility (and Example)

The following sample job to activate the utility can be found in member CTMBGRP in the CTM.JCL library:

```
//          JOB CARD PARAMETERS
//BUILD1   EXEC PGM=CTMGRB,REGION=0M
//STEPLIB DD DISP=SHR,DSN=IOA.Vnnn.LOAD
//DALOG    DD DISP=SHR,DSN=IOA.Vnnn.
*-----*
*          FIRST TABLE.          *
*-----*
IN-TABLE=OLD.DSN(OLDTABLE)
OUT-TABLE=NEW.DSN(NEWTABLE)
GROUP=NEW-GROUP-NAME
*-----*
*          SECOND TABLE.         *
*-----*
IN-TABLE=OLD.DSN2(OLDTAB2)
OUT-TABLE=NEW.DSN2(NEWTAB2)
GROUP=NEW-GROUP-NAME-2
*-----*
*          THIRD TABLE, ETC.     *
*-----*
```

Return Codes

- | | |
|----|--|
| 0 | All scheduling tables were successfully converted |
| 4 | All scheduling tables were successfully converted, but at least one of the tables contained at least one compress job. Compress jobs were not assigned SCHEDULE TAG fields during conversion |
| 8 | At least one table was not converted due to error. An error message identifying the unconverted table, and the error, was issued for each unconverted job |
| 16 | Initialization failed. No tables were converted. A message indicating the error was issued |
| 20 | Insufficient memory. No tables were converted |

CTMCAJF - Maintain the Active Jobs File

Utility CTMCAJF performs special maintenance functions on the Active Jobs file. The operator is warned in advance when the Active Jobs file is nearly full. The operator can use utility CTMCAJF to perform any of the following actions:

- Compress the file.
- Clean unwanted entries from the Status screen.
- Change the size of the file by copying the file to a different size file.

Statements and Parameters

The utility receives parameters, which designate the required function, via DD statement DACOPPRM (or SYSIN). Any of the following functions can be requested:

COMPRESS	Makes room in the Active Jobs file by physically deleting all jobs marked DELETED or ENDED OK.
CLEANUP	frees space in the Active Jobs file by logically deleting all jobs marked DELETED or ENDED OK. Although these jobs remain in the Active Jobs file until the New Day procedure is run, the space they occupy in the file can be overwritten by new information.
COPY	Copies the Active Jobs file to a different size file, generally a larger file which effectively increases the size of the Active Jobs file.

Notes *The COMPRESS and COPY functions cannot run in parallel to the CONTROL-M monitor. If the monitor is not shut down, the utility will terminate. Online users may remain logged on.*

The CLEAN function must run simultaneously with the CONTROL-M monitor. Do not shut down the monitor when running the utility with this function.

IGNORE/SELECT Statements

IGNORE/SELECT statements are optional and can only be specified for COMPRESS and CLEANUP functions. They identify jobs which should be deleted or not deleted, as follows:

IGNORE statements	Identify jobs which should not be deleted.
SELECT statements	Identify jobs which should be deleted.

Note *If a CLEANUP is being performed, IGNORE/SELECT statements are applied in addition to (not instead of) the default IGNORE/SELECT statements specified in the New Day procedure. New Day procedure default IGNORE/SELECT statements are applied by this utility.*

Up to 500 IGNORE/SELECT statements can be specified. Jobs are handled according to the first statement for which the criteria are met.

One or more of the following parameters may be specified in any IGNORE/SELECT statement (in any order):

GROUP groupname	Name of the group appearing in the job scheduling definition.
JOB NAME jobname/ MEMBER membername	<p>jobname Job name appearing in both the job statement and the JOB NAME field of the Status screen.</p> <p>membername Job scheduling definition name (name of the JCL file).</p> <p>jobname or membername (but not both) can be specified in one statement.</p>
STATUS status	<p>Specify one of the following job statuses:</p> <p>ENDOK</p> <p>ENDNOTOK</p> <p>HELD</p> <p>WAITSCHEd</p>
FROM date	Starting ODATE in yymmdd format
TO date	Ending ODATE in yymmdd format

Note A prefix can be specified instead of a full group name, job name or member name by placing an asterisk (*) at the end of the string (e.g., PROD*).

Specifying AutoEdit Variables and Functions

AutoEdit variables can be specified for any of the above parameters. These variables are especially useful for specifying relative date ranges in the FROM and TO parameters.

AutoEdit functions can be used to set variables and these variables can then be used in the parameters of the utility. The lines in which the variables are set must have an asterisk (*) in the first column of the line. Such lines are resolved by the AutoEdit facility, and are interpreted as comments by the utility. Regular comments can also be specified in lines with an asterisk in the first column.

Example

```
* IGNORE MEMBERS PREFIXED WITH BR14 WHICH WERE ORDERED
* WITHIN THE LAST THREE DAYS.
* %%SET %%F = %%CALCDATE %%DATE -3
  IGNORE MEMBER BR14*
    FROM %%F
```

Note The following AutoEdit terms cannot be used in the input for this utility: %%ODATE, %%OYEAR, %%OMONTH, %%ODAY, %%OWDAY, %%INCLIB, and %%INCMEM.

File Statements

File statements are mandatory for COPY functions. They cannot be specified for COMPRESS or CLEANUP functions. The following parameters must be specified for COPY functions:

OLDAJF	Name of the old Active Jobs file
NEWAJF	Name of the new Active Jobs file

Compressing the Active Jobs File

The Active Jobs file often contains entries for jobs which are no longer needed (jobs with a status of DELETED or ENDED OK). Compression makes room in the Active Jobs file by physically removing these unneeded entries from the file. Removed entries no longer appear on the Status screen.

Cleaning the Active Jobs File

Normally, the Active Jobs file is cleaned (i.e., unneeded job entries are deleted) once a day by the New Day procedure. By default, all jobs with status ENDOK which are not being held are deleted during this cleanup. The defaults can be overridden by using SELECT and IGNORE statements (which select or ignore jobs for deletion).

Utility CTMCAJF can be run as often as necessary to clean the Active Jobs file.

Unlike the cleanup performed by the New Day procedure, utility CTMCAJF does not actually delete the unneeded jobs. Instead, it marks the job entries as deleted so that they are ignored by CONTROL-M and so that, by default, the job entries do not appear in the Status screen. (To display these logically deleted jobs in the Status screen, specify Y in the DELETED field in the Show Option Window of the Status screen.)

These jobs will be physically deleted the next time the New Day procedure is run.

When utility CTMCAJF is run to clean up the Active Jobs file, it applies the cleanup criteria of the New Day procedure in addition to the cleanup criteria specified with the utility. If the cleanup criteria of the New Day procedure are not desired when running utility CTMCAJF, they should be overridden by SELECT/IGNORE statements specified in the utility.

Changing the Size of the Active Jobs File

Note *Online users should not access the CONTROL-M Online facility while the size of the Active Jobs file is being changed.*

To change the size of the CONTROL-M Active Jobs file, perform the following steps:

1. Change CONTROL-M installation parameter CKPSIZE in member CTMPARM to the required new Active Jobs file size.

Note *After step 2, every access to the Active Jobs file (Online, batch, or monitor) will try to utilize the new size and may abend. Therefore, all access to the Active Jobs file should be stopped, and the monitor should be shut down until the process is complete.*

2. Reassemble member CTMPARM using job CTMPARMJ.
3. Change parameter CKPSIZE in procedure CTMFRCKP to the new size of the Active Jobs file.

4. Run member FORMCKP in the IOA INSTCTM library after changing the dataset names (to avoid overriding the existing Active Jobs file and its backup file).
5. If you have a backup file, allocate a new backup file and run member FORMCKP against the new backup file. If you are working in dual checkpoint mode, run member FORMDCKP.

Note If you are working in dual checkpoint mode, the Mirror file must also be allocated with the new size before starting the CONTROL-M monitor. Otherwise, an abend of the CONTROL-M monitor may occur.

6. Run utility CTMCAJF. The utility copies the current Active Jobs file to the new Active Jobs file.
7. Rename the old Active Jobs file (and its backup file - suffix BKP).
8. Rename the new Active Jobs file (and its backup file - suffix BKP) to the former names of the old files.
9. Restart the CONTROL-M monitor and resume operation. The utility must terminate with a return code of 0.

Activating the COMPRESS Function

```
//COMP EXEC CTMCAJF
      COMPRESS
//
```

Activating the CLEANUP Function

```
//CLEANUP EXEC CTMCAJF
      CLEANUP
//
```

Activating the COPY Function (and Example)

```
//COPY EXEC CTMCAJF,
//      OLDAJF='CTM.PROD.CKP',      OLD ACTIVE JOBS FILE
//      NEWAJF='CTM.PROD.NEWCKP'    NEW ACTIVE JOBS FILE
      COPY
//
```

Return Codes

0	Operation performed successfully
other	Operation failed

Examples

```
//COMP EXEC CTMCAJF
      COMPRESS
      IGNORE STATUS WAITSCHED
      SELECT GROUP NIGHT-BATCH
```

In the above example, all jobs from the night batch, except those with a WAIT SCHEDULE status, are deleted.

CTMJSA - Job Statistics Accumulation

General

To use the Simulation and Forecasting facility the job's statistical data must be periodically retrieved from statistical messages in the IOA Log and placed and accumulated in a special statistical data file called the CONTROL-M Job Execution Statistics file. This is done using utility CTMJSA.

Parameters

FROMDATE Date from which statistical data are accumulated from the Log. The date can be a specific date (in mmddy, ddmyy, or yymmdd format, depending on the site standard), or the date can be a relative negative number of days from the date defined by the ODATE parameter.

TODATE Date up to which statistical data are accumulated from the Log.

ODATE Original date (for reference by relative FROMDATE/TODATE parameters).

Activating the Utility

It is highly recommended that the utility be activated periodically, for example, once a day, or once a week. The following is an example of a daily accumulation:

```
//D4JSA JOB...
// EXEC CTMJSA
   ODATE %%OMONTH.%%ODAY.%%OYEAR
   FROMDATE -3
   TODATE -1
//
```

The statistical accumulation for the above example will always work on jobs which have executed from three days ago until the day before today. There is no danger that the same job's data will be accumulated twice.

Note *The above sample job contains AutoEdit variables and must therefore be submitted directly by CONTROL-M.*

CONTROL-M accumulates data for up to 20 successful runs of a job. If a job finished executing NOTOK, or has been queued for re-execution by a \$EJ command, its statistical data is not accumulated.

CONTROL-B Utilities

CTBABI - Initialize Active Balancing File

Utility CTBABI initializes both the Active Balancing file and its backup file, ABFBKP. Utility CTBABI should be activated whenever these files are reformatted by utility IOADBF.

Parameters

The following parameter must be specified in the EXEC statement which invokes the CTBABI JCL procedure.

DBFILEB	Name of the file to initialize
ABF	Active Balancing file
ABFBKP	Backup file ABFBKP of the Active Balancing file

Activating the Utility

```
// EXEC CTBABI
```

Return Codes

0	Operation successful
other	Operation failed

Example

```
// EXEC CTBABI, DBFILEB=ABF
```

This utility is used by jobs CREFABF and CREFBKP when these files are created at time of installation. See these jobs for additional examples.

CTBCMP - Compile Rule Definitions

Utility CTBCMP checks the syntax of rule definitions. It is especially useful for checking rule definitions created via a standard editor.

Parameters

Parm='-irulename'

Name of the rule definition which must be checked for syntax errors. The rule name is preceded by the -i prefix (which can be uppercase or lowercase). The rule name and the -i prefix must be specified between quotes.

Activating the Utility

```
// EXEC CTBCMP, PARM='-IRULENAME'
```

Return Codes

0	No syntax errors were found
other	Syntax errors were found

Example

```
// EXEC CTBCMP, PARM='-ICOMPTOTS'  
//DABRUL DD DISP=SHR, DSN=CTB.PROD.RULES
```

CTBDBIB - Rebuild a CONTROL-B Index File

Utility CTBDBIB rebuilds index files for various CONTROL-B data files. The new index files correspond to information contained in the data files.

This utility rebuilds the indexes of a specific data file. Previous contents of the index file are erased.

Note *The index file that is to be rebuilt must be allocated and formatted prior to activating the utility.*

This utility should be used:

- When the space remaining for the existing index file is insufficient. In this case:
 - Delete the index file.
 - Redefine the index file with the required amount of space.
 - Format the index file with utility IOADBF.
 - Rebuild the index file with utility CTBDBIB.
- In case of index file integrity problems: index file integrity problems can be detected using utility CTBDBII, which is described later in this section.
- In case of data file integrity problems: correct the data file and reindex it with utility CTBDBIB to reproduce the corresponding index file.

Parameter

The following parameter can be specified in the EXEC statement which invokes the CTBDBIB JCL procedure.

DBFILE Name of data file to be processed.

Note *SYSIN parameters are provided in the keyxxx members in the PARM library for your convenience. File name specifications are hard-coded (predefined) in these members.*

Activating the Utility

```
// EXEC CTBDBIB
```

Return Codes

0	Operation performed successfully
other	Operation failed

Example

```
//      EXEC CTBDBIB, DBFILE=GRPD  
//SYSIN DD DSN=CTB.PROD.PARM(KEYGRP), DISP=SHR  
//DAGRP DD DSN=CTB.PROD.GRPI, DISP=SHR
```

Utility CTBDBIB is used by the following jobs in the JCL library. Refer to these jobs for additional examples:

JOBBDBM	Rebuilds the index of the Database Variable Basic file.
JOBBDBV	Rebuilds the index of the Generation file.
JOBBGRP	Rebuilds the index of the Group file.
JOBBJAF	Rebuilds the index of the Rule Activity file.
JOBBREP	Rebuilds the index of the Report file.

CTBDBVCG - Change Number of Variable Generations

This utility changes the maximum number of generations specified for one or more Database Variables. It is useful whenever the number of generations must be changed but is most often used when changing the status (mode) of variables from test to production. The utility can be invoked in batch mode as described in "Activating the Utility."

The number of generations to be retained in the database can be increased or decreased. The content of existing variable generations can be retained or deleted. All other relevant aspects of the variable(s) are retained, including definitions and group information.

It is possible to change the number of generations of all variables, groups of variables (with a common group name prefix), or one or more variables of the same group (with a common variable name prefix) in one activation of utility CTBDVCG.

Parameters

The following parameters can be specified in the EXEC statement which invokes the CTBDBVCG JCL procedure. These parameters dynamically specify source (input) file names and indicate whether the current content of existing generations should be retained or deleted.

DIMOD		Prefix of the third qualifier of the name of the source (input) Database Variable Basic Data file.
DIVAR		Prefix of the third qualifier of the name of the source (input) Database Variable Generation Data file.
DIGRP		Prefix of the third qualifier of the name of the source (input) Group file.
COPY	YES	Retain current contents of existing generations. Default.
	NO	Delete current contents, if any, of existing generations.

Parameters are also supplied via DD statement SYSIN. The following parameters are mandatory:

GROUP	Group name (or prefix) of variables to be affected. A group name prefix can be specified only if parameter VARIABLE contains an asterisk (*).
-------	---

Example

GROUP=INV	Variables of group INV are changed.
GROUP=INV*	Variables of all groups that begin with INV (e.g., INV,INVENTORY, INVESTMENTS) are changed.

To change the number of variables for all groups, the SYSIN input file should be empty (and no GROUP or VARIABLE names should be specified).

Note Parameter *GROUP* may be specified only if an asterisk (*) was specified in parameter *VARIABLE*.

VARIABLE Variable name (or prefix) of variable(s) whose number of generations should be changed.

Examples

GROUP=INV Variable TOT of group INV is changed.
VARIABLE=TOT

GROUP=INV Variables of group INV that begin with
TOTVARIABLE=TOT* TOT (e.g., TOT, TOTAL, TOT01, TOT02) are changed.

NUMGEN New number of generations to be specified for the variable. If not specified, NUMGEN is set to the default number of generations in System Variable SYSDBVARGENS.

Activating the Utility

```
//      EXEC  CTBDBVCG, DIMOD=mod, DIVAR=var, DIGRP=grp, COPY=yes/no
//SYSIN DD *
      GROUP=group
      VARIABLE=variable
      NUMGEN=NEW-MAXIMUM-#-OF-GENERATIONS
//
```

Return Codes

0: Operation performed successfully
other Execution error, operation failed

Example

```
//      EXEC  CTBDBVCG, DIMOD=MGMT, DIVAR=ACTG, DIGRP=TAX, COPY=YES
//SYSIN DD *
      GROUP=BONDS
      VARIABLE=COST*
      NUMGEN=24
//
```

CTBDBVCP - Copy Variables or Groups to Another File

This utility copies the contents of specified group/variable files (i.e., data from source files) to another set of group/variable files (i.e., target files). All relevant aspects of the variables are copied, including definitions, generations, and group information.

The sizes of the source and target files may differ. Therefore, utility CTBDBVCP is useful for increasing the size of variable files. Usually, this utility copies variables from source files to larger target files.

Copying variables is composed of the following stages:

- Create and format a new set of group/variable files (target files).
- Copy the data from the source files to the newly-created target group/variable files.
- Delete the source files.

It is possible to copy all variables, groups of variables (with a common group name prefix), or one or more variables of the same group (with a common variable name prefix) in one activation of utility CTBDBVCP.

Parameters

The following parameters can be specified in the EXEC Statement which invokes the CTBDBVCP JCL procedure. These parameters dynamically specify the prefix of the third qualifier for the source and destination files listed below:

DIMOD	Source (input) Database Variable Basic Data file.
DIVAR	Source (input) Database Variable Generation Data file.
DIGRP	Source (input) Group file.
DOMOD	Destination (output) Database Variable Basic file.
DOVAR	Destination (output) Database Variable Generation file.
DOGRP	Destination (output) Group file.
REPLACE	Flag which indicates whether or not the source variable should overwrite an existing variable of the same name in the target.
	YES Target variable should be overwritten.
	NO Target variable should not be overwritten.
COPY	Flag which indicates whether entire records or only variables should be copied.
	YES Copy entire records from the Group file.
	NO Copy only variables.

Parameters are also supplied via DD statement SYSIN. The following parameters are mandatory:

GROUP	Group name (or prefix) of variables to be copied. A group name prefix may be specified only if parameter VARIABLE contains an *.
-------	--

Examples

GROUP=INV Variables of group INV are copied.

GROUP=INV* Variables of all groups that begin with INV (e.g., INV, INVENTORY, INVESTMENTS) are copied.

To copy all groups, the SYSIN input file should be empty (and no GROUP or VARIABLE names should be specified).

Note Parameter GROUP may be specified only if '*' was specified in parameter VARIABLE.

VARIABLE Variable name (or prefix) to be copied.

Examples

GROUP=INV Variable TOT of group INV is copied.
VARIABLE=TOT

GROUP=INV Variables of group INV that begin with TOT (e.g.,
VARIABLE=TOT* TOT, TOTAL, TOT01, TOT02) are copied.

NUMGEN Number of generations to be created for the variable during the copy. If not specified, the number matches the source variable's number of generations.

FROMGEN Most recent generation to copy from the source to the target. If FROMGEN is not specified, CONTROL-B begins copying with the current (most recent) source generation.

TOGEN Oldest generation to copy from the source to the target. If TOGEN is not specified, CONTROL-B begins copying with the current (most recent) source generation.

Example

```
GROUP=INV
VARIABLE=TOT
NUMGEN=20
FROMGEN=5
TOGEN=10
```

Assume the source variable has at least 10 generations.

The target variable can have a maximum of 20 generations. The first six generations of the target variable will contain the values of the 5th through 10th generations of the source variable.

Note If no input statement is specified (i.e., the input file is empty), all records from the source files are copied.

Activating the Utility

```
// EXEC CTBDBVCP
```

Return Codes

0	Copy operation performed successfully
other	Descriptive execution error, copy operation failed

Example

```
// EXEC CTBDBVCP, DIMOD=MOD, DIVAR=VAR, DIGRP=GRP,  
// DOMOD=TARGMOD, DOVAR=TARGVAR, DOGRP=TARGGRP  
//SYSIN DD *  
GROUP=CURRENCY  
VARIABLE=TOT*  
//
```

Refer to sample member JOBDBVCP in the JCL library for additional information.

CTBDBVDL - Delete Variables/Groups from Variable/Group Files

Utility CTBDBVDL deletes specified records from CONTROL-B Database Variable files.

Parameters

The following parameters can be specified in the EXEC statement which invokes the CTBDBVDL JCL procedure. These parameters provide dynamic specification of source and destination file names:

DMOD	Prefix of the third qualifier of the name of the Database Variable Basic file.
DVAR	Prefix of the third qualifier of the name of the Database Variable Generation file.
DGRP	Prefix of the third qualifier of the name of the Group file.
DELGR	Flag which indicates whether or not the entire group should be deleted.
YES	Delete the entire group if no variables exist for this group.
NO	Do not delete the entire group. Default.

Parameters are also supplied via DD statement SYSIN:

GROUP	Group name (or prefix) of variables to be deleted. A group name prefix can be specified only if parameter VARIABLE contains an *.
-------	---

Example

GROUP=INV	Variables of group INV are deleted.
GROUP=INV*	Variables of all groups that begin with INV (e.g., INV, INVENTORY, INVESTMENTS) are deleted.

Note GROUP=* cannot be specified.

VARIABLE	Variable name (or prefix) to be deleted.
----------	--

Note Parameter VARIABLE may be specified only if '*' was not specified with parameter GROUP.

Example

GROUP=INV VARIABLE=TOT	Variable TOT of group INV is deleted.
GROUP=INV VARIABLE=TOT*	Variables of group INV that begin with TOT (e.g., TOT, TOTAL, TOT01, TOT02) are deleted.

Activating the Utility

```
// EXEC CTBDBVDL
```

Return Codes

0 Operation performed successfully
other Operation failed

Example

```
//        EXEC CTBDBVDL, DMOD=MOD, DVAR=VAR, DGRP=GRP, DELGR=YES  
//SYSIN DD *  
          GROUP=CURRENCY  
          VARIABLE=TOT*  
//
```

Refer to sample member JOBDBVDL in the JCL library for additional information.

CTBDBVIG - Check the Integrity of Variable/Group Files

Utility CTBDBVIG checks the integrity of the Database Variable Basic and Generation files. The utility reports errors which are detected in the logical organization of these files. If indicated, CTBDBVIG corrects errors that are detected. file integrity errors are listed in the sysout of the job.

Parameter

The following parameter can be specified in the EXEC statement which invokes the CTBDBVIG JCL procedure:

ACT	Flag which indicates whether or not integrity errors should be corrected.
R	Report errors, but do not correct them. Default.
W	Report errors and correct them.

Notes *Index file integrity is not checked. Therefore, after running utility CTBDBVIG, run jobs JOBBDBM and JOBBDBV to rebuild the index files according to the corrected state of the data file.*

It is recommended that utility CTBDBVIG first be run without requesting corrections (ACT=R).

Activating the Utility

```
// EXEC CTBDBVIG
```

Return Codes

0	Operation performed successfully
other	Operation failed

Example

```
// EXEC CTBDBVIG,ACT=R
```


CTBFRM - Compress the Active Balancing File

Utility CTBFRM compresses the Active Balancing file. The utility usually runs as part of the New Day procedure before potential missions are placed as entries in the Active Balancing file, but can also be run as a single job, using the FORCE parameter.

The utility deletes entries from the Active Balancing file which will no longer be processed by CONTROL-B. The Active Balancing file is then compressed.

A backup of the Active Balancing file is created automatically by this utility. The default backup file name is ABFBKP.

Parameters

The following parameters can be specified in the EXEC statement which invokes the CTBFRM JCL procedure.

DBABFB=ABF Third qualifier of the Active Balancing file name.

DBBKPB=ABFBKP Third qualifier of Backup file name.

FORCE Y (Yes) Run the utility in the stand alone mode, without checking corresponding dates in member CTBDATE in the CONTROL-B PARM library. Refer to the CONTROL-B New Day procedure.

 N (No) Default value. The utility is run when invoked by the CONTROL-B New Day procedure. Default.

Activating the Utility

```
// EXEC CTBFRM, PARM='&FORCE'
```

Return Codes

0 Operation performed successfully

other Operation failed

Example

```
// EXEC CTBFRM, FORCE=N, DBABFB=ABF, DBBKPB=ABFBKP
```

Refer to sample member JOBFRM in the JCL library for additional information.

CTBJAFCP - Copies Entries in Rule Activity/Report Files

Utility CTBJAFCP copies invocation and report entries to other Rule Activity and/or Report files.

Parameters

FILE	File to be copied.
JAF	Only the Rule Activity file is copied.
REP	Only the Report file is copied.
ALL	Both the Rule Activity file and corresponding reports from the Report file are copied.

The following parameters can be specified in the EXEC statement which invokes the CTBJAFCP JCL procedure:

DBJAFT	Prefix of the third qualifier of the source (input) Rule Activity file name.
DBREPI	Prefix of the third qualifier of the source (input) Report file name.
DBJAFO	Prefix of the third qualifier of the destination (output) Rule Activity file name.
DBREPO	Prefix of the third qualifier of the destination (output) Report file name.

The following utility parameters are supplied via DD statement SYSIN. The suffix * is permitted at the end of any of these parameters. * represents any number of characters (including no characters).

INCLUDE/EXCLUDE Statements

The scope of the copy can be limited by one or more INCLUDE or EXCLUDE statements. Each INCLUDE/EXCLUDE statement contains job name prefixes which limit the entries which are copied. If one or more INCLUDE statements are specified, only entries specified in the INCLUDE statements are copied. If one or more EXCLUDE statements are specified, the entries specified in the EXCLUDE statements are not copied. There is an or relationship between multiple INCLUDE statements and between multiple EXCLUDE statements.

Parameter job name can be specified in INCLUDE/EXCLUDE statements. This parameter represents the job name.

If no INCLUDE/EXCLUDE criteria are specified, the entire file is copied.

CTBJAFCP parameters can be specified on several lines. In this case, each parameter should be listed on a separate line.

Note *If no input statement is specified (i.e., the input file is empty), all records from the source file are copied.*

Activating the Utility

```
// EXEC CTBJAFCP
//SYSIN DD *
//
```

Return Codes

8	Syntax error in the input parameters stream (SYSIN cards)
12	Insufficient memory to run the utility
16	Unable to open a CONTROL-B Database file
100+	Internal error

Example

```
// EXEC CTBJAFCP, FILE=ALL, DBJAFI=JAF, DBREPI=REP, DBJAFO
  =JAFNEW, DBREPO=REPNEW
//SYSIN DD *
  INCLUDE JOB2*
  EXCLUDE JOB22
  EXCLUDE JOB23*
//
```

To copy the files, perform the following steps:

1. Allocate a new set of Rule Activity and Report files via utility CTBCRDF.
2. Copy the existing files, or a portion of these files (e.g., only relatively new invocations), into the new files via utility CTBJAFCP.
3. Delete the old files.
4. Rename the new files to the names of the original files.

Refer to sample member JOBJAFCP in the JCL library for additional information.

CTBJAFDL - Deletes Entries From Rule Activity/Report Files

Utility CTBJAFDL cleans unnecessary entries from the Rule Activity file and/or the Report file.

Parameters

The following parameters can be specified in the EXEC statement which invokes the CTBJAFDL JCL procedure:

DBJAFB	Prefix of the third qualifier of the Rule Activity file name.
DBREPB	Prefix of the third qualifier of the Report file name.
PRINT	Flag which indicates whether or not deleted entries are printed.
PRINT	Information about deleted entries is printed.
NOPRINT	Information about deleted entries is not printed.

The following parameters are supplied via DD statement SYSIN. The suffix * is permitted at the end of any of these parameters. * represents any number of characters (including no characters).

DAYS n: If 0 is specified or if DAYS is specified without any accompanying subparameters, utility CTBJAFDL operates on all entries in the Rule Activity file/Report file. n is the retention period before deletion of entries from the Rule Activity file/Report file. n cannot be a negative number. To delete all entries, specify DAYS=0. Mandatory.

INCLUDE/EXCLUDE: The scope of the deletion can be limited by one or more INCLUDE or EXCLUDE statements. Each INCLUDE/EXCLUDE statement contains job name prefixes which limit the entries which are deleted. If one or more INCLUDE statements are specified, only entries specified in the INCLUDE statements are deleted. If one or more EXCLUDE statements are specified, the entries specified in the EXCLUDE statements are not deleted. There is an OR relationship between multiple INCLUDE statements and between multiple EXCLUDE statements.

Parameter JOB NAME can be specified in INCLUDE/EXCLUDE statements. This parameter represents the job name.

CTBJAFDL parameters may be specified on several lines. In this case, each parameter should be listed on a separate line.

Activating the Utility

```
// EXEC CTBJAFDL
//SYSIN DD *
DAYS n
[INCLUDE {PARAMETERS}]
[EXCLUDE {PARAMETERS}]
//
```

Return Codes

0	Operation performed successfully
other	Operation failed

Example

```
// EXEC CTBJAFDL,DBJAFB=JAF,DBREPB=REP,PRINT=PRINT
//SYSIN DD *
DAYS 4
INCLUDE JOB2*
EXCLUDE JOB22
EXCLUDE JOB23*
//
```

CTBJAFIG - Check the Integrity of Rule Activity/Report Files

Utility CTBJAFIG checks the integrity of the Rule Activity and Report files. The utility reports errors detected in the logical organization of these files. If requested, utility CTBJAFIG corrects the errors that are detected. file integrity errors are listed in the sysout of the job.

Parameters

The following parameters can be specified in the EXEC statement which invokes the CTBJAFIG JCL procedure:

FILE	File to be checked/corrected.
JAF	Rule Activity file. Default.
REP	Report file.
BTH	Both of the files (without logical links).
ALL	Both the Job Activity and Report files (and their logical links).
C	Flag to indicate whether or not integrity errors should be corrected.
Y (Yes)	Correction is requested.
N (No)	Correction is not requested. Default.

Notes Index file integrity is not checked. Therefore, after running utility CTBJAFIG with option C=Y, run jobs JOBBJAF and JOBBREP to rebuild the index files based on the corrected data file.

It is recommended that utility CTBJAFIG first be run without requesting correction (C=N).

Activating the Utility

```
// EXEC CTBJAFIG
```

Return Codes

0	Operation performed successfully
other	Operation failed

Example

```
// EXEC CTBJAFIG, FILE=ALL, C=N
```

Refer to sample member JOBJAFIG in the JCL library for additional information.

CONTROL-D/V Utilities

CTDCAMF - Compress Active Missions File

This utility can compress the Active Missions file or change its size.

Message CTDF63W is issued if the Active Missions file is about to become full. Message CTD533S is issued if the Active Missions file does become full. There are two solutions: Increase the size of the file or compress the file.

Utility CTDCAMF receives control parameters via DD statement DACOPPRM (or SYSIN). The parameters designate the required function (compress or copy).

Note This utility cannot run simultaneously with the CONTROL-D monitor. Results will be unpredictable if the monitor is not shut down. Online users can remain logged on.

Compressing the Active Missions File

The Active Missions file contains entries for jobs which are no longer required (e.g., missions whose status is DELETED or ENDED OK). By removing the unneeded entries from the file, utility CTDCAMF provides space for new entries.

Activating the Utility

```
//COMP EXEC CTDCAMF
  COMPRESS
//
```

Refer to sample job CTDCAMF in the CONTROL-D JCL library for an example of how to implement this utility.

Changing the Size of the Active Missions File

Note Online users should not access the Online facility while the size of the Active Missions file is being changed.

Perform the following steps to change the size of the Active Missions file:

1. Change CONTROL-D Installation parameter AMFSIZE in member CTDPARM to the required new Active Missions file size. Assemble the CONTROL-D Installation parameters. From this point on, every access to the Active Missions file (Online or batch) will use the new size and may abend. Therefore, it is recommended to stop all access to the Active Missions file until the process is complete.
2. Change parameter AMFSIZE in procedure CTDFRAMF to the new size of the Active Missions file.

3. Run member FORMAMF in the IOA INSTCTD library. Change the dataset names so you do not override the existing Active Missions file and its backup file (e.g., parameter TYPE should be changed to NEWNAME).
4. Run utility CTDCAMF with the following parameters:

```
//EXEC CTDCAMF,  
// OLDAMF='CTD.PROD.AMF', OLD ACTIVE MISSIONS FILE  
// NEWAMF='CTD.PROD.NEWAMF' NEW ACTIVE MISSIONS FILE  
COPY  
//
```

Utility CTDCAMF copies the current Active Missions file to the new Active Missions file. The utility must terminate with a return code of 0.

5. Rename the old Active Missions file (and its backup file - suffix AMB).
6. Rename the new Active Missions file (and its backup file - suffix AMB) to the former names of the old files.
7. Restart the CONTROL-D monitor and resume operation.

CTDCATF - Compress Active Transfer File

This utility can compress the Active Transfer file or change its size.

If the Active Transfer file becomes full, there are two solutions: Increase the size of the file or compress the file.

Utility CTDCATF receives control parameters via the EXEC statement. The parameters designate the required function (compress or copy).

Note *This utility cannot run simultaneously with the CONTROL-D monitor. Results will be unpredictable if the monitor is not shut down. Online users can remain logged on.*

Compressing the Active Transfer File

The Active Transfer file contains entries for packets which are no longer required (e.g., packets whose status is DELETED or TRANSFERRED OK). By removing the unneeded entries from the file, utility CTDCATF provides space for new entries.

Example

```
//COMP EXEC CTDCATF
//
```

Changing the Size of the Active Transfer File

Note *It is highly recommended that online users do not access the Online facility while the size of the Active Transfer file is being changed.*

Perform the following steps to change the size of the Active Transfer file:

1. Change CONTROL-D installation parameter ATFBLK in member CTDPARM to the required new Active Transfer file size. Assemble the CONTROL-D installation parameters. After you do this, every access to the Active Transfer file (online or batch) will use the new size and may abend. Therefore, it is recommended to stop all access to the Active Transfer file until the process is complete.
2. Change parameter ATFSIZE in procedure CTDFRATF to the new size of the Active Transfer file.
3. Run member FORMATF in the IOA INSTCTD library. Change the dataset names so you will not override the existing Active Transfer file and its backup file.
4. Run utility CTDCATF with the following parameters:

```
// EXEC CTDCATF,P=COPY,
// INATF='CTD.PROD.ATF',    OLD ACTIVE TRANSFER FILE
// OUTATF='CTD.PROD.NEWATF' NEW ACTIVE TRANSFER FILE
//
```

Utility CTDCATF copies the current Active Transfer file to the new Active Transfer file. The utility must terminate with a return code of 0.

5. Rename the old Active Transfer file (and its backup file - suffix ATB).
6. Rename the new Active Transfer file (and its backup file - suffix ATB) to the former names of the old files.
7. Restart the CONTROL-D monitor and resume operation.

CTDCA2P - Copy Active User Report File to Permanent

This utility copies new user/report/job/category entries from the Active User Report List file to the Permanent User Report List file. The Permanent User Report List file contains, for each user, a list of all the reports which he/she/she may receive at any point in time. When a new report is defined, or when a user becomes a report recipient (e.g., because of new data from the application), this information is placed in the Active User Report List file after decollation.

This utility copies new entries from the Active User Report list file to the Permanent User Report List file. In this way, the list of reports which the user may get is maintained automatically.

It is recommended that utility CTDCA2P be run on a daily basis. See "Permanent User (Report List) File" in Section 4 of this guide.

INCLUDE/EXCLUDE Statements

The scope of the copy can be limited by one or more INCLUDE or EXCLUDE statements. Each INCLUDE/EXCLUDE statement contains parameters which limit the user/report list entries which will be copied. See "Record Selection Logic (INCLUDE/EXCLUDE Statements)" in this section for information about prefixing, masking, and the INCLUDE and EXCLUDE logic used to determine record selection.

Parameters in INCLUDE/EXCLUDE Statements

The parameters which can be specified in INCLUDE/EXCLUDE statements are:

JOB NAME	Job name
USER	User name
REPNAME	Report name
CATEGORY	Category name

Activating the Utility

```
// EXEC CTDCA2P
  [INCLUDE {parameters}]
  [EXCLUDE {parameters}]
```

The new report entries are assigned the current CONTROL-D working date (not the computer date).

This utility may be run simultaneously with the CONTROL-D monitor.

Return Codes

0	Operation performed successfully
8	I/O error, execution stopped

Example 1

```
// EXEC CTDCA2P
   INCLUDE REPNAM=PROD-JCL
```

The following is the status of the files before the utility is run:

Active User Report List File

--USER--	-----REPORT-----	---JOB--	----CATEGORY----
ACCTSRCV	ACCOUNTS-RECEIV-REPT	D4ACCRCV	QUARTERLY
ACCTSRCV	ACCOUNTS-RECEIV-REPT	D4ACCRCV	WEEKLY
BRANCH01	DAILY-BALANCE-REPORT	D4DAILYB	DAILY
BRANCH02	DAILY-BALANCE-REPORT	D4DAILYB	MONTHLY

Permanent User Report List File

--USER--	-----REPORT-----	---JOB--	----CATEGORY----
BRANCH01	DAILY-BALANCE-REPORT	D4DAILYB	DAILY
BRANCH02	DAILY-BALANCE-REPORT	D4DAILYB	MONTHLY
BRANCH03	DAILY-BALANCE-REPORT	D4DAILYB	DAILY

The following is the status of the Permanent User Report List file after the utility is run:

--USER--	-----REPORT-----	---JOB--	----CATEGORY----
01ACCTSRCV	ACCOUNTS-RECEIV-REPT	D4ACCRCV	QUARTERLY
01ACCTSRCV	ACCOUNTS-RECEIV-REPT	D4ACCRCV	WEEKLY
02BRANCH01	DAILY-BALANCE-REPORT	D4DAILYB	DAILY
02BRANCH02	DAILY-BALANCE-REPORT	D4DAILYB	MONTHLY
03BRANCH03	DAILY-BALANCE-REPORT	D4DAILYB	DAILY

- 01 These entries were copied from the Active User Report List file since they did not exist on the Permanent User Report List file.
- 02 These entries were not copied from the Active User Report List file since they already existed in the Permanent User Report List file.
- 03 This entry was not modified.

CTDCCU - Delete Unneeded CDAM Files

This utility identifies CDAM files not referenced by the Active User Report List file (by the SYSDATA entries) so that they can be deleted, and optionally deletes them.

CDAM files created using JCL parameter SUBSYS may not always be decollated. In these cases, CTDELRP does not delete these CDAM files since they are not referenced by the Active User Report List file (ACTUSR). This utility identifies/deletes these files.

Parameters

DATE Date range for selecting CDAM files. Format: fromdate - todate. fromdate is mandatory; todate is optional.

Date is in format mmddy, ddmyy or yymmdd, depending on the site standard. todate may not exceed the run date minus one. If todate is not specified, it defaults to the specified fromdate.

PREFIX CDAM prefix. At least one PREFIX command is required. Multiple PREFIX commands are supported.

SIMULATION Whether or not CDAM file should be deleted during the run of the utility.

YES Report on CDAM files to be deleted. Do not perform the deletion. Default.

NO Report on CDAM files to be deleted. Perform the deletion.

Note It is recommended that utility CTDCCU be run first in simulation mode to verify that only the correct files are to be deleted.

VERIFY-IN-HISTORY Whether or not the History User Report file should be searched for references to CDAM files not in the Active User file.

YES Search the History User Report list file for references to CDAM files not in the Active User file. Issue a message that indicates whether or not such references were found. This parameter does not influence the identification of the CDAM files to be deleted.

NO Do not search the History User Report List file.

Activating the Utility

```
// EXEC CTDCCU
   DATE=fromdate [- todate]
   PREFIX=prefix
   [SIMULATION={YES/NO}]
   [VERIFY-IN-HISTORY={YES/NO}]
```

Example 1

In the following example, any CDAM file satisfying all of the following criteria is listed in the report but not deleted:

- Prefixed CTDRSV.
- Created August 21 - 30, 1995 (USA format).
- Not referenced by the Active User Report List file.
- Still on disk.

```
// EXEC CTDCCU  
    DATE=082196-083096  
    PREFIX=CTDRSV
```

Example 2

In the following example, any CDAM file satisfying all of the following criteria is deleted:

- Prefixed by CTDRSV or CTDALL.
- Created 21 - 30 August 1995 (International format).
- Not referenced by the Active User Report List file.
- Still on disk.

Deleted CDAM files are checked against the History Report list file. The report indicates whether or not the deleted CDAM files are referenced in the History Report List file.

```
// EXEC CTDCCU  
    DATE=210896-300896  
    PREFIX=CTDRSV  
    PREFIX=CTDALL  
    SIMULATION=NO  
    VERIFY-IN-HISTORY=YES
```

CTDCLHIS - Clean History User Report List File

This utility cleans entries from the History User Report List file. The History User Report List file contains entries describing backed up user reports. Each report has a retention period. The utility scans the History User Report List file for all the reports whose retention period has passed (i.e., the backup is no longer needed) and deletes them from the History User Report List file.

This utility produces a list of all tapes which are no longer needed and can therefore be returned to the global tape pool.

MODE Parameter

This optional parameter in the EXEC Statement determines whether utility CTDCLHIS is run in Production mode or Simulation (Test) mode. If the parameter is not specified, the default is Production mode. It is recommended that this utility be run in Simulation mode prior to being run in Production mode.

PROD	The utility runs in Production mode, changing and deleting VSAM and CDAM files which satisfy the requirements specified by other parameters. Default.
TEST	The utility runs in Simulation mode, performing analysis and reporting but not changing or deleting any files.

Activating the Utility

```
// EXEC CTDCLHIS [MODE=TEST|PROD]
```

Return Codes

0	Operation performed successfully
4	Operation performed successfully (for HSM backup utility)
8	Execution error
11	Operation performed successfully (in TEST mode)
12	Invalid input parameters

CTDCP2A - Copy Permanent User Report File to Active

This utility copies job/user/report/category entries from the Permanent User Report List file to the Active User Report List file. This process is performed once a day to create new empty entries (in WAIT DECOLLATION status) for each job/user/report/category/odate (original scheduling date).

Any job/user/report/category/odate entries which are already present on the Active User Report List file are not copied (i.e., are not duplicated).

INCLUDE/EXCLUDE Statements

The scope of the copy can be limited by one or more INCLUDE or EXCLUDE statements. Each INCLUDE/EXCLUDE statement contains parameters which limit the user/report list entries which will be copied. See "Record Selection Logic (INCLUDE/EXCLUDE Statements)" earlier in this section for information about prefixing, masking, and the INCLUDE and EXCLUDE logic used to determine record selection.

Parameters in INCLUDE/EXCLUDE Statements

The parameters which can be specified in INCLUDE/EXCLUDE statements are:

JOB NAME	Job name
USER	User name
REPNAME	Report name
CATEGORY	Category name

Activating the Utility

```
// EXEC CTDCP2A
  [INCLUDE {PARAMETERS}]
  [EXCLUDE {PARAMETERS}]
```

The new report entries are assigned the current CONTROL-D working date (not the computer date). It is possible to specify a specific date via parameter PARM=date (format mmddy, ddmyy or yymmdd, depending on the site standard).

The utility may be run simultaneously with the CONTROL-D monitor.

Return Codes

0	Operation performed successfully
8	I/O error, execution stopped

Example

```
// EXEC CTDCP2A, PARM=date
      INCLUDE REPNAME=PROD-JCL
```

The following is the status of the files before the utility is run:

Permanent User Report list file

--USER--	-----REPORT-----	---JOB--	----CATEGORY----
BRANCH01	DAILY-BALANCE-REPORT	D4DAILYB	DAILY
BRANCH02	DAILY-BALANCE-REPORT	D4DAILYB	MONTHLY
BRANCH03	DAILY-BALANCE-REPORT	D4DAILYB	DAILY

Active User Report list file

--USER--	-----REPORT-----	---JOB--	----CATEGORY----	-ODATE-
ACCTSRCV	ACCOUNTS-RECEIV-REPT	D4ACCRCV	QUARTERLY	070796
ACCTSRCV	ACCOUNTS-RECEIV-REPT	D4ACCRCV	WEEKLY	070796
BRANCH01	DAILY-BALANCE-REPORT	D4DAILYB	DAILY	070796
BRANCH02	DAILY-BALANCE-REPORT	D4DAILYB	MONTHLY	070796

The following is the status of the Active User Report List file after the utility is run on the 5th of February:

--USER--	-----REPORT-----	---JOB--	----CATEGORY----	-ODATE-
ACCTSRCV	ACCOUNTS-RECEIV-REPT	D4ACCRCV	QUARTERLY	070796
ACCTSRCV	ACCOUNTS-RECEIV-REPT	D4ACCRCV	WEEKLY	070796
BRANCH01	DAILY-BALANCE-REPORT	D4DAILYB	DAILY	070796
BRANCH02	DAILY-BALANCE-REPORT	D4DAILYB	MONTHLY	070796
BRANCH02	DAILY-BALANCE-REPORT	D4DAILYB	MONTHLY	070796
BRANCH03	DAILY-BALANCE-REPORT	D4DAILYB	DAILY	070796

CTDDELRP - Delete Reports Which Are Not Needed

This utility cleans unneeded user/report/job entries from the Active User Report list file and erases from the disk compressed (CDAM) datasets which have no entries in the Active User Report List file.

This utility also moves a report entry from the Active to the History User file or the Migrated User Report List file when the DASD retention period for that report has expired and the report has been backed up or migrated successfully.

The utility determines which reports are no longer needed (i.e., for backup, migration, Online viewing, or printing) by means of user-specified selection groups. A selection group consists of a DAYS statement and any number of optional INCLUDE and/or EXCLUDE statements. The DAYS statement must contain a retention period which specifies the number of days the reports should remain online, after which they should be deleted by the utility. The optional INCLUDE/EXCLUDE statements further qualify which reports belong to the selection group.

By specifying multiple selection groups, each with its own retention period (and INCLUDE/EXCLUDE statements), groups of entries with differing retention periods can be deleted by a single execution of CTDDELRP. See "Record Selection Logic (INCLUDE/EXCLUDE Statements)" in this section for information about prefixing, masking, and the INCLUDE and EXCLUDE logic used to determine record selection.

Selection groups are scanned sequentially. The processing of a report is determined by the first selection group whose INCLUDE or EXCLUDE criteria match the report. INCLUDE/EXCLUDE criteria in other selection groups are ignored for that report.

Utility CTDDELRP performs the following sequence of steps:

1. Records to be deleted from the Active User file are selected (according to the parameters specified for the utility) and written to a temporary file.

Records for reports which have been backed up, are written to the History file.

CDAM files to be deleted are written to an intermediate file.

2. The temporary file is read and the records it contains are deleted from the Active User file.
3. CDAM files stored in the intermediate file (in Step 1) are deleted.

Parameters

MODE	This optional parameter in the EXEC statement determines whether utility CTDDELRP is run in Production mode or Simulation mode. If the parameter is not specified, the default is Production mode. It is recommended that this utility be run in Simulation mode prior to being run in Production mode.
PROD	The utility runs in Production mode, changing and deleting VSAM and CDAM files which satisfy the requirements specified by other parameters. Default.
TEST	The utility runs in Simulation mode, performing analysis and reporting but not changing or deleting any files.

DAYS Statement

The DAYS statement can contain the following parameters:

n Retention period (in days), after which the report expires and is selected for deletion from the Active User Report List file by the selection group.

By default, the report expires *n* days from the original scheduling date (ODATE) of the report decollating mission. The IOA administrator can change the default so that *n* is calculated from the decollation date. Refer to optional wish WD1377 in the IOA SECUDATA library.

n can be a negative number (to delete reports created on future ODATEs) and cannot exceed 9999.

This parameter is mandatory and must be the first parameter in the statement.

FORCE When not specified, reports with WAIT BACKUP status are not deleted. If specified, reports waiting backup are also deleted.

WAITPRINT If specified, reports with WAIT PRINT status are not deleted. If not specified, reports waiting to be printed are also deleted.

**WAITDEC/
NOTWD** WAITDEC deletes only reports having Wait Decollation status. NOTWD excludes from the selection group reports having Wait Decollation status. If neither parameter is specified, Wait Decollation status does not affect report deletion. Both parameters cannot be specified in the same selection group.

**RESTORED/
NOTRST** RESTORED deletes only reports having Restored status. NOTRST excludes from the selection group reports having Restored status. If neither parameter is specified, Restored status does not affect report deletion. Both parameters cannot be specified in the same selection group.

Note Parameters **FORCE**, **WAITPRINT**, **WAITDEC**, **NOTWD**, **RESTORED** and **NOTRST** apply only to the specific selection group in which they appear. They may be specified for as many selection groups as necessary.

NOREPORT Deletes any SYSDATA entries and their CDAM datasets (original reports data) if the SYSDATA is not pointed to by any user entries (e.g., the user entries have been manually deleted), regardless of the retention value specified in the DAYS parameter and all other selection group parameters.

KEEPDSN Deletes reports from the Active User Report List file but does not delete CDAM datasets. The CDAM datasets can then be re-decollated.

Note Parameters **NOREPORT** and **KEEPDSN** apply to all selection groups regardless of the criteria specified for them.

NOINDEX Prevents indexes from being deleted from the DASD (when a report is deleted), even if the indexes are defined as non-resident. This parameter is valid only for CONTROL-V. Indexes which are not deleted by utility CTDDELRP can be deleted at a later stage using utility CTVDELI.

INCLUDE/EXCLUDE Statements

If multiple INCLUDE/EXCLUDE statements are specified, each statement can contain any or all of the parameters listed below separated by a space. However, each parameter can appear no more than once in each statement.

See "Record Selection Logic (INCLUDE/EXCLUDE Statements)" in this section for more information about INCLUDE and EXCLUDE statements.

General Parameters in INCLUDE/EXCLUDE Statements

Each INCLUDE/EXCLUDE statement must contain at least one of the following parameters:

JOB NAME	Job name (or prefix).
JOBID	Job ID (format Jnnnnn).
USER	User name (or prefix).
CLASS	Class.
REPNAME	Report name (or prefix). If the report name contains embedded spaces, replace each embedded space with the @ character.
CATEGORY	Category of report decollation mission.

By default, the values specified for JOB NAME, USER and REPNAME are treated as prefixes. To prevent a value from being considered a prefix, place the special character < after the value.

CDAM Dataset Parameters in INCLUDE/EXCLUDE Statements

The following additional parameters can be specified for CDAM datasets created directly via job JCL SUBSYS parameters:

PGMSTEP	Program step name (which created the report).
PROCSTEP	Procedure step name (which created the report).
DDNAME	DD name (which created the report).

Notes *PGMSTEP refers to the step name specified in a //STEP01 EXEC PGM=pgmname statement.*

PROCSTEP refers to the step name specified in a //UPB01 EXEC procname statement which invokes a procedure.

CTDDELRP Deletion Rules

CONTROL-D/V reports are identified by user entries, SYSDATA entries and CDAM compressed datasets. Each user entry may point to one or more SYSDATA entries. Each SYSDATA entry points to one CDAM dataset and is pointed to by one or more user entries. Each CDAM dataset is pointed to by one or more SYSDATA entries.

When deleting reports, CTDDELRP maintains the integrity of this data structure according to the following rules:

- A CDAM compressed dataset is deleted when all SYSDATA entries pointing to it are deleted.
- A SYSDATA entry is deleted when all user entries pointing to it are selected for deletion.
- A user entry is deleted when all SYSDATA entries it points to are selected for deletion.
- A user entry is also deleted when all SYSDATA entries to which it points do not exist (i.e., the SYSDATA entries were deleted via the CONTROL-D/V Online facility).

CTDDELRP issues a message if it finds a SYSDATA entry which is not pointed to by any user entry (i.e., the user entries were deleted via the CONTROL-D/V Online facility). If parameter NOREPORT was specified, this SYSDATA entry is automatically deleted.

Activating the Utility

```
//          EXEC CTDDELRP[,MODE=TEST|PROD]
      DAYS n [FORCE] [WAITPRINT] [WAITDE|NOTWD] [RESTORED|NOTRST] [NOREPORT]
           [KEEPDSN]
[INCLUDE [JOB NAME=Job name] [JOBID=jobid] [USER=user] [CLASS=class]
        [REPNAME=repname] [PGMSTEP=pgmstep] [PROCSTEP=procstep]
        [DDNAME=ddname]
        .
        .
[EXCLUDE [JOB NAME=Job name] [JOBID=jobid] [USER=user] [CLASS=class]
        [REPNAME=repname] [PGMSTEP=pgmstep] [PROCSTEP=procstep]
        [DDNAME=ddname]
        .
        .
[
  DAYS statement
  [INCLUDE statement]
        .
        .
  [EXCLUDE statement]
        .
        .
        ]
        .
        .
```

Examples

Example 1

```
//S1 EXEC CTDDELRP
DAYS 3
```

The utility deletes reports which are older than three days if they have been backed up and have either WAIT PRINT or PRINTED status.

Example 2

```
//S1 EXEC CTDDELRP
DAYS 0 WAITPRINT
```

A disk space saver. All reports that were backed up and printed are deleted. Reports waiting for migration, print and/or backup are not deleted.

Example 3

```
//S1 EXEC CTDDELRP
DAYS 5
INCLUDE REPNAME=PROD-JCL
```

All reports with name prefix PROD-JCL which are more than five days old are deleted unless they have not yet been backed up.

Example 4

```
//S1 EXEC CTDDELRP
DAYS 0 FORCE
INCLUDE JOB NAME=Y99BUP01 JOBID=J01923
```

All reports of the specified job name and Job ID are deleted, regardless of WAIT BACKUP/WAIT PRINT status.

Example 5

Assume the following report names exist: ACCT, ACCT-REC, ACCT-PAY.

```
//S1 EXEC CTDDELRP
DAYS 0
INCLUDE REPNAME=ACCT*
```

All of the above reports are deleted.

```
//S1 EXEC CTDDELRP
DAYS 0
INCLUDE REPNAME=ACCT<
```

Only report ACCT is deleted.

Example 6

```
//S1 EXEC CTDDELRP
DAYS 0 WAITPRINT
EXCLUDE REPNAME=PROD-JCL
EXCLUDE CLASS=X
EXCLUDE JOB NAME=P00*
```

All reports which have already been migrated, printed and backed up (if backup was requested) are deleted from DASD, except:

- The production JCL reports.
- All reports produced on output Class X.
- All reports produced by jobs with names beginning P00.

Example 7

```
//S1 EXEC CTDDELRP
DAYS 7 WAITPRINT
INCLUDE JOB NAME=PROD*
DAYS 3 FORCE
INCLUDE JOB NAME=TEST*
```

All reports older than seven days and created by jobs with names beginning PROD are deleted except reports waiting for print.

All reports older than three days and created by jobs with names beginning TEST are deleted, regardless of WAIT BACKUP/WAIT PRINT status.

Example 8

```
//S1 EXEC CTDDELRP
DAYS 3 WAITPRINT RESTORED
DAYS 1 WAITDEC
DAYS 10 WAITPRINT NOT RST NOT WD
DAYS 0
INCLUDE USER=SCRAP
```

All reports that were restored at least three days ago are deleted except reports still waiting for print.

All report entries with a status of WAIT DECOLLATION (created via utility CTDCP2A) that are older than one day are deleted.

All reports older than ten days with neither RESTORE nor WAIT DECOLLATION status and which already printed are deleted.

All report entries belonging to user name SCRAP are deleted.

Example 9

```
//S1 EXEC CTDDELRP
DAYS 4 NOREPORT
```

All reports which are older than four days will be deleted.

All reports which have only a SYSDATA record (no user record points to them) will be deleted, even the SYSDATA record is one day old.

Note *This utility is fully recoverable. If an abend occurs at any step of the utility, fix the problem and rerun the utility from the first step.*

CTDDIB - Rebuild Index Component of a User Report List File

Utility CTDDIB is used to rebuild a CONTROL-D User Report List file's index component. The new index component corresponds to information currently contained in the IOA Access Method file's data component. Previous contents of the index component are erased.

Note The index component that is to be rebuilt must be allocated and formatted prior to activating the utility.

This utility should be used when an integrity problem is detected in an IOA Access Method file's index component. In this case:

- Delete and reallocate the index component.
- Format the index component with utility IOADBF.
- Rebuild the index component with utility CTDDIB.

Activating the Utility

```
//BUILD EXEC CTDDIB
```

Parameters

The following parameter can be specified to the utility in the EXEC statement which invokes the CTDDIB JCL procedure.

DBFILE	CONTROL-D IOA Access Method file's dataset name suffix.
ACT	Active User Report List file.
PRM	Permanent User Report List file.
HST	History User Report List file.
MIG	Migrated User Report List file (CONTROL-V only)

Example

```
//BUILD EXEC CTDDIB,DBFILE=HST
```

Refer to sample job CTDUFDIB in the CONTROL-D JCL library for an example of how to rebuild a IOA Access Method file index component.

CTDUPBKP - Update Backup Details

This utility is primarily used to update retention period and backup volume serial numbers in History/Migration/Active User Report files.

This utility can also be used to switch PRIMARY and SECONDARY information in the Migration User Report List file.

The utility receives the following groups of parameters:

- Selection parameters
- CHANGETO parameters
- Report parameters

The utility scans the file(s) that are specified via the Selection parameters and modifies the specified records in those files according to the CHANGETO parameters. Report parameters specify characteristics of the report to be produced.

Activating the Utility

```
//UPDATE EXEC CTDUPBKP
//DAVIN DD *
SELECT
    [FILE=ACT]
    [FILE=HST]
    [FILE=MIG]
    [DECFROM=ddmmyy,hhmm]
    [DECTO=ddmmyy,hhmm]
    [BKPFROM=ddmmyy,hhmm]
    [BKPTO=ddmmyy,hhmm]
    [BKPMIS=backup/migration mission name]
    [PLATTER=platter name]
    [VOLSER=old volser]
    [SIMULATE=YES/NO]
CHANGE TO
    [GENER=number of generation/NONE]
    [RETPD=retention period]
    [VOLSER=new volser]
    [SECONDARY]
    [PRIMARY]
/*
//DAULST DD *
    [PRINTSYS]
    [DETAIL]
    [UNMATCH=ALL/VOLSER]
/*
//
```

Selection Parameters

The following parameters can be specified in the SELECT statement:

- FILE** Type of file to be acted upon. Specify at least one of the following three files:
- ACT** Active User file
- HST** History User file
- MIG** Migration User file
- This parameter is mandatory.
- DECFROM** Earliest decollation date and time of reports to be selected for update.
Format: date, time. Both parameters are optional, but time cannot be specified without specifying date.
- Date format is mmddyy, ddmmyy or yymmdd (depending on the site standard defined in installation member IOAPARM).
- Time format is hhmmss.
- DECTO** Latest decollation date and time of reports to be selected for update.
Format: date, time. Both parameters are optional, but time cannot be specified without specifying date.
- Date format is mmddyy, ddmmyy or yymmdd (depending on the site standard defined in installation member IOAPARM).
- Time format is hhmmss.
- BKPFROM** Earliest backup date and time of reports to be selected for update.
Format: date, time. Both parameters are optional, but time cannot be specified without specifying date.
- Date format is mmddyy, ddmmyy or yymmdd (depending on the site standard defined in installation member IOAPARM).
- Time format is hhmmss.
- BKPTO** Latest backup date and time of reports to be selected for update.
Format: date, time. Both parameters are optional, but time cannot be specified without specifying date.
- Date format is mmddyy, ddmmyy or yymmdd (depending on the site standard defined in installation member IOAPARM).
- Time format is hhmmss.
- BKPMIS** Backup mission name. Validity checks are identical to DO BACKUP in the decollation. Optional.
- VOLSER** Volume serial number to be updated. A maximum of 10 volume serial numbers can be specified. Optional. Up to 6 characters can be specified. Example: 444444, 555555, 666666.

PLATTER		Platter name. A maximum of 9 characters. This parameter is used as a selection parameter for migration files, and can be specified for Kodak OptiStar Storage System (OSS) only.
SIMULATE	YES	Files are not updated. Only a report of all records to be updated is produced. Default.
	NO	Files are actually updated.

CHANGETO Parameters

The following parameters can be specified in the CHANGETO statement:

GENER	Number of generations. Valid values are:
	<p>nnnn Valid numeric value greater than zero.</p> <p>Only parameters FILE=HST and BKPMIS=BKPNAME can be defined in the SELECT group.</p> <p>Parameter RETPD must not be defined in the CHANGETO group. Reports belonging to the corresponding Backup mission are switched from "DAYS TO KEEP" method to the "GENERATIONS TO KEEP" method. The value of "DAYS TO KEEP" is automatically set to 0000. If there is no corresponding Backup control record in the History file, this utility creates it.</p>
	<p>NONE Parameter RETPD must be defined in the CHANGETO group.</p> <p>Reports belonging to the corresponding Backup mission are switched from "GENERATIONS TO KEEP" method to the "DAYS TO KEEP" method.</p>
RETPD	<p>Four-digit value identical to the "DAYS TO KEEP" value in the Backup mission.</p> <p>If this parameter is specified without specifying parameter GENER, the "GENERATIONS TO KEEP" should be 0 in the corresponding Backup control record in the History file. Otherwise, utility CTDUPBKP produces an error message.</p>
VOLSER	A new volume serial number for the volume be updated. Each volsers specified can be up to 6 characters in length. A maximum of 10 volsers can be specified (e.g., 111111, 222222, 333333). The number of volsers specified must be identical to the number of volsers selected (via the CHANGETO parameters described above).
PRIMARY	If utility CTDUPBKP was run with parameter SECONDARY specified, running this utility again with parameter PRIMARY causes the primary migrated copy of CDAM and index files to be used when accessing the migrated report. FILE=MIG must be specified in the SELECT group.

SECONDARY If utility CTDUPBKP was run with parameter SECONDARY specified, the secondary migrated copy of CDAM and index files will be used when accessing the migrated report. FILE=MIG must be specified in the SELECT group. If there is no secondary migration information, information in SYSDATA or the index record, this parameter is ignored.

Note Parameters PRIMARY and SECONDARY are relevant if the SECONDARY parameter was specified in the migration mission definition screen.

Report Parameters

DETAIL Print a detailed report for all volsers found in the file(s), including the backup job name, date and time. This report is useful only when a new volser was requested.

UNMATCH Produce a report listing all volsers in the file(s) that did not match any of the old volsers defined in the DAUIN DD input file.

ALL In any case.

VOLSER Prints a report only if volser update was requested.

PRINTSYS Produce a report detailing all sysdata records.

Example 1

```
//UPDATE EXEC CTDUPBKP
//DAUIN DD *
  SELECT
    FILE=HST
    DECFROM=250696,1000
    DECTO=270696,1500
    BKPFROM=260696,1000
    BKPTO=280696,1500
    BKPMIS=BKP001
    SIMULATE=YES
  CHANGE TO
/*
//DAULST DD *
//
```

The utility creates a report for the History file. Every SYSDATA which was created on 25/06/96 at 10 o'clock until 27/06/96 at 3 o'clock and was backed up on 26/06/96 at 10 o'clock until 28/06/96 at 3 o'clock by the backup mission BKP001 will be printed in this report.

Example 2

```

//UPDATE   EXEC CTDUPBKP
//DAVIN    DD *
SELECT
    FILE=ACT
    FILE=HST
    BKPFROM=260696,1000
    BKPTO=280696,1500
    BKPMIS=BKP001
    VOLSER=VOL001
    SIMULATE=NO
CHANGE TO
    RETPD=0100
    VOLSER=VOL002
//DAULST   DD *
PRINTSYS
DETAIL

```

If a CDAM was backed up to volser VOL001 on 26/06/96 from 10 o'clock until 28/06/96 at 3 o'clock by backup mission BKP001, the corresponding SYSDATA from the Active User file and History User file will be changed. The new value of RETPD is 100 and the new value of VOLSER is VOL002. A report about every SYDATA from the Active User file and the History User file and a report about every changed SYSDATA (detail) will be produced.

Example 3

```

//UPDATE   EXEC CTDUPBKP
//DAVIN    DD *
SELECT
    FILE=MIG
    BKPMIS=MIG001
    PLATTER=PLATTER01
    SIMULATE=YES
CHANGE TO
    SECONDARY
//DAULST   DD *

```

In this example, a simulation is performed for switching SYSDATA and INDEX records of the Migrated User file to use the secondary migrated CDAM files. Only the SYSDATA and INDEX records for CDAM files which were migrated to platter PLATTER01 by migration mission MIG001 will be selected. The result of this simulation is a report of the selected SYSDATA records. No change is made to the records themselves.

Example 4

```
//UPDATE EXEC CTDUPBKP
//DAUIN DD *
  SELECT
    FILE=HST
    BKPMIS=BKP001
    SIMULATE=NO
  CHANGE TO
    GENER=0005
//DAULST DD *
```

SYSDATA records referring to backup mission BKP001 are set to use the “# of GENERAITONS TO KEEP” method (In this example, 5 generations should be used.) A report for the modified SYSDATA records is produced.

See sample job CTDUBKPJ in the CONTROL-D JCL library for an example of how to update backup information in CONTROL-D User Report List files.

CTVCLMIG - Clean Migrated CDAM Files

This CONTROL-V utility cleans entries from the Migrated User Report List file and erases and/or uncatalogs the CDAM datasets and index files that belong to these entries.

The Migrated User Report list file contains entries describing migrated user reports and indexes. Each report has a retention period. The utility scans the Migrated User Report List file for all the reports whose retention period has passed and deletes them from the Migrated User Report List file.

For non-DASD reports, the utility produces a list of cartridges which can be returned to the global tape pool.

For FileTek migrated reports, a temporary file with report names and their MEDIA type is created. This file is processed by the next step which deletes them from the FileTek SM.

Activating the Utility

```
// EXEC CTVCLMIG[,MODE=TEST|PROD]
```

Return Codes

0	Operation performed successfully
4	Operation performed successfully (for HSM backup utility)
8	Execution error
11	Operation performed successfully (in TEST mode)
12	Invalid input parameters

CTVDELI - Delete Index Files from DASD

This CONTROL-V utility scans the Migrated User Report list file and erases from the disk index files which are no longer needed on DASD. The utility determines which indexes to delete based on user-specified selection groups.

Note Indexes that have been deleted from DASD by utility CTVDELI will be retrieved from the migrated media when required.

A selection group consists of a DAYS statement and any number of optional INCLUDE and/or EXCLUDE statements. The DAYS statement must contain a retention period which specifies the number of days the indexes should remain on DASD, after which they are deleted by the utility. The optional INCLUDE/EXCLUDE statements further qualify which indexes belong to the selection group.

By specifying multiple selection groups, each with its own retention period (and INCLUDE/EXCLUDE statements), groups of entries with differing retention periods can be deleted by a single execution of utility CTVDELI. See "Record Selection Logic (INCLUDE/EXCLUDE Statements)" in this section for information about prefixing, masking, and the INCLUDE and EXCLUDE logic used to determine record selection.

Selection groups are scanned sequentially. The processing of a report is determined by the first selection group whose INCLUDE or EXCLUDE criteria match the index. INCLUDE/EXCLUDE criteria in other selection groups are ignored for that index.

\$INDEX records are not deleted from the Migrated User Report list file because they are needed for accessing the migrated index file.

After an index file is deleted, a flag indicating INDEX DELETED FROM DASD is placed in the corresponding \$INDEX record.

EXEC Statement

The EXEC statement can contain the following parameter:

MODE	This optional parameter on the EXEC Statement determines whether utility CTVDELI is run in Production Mode or Simulation Mode. If the parameter is not specified, the default is Production Mode. It is recommended that this utility be run in Simulation Mode prior to being run in Production Mode.
PROD	The utility runs in Production Mode, deleting index files which satisfy the requirements specified by other parameters. Default.
TEST	The utility runs in Simulation Mode, performing analysis and reporting but not deleting any indexes.

DAYS Statement

Parameters in the DAYS Statement

The DAYS statement can contain the following parameters:

n Retention period (in days) after which the index expires and is selected for deletion from the Active User Report List file by the selection group.

By default, the index expires *n* days from the Original Scheduling Date (ODATE) of the Report decollating mission. The IOA administrator can change the default so that *n* is calculated from the decollation date.

n can be a negative number (to delete reports created on future ODATEs) and cannot exceed 9999.

This parameter is mandatory and must be the first parameter in the statement.

RESIDENT If RESIDENT is specified, resident indexes are deleted. Otherwise, resident indexes are not deleted.

Parameter RESIDENT only applies to the specific selection group in which it appears. It may be specified for as many selection groups as necessary.

INCLUDE/EXCLUDE Statements

If multiple INCLUDE/EXCLUDE statements are specified, each statement can contain any or all of the parameters listed below separated by a space. However, each parameter can appear no more than once in each statement.

See "Record Selection Logic (INCLUDE/EXCLUDE Statements)" above for more information about INCLUDE and EXCLUDE statements.

Parameters in INCLUDE/EXCLUDE Statements

Each INCLUDE/EXCLUDE statement must contain at least one of the following parameters:

JOB NAME Job name or prefix.

JOBID Job ID (format Jnnnnn).

REPNAME Report name or prefix.

If the report name contains embedded spaces, replace each embedded space with the @ character.

IXNAME Index name or prefix.

By default, the values specified for JOB NAME, REPNAME and IXNAME are treated as prefixes. To prevent a value from being considered a prefix, place the special character < after the value.

Activating the Utility

```
//      EXEC  CTVDELI [,MODE=TEST|PROD]
      DAYS  n      [RESIDENT]
      [INCLUDE  [JOB NAME=Jobname] [JOBID=jobid] [IXNAME=indexname]
              [REPNAME=repname]
      [EXCLUDE  [JOB NAME=Jobname] [JOBID=jobid] [IXNAME=indexname]
              [REPNAME=repname]

      DAYS statement
      [INCLUDE statement]
      .
      .
      [EXCLUDE statement]
      .
      .
      ]
      .
      .
```

Example 1

```
//S1 EXEC CTVDELI
      DAYS 3
```

All non-resident indexes which are older than three days are deleted.

Example 2

```
//S1 EXEC CTVDELI
      DAYS 0
```

A disk space saver. All non-resident indexes are deleted.

Example 3

```
//S1 EXEC CTVDELI
      DAYS 5
      INCLUDE REPNAME=PROD-JCL
```

All indexes of reports with name prefix PROD-JCL which are more than five days old are deleted.

Example 4

```
//S1 EXEC CTVDELI
      DAYS 0      RESIDENT
      INCLUDE JOBNAME=Y99BUP01 JOBID=01923
```

All indexes (including resident indexes) with the specified job name and Job ID are deleted.

Example 5

Assume the following index names exist: ACCT, ACCT-REC, ACCT-PAY.

```
//S1 EXEC CTVDELI
      DAYS 0
      INCLUDE IXNAME=ACCT
```

All of the above indexes are deleted.

```
//S1 EXEC CTVDELI
      DAYS 0
      INCLUDE IXNAME=ACCT<
```

Only index ACCT is deleted.

Example 6

```
//S1 EXEC CTVDELI
      DAYS 0
      EXCLUDE IXNAME=PROD-NDX JOB NAME=P01
```

All non-resident indexes are deleted from DASD except:

- Indexes whose name begins with PROD-NDX.
- Indexes for reports produced by jobs whose name begins with P01.

CONTROL-O Utilities

CTOCSF - Copying the Message Statistics File

Utility CTOCSF copies the Message Statistics file. The utility is generally used to create a new Message Statistics file whose size will be better adapted to the installation's needs.

Before copying the file, the following steps must be followed:

1. If the CONTROL-O monitor is active, stop the CONTROL-O statistics collection task using the command:


```
F CONTROL0,STOPSTATS
```

 Message CTO189I indicates that statistics accumulation has stopped.
2. Verify that no users have allocated the Message Statistics file using CONTROL-O screen OM.
3. Rename the current Message Statistics file (input file for the copy operation), for example, via ISPF 3.2/3.4.
4. Perform the following steps in ICE:
 - Verify that CTO is specified as the Product ID.
 - Select the CUSTOMIZE activity from the menu.
 - Select Major step "CTOPARM Post-Installation."
 - Select Minor step "Update CTOPARM Variables."
 - Change parameter STREC# from 10000 records to 20000 records to change the size of the new Message Statistics file.
 - Select Minor step "Save Parameters Into Product Libraries."
 - Select Minor step "Compile and Link CTOPARM."
5. Allocate and format the new Message Statistics file (output file for the copy operation) using member DEFSTAT in the IOA INSTCTO library.
6. Edit member CTOCSFJ in the CONTROL-O JCL library. Change DD statement DASTF DSN to match the name specified in Step 3.

Example

```
//COPYST JOB
//COPYFILE EXEC CTOCSF
//DASTF DD DISP=SHR,DSN=old-dsname
//
```

Where:

old-dsname is the name specified in Step 3 above.

After the Message Statistics file has been copied, the statistics collection task may be restarted using the following command:

```
F CONTROL0,STARTSTATS
```

Message CTO184I indicates that statistics accumulation has started.

CTORSTM - Statistics Report Utility

Utility CTORSTM generates statistical reports based on information provided by the Message Statistics Facility. CTORSTM output can be used to analyze site automation requirements and the progress of the automation process. CTORSTM output can also be used as input for report generator utilities (e.g., SAS, SP/SS, or EASYTRIEVE).

A sample JCL which invokes procedure CTORSTM can be found in member CTORSTMJ in the CONTROL-O JCL library. The output report is written to SYSPRINT.

Report parameters are passed to utility CTORSTM via DD statement DAREPIN.

SORT Parameter

The following parameter can be specified to sort the report data.

SORTBY	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">MSGID</td> </tr> <tr> <td style="padding: 2px 5px;">COUNT</td> </tr> </table>	MSGID	COUNT	Indicates whether the report should be sorted according to message ID (MSGID) or message count (COUNT), with the most frequent messages appearing first. Default: SORTBY COUNT.
MSGID				
COUNT				

INCLUDE/EXCLUDE Statements

The scope of the copy can be limited by one or more INCLUDE or EXCLUDE statements. Each INCLUDE/EXCLUDE statement contains parameters which limit the messages for which statistics are collected. See "Record Selection Logic (INCLUDE/EXCLUDE Statements)" earlier in this section for information about prefixing, masking, and the INCLUDE and EXCLUDE logic used to determine record selection.

Parameters in INCLUDE/EXCLUDE Statements

The parameters which can be specified in INCLUDE/EXCLUDE statements are:

msg	Message ID (e.g., DFH1799) or mask (e.g., DFH*). Defaults:
	INCLUDE * Include all messages.
	EXCLUDE blank Do not exclude any messages.

Example

```
//REPORT EXEC CTORSTM
//DAREPIN DD *
INCLUDE DFH*
INCLUDE $HASP*
EXCLUDE $HASP100
SORT BY MSGID
/*
```

CTOALOCP - Backing Up the Automation Log

Utility CTOALOCP copies the contents of one Automation Log file to another Automation Log file or to a sequential dataset. The utility may be used to perform any of the following functions, depending on the parameters submitted:

- Backing up the Automation Log to another Automation Log. Such Automation Log backup files can be displayed, just like the current Automation Log file, via the Automation Log screen (option OL).
- Increasing the size of the current Automation Log file.
- Copying the Automation Log to a sequential file. Such a sequential file can be processed by a report generator utility or restored and displayed later as an Automation Log file.
- Restoring a sequential backup file to the format of an Automation Log file.
- Changing the dataset organization of an Automation Log.

The parameters are passed to utility CTOALOCP via DD statement DAALPPRM.

The following parameters can be specified:

STARTDATE	Earliest date of the log records to be copied. Format: yymmdd. Default: Date of the earliest record in the log.
STARTTIME	Earliest time of the log records to be copied. Format: hhmmss. Default: 000000.
ENDDATE	Latest date of the log records to be copied. Format: yymmdd. Default: Date of the latest record in the log.
ENDTIME	Latest time of the log records to be copied. Format: hhmmss. Default: 240000.
FROMDSN	Name of the input dataset for the copy operation. Default: Name of the Operational Automation Log file.
FROMTYPE	Organization of the input dataset. Valid values are S and blank. Type S must be specified if the input dataset is a sequential file. For all other input dataset types, the field must be blank.
TODSN	Name of the output dataset for the copy operation.
TOTYPE	Organization of the output dataset. Valid values are: V DIV (VSAM Linear) dataset. D Direct Access (BDAM) dataset. S Sequential file. The default is the type of the Operational Automation Log (i.e., the value of parameter AUTOMLOG in member CTOPARM). Refer to the IOA Installation Guide.
TOVOLUME	Name (volser) of the volume on which the dataset named in the parameter TODSN will reside. This parameter is mandatory for an output dataset of type V (as specified explicitly with parameter TODSN or by default).

- TOUNIT** Name of the unit to which the output dataset will be allocated.
Default: SYSALLDA.
- TORECNUM** Number of records allocated to the output file. (This value is used to determine the size of the output dataset.)
- Valid values are:
- nnnnnnnn** Specified number of records.
- MINIMAL** The output file is allocated the exact number of records, copied from the input dataset, based on date and time specified limits.
- If no value was specified, the default is the number of records in the input file.

Automation Log Backup Utility - Examples

- Copy the entire Operational Automation Log to another file of same size:


```
TODSN      SYS3.MYLOG
TOVOLUME   WORK02
TOUNIT     DISK
```
- Copy the entire Operational Automation Log to a larger file:


```
TODSN      SYS3.MYLOG
TOVOLUME   WORK02
TOUNIT     DISK
TORECNUM   100000
```
- Restore a sequential file to the format of a direct access (BDAM) Automation Log file:


```
FROMTYPE   S
TODSN      SYS3.MYLOG.DTYPE
TOTYPE     D
TOVOLUME   WORK02
TOUNIT     DISK
```
- Back up part of the Operational Automation Log to a file on the basis of date/time limits. (The output file will contain only the records selected.) With CONTROL-M, the date/time may be set using the CONTROL-M AutoEdit facility.


```
STARTDATE  951027
STARTTIME  150000
ENDDATE    951027
ENDTIME    200000
TODSN      SYS3.MYLOG.D2710.F15T20
TOVOLUME   WORK02
TOUNIT     DISK
TORECNUM   MINIMAL
```
- Copy part of the Operational Automation Log file to a sequential file:


```
STARTDATE  951020
ENDDATE    951021
TODSN      SYS6.MYLOG.F20T21
TOTYPE     S
TOVOLUME   WORK02
TOUNIT     DISK
TORECNUM   MINIMAL
```

CONTROL-T Utilities

Record Selection Logic (INCLUDE/EXCLUDE Statements)

General

Certain CONTROL-T utilities accept selection criteria that determine which records should be processed when the utility runs. These selection criteria are specified as expressions in INCLUDE and EXCLUDE statements which are supplied as parameters to the utility.

INCLUDE and EXCLUDE statements are grouped into blocks. An INCLUDE/ EXCLUDE block consists of no more than one INCLUDE statement followed by any number of corresponding EXCLUDE statements.

INCLUDE and EXCLUDE expressions, statements and blocks are described on the following pages.

Expressions

Expressions are used to specify selection criteria. The format of an expression is:

key operator value

where:

key	A parameter (field) in the record. Each key listed in each INCLUDE/EXCLUDE statement must be unique.
operator	Relational symbols used to compare key to value (e.g., =, - or !, <, >).
value	Number or character string (which may include mask characters) against which to compare key.

Note See "Logical Field Names for the CONTROL-T Repository" in Appendix D of this guide for valid values that can be specified in INCLUDE/EXCLUDE statements in CONTROL-T.

Only the = or - operators may be specified when key and value are of character type.

Mask characters can be specified in expressions whose operator is either = or -. For additional information on masking, see "Character Masking" in Section 2 of the appropriate user manual.

INCLUDE Statements

INCLUDE statements determine which records should be processed by the utility according to selection criteria.

Section 9: Utilities

INCLUDE statements are composed of expressions (at least one) separated by commas.

The format of an INCLUDE statement is:

INCLUDE Expression1, Expression2, ... , ExpressionN

All the expressions in an INCLUDE statement must be true for the statement to be true (i.e., the relationship between the expressions within one INCLUDE statement is AND).

EXCLUDE Statements

EXCLUDE statements determine which records should not be processed by the utility according to selection criteria.

EXCLUDE statements are composed of expressions (at least one) separated by commas.

The format of an EXCLUDE statement is:

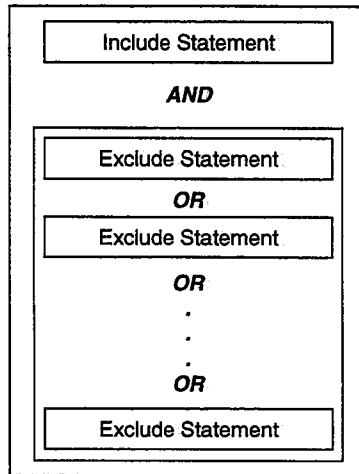
EXCLUDE Expression1, Expression2, ... , ExpressionN

All the expressions in an EXCLUDE statement must be true for the EXCLUDE statement to be true (i.e., the relationship between the expressions within one EXCLUDE statement is AND).

INCLUDE/EXCLUDE Block

An INCLUDE/EXCLUDE block is composed of no more than one INCLUDE statement followed by any number of its corresponding EXCLUDE statements.

The format of an INCLUDE/EXCLUDE block is:



The relationship between multiple EXCLUDE statements in a block is OR.

The relationship between an INCLUDE statement and the following EXCLUDE statement(s) in a block is AND.

Single INCLUDE/EXCLUDE Block Processing

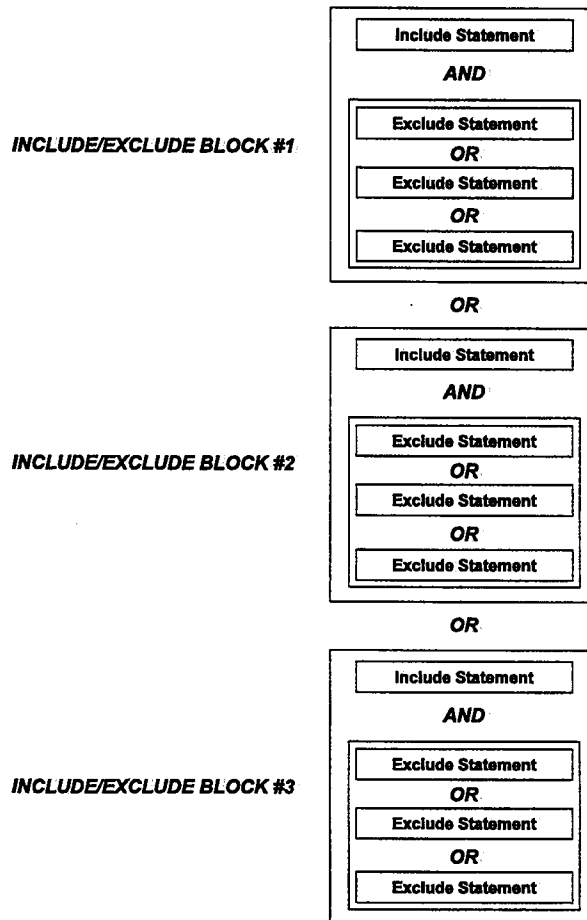
If even one EXCLUDE statement of a block is true (i.e., a record satisfies the EXCLUDE statement), the record is not selected, and the next record is examined.

If all EXCLUDE statements in the block are false, the INCLUDE statement is evaluated.

- If the INCLUDE statement is true, the record is selected.
- If the INCLUDE statement is false, the record is not selected and the next record is examined.

Multiple INCLUDE/EXCLUDE Blocks

An unlimited number of INCLUDE/EXCLUDE blocks may be specified. If multiple INCLUDE/EXCLUDE blocks are specified, the relationship between the INCLUDE/EXCLUDE blocks is OR and blocks are checked sequentially. If a record is not selected based on an INCLUDE/EXCLUDE block, it is checked against the next INCLUDE/EXCLUDE block. Once a record is selected based on an INCLUDE/EXCLUDE block, the record is included and not checked against other blocks.



INCLUDE/EXCLUDE Examples

The following examples illustrate include/exclude logic. For all but the last example, assume the following expressions, all of which evaluate to True.

- Expression 1: Key1=Value1 True
- Expression 2: Key2=Value2 True

Section 9: Utilities

Expression 3: **Key3=Value3 True**

Example 1

Select all records which match these expressions:

INCLUDE Key1=Value1,Key2=Value2,Key3=Value3

Example 2

Select all records which match these expressions, using another method:

INCLUDE Key1=Value1

EXCLUDE Key2=Value2

EXCLUDE Key3=Value3

Example 3

Select all records which match the first expression, and either the second or third expressions:

INCLUDE Key1=Value1

EXCLUDE Key2=Value2,Key3=Value3

Example 4

Select all records which do not match these expressions:

INCLUDE Key1=Value1

INCLUDE Key2=Value2

INCLUDE Key3=Value3

Example 5

Select all records which match either the second or third expression, but do not match the first expression:

INCLUDE Key2=Value2

EXCLUDE Key1=Value1

INCLUDE Key3=Value3

EXCLUDE Key1=Value1

Example 6

The following example utilizes actual values (instead of the terms KeyN and ValueN).

Select the records of all volumes starting with FD whose dataset name starts with BACKUP, except for those volumes starting with FD but ending with 102. Furthermore, independent of the preceding selection criteria, select records created by jobs whose job names start with CTT.

INCLUDE VOLSER=FD*,DSNAME=BACKUP*

EXCLUDE VOLSER=FD?102

INCLUDE CREJBN=CTT*

CTTACP - Copy Trace File

This utility can be used to:

- Copy a Trace file to another pre-formatted Trace file.
- Copy a Trace file to a sequential variable-blocked file, enabling the output to be formatted to a Hexadecimal or Dump display format.
- Copy a sequential, variable-blocked file containing Trace data back to a formatted Trace dataset. If this Trace dataset contains data, the sequential file's contents will be appended to it.

Note *The Trace file that is to be created must be formatted prior to activating the utility.*

Activating the Utility

```
// EXEC      CTTACP
//SYSIN      DD      *
COPY         FROM=TRACE|SEQ, TO=TRACE|SEQ
<INCLUDE/EXCLUDE STATEMENTS>
OUTPUT      TYPE=T|X|D|C|F, LINESIZE=size
//
```

Parameters

The following parameters can be specified to the utility in the EXEC statement which invokes the CTTACP JCL procedure:

TRCIN Name of source file.

TRCOUT Name of target file.

Parameters are also supplied via DD statement SYSIN:

COPY Copy statement type.

FROM Source dataset type:

TRACE Source is a CONTROL-T Trace file.

SEQ Source is a sequential variable-blocked file.

TO Target dataset type:

TRACE Target is a CONTROL-T Trace file.

SEQ Target is a sequential, variable-blocked file.

INCLUDE/
EXCLUDE The records written to the output dataset can be filtered by one or more INCLUDE/EXCLUDE statements. For more information about INCLUDE/EXCLUDE statements, refer to "Logical Field Names for the CONTROL-T Repository" in Appendix D.

OUTPUT	Output formatting.
TYPE	Output format type:
T	Trace file records are unchanged. Default.
C	Output contains only printable characters.
X	Output is written in Hexadecimal format.
D	Output is written in dump format (a combination of C and X types).
F	Output is formatted. A comprehensive list of Logical Field Names and relevant values is written for each volume/dataset record.
LINESIZE	Output record size. Valid values are from 70 to 132 characters; default is 80.

Note: The LINESIZE parameter and C, X, F, and D values for the TYPE parameter are only valid when the output dataset is a sequential dataset.

Return Codes

0	Operation ended successfully
8	Missing DD statement(s) or open failed
12	Control card error
16	Load of CONTROL-T routines failed
20	I/O error on input/output files

Example 1

Copy a Trace file to a second preformatted Trace file:

```
// EXEC      CTTACP,TRCOUT='CTT.BACKUP.TRC'
//SYSIN      DD      *
COPY        FROM=TRACE,TO=TRACE
//
```

Example 2

Copy the Trace file to a sequential variable-blocked file. Only records from the job TAPEJOB are included. The output should be in dump format. The line size is 132 bytes.

```
// EXEC      CTTACP,TRCOUT=CTT.JOB.ACTIVITY
//SYSIN      DD      *
COPY        FROM=TRACE,TO=SEQ
INCLUDE     USERID=TAPEJOB
OUTPUT     TYPE=D,LINESIZE=132
//
```

CTTAFR - Format and Initialize Trace File

Utility CTTAFR formats and initializes the Trace file.

The required number of trace records is specified in the TRCSIZE parameter in member CTT Parm. The maximum length of a trace record is 600 bytes. Assuming the dataset blocksize is 3600 (the supplied installation default), one 3380 cylinder contains about 1000 trace records.

Activating the Utility

```
// EXEC CTTAFR  
//
```

Parameters

No parameters are specified for this utility.

Return Codes

0	Operation performed successfully
8	DCB open error or TCT load failure occurred

Example

Format the Trace file.

```
// EXEC CTTAFR  
//
```

CTTBIX - Rebuild Media Database Index

This utility rebuilds the index file for the Media Database. The new index file corresponds to information currently contained in the Media Database data file.

Notes *The index file to be rebuilt must be allocated and formatted prior to activating the utility.*

CTTBIX will usually run only if CONTROL-T is down or suspended. See parameter TYPERUN for further information.

This utility should be used:

- When the amount of space remaining for the existing index file is insufficient. In this case:
 - Delete the index file.
 - Redefine the index file with the required amount of space.
 - Format the index file.
 - Rebuild the index file.
- For performance improvement.
- After recovery.
- After converting volumes from other tape management systems (e.g., CA-1, CA-TLMS) to CONTROL-T.

Activating the Utility

```
// EXEC CTTBIX, TYPERUN=B|C|F
//
```

Parameters

Utility parameters are supplied via procedure parameters:

TYPERUN	Statement type.
B	Utility is run in BUILD mode; the index file is rebuilt. If CONTROL-T is active, utility CTTBIX will not run.
C	Utility is run in CONVERT mode. This mode is specified when converting the database from another tape management system to CONTROL-T. If CONTROL-T is active, utility CTTBIX will not run.

F Utility is run in FORCED BUILD mode; the index file is rebuilt. No checking is performed for CONTROL-T status. This mode forces building of the index file.

This value is useful to initiate a rebuild in a test environment on a test Media Database while the production environment is still active.

Utility CTTBIX is run in FORCED CONVERT mode. No checking is performed for CONTROL-T status. This mode forces the building of the Index file.

This mode is specified when converting the database from another tape management system to CONTROL-T. This index file can be built and checked in a test environment while the production environment is still active.

Note Do not use type `TYPERRUN=F` if utility CTTBIX is running on the production Media Database (i.e., CONTROL-T is active). Running this utility on the production Media Database can cause unpredictable results.

Return Codes

0	Operation performed successfully
8	Inconsistency error detected - Execution continues - Index file built successfully (Logical errors in the data should be manually corrected.)
12	Error in parameters - Execution terminates
16	Error opening DCB - Execution terminates
24	Unsuccessful sort
28	Media Database access error
32	Load of CONTROL-T TCT failed
40	Index not empty
44	CONTROL-T is active - Utility CTTBIX cannot run in the specified mode while CONTROL-T is active

Example

```
// EXEC CTTBIX, TYPERRUN=B
//
```


CTTCRDF - Allocate and Format Data/Index Files

CTTCRDF creates and formats various CONTROL-T files. It can also increase or decrease the size of an index file to provide more space for the file or reduce the space used by the file.

Activating the Utility

```
// EXEC CTTCRDF
```

Other utilities are provided for copying entries from the original data file to the newly-formatted data file. These other utilities are described later in this section.

This utility is used during CONTROL-T installation by jobs that create CONTROL-T files. Each file creation job builds a newly-formatted data file and a corresponding index file.

Note *This utility is activated during CONTROL-T installation.*

Parameters

There are two categories of CTTCRDF parameters: Data file parameters and Index file parameters. Parameters of both categories specify precisely the type of file to be created.

Parameters are supplied to the utility via a JCL procedure. Each file creation job creates and formats a data/index file by receiving one or more of the following parameters:

Parameters for Data File Creation

DBPREFT	Highest qualifier of the dataset name, which usually specifies the prefix of the dataset.
DBVERT	Second highest qualifier of the dataset name, which usually specifies the version of the dataset.
DBFILET	Third qualifier of the name of the data file to be created. Valid values are: MDBD Media Database file STKD Stacking Statistics file
DBUNITT	Type of unit on which the data file should reside (3380, 3390).
DBVOLT	Volume on which the data file should reside.
NRECS	Number of records that the data file should contain. (To calculate the required number of records, refer to Disk Storage Requirements in the Considerations section in the IOA Installation Guide.)
NBLKS	Number of blocks that the data file should contain. (To calculate the required number of blocks, refer to Disk Storage Requirements in the Considerations section in the IOA Installation Guide.)

KEYLEN	For data files, KEYLEN is always zero.
BLK	Block size. This value should not be changed.
LREC	Record length. This value should not be changed.

Parameters for Index File Creation

DBPREFT	Highest qualifier of the index file name, which usually specifies the prefix of the index file.				
DBVERT	Second highest qualifier of the index file name, which usually specifies the version of the index file.				
DBFILET	Third qualifier of the name of the index file to be created. Valid values are: <table> <tr> <td>MDBI</td> <td>Media Database file</td> </tr> <tr> <td>STKI</td> <td>Stacking Statistics file</td> </tr> </table>	MDBI	Media Database file	STKI	Stacking Statistics file
MDBI	Media Database file				
STKI	Stacking Statistics file				
DBUNITT	Type of unit on which the index file should reside (3380, 3390).				
DBVOLT	Volume on which the index file should reside.				
NRECS	Number of records that the index file should contain. (To calculate the required number of records, refer to Disk Storage Requirements in the Considerations section in the IOA Installation Guide.)				
KEYLEN	Index file's preset key length. This value should not be changed.				
BLK	Block size. This value should not be changed.				
LREC	Same as block size length. This value should not be changed.				
NBLKS	Number of blocks that the index file should contain. (To calculate the required number of blocks, refer to "Disk Storage Requirements" in the Considerations section in the IOA Installation Guide.)				

Parameters DBPREFT, DBVERT, and DBFILET specify the name of the file to be created (e.g., if DBPREFT is CTT, DBVERT is V500, and DBFILET is MDBI, the name of the newly-created file is CTT.V500.MDBI).

Return Codes

0	Operation performed successfully
8	Invalid parameters
12	LRECL not equal to BLKSIZE (Index file)
16	Open of output file failed
20	SYSPRINT open failed
24	Insufficient storage (REGION)
28	Maximum block count was exceeded
32	Invalid key length

Example

```
// EXEC CTTCRDF,DBPREFT='CTT',DBVERT='V500',DBFILET='MDBI',  
      DBUNITT=3390,KEYLEN=30
```

Utility CTTCRDF is used by the following jobs in the IOA INSTCTT library. Refer to these jobs for additional examples.

- CTTCMDB Allocates and formats the Media Database file (described in detail in the IOA Installation Guide).
- CTTCSTK Allocates and formats the Stacking Statistics file (described in detail in the IOA Installation Guide).

CTTCTRC - Allocate and Format Trace File

CTTCTRC creates and formats CONTROL-T Trace files.

Activating the Utility

```
// EXEC CTTCTRC
```

This utility is used during CONTROL-T installation by the job that creates the CONTROL-T Trace file.

Parameters

Parameters are supplied to the utility via the invocation of a JCL procedure.

Parameters for Data File Creation

DBPREFT	Highest qualifier of the dataset name, which usually specifies the prefix of the dataset.
DBVERT	Second highest qualifier of the dataset name, which usually specifies the version of the dataset.
DBUNITT	Type of unit on which the file should reside (3350, 3380, 3390).
TRCVOL	Volume on which the file should reside.
TRCBLKS	Number of blocks that the Trace file should contain. (To calculate the required number of blocks, refer to "Disk Storage Requirements" in the Considerations section in the IOA Installation Guide.)

Parameters DBPREFT and DBVERT specify the name of the file to be created (e.g., if DBPREFT is CTT and DBVERT is V500, the name of the newly-created file is CTT.V500.TRC).

Return Codes

0	Operation performed successfully
8	DCB open error or CONTROL-T Control table load failure

Example

```
// EXEC CTTCTRC, DBPREFT='CTT', DBVERT='V500', DBUNITT=3390,
      TRCBLKS=1700
```

CTTDBDLD - Load Media Database/Stacking Statistics File Data Components

Utility CTTDBDLD loads Media Database (MDB) or Stacking Statistics file data components from a sequential file (which was previously created by utility CTTDBDUL). The data file must be formatted via utility CTTDBF before running this utility.

Note The Media Database index component can be rebuilt from the data component of the database using utility CTTBIX. The Stacking Database index component can be rebuilt using utility CTTDBIB.

Activating the Utility

```
// EXEC CTTDBDLD
```

Parameters

The following parameters can be specified to the utility in the EXEC statement which invokes the CTTDBDLD JCL procedure:

DBFILEI	Sequential file name suffix (input).
DBFILEO	Output file name suffix. Valid values are:
	MDBD Media Database file
	STKD Stacking Statistics file

Return Codes

0	Operation performed successfully
8	Sequential file operation error
12	Insufficient storage (REGION)
16	Invalid parameters
20	IOAPARM load failed
100-199	Index component errors
200-299	Data component errors

Example

```
// EXEC CTTDBDLD, DBFILEI=SEQ, DBFILEO=MDBD
```

CTTDBDUL - Unload Media Database/Stacking Statistics File Data Components

Utility CTTDBDUL unloads Media Database (MDB) or Stacking Statistics file data components to a sequential file. The sequential file must be previously allocated as fixed block, with a record length equal to the record length of the database file being unloaded.

Activating the Utility

```
// EXEC CTTDBDUL
```

Parameters

The following parameters can be specified to the utility in the EXEC statement which invokes the CTTDBDUL JCL procedure:

DBFILEI	Third qualifier of the input dataset. Valid values are:
	MDBD Media Database file
	STKD Stacking Statistics file
DBFILEO	Sequential file name suffix (output).

Return Codes

0	Operation performed successfully
8	File operation error
12	Insufficient storage (REGION)
16	IOAPARM load failed

Example

```
// EXEC CTTDBDUL,DBFILEI=MDB,DBFILEO=SEQ
```

CTTDBF - Format Existing Data/Index Files

This utility formats an existing CONTROL-T data or index file.

Note Utility CTTDBF is similar to utility CTTCRDF (described in this section). However, utility CTTDBF does not actually create data/index files. The files must exist on the disk before the utility is run.

Activating the Utility

```
// EXEC CTTDBF
```

Parameters

The following parameters can be specified to the utility in the EXEC statement which invokes the CTTDBF JCL procedure:

DBFILET	File name suffix. Valid values are:
	MDBD Media Database Data file
	MDBI Media Database Index file
	STKD Stacking Statistics Data file
	STKI Stacking Statistics Index file
NRECS	Number of records. This number should be the same number that utility CTTCRDF received via parameter NRECS when the file was created.
KEYLEN	Record key length. This number should be the same number that utility CTTCRDF received via parameter KEYLEN when the file was created. For data files, KEYLEN is always zero.

Return Codes

0	Operation performed successfully
8	Invalid parameters
12	LRECL not equal to BLKSIZE (Index file)
16	Open of output file failed
20	SYSPRINT open failed
24	Insufficient storage (REGION)
28	Maximum block count was exceeded
32	Invalid key length

Example

```
// EXEC CTTDBF,DBFILET=STKD,NRECS=1000,KEYLEN=0
```

See the following jobs in the JCL library for additional examples.

CTTFMDB	Formats Media Database data and index files.
CTTFSTK	Formats Stacking Statistics data and index files.

CTTDBIB - Rebuild Indexes of Stacking Statistics File

Utility CTTDBIB rebuilds an index file from the CONTROL-T Stacking Statistics Data file. The new index file corresponds to information currently contained in the Stacking Statistics Data file.

Previous contents of the index file are erased.

Note *The index file to be rebuilt must be allocated and formatted prior to activating the utility.*

This utility should be used:

- When the amount of space remaining in the existing index file is insufficient. In this case:
 - Delete the index file.
 - Redefine the index file with the required amount of space.
 - Format the index file with utility CTTDBF.
 - Rebuild the index file with utility CTTDBIB.
- Performance improvement.

Activating the Utility

```
// EXEC CTTDBIB
```

Parameters

The following parameter can be specified to the utility in the EXEC statement which invokes the CTTDBIB JCL procedure:

DBFILET	Data file name to be processed. Valid values are:
	STKD Stacking Statistics Data file

Note *SYSIN parameters are provided in the KEYxxx members in the PARM library for your convenience. Relevant file name specifications are hard-coded (predefined) in these members.*

Return Codes

0	Operation performed successfully
other	Operation failed

Example

```
// EXEC CTTDBIB, DBFILET=STKD
//SYSIN DD DSN=CTT.PROD.PARM(KEYSTK), DISP=SHR
//DAIDXSTK DD DSN=CTT.PROD.STKI, DISP=SHR
```


CTTDBID - Check Integrity of Data File

Utility CTTDBID checks data file integrity.

Utility CTTDBID checks the physical organization of a data file and reports errors which are detected in the data file structure. If specified, the utility corrects errors that are detected.

Activating the Utility

```
// EXEC CTTDBID
```

Data file integrity errors are listed in the sysout of the job.

Parameters

The following parameters can be specified to the utility in the EXEC statement which invokes the CTTDBID JCL procedure.

DBFILET Third qualifier of the name of the data file to be processed. Valid values are:

MDBD Media Database file

STKD Stacking Statistics file

ACT Action to be performed on the file.

R Report on detected errors

W Report on and correct detected errors

Note It is recommended that utility CTTDBID be run first with ACT=R (i.e., without requesting correction). If, based on such a run, it is determined that corrections are necessary, rerun the utility with ACT=W (i.e., requesting corrections).

Return Codes

0 Operation performed successfully

other Operation failed

Example

```
// EXEC CTTDBID,DBFILET=MDBD,ACT=R
```

CTTDBII - Check Integrity of Index File

Utility CTTDBII checks index file integrity.

Utility CTTDBII checks the physical organization of an index file and reports errors that are detected in the index file structure.

If errors are detected, the corresponding index file should be rebuilt using utility CTTDBIB (or utility CTTBIX if the file is the Media Database file).

Activating the Utility

```
// EXEC CTTDBII
```

Index file integrity errors are listed in the sysout of the job.

Parameters

The following parameter can be specified to the utility in the EXEC statement which invokes the CTTDBII JCL procedure:

DBFILET	Third qualifier of the name of the index file to be processed. Valid values are:
	MDBI Media Database file
	STKI Stacking Statistics file

Return Codes

0	Operation performed successfully
other	Operation failed

Example

```
// EXEC CTTDBII,DBFILET=STKI
```

CTTDBPRT - Print Data Records

This utility prints any CONTROL-T data file and its corresponding index file in SNAP format.

Activating the Utility

```
// EXEC CTTDBPRT
```

Parameters

The following parameters can be specified to the utility in the EXEC statement which invokes the CTTDBPRT JCL procedure:

DBFILET	Third qualifier of the name of the data file. Valid values are:
	MDB Media Database file
	STK Stacking Statistics file
FROM	Value of key from which to print.
TO	Value of key until which to print.

Notes *This utility works with a pair of files (both data and index files).*

Parameters FROM and TO are keys that should be found in the index and according to which data are printed.

The actual key is prefixed with one byte which identifies the record type:

D	Dataset
V	Volume
K	Stacking

Either FROM and TO, or neither FROM nor TO can be specified. If FROM and TO parameters are not specified or are specified as blank, all records are printed.

Example 1

Print the entire Media Database file.

```
// EXEC CTTDBPRT,DBFILET=MDB
```

Example 2

Print the Media Database volume records for volumes 100,000 through 999,999.

```
// EXEC CTTDBPRT,DBFILET=MDB,FROM=V100000,TO=V999999
```

Example 3

Print the Media Database dataset records for datasets BACKUP.G0012V00 through BACKUP9.

```
// EXEC CTTDBPRT,DBFILET=MDB,FROM=DBACKUP.G0012V00,TO=DBACKUP9
```

Example 4

Print the Stacking Statistics file records created by job M77BKUP.

```
// EXEC CTTDBPRT,DBFILET=STK,FROM=KM77BKUP,TO=KM77BKUP
```

CTTDLD - Add/Delete/Convert Volumes

This utility adds volumes to, and deletes volumes from, the Media Database. It can also convert volumes from other tape management systems (e.g., CA-1, CA-TLMS).

This utility should be used:

- To add a range of scratch/non-scratch volumes to the Active library (MAINLIB). Records for these volumes are added to the Media Database. This is called ADD mode.
- To delete a range of scratch volumes from the Active Library (MAINLIB). Records for these volumes are deleted from the Media Database. This is called DELETE mode.
- To build data from an external source of information and add it to the data component of the Media Database. This is called CONVERT mode. Usually CONVERT mode is used to convert another tape management system's database (e.g., CA-1, CA-TLMS) to CONTROL-T.

Activating the Utility

ADD mode:

```
// EXEC      CTTDLD
//SYSIN      DD      *
      TYPERUN  MODE=ADD
      RANGE    FIRST=firstvol, LAST=lastvol, MEDIA=type,
              SCRATCH=Y|N, VENDOR=vendor, RENAME=prefix
//
```

DELETE mode:

```
// EXEC      CTTDLD
//SYSIN      DD      *
      TYPERUN  MODE=DELETE
      RANGE    FIRST=firstvol, LAST=lastvol
//
```

CONVERT mode:

```
// EXEC      CTTDLD, INP='source-dataset'
//SYSIN      DD      *
      TYPERUN  MODE=CONVERT
      RANGE    FIRST=firstvol, LAST=lastvol, MEDIA=type,
              VENDOR=vendor, RENAME=prefix
//
```

Note Parameter *INP* in procedure *CTTDLD* is mandatory and valid only in *CONVERT* mode.

Utility CTTDLD is used by the following jobs in the INSTCTT library. Refer to these jobs for additional examples.

CONVTMDB Converts CA-TLMS Tape Management System reports to the CONTROL-T Media Database.

CONVC4MD/CONVC5MD

Converts from CA-1 (versions 4 or 5, respectively) to the CONTROL-T Media Database.

CONVLCTL

Converts from MVS Catalog to the CONTROL-T Media Database.

CONVRMDB

Converts from RMM to the CONTROL-T Media Database.

Parameters

The following parameter is specified in the EXEC statement:

INP Name of the input file containing the data to be converted. Mandatory and valid only in CONVERT mode.

The following parameters are supplied via DD statement SYSIN:

TYPERUN Statement type...

MODE Operation mode.

ADD Add volume entries to the Media Database.

DELETE Delete scratch volume entries from the Media Database.

CONVERT Convert volume entries from another tape management product's database to CONTROL-T Media Database format.

The following parameters are supplied in the RANGE statement:

RANGE Statement type:

FIRST First volume serial number in the range. Mandatory.

LAST Last volume serial number in the range. Mandatory.

MEDIA One of the media types defined in member CTTPARM. Mandatory and valid only in ADD and CONVERT modes.

SCRATCH Indication of whether or not the volume should be added as scratch. Valid for ADD mode only. Optional.

Y (Yes) Volume should be added as scratch.

N (No) Volume should not be added as scratch.

VENDOR Vendor of the media. Optional. Valid only in ADD and CONVERT modes.

RENAME A string used to replace the prefix of the volume serial number (volser). The SL-name of the volume remains unchanged. Optional. RENAME=prefix is valid only in ADD and CONVERT modes. The length of this string must not exceed the length of the corresponding part of the FIRST and LAST prefix.

Return Codes

0	Operation performed successfully
4	Minor errors occurred - Operation continues
12	Error in parameters - Execution terminates
16	Error opening DCB - Execution terminates
Greater than 16	Other error - the error is accompanied by an appropriate message indicating the problem - Execution terminates

Example 1

Add 3400 type media non-scratch volumes numbered K00100-K00500, made by vendor IBM.

```
// EXEC      CTTDLD
//SYSIN     DD *
      TYPERUN  MODE=ADD
      RANGE   FIRST=K00100, LAST=K00500, MEDIA=3400,
              SCRATCH=N, VENDOR=IBM
//
```

Example 2

Delete volumes numbered K00100-K00500.

```
// EXEC      CTTDLD
//SYSIN     DD *
      TYPERUN  MODE=DELETE
      RANGE   FIRST=K00100, LAST=K00500
//
```

Example 3

Convert 3480 type volumes numbered K00100-K00500 to CONTROL-T.

```
// EXEC      CTTDLD, INP='TAPE.CONVERT'
//SYSIN     DD *
      TYPERUN  MODE=CONVERT
      RANGE   FIRST=K00100, LAST=K00500, MEDIA=3480
//
```

Example 4

If you are consolidating two tape libraries and both libraries contain tapes with the same volsers, you can rename the volsers for one set of tapes so that no conflict will appear in the Media Database.

In this example, active volumes of the 3480 media type made by vendor IBM with volsers FX0001-FX0100 are converted to CONTROL-T, renamed, and added to the Media Database.

Volsers FX0001-FX0100 are renamed MA0001-MA0100. The SL-names (the internal labels of the volumes) remain FX0001-FX0100.

```
// EXEC      CTTDLD
//SYSIN     DD *
      TYPERUN  MODE=CONVERT
      RANGE   FIRST=FX0001, LAST=FX0100, MEDIA=3480,
              VENDOR=IBM, RENAME=MA
//
```

CTTGVL - User Interface to Stacking Facility

Utility CTTGVL assigns a volume to a given dataset before the dataset is allocated. The utility returns a volume serial number (according to user specifications) for the file that the user wishes to create.

To select the appropriate volume, the utility scans the CONTROL-T rules (according to supplied parameters) so that the volume which is returned is allocated from the appropriate pool.

The utility can optionally activate the CONTROL-T Stacking Facility, which searches the Media Database to locate a volume which has enough free space to hold the new file. Upon user request, or if such a volume cannot be located, a scratch volume can be returned instead.

Utility CTTGVL uses the active CONTROL-T environment (in ECSA). Therefore, the utility can be called only on a CPU on which the CONTROL-T environment is initialized.

The utility requires APF authorization. Therefore, it must reside in an authorized library, and the load module that contains it (if linked to the user program) must also be authorized (AC=1).

Activating the Utility

This utility is not a standalone utility. It must be called by an application using one of the following methods. See sample CTTGVLD in the IOA SAMPLE library.

Using a Call Command

```
CALL CTTGVL, (dataset, Job name, stackid, volser, label), VL
```

Using a Link command

```
LINK EP=CTTGVL, PARAM=(dataset, Job name, stackid, volser, label), VL
```

Parameters

Utility CTTGVL receives parameters in the standard method (Register 1 points to a list of addresses). The following parameters should be passed to the utility:

Input Parameters

DATASET	Name of the dataset. A maximum of 44 characters can be specified. Mandatory.
JOB NAME	Name of the job. A maximum of eight characters can be specified. Optional.
STACKID	Indicator (one character in length) specifying whether stacking is enabled/disabled:
	Y (Yes) Stacking is enabled.
	N (No) Stacking is disabled.

Output Parameters

VOLSER Volume serial number. Six characters must be specified.

LABEL Hexadecimal representation (two bytes in length) of the label number.

Return Codes

0 Function completed successfully

4 No matching volume found

8 Invalid stacking indicator

12 CONTROL-T is not active

16 Database access failed - a detailed message can be found in the job log or system log

Example

This utility must be called from an application program using one of the following methods:

Using a Call command:

```
CALL CTTGVL, (DSN, JOB NAME, STACKID, VOLSER, LABEL), VL
```

Using a Link command:

```
LINK EP=CTTGVL, PARAM=(DSN, JOB NAME, STACKID, VOLSER, LABEL), VL
```

Assume the following definitions for either of the above methods:

DSN	DC	CL44 'MY.DSN'	Dataset name
JOB NAME	DC	CL8 'MYJOB'	Job name
STACKID	DC	C 'Y'	Stacking indicator
VOLSER	DS	CL6	Volume serial number
LABEL	DS	H	Label

CTTIDB - Check Integrity of the Media Database

Utility CTTIDB checks the integrity of the Media Database (MDB) and provides statistical information. Warning messages are provided if database inconsistencies are detected. Upon user request, CTTIDB produces a transaction file which can be used by utility CTTUDB to repair some types of detected inconsistencies.

These inconsistencies may occur during conversion from another database, or as a result of system crashes, utility abends, etc., during processing.

Examples of inconsistencies are:

- Errors in the chain links which connect volumes in a multi-volume chain
- Existence of active volumes which do not contain datasets
- Existence of scratch volumes which contain datasets
- Discrepancies between various Media Database records

Upon completion, utility CTTIDB provides statistics about the Media Database. The utility can be run for statistical purposes. The statistics provided include:

- Number of used data records
- Number of used index records
- Number of free data records
- Number of free index records
- Totals by type of records (volume, dataset, scratch)

It is recommended that this utility be run immediately after conversion from another database, after a system crash, or after a CONTROL-T utility abends during processing. This utility can be run whenever Media Database statistics are desired.

Activating the Utility

Note *The SYSIN statement and the MTU parameter are optional.*

```
// EXEC     CTTIDB,MTU='transaction-file'
//SYSIN     DD *
TYPERUN     MODE=CHECK|UPDATE(,USEDAYS=n)
```

Parameters

The following parameter can be specified in the EXEC statement:

MTU Name of the transaction file to be used by utility CTTUDB. Optional. If not specified, an installation-defined default is used.

The following parameters are supplied via DD statement SYSIN:

TYPERRUN	Statement type.
MODE	Operation mode. Mandatory.
CHECK	Only messages about inconsistencies are provided. Default.
UPDATE	Transaction records for utility CTTUDB are provided in addition to inconsistency messages.
USEDAYS	Threshold number of days since the volume was last accessed. (For MODE=UPDATE only.) For those volumes not accessed since the specified number of days have passed, transaction records are generated for utility CTTUDB to remove the IN-USE indication from the volumes. Appropriate messages are also generated. Default is 7 (days).

Return Codes

0	Operation performed successfully
8	Problem detected in the Media Database - Execution continues
12	Invalid control parameter given - Execution terminates
16	Error opening DCB - Execution terminates
Greater than 16	Severe error - the error is accompanied by an appropriate message indicating the problem - Execution terminates

Example

In the following example, utility CTTIDB produces a transaction file of problems and issues appropriate messages. It also produces transaction records which remove the IN-USE indication from every volume which has not been accessed for more than 6 days.

```
// EXEC    CTTIDB
//SYSIN   DD *
TYPERRUNMODE=UPDATE,USEDAYS=6
```

CTTMER - Merge Records Into the Media Database

Utility CTTMER is used to merge Media Database records into a target Media Database via an extract file created by utility CTTSPPL.

Notes *There is no need to shut down CONTROL-T while running this utility.*

Utility CTTMER is used together with utility CTTSPPL in the split and merge process. Refer to the "The SPLIT and MERGE Process" in the description of the CTTSPPL utility in this section before using utility CTTMER.

After execution of utility CTTMER it is highly recommended that you run utility CTTVTM in SLOTBLD mode to rebuild the slot definitions. After completion of utility CTTVTM use utility CTTIDB to check the integrity of the target Media Database.

Utility CTTMER automatically updates the Media Database index component. There is no need to run utility CTTBIX after the run of this utility.

Activating the Utility

```
// EXEC CTTMER,MRGIN=repdata-file
//DAIN DD *
    TYPERUN MODE=NORMAL|SIMULATE|RSIMULAT[,DBGLEVEL=XXX]
/*
```

Parameters

repdata-file Extract file created by utility CTTSPPL. The name of this file is supplied via parameter MRGIN. Mandatory.

The extract file is a predefined sequential file (olpreft.olvert.REPDATA) allocated as one of the CONTROL-T operations libraries at time of CONTROL-T installation (at the site containing the source Media Database).

TYPERUN The TYPERUN statement is specified via DD statement DAIN. The format for this statement is:

```
TYPERUN MODE=NORMAL|SIMULATE|RSIMULAT[,DBGLEVEL=XXX]
```

Where:

MODE	Operation mode. Mandatory.
NORMAL	Normal operation mode. Default.
SIMULATE	Simulation mode. No changes are made to the Media Database. Messages describing the simulated operations are sent to the utility log.

RSIMULAT Restart simulation. If a first run of utility CTTMER abended, the utility can be rerun in this mode prior to rerunning it in NORMAL mode, and thus causing an automatic restart. Only the deletion of records already merged into the Media Database is simulated.

Note Only the deletion of data records, V-keys, D-keys and T-keys is simulated. Deletion of L-keys is not simulated.

DBGLEVEL Debug level. A numeric value from 1 to 255 can be specified. Optional.

Note This parameter should only be used if requested by your IOA representative.

Example

In this example utility CTTMER merges all records in the extract file (CTT.V500.REPDATA) into the Media Database.

```
// EXEC CTTMER,MRGIN=CTT.V500.REPDATA
//DAIN DD *
      TYPERUN MODE=NORMAL
/*
```

Return Codes

0	Operation performed successfully
8	Initialization problem.
12	Merge failed - some of the records to be merged already exist in the target Media Database
16	Unable to access the restart information from the Trace file
20	Simulation mode is not allowed while restart is required
24	Another utility is currently executing - the Media Database is locked
28	Restart processing failed
Greater than 28	Severe error - the error is accompanied by a message describing the problem

CTTMUP - Manually Update the Media Database

Utility CTTMUP updates fields of a volume or dataset record in the Media Database (MDB). Update is performed independently for volume and/or dataset records. When updating a volume record, CTTMUP does not automatically update the corresponding dataset records, and vice versa. (For example, the dataset count in a volume record is not updated when CTTMUP adds a dataset.) Therefore, when using this utility, be sure to update all connected records.

When a dataset is added, the utility searches the rules (as if the dataset is being created on removable media) to determine retention and vaulting criteria to be assigned to the dataset and the volume containing the dataset. Refer to "Manual Update of the Media Database" in Section 7 of this guide for a general description of this utility.

Notes *Utility CTTMUP is an extremely powerful utility which modifies your database. Great care should be taken that it is being used correctly. Therefore, before specifying parameters for use by this utility, read the section "Special Notes Regarding Control Statements" following the parameter descriptions.*

Before using utility CTTMUP, it is important to familiarize yourself with the structure of the Media Database (described in Section 7 of this guide). Attempting to update the Media Database with proper understanding of the Media Database structure can lead to integrity errors.

Activating the Utility

```
// EXEC      CTTMUP
//SYSIN      DD *
  TYPERUN    MODE=NORMAL | SIMULATION [, RULEINFO=YES | NO] [, CHECK=YES | NO]
             [, APPLY=GROUP]
  <CONTROL STATEMENTS>
```

Parameters

TYPERUN	Operation type.
MODE	Operation Mode. Mandatory.
NORMAL	Normal operation is performed.
SIMULATION	Simulation mode is performed.
RULEINFO	Whether or not to apply rule information (e.g., Retention criteria) when adding a dataset to the Media Database.
YES	Apply the rule information to the Media Database record. Default.
NO	Do not apply the information to the Media Database record.

CHECK	Whether or not to check Media Database integrity after performing the requested changes.
YES	Check the integrity of the Media Database. Default.
NO	Do not check the integrity of the Media Database.
APPLY	Determines which volumes will be affected by the OLUPD control statement. Optional.
GROUP	The VOLUPD statement is applied to all volumes in the multi-volume chain containing the volume specified in the VOLSER field of the VOLUPD statement.

Note *If parameter APPLY is not specified, the VOLUPD statement is applied only to the volume specified in the VOLSER field of the VOLUPD statement.*

Control Statements

Control Statements specify the update function to be performed, the Media Database fields to be updated and the update values. Some functions are valid for volume type records only, and some functions are valid for dataset records only.

Valid Control Statement functions are listed below, followed by information on specifying fields and values.

Note *Fields which can be updated by this utility are listed in Table 1 and Table 2 in "Logical Field names for the CONTROL-T Repository" in Appendix D. (They are the same fields that can be specified in INCLUDE/EXCLUDE statements of other utilities.)*

VOLADD	Add volume. One field is mandatory for this function: VOLSER Volume serial number.
VOLUPD	Update volume. The VOLSER field is mandatory for this function: VOLSER Volume serial number.
VOLDEL	Delete volume. The volume to be deleted must be identified by one of two fields (either field is mandatory): Either: VOLSER Volume serial number. or: DVLRBA Internal CONTROL-T record identifier. If DVLRBA is used to identify the volume, VOLSER can be used for verification.

- VOLSCR** Delete all the datasets in the volume and change the status of the volume to scratch. One field is mandatory for this function:
- VOLSER** Volume serial number.
- VOLBIX** Rebuild the indexes of a volume and of all its datasets. The volume must be identified by one of two fields (either field is mandatory):
- Either:
- VOLSER** Volume serial number.
- or:
- DVLRBA** Internal CONTROL-T record identifier.
- If DVLRBA is used to identify the volume, VOLSER can be used for verification.
- GRPSCR** Delete all datasets of all the volumes in the group, and change the status of those volumes to scratch. One field is mandatory for this function:
- VOLSER** Volume serial number of one of the volumes in the group.
- GRPBIX** Rebuild the indexes of all the volumes and of all the datasets in the group. One of the volumes in the group must be identified by one of two fields (either field is mandatory):
- Either:
- VOLSER** Volume serial number of a volume in the group.
- or:
- DVLRBA** Internal CONTROL-T record identifier of a volume in the group.
- If DVLRBA is used to identify the volume, VOLSER can be used for verification.
- DSNUPD** Update dataset record. The record to be updated must be identified in one of two ways (either set of fields is mandatory):
- Either:
- DSVOLSER** Dataset's volume serial number.
- DSLABEL** Dataset's label number (file sequence).
- or:
- DDSRBA** Internal CONTROL-T record identifier.
- If DSNAME is specified, it is used for verification only. The dataset name cannot be modified.

- DSNUPDN** Update dataset name (the DSNAME field in the dataset record). Only the parameters shown below are valid for this control statement.
- NEWNAME** New dataset name. Mandatory.
- DSNAME** The current dataset name (before update). This parameter is used for verification only. The dataset cannot be modified.
- The record to be updated must be identified in one of two ways (either set of fields is mandatory):
- Either:
- DSVOLSER** Dataset's volume serial number.
- DSLABEL** Dataset's label number (file sequence).
- or:
- DDSRBA** Internal CONTROL-T record identifier.
- DSNADD** Add dataset record. Three fields are mandatory for this function:
- DSVOLSER** Dataset's volume serial number.
- DSLABEL** Dataset's label number.
- DSNAME** Dataset name.
- DSNDEL** Delete dataset record. The record to be deleted must be identified in one of two ways (either set of fields is mandatory):
- Either:
- DSVOLSER** Dataset's first volume.
- DSLABEL** Dataset's label number.
- or:
- DDSRBA** Internal CONTROL-T record identifier.
- If DDSRBA is used to identify the record, DSVOLSER and DSLABEL can be used for verification.
- If DSNAME is specified, it is used for verification only.
- DSDELL** Delete L-type key for a volume. Two fields are mandatory for this function:
- DSVOLSER** Dataset's volume serial number.
- DSLABEL** Dataset's label number.
- RECDEL** Delete Media Database record (any type). One field is mandatory for this function:
- RBA** Internal CONTROL-T record identifier.

RECUPD Modify records of any type in the Media Database (e.g., Restart Control records, Slot Definition records).

This function includes the following control statements:

```
RECUPD RBA=rba
VER    offset data
REP    offset data
ENDUPD
```

Statement **RECUPD** selects a records from the Media Database for update. It must be the first statement specified. This statement includes the following parameters:

RBA Internal CONTROL-T record identifier of the record to be updated.

VER Identifies existing data to be verified before updating the record.

REP Identifies the new data that should replace existing data.

ENDUPD Identifies the end of the **RECUPD** function. It must be the last statement in this function.

An unlimited number of paired **VER** and **REP** statements can follow the **RECUPD** statement. These statements are formatted like the **VER** and **REP** statements of MVS utility **AMASPZAP**:

Data is provided in both the **VER** and **REP** statements, as follows:

offset Offset of the data from the beginning of the record (four digits in hexadecimal format, from 0000 to 01CC).

data Record data for verification or replacement (in hexadecimal ZAP format).

Example

This example updates the volume record with **RBA=000104**. Two status bytes are verified on offset 0005, and status **IN-USE** then set.

```
RECUPD RBA=X000104
VER    0005 8000
REP    0005 8002
ENDUPD
```

UPDQNM Change the **QNAME** in the Media Database (i.e., in the following files: **MDBD**, **MDBI**, **TRC**, **STKD**, **STKI**). One field is mandatory for this function:

QNAME New **QNAME**.

VOLSCRF Scratch a volume in the Media Database regardless of integrity errors. One field is mandatory for this function:

VOLSER Volume serial number.

Special Notes Regarding Control Statements

Note Be sure to read the following notes before specifying control statements for this utility.

Function VOLBIX

This function is intended for use only when utility CTTIDB detected errors with V-indexes (volume indexes). This function should not be used for L-index problems.

Function GRPBIX

Be sure that you have corrected all V-indexes before using this function.

This function corrects all problems of L-indexes and D-indexes. It can be applied for a single volume as well as for a multi-volume group.

Function VOLUPD

- When this function is used to update fields MEDIA, LOCATION or SLOTNUM of the volume, the slot in which a vaulted volume resided remains in status IN USE.

In this case, to synchronize the Slot Definition Records in the Media Database, run utility CTTVTM with the following control statement:

```
TYPEVLT MODE=SLOTBLD
```

- A single status bit can be set in a specific status byte of a volume/dataset record.

For example to turn on status PVL (Potential Vault) on a volume record, include the following statement:

```
VOLUPD. . . . ., VOLIND=+PVL
```

To turn off the above status bit include the following:

```
VOLUPD. . . . ., VOLIND=-PVL
```

Function VOLSCRF

This function is intended for use only after integrity errors occurred while trying to scratch records via function VOLSCR or GRPSCR.

This function affects single volumes only. To scratch a multi-volume group, specify this function for each volume separately.

Specifying Fields and Values

The following points should be considered when specifying fields and values for this utility:

Date Fields	Format is yymmdd or yyyyymmdd.
Time Fields	Format is hhmmss.
Numeric Fields	Leading zeroes are not required.
Embedded Blanks	Should only be used if the value is enclosed in either single or double quotes.

Quotation Marks (' or ")	If the value begins and ends with either single or double quotes, the quotation marks are interpreted as markers of the beginning and end of the value. To include a quote mark in a string (value), use the other type of quote mark to mark the beginning and end of the string. For example, a single quote (apostrophe) can only be included in the value if the value is enclosed in double quotes (").
Null Values	Specify a null value for a field by specifying the field without a value (e.g., VOLSER,).
Retention Criteria Fields	Specification of retention criteria requires two fields/values: <ul style="list-style-type: none"> - Retention type. - Retention data. If retention data is specified without specifying the retention type, the retention type defaults to DATE.
Fields with Predefined Valid Values	Certain fields have a predefined set of valid values (e.g., valid values for field VOLSTAT are ACTIVE, VAULTED). Valid values for these fields are listed in Table 1 and Table 2 in "Logical Field Names for the CONTROL-T Repository" in Appendix D. Values can also be specified in hexadecimal format (e.g., VOLSTAT=X88, which means Active and Vaulted).
Status Bits	A single status bit can be set in a specific status byte of a volume/dataset record. For example, to turn on status PVLТ (Potential Vault) on a volume record, include the following statement: VOLUPD..... , VOLIND=+PVLТ To turn off the above status bit, include the following: VOLUPD..... , VOLIND=-PVLТ

Specific Field Recommendations

LBLNUM	Last label number on the volume. This value cannot be less than the value of ACTIVEDS (number of active datasets on the volume). If not specified, LBLNUM defaults to the value of ACTIVEDS.
--------	--

Return Codes

0	Operation performed successfully
4	Minor errors occurred. Operation continues
12	Load of other CONTROL-T programs failed
16	Media Database I/O error
20	Input Control Statement error
24	Authorization failure (by security exit)
28	Invalid return code from routine CTTEXP

Example 1

Convert scratch volume DDD020 to active status and add dataset FILE7 to the volume.

```
// EXEC      CTTMUP
//SYSIN      DD *
  TYPERUN    MODE=NORMAL
  VOLUPD     VOLSER=DDD020,ACTIVEDS=1,VOLSTAT=ACTIVE
  DSNADD     DSVOLSER=DDD020,DSLABEL=1,DSNAME=FILE7,
             CREDIT=950622,CRETM=142500,CREJBN=JOB1
```

Example 2

Rebuild a chain of two volumes. The first volume, PRD001, has two datasets on it. The second dataset reached the end of the first volume and continued on the second volume, PRD002.

```
//JOB NAME   JOB: ....
//CTTMUP     EXEC CTTMUP
//SYSIN      DD *
  TYPERUN    MODE=NORMAL, RULEINFO=YES, CHECK=NO
  VOLUPD     VOLSER=PRD001, VOLSTAT=ACTIVE, VOLSEQ=1, ACTIVEDS=2, LBLNUM=2,
             MEDIA=3480, LOCATION=MAINLIB, FIRSTVOL=PRD001,
             NEXTVOL=PRD002, PREVVOL=
  VOLUPD     VOLSER=PRD002, VOLSTAT=ACTIVE, VOLSEQ=2, ACTIVEDS=1, LBLNUM=2,
             MEDIA=3480, LOCATION=MAINLIB, FIRSTVOL=PRD001,
             NEXTVOL=, PREVVOL=PRD001
  DSNADD     DSVOLSER=PRD001, DSLABEL=1, DSNAME=PROD.BACKUP.G0001V00,
             VOLSNUM=1
  DSNADD     DSVOLSER=PRD001, DSLABEL=2, DSNAME=PROD.BACKUP.G0002V00,
             VOLSNUM=2
//
```

Considerations

- This example assumes that PRD001 and PRD002 are defined in the Media Database and are in SCRATCH status. If these volumes are not defined in the Media Database, you must change the first two control cards from VOLUPD to VOLADD.
- Parameter ACTIVEDS is set to 2 for PRD001 because PRD002 contains the two datasets. PRD002 contains only the second dataset, so ACTIVEDS is set to 1 for PRD002.
- LBLNUM is set to 2 for both volumes because the highest label number on both of volumes is label 2.
- The second dataset, PROD.BACKUP.G0002V00 has VOLSNUM=2 because it spans the two volumes.
- RULEINFO=YES indicates that retention and vaulting attributes are automatically taken from the rules. It is therefore unnecessary to specify retention and vaulting information.

CTTRCV - Recover the Media Database Using Trace File

Utility CTTRCV recovers damaged Media Database (MDB) contents according to trace data recorded during Media Database update operations. The utility can perform either a physical or a logical recovery.

- Physical Recovery (Roll Forward)

Physical recovery assumes that the last available backed-up Media Database was restored to disk (see utility CTTTRB in this section). From this point forward, the recovery is performed according to trace data.

Trace records are read starting with the first record after the Media Database backup record. The recovery continues until the last trace record is reached. When the process is finished, the Media Database will have been brought back to the condition it was in before it was damaged.

In situations where no Media Database backup has been performed, the recovery may be started from the first trace record.

While the physical recovery process is running, the CONTROL-T Environment should be inactive (or unpredictable results may occur).

- Logical Recovery (Roll Back)

Logical recovery rolls back selected changes from the current Media Database. This type of recovery can be performed when a user modifies the status of one or more Media Database records, and subsequently discovers that these modifications should be reversed. A good example of this scenario is the run of a job which scratches a series of active volumes or moves a series of volumes to a vault incorrectly.

When this type of recovery is used, the trace records are read backward (starting from the last record towards the first record). The improper modifications are reversed in a way that restores the Media Database to the condition it was in prior to the improper modifications.

To maintain the logical consistency of the Media Database, the utility must verify that the current contents of the Media Database match the information in the Trace file. Therefore, before a Media Database operation is reversed, the utility checks that the current Media Database reflects that operation. For example, if the status of a volume was changed from Active to Scratch by TSO user ABC, and a logical recovery is performed on all operations performed by user ABC, the utility verifies that the current volume status in the Media Database is Scratch before the recovery reverses the operation. The check can be bypassed (by specifying FORCE=YES) but this may lead to unpredictable results.

Activating the Utility

```
// EXEC      CTTRCV
//SYSIN      DD *
RECOVER TYPE=PHYSICAL|LOGICAL [MODE=SIMULATION|NORMAL]
          [,FROMREC=LASTBKP|FIRSTREC] [,CHECKACT=YES|NO] [,RECAT=YES|NO]
          [,FORCE=YES|NO]
<INCLUDE/EXCLUDE STATEMENTS>
//
```

Parameters

Utility parameters are supplied via DD statement SYSIN:

RECOVER	Determines the type of recovery that should be performed.	
	TYPE	Type of the recovery. Mandatory.
	PHYSICAL	A physical (roll forward) recovery is to be performed.
	LOGICAL	A logical (roll back) recovery is to be performed.
	MODE	Mode of recovery. Optional.
	SIMULATION	Simulation mode. Trace records and Media Database records (when needed) are read, but the Media Database is not updated and datasets are not recataloged.
	NORMAL	Normal mode. This is the default. Trace records and Media Database records (when needed) are read, and updates are actually performed on the Media Database (i.e., the Media Database is updated).
FROMREC	Starting point of the recovery. Can only (and must) be specified when TYPE=PHYSICAL .	
	LASTBKP	The recovery starts from the point where the Media Database was last backed up.
	FIRSTREC	The recovery starts from the first Trace record. This option is useful for new sites which have not yet performed any Media Database backups.

CHECKACT Check whether or not CONTROL-T is active before starting the recovery. When recovery is performed, the CONTROL-T Real-time Environment should not be active (i.e., it should be down or suspended). In simulation mode, this parameter has no effect.

YES If, before starting the recovery, CONTROL-T is active, recovery will not be started.

NO Recovery will be started whether or not CONTROL-T is active.

Note Specifying **NO** can cause unpredictable results and the loss of Media Database integrity, if CONTROL-T is active.

This value is useful to initiate a recovery in a test environment on a test Media Database while the production environment is still active.

RECAT In logical (roll back) recovery, indicates whether datasets that were uncataloged by utility CTTRTM should be recataloged.

YES Recatalog all datasets that were uncataloged.

Note The same catalog environment that existed for CTTRTM is required by CTTRCV.

NO Do not recatalog datasets.

FORCE In logical (roll back) recovery, indicates whether a check will be performed to verify that the current contents of the Media Database match the operation that should be reversed.

YES Perform the operation reversal without performing the check.

Note Specifying **YES** can cause unpredictable results and the loss of Media Database integrity.

NO Perform the operation reversal only if all checks were performed successfully.

**INCLUDE/
EXCLUDE**

The scope of the utility can be limited by one or more INCLUDE or EXCLUDE statements. INCLUDE/EXCLUDE statements are optional, and are valid only when a logical (roll back) recovery is performed (i.e., TYPE=LOGICAL).

For more information about INCLUDE/ EXCLUDE statements for this utility, refer to "Logical Field Names for the CONTROL-T Repository" in Appendix D.

For additional information on INCLUDE and EXCLUDE statements, see "Record Selection Logic (INCLUDE/EXCLUDE Statements)" earlier in this section.

Return Codes

0	Recovery operation performed successfully
4	Input statement(s) missing
8	DCB open error occurred
12	Input statement parsing error or validation error occurred
16	CONTROL-T Control Table (TCT) load failed
24	Media Database/Trace file I/O error occurred
28	Logical error in Media Database (for TYPE=LOGICAL)
32	CONTROL-T Real-time Environment is active

Example 1

Recover a damaged Media Database. The last available Media Database backup has been restored to disk.

```
//EXEC      CTTRCV
//SYSIN     DD *
            RECOVERTYPE=PHYSICAL, FROMREC=LASTBKP
```

Example 2

Reverse all changes performed by user M27A from July 19th, 13.40.00, to July 22nd, 12.24.45:

```
// EXEC      CTTRCV
//SYSIN     DD *
            RECOVER TYPE=LOGICAL, RECAT=YES
            INCLUDE USERID=M27A, DATE=950719, TIME>134000
            INCLUDE USERID=M27A, DATE>950719
            EXCLUDE DATE>950721
            INCLUDE USERID=M27A, DATE=950722, TIME<122445
```


CTTRPT - Database Extraction and Report Utility

Utility CTTRPT extracts data from the Media Database (MDB) via Extract parameters, and generates reports from the extracted data via Report Formatting parameters.

The utility can be used in either of two ways.

- Data extraction and report generation are performed in a single utility run. Both Extraction parameters and Report Formatting parameters are specified in the utility run.

Instead of extracting entire records, the utility only extracts information required for the report from the Media Database (or from the extracted dataset described above), and produces the report. There is no need to create a dataset for saving the extracted information, although this can be done if desired.

This method of report generation is the most efficient when a single report is required.

It is also most efficient when all the data requirements of subsequent reports can be satisfied from the extracted data (i.e., when no additional data is required by subsequent reports). In this case, the extracted data should be saved in a dataset for subsequent runs of the utility.

- Data extraction and report generation are performed in separate utility runs.

One run of the utility uses Extract parameters to select and extract entire volume and/or dataset records from the Media Database which the utility saves in a dataset.

Subsequent runs of the utility can then use Report Formatting parameters (REPORT, FIELDS and SORTBY statements) to generate reports in different formats from the extracted dataset.

This method is most efficient when different reports use the same records but require data from different fields.

At least one group of parameters (i.e., Extract parameters, or Report Formatting parameters) must be specified.

The dataset for holding extracted information is specified in DD statement REPDATA.

Primary Key, Primary Fields and Primary Lines in Report Generation

The user can define up to two lines which comprise the report output (i.e., the user determines which fields are included in the report lines).

Many fields in a report maintain the same value for many records (e.g., vault name for a multi-volume group). These fields are called primary fields, and one of them is the primary key. (The field which is the primary key and the fields which are the primary fields are determined by the type of report being generated.)

Via parameter ID=P|S, the user can designate a report line as a primary line. Primary lines are only printed when the primary key changes value.

The primary key and the primary fields are determined as follows:

- In the GENERAL report, the primary key is determined according to PATH parameter specified in EXTRACT statement.

If PATH=VOLUME/ALLDATASETS or PATH=VOLUME/FIRSTDATASET is specified, the primary key is the volser and the primary fields are the fields extracted from volume record.

If PATH=DATASET/ALLVOLUMES or PATH=DATASET/FIRSTVOLUME is specified, the primary key is the dataset's volser and dataset's sequence number; and primary fields are the fields extracted from the dataset record.

- In the DISTRIB report, the primary key is the volume's volser (or the first volume's volser in a multi-volume chain) and the primary fields are the fields common for all volumes in the chain (e.g., From Location and To Location), and the fields extracted from the dataset record which determined the vaulting pattern.
- In the SCRATCH report, the primary key is the volume's volser (or the first volume's volser in a multi-volume chain) and the primary fields extracted from the expiration dataset record.

Activating the Utility

```
// EXEC CTTRPT
//REPDATA DD DSN=extract-dataset
//SYSIN DD *
[
<EXTRACT parameters>
<REPORT parameters>
]
```

Parameters

Utility parameters are supplied via DD statement SYSIN:

Extract Parameters

```
EXTRACT PATH=VOLUME |
    VOLUME/FIRSTDATASET |
    VOLUME/ALLDATASETS |
    DATASET |
    DATASET/FIRSTVOLUME |
    DATASET/ALLVOLUMES
<INCLUDE/EXCLUDE statements>
```

EXTRACT PATH=rectype Specifies the type of records to be extracted. Valid values are:

VOLUME	Volume records are extracted.
VOLUME/ FIRSTDATASET	Volume records with only their first dataset records are extracted. Abbreviation: VOLUME/FIRSTDS
VOLUME/ ALLDATASETS	Volume records with all their dataset records are extracted. Abbreviation: VOLUME/ALLDS
DATASET	Dataset records are extracted.
DATASET/ FIRSTVOLUME	Dataset records with only their first volume records are extracted. Abbreviation: DATASET/FIRSTVOL
DATASET/ ALLVOLUMES	Dataset records with all their volume records are extracted. Abbreviation: DATASET/ALLVOLS

Multiple **EXTRACT** statements can be specified in DD statement **SYSIN**, but each additional **EXTRACT** statement must follow the **INCLUDE/EXCLUDE** statements (if any) and the Report Parameter statements of the preceding **EXTRACT** statement. When saving extracted data in a dataset, if several **EXTRACT** statements are specified, only the data extracted by the last **EXTRACT** statement is stored.

INCLUDE/EXCLUDE Statements

The scope of extraction can be limited by one or more **INCLUDE** or **EXCLUDE** statements.

INCLUDE/EXCLUDE statements can be specified only immediately after the **EXTRACT** statement.

Subparameters specified in **INCLUDE/EXCLUDE** statements must not contradict the **PATH** parameter of the **EXTRACT** statement.

For example, if **PATH=DATASET** is specified, subparameters specified in the **INCLUDE/EXCLUDE** statements must not include fields from volume records.

The subparameters which can be specified in **INCLUDE/EXCLUDE** statements are listed in Tables 1 and 2 in Appendix D of this manual. Refer to this appendix for more information.

For additional information concerning **INCLUDE/EXCLUDE** statements, see "Record Selection Logic (**INCLUDE/EXCLUDE** Statements)" earlier in this section.

Report Formatting Parameters

```
REPORT NAME=GENERAL, TITLE='text',
        PAGESIZE=p, LINESIZE=s, MARGINS=m-n, SUMMARY=YES|NO, OUTDD=ddname
FIELDS field 1/options-1, field 2/options-2, ..., field n/options-n, ID=P|S
SORTBY field 1/options-1, field 2/options-2, ..., field n/options-n
ATTR   FIELD=f, LENGTH=t, TITLE='text', EDIT='mask'
BREAK  FIELD=f, COLUMN=c, SUMTITLE='text', EDIT='numeric mask'
```

REPORT	Activates report formatting. The following subparameters can be specified:
NAME	Report name. Mandatory. Valid value is:
	GENERAL General report of information extracted from the Media Database.
TITLE	Report title. Optional. The maximum length of text is 64 characters. Default: 'General Report'
PAGESIZE	Maximum lines in a page, where p is a numeric value greater than 20. Optional. Default: 63.
LINESIZE	Maximum characters in a line, where s is a numeric value not greater than 255. Optional. Default: 127.
MARGINS=m-n	Lowest (m) and highest (n) columns in a line, where m is a numeric value less than s (the value of LINESIZE), and n is a numeric value greater than m but not greater than s. Optional. Defaults: m=1 and n=s.

SUMMARY= Indicates whether or not to show a summary line for each field break. Optional. Default: NO.
YES|NO

A field break occurs when there is a change in value for a field which is specified in a SORTBY statement with option /BRK.

OUTDD JCL DD name of the output. Default: REPOUT.
=ddname

FIELDS Output fields (i.e., fields required in the report), and the options that can be applied to each field. Mandatory.

The GENERAL report can contain volume and dataset fields which can be specified as INCLUDE/EXCLUDE subparameters and special fields which cannot be specified as INCLUDE/EXCLUDE subparameters (described with their attributes later in this utility).

Options Options can be added as a suffix to any output field in the list (optional for each field). The slash (/) preceding the option is required to distinguish the option from the field. Any number of options can be added to an output field. Available options are:

/Ln Length of the output field. n must be a numeric value which does not exceed the length defined in the list of available fields. This option should be used for Alphanumeric and Hexadecimal fields only. Optional.

This value, if specified, overrides a LENGTH value specified in the ATTR.

/Cn Column in which the field should begin in the printed output. If specified for a field, it must be specified for all preceding fields. n must be a numeric value greater than the last column of the preceding field (i.e., determined by the preceding field's beginning column and field length).

ID=P|S Output line indicator. This parameter designates an output line as a Primary Line (P) or Secondary Line (S). These values determine whether or not duplicate lines are printed from Volume and Dataset data.

P Primary line indicator. The primary fields in the line are only printed once, and printed again only when the primary key changes value.

S Secondary line indicator. The line is printed for each record.

If this parameter is not specified, a line is printed for every input record.

SORTBY

Sort order of the output fields (see FIELDS above). Optional.

Options Options can be added as a suffix to any output field in the list (optional for each field). The slash (/) preceding the option is required to distinguish the option from the field. Any number of options can be added to an output field. Available options are:

<code>/BRK</code> or <code>/B</code>	Insert a page break in the report when the field changes its value. If specified for a field, it must be specified for all preceding fields.
<code>/ASC</code>	Sort the field in ascending order. Default.
<code>/DSC</code>	Sort the field in descending order.
<code>/RSR</code>	Reset the row number whenever the field changes values. This option is effective only when field ROWID is specified in the FIELDS statement.
<code>/RSP</code>	Reset the page number whenever the field changes values. This option also automatically activates option /RSR.

Note *Primary fields appearing in a SORTBY statement should appear before non-primary fields.*

Examples of SORTBY usage are provided below.

Example 1 SORTBY LOCATION
/BRK,POOL/BRK/DSC,MEDIA/ASC

Output is sorted by vault (LOCATION) in ascending order (the default), by pool (POOL) within a vault in descending order, and by media (MEDIA) within a pool in ascending order. A page break is inserted each time the vault or pool changes.

Example 2 SORTBY LOCATION
/BRK/RSP,MEDIA/BRK/RSR,EXPDT/DSC

Output is sorted by vault (LOCATION) in ascending order (the default), by media (MEDIA) within a vault in ascending order (by default), and by expiration date (EXPDT) within a media in descending order. A page break is inserted for every change in vault and media, and the page number (and, by default, the row number) is reset each time the vault changes. The row number is reset each time the media changes.

ATTR

Overrides default attributes defined for a field. Optional.

At least one optional parameter must be specified for statement ATTR.

An ATTR statement can be specified for any field, even a field not included in a FIELD statement. The report, however, is not affected by an ATTR for a field until the field is included in a FIELD statement. (This means that an ATTR statement can be defined in advance for each field, as desired, and be ready for use when the field appears in a FIELD statement.)

Member ATTRGEN in the CONTROL-T PARM library contain sample ATTR statements for the GENERAL report.

The following parameters apply to the field whose attributes should be changed:

FIELD=name Name of the field. Mandatory.

TITLE='text' Field header. 'text' is any character string (enclosed in apostrophes). To split the header between lines, insert a slash (/) before each line break. The number of header lines in the report is determined by the field whose TITLE parameter text has the highest number of lines. Optional.

For example, if the following is specified:

```
ATTR          FIELD=VOLSER
              TITLE="Volume/Serial/Number"
```

The header for field VOLSER will span three output lines as follows:

```
Volume
Serial
Number
```

LENGTH=t Length of the output field. t must be a numeric value which must not exceed the field length specified in the list of available fields. Optional. An /Ln value, if specified for the same field in the FIELDS statement, overrides this value.

EDIT='mask' Field format, as specified through a mask. The mask must be enclosed in apostrophes. Parameter EDIT cannot be specified for character and hexadecimal fields. Valid mask formats are:

Numeric Format

For numeric fields only. The mask can consist of Z's and 9's and commas, where 9 represents any digit, Z represents a digit which is suppressed if its value is a leading zero, and comma is a separator between digits.

When specifying a numeric format, the following rules apply:

- A 9 cannot precede a Z in the mask.
- A comma cannot be the first or last character of the mask.

Example

Assume number 1234 is being edited (and b indicates blank)

<u>Mask Specified</u>	<u>Result</u>
'ZZ,ZZ9'	b1,234
'Z,999'	1,234
'999999'	001234
'ZZZ,ZZZ'	bb1,234

Date Format

For Date fields. Valid mask combinations can have no more than one day symbol, one month symbol, and one year symbol.

Valid symbols are:

DD	Day (i.e., values 1 - 31).
MM	Month (i.e., values 1 - 12).
MMM	Month (i.e., values Jun - Dec).
YY	Year (last two digits, i.e., 00 - 99).
YYYY	4-digit year (e.g., 1995).

Symbols can be optionally separated by one or more separation characters: '/', '.', '-', blank, etc.

Example

Assume the date being formatted is August 23, 1994

<u>Mask Specified</u>	<u>Result</u>
'DD/MM/YY'	23/08/94
'YYYY.MM.DD'	1994.08.23
'MM-DD-YYYY'	08-23-1994
'DD MMM YYYY'	23 Aug 1994
'MMM - DD - YYYY'	Aug - 23 - 1994
'YY/MM'	94/08

Boolean Format

For Boolean fields (i.e., fields that contain a True/False or Yes/No condition). The specified FIELD is checked for a true/false condition in the record.

The mask can contain either one value, or two values separated by a slash (/). A null value can be specified on either side of the slash.

When one mask value is specified, the value is printed in the report only if the FIELD checks "true" in the record. (If the FIELD checks "false", the mask value is not printed; blanks are printed instead.)

When two values are specified, the first value is printed in the report if the FIELD checks "true" in the record. The second value is printed if the record checks "false".

Examples

- (1) ATTR FIELD=SCRATCH,
TITLE='Scratch',EDIT='Yes/No'
- (2) ATTR FIELD=ACTIVE,
TITLE='Type',EDIT='Active/Scratch'
- (3) ATTR FIELD=VAULTED, TITLE='V',EDIT='v'
- (4) ATTR FIELD=EDM, TITLE='Edm',EDIT='/No'

The following table shows the results of Boolean editing for these examples.

Example #	Indication Tested	Printed Result if TRUE	Printed Result if FALSE	Field Title
(1)	SCRATCH	Yes	No	Scratch
(2)	ACTIVE	Active	Scratch	Type
(3)	VAULTED	v	(blank)	V
(4)	EDM	(blank)	No	Edm

BREAK

Attributes of the summary line printed at the bottom of the page when a field break occurs. Optional.

For each field break, the user can specify a header and variables which indicate the number of primary and secondary lines printed, and the value of the field before the break occurred.

At least one optional parameter of the BREAK statement must be specified.

A BREAK statement can be specified for any field, even a field that is not included in a SORTBY statement. The report, however, is not affected by a BREAK for a field until the field is included in a SORTBY statement with option /BRK, and SUMMARY=YES has been specified in the REPORT statement.

(This means that a BREAK statement can be defined in advance for each field, as desired, and be ready for use when the field appears in a SORTBY statement and report summaries are included.)

FIELD=f Name of the field. Mandatory.

SUMTITLE= Summary text to be printed when the value of a field
'text' changes. text can be any string of characters. It must be
enclosed in apostrophes and can contain one or more of the
following variables:

%FIELD Value of the field before the break occurred.

%COUNTP Number of primary lines printed before the break occurred.

%COUNT Number of secondary lines printed before the break
occurred.

The default value of SUMTITLE 'text' is:

'TOTAL IN %FIELD IS %COUNT'

Note If parameter ID is not specified in the FIELDS statement or there
is only one FIELDS statement, the value of %COUNT is equal to
the value of %COUNTP.

COLUMN=c Column of the output line in which the summary title will
begin printing. c must be numeric. Optional. The default is
the report start margin specified in parameter MARGINS of
the REPORT statement.

EDIT= Required numeric format of %COUNT and %COUNTP
'numeric variables. Optional. The default is 'ZZ,ZZ9'. (See the
mask' Numeric format example for parameter EDIT in the ATTR
statement above.)

Example

```
REPORT NAME=DISTRIB,SUMMARY=YES
FIELDS ...
SORTBY                    TOLOC/BRK/RSR/RSP,
                          FROMLOC/BRK,
                          TOSLOT,FROMSLOT,
                          VOLSER

BREAK                    FIELD=TOLOC,
                          SUMTITLE='Total Volumes
                          Moved To %FIELD is
                          %COUNT', EDIT='Z,ZZ9'

BREAK                    FIELD=FROMLOC,
                          SUMTITLE='Total Volumes
                          Moved From %FIELD is
                          %COUNT', EDIT='Z,ZZ9'

ATTR ...
```

Special Fields

The following table contains a list of derived fields that can be included in GENERAL report. These fields do not exist in the Media Database records, but their values are derived from either volume or dataset records, as indicated, where:

V Indicates the source is a volume record

D Indicates the source is a dataset record

ROWID is an internal field and therefore no source is indicated.

These fields cannot be subparameters of INCLUDE/EXCLUDE statements.

Field Name	Source	Type	Default Size in Bytes	Description
ROWID		Numeric	5	Row (line) number. It can be specified for either Primary or Secondary lines but not for both.
VOLSTAT	V	Character	12	Volume statuses as shown below the header STATUS in Screen TI. For more information, refer to the description of Screen TI in Section 2 of this manual.
DSSTAT	D	Character	12	Dataset statuses as shown below the header STATUS in Screen TI. For more information, refer to the description of Screen TI in Section 2 of this manual.
VSTATX	V	Hexadecimal	8	Volume statuses in Hexadecimal representation.
DSTATX	D	Hexadecimal	4	Dataset statuses in Hexadecimal representation.
POOL	V	Character	15	Pool name derived from the volser of the volume record.
DSPOOL	D	Character	15	Pool name derived from the volser of the dataset record.

Return Codes

- 0 Operation performed successfully
- 8 Missing DD statement(s), or open error occurred
- 12 Control card error
- 16 Operation code not found
- 20 Invalid operation code
- 24 Syntax error in FIELDS or SORTBY statement
- 28 Field name specified in FIELDS or SORTBY list is undefined
- 32 Same parameter specified more than once
- 36 Length of field name exceeds maximum
- 40 Invalid relation specified (must always be equal sign)
- 44 Report NAME is undefined
- 48 /BRK not specified for the preceding field in SORTBY list
- 52 FIELDS statement not specified
- 56 Total of all field lengths requested in FIELDS statement exceeds the line size
- 60 Continuation card expected but not received
- 64 Invalid numeric value
- 68 Option in FIELDS or SORTBY statement already specified
- 72 ROWID specified in BREAK or SORTBY statement
- 76 EDIT parameter in ATTR statement is invalid for specified FIELDS
- 80 EDIT mask is not valid
- 84 Specified number not in permitted range
For example, end column of a MARGIN statement exceeds the LINESIZE, or a column in a FIELDS statement is outside the report's margins
- 88 Column option (/C) not specified in the preceding field
- 92 Specified column in a FIELDS statement is less than the sum of the previous field's column and length
- 96 Second EXTRACT statement did not come after report statements
- 100 Load of CONTROL-T programs failed
- 104 Load of POOL table failed
- 108 Insufficient memory
- 200 Fields in FIELDS or SORTBY statements do not fit the PATH parameter of the EXTRACT parameters

Sample Reports

The reports listed below can be generated via utility CTTRPT, and are located in the IOA SAMPLE library.

Some of these sample reports contain question marks (??) where customization is necessary. Replace the question marks with the desired values and use the appropriate job to produce the report.

Reports

CTTRFSCR	Full-scratch report. Lists all volumes in scratch status in the Media Database.
CTTRINV	Vault inventory report. Lists all vaulted volumes and datasets in a specified location.
CTTRINVT	Comprehensive inventory report. Lists all volumes and datasets in the Media Database.
CTTRSCRD	Lists all volumes scratched on a specified date, and the datasets residing on these volumes.
CTTRDELV	Lists all volumes in DELETED status.
CTTRCLN	Clean report. Lists volumes that need to be cleaned (i.e., each volume that has more than a specified threshold number of permanent I/O errors).
CTTRRTNP	Lists all datasets (and their volumes) according to expiration date type (e.g., CYCLE, or CATALOG).
CTTRLACV	Lists all volumes accessed today.
CTTRDCRE	Lists all datasets created yesterday.
CTTRMOVW	Lists all volumes moved in the past week (i.e., within the last seven days).
CTTRCHKI	Lists all external volumes checked-in in the past month (i.e., within the last 30 days).
CTTRUTIL	Lists all volumes that are under-utilized (less than 50 feet).
CTTRLAVJ	Lists all volumes that were last accessed by a certain job since a specified date.
CTTRWDCP	Lists all datasets that were written in a certain system (CPU) since a specified date.
CTTROUTL	Lists all volumes in an out-location (borrower volumes).
CTTRLRTV	Lists all volumes with high retention dates (e.g., after year 2000).
CTTRM10V	Lists all multi-volume datasets which span more than ten volumes.
CTTRMACV	Lists the most active volumes (accessed more than ten times since they became active).
CTTREDM	Lists all EDM volumes.
CTTRRCLV	Lists all RECALLED volumes.

CTTRVUPD	Lists all volumes that were manually updated by a specified user.
CTTRDUPD	Lists all datasets that were manually updated by a specified user.
CTTRM25D	Lists all volumes containing more than 25 active datasets.
CTTRVBOX	Lists the volumes stored in a specified box.
CTTRBXLC	Lists all volumes which are stored in boxes, according to location and BOXID.
CTTRNRTR	Lists all borrowed volumes which have not yet been returned.

Jobs for Report Production

The above reports are produced via the following jobs (which are located in the JCL library):

REPFSCR	Used to produce report CTTRFSCR (full-scratch report).
REPINV	Used to produce report CTTRINV (comprehensive inventory report).
REPSJOB	Used for all other reports. This job requires minor customizations that it refers to the desired report.

Example: Extraction and Report Generation in a Single Utility Run

Extract all datasets from location VAULT1 and generate a report showing all VAULT1 datasets and their volumes sorted by their pool name (DSPOOL) which is derived from dataset volser.

```
//EXTRACT EXEC CTTRPT
//SYSIN DD *
  EXTRACT PATH=DATASET/ALLVOLS
  INCLUDE LOCATION=VAULT1
  REPORT NAME=GENERAL, SUMMARY=YES, MARGINS=1-080, LINESIZE=080,
    TITLE='Datasets in Vault VAULT1'
  FIELDS ROWID, FIRSTVOL,
    DSNAME, DSLABEL, CREJBN, CREDIT, LRECL, RECFM,
    ID=P
  FIELDS VOLSER/C13, MOVEDATE, VOLSEQ, SLOTNUM,
    ID=S
  SORTBY DSPOOL/BRK/RSP, FIRSTVOL, DSLABEL, VOLSEQ, SLOTNUM
  BREAK FIELD=DSPOOL, COLUMN=10, EDIT='Z, ZZ9',
    SUMTITLE='%FIELD Datasets in VAULT1: %COUNTP'
  ATTR FIELD=ROWID, TITLE='Row/Id', EDIT='ZZ, ZZ9'
  ATTR FIELD=FIRSTVOL, TITLE='First/Volume'
  ATTR FIELD=DSNAME, TITLE='/Dataset Name', LENGTH=24
  ATTR FIELD=DSLABEL, TITLE='Ds/Seq. ', EDIT='ZZ9'
  ATTR FIELD=CREJBN, TITLE='Create/Job'
  ATTR FIELD=CREDIT, TITLE='Create/Date', EDIT='DD MMM YY'
  ATTR FIELD=LRECL, TITLE='Record/Length', EDIT='ZZ, ZZ9'
  ATTR FIELD=RECFM, TITLE='Record/Format'
  ATTR FIELD=DSPOOL, TITLE='Pool/Name'
  ATTR FIELD=VOLSER, TITLE='/Volser'
  ATTR FIELD=MOVEDATE, TITLE='Move/Date', EDIT='DD MMM YYYY'
  ATTR FIELD=VOLSEQ, TITLE='Vol/Seq. ', EDIT='Z9'
  ATTR FIELD=SLOTNUM, TITLE='Slot/No. ', EDIT='Z, ZZ9'
```

A sample of the output report from the above example is illustrated below:

NEW DIMENSION SOFTWARE, LTD. CONTROL-T REPORT (REL 5.0.0)							
Datasets in Vault VAULT1							

Pool Name: POOL-NUMBER-1							
Row	First	Volume	Dataset Name	Move	Vol	Record	Record
Id	Volume	Dataset	Name	Seq.	Job	Date	Length
	Volser			Date	Seq.		Format
						Slot	
						No.	

58	016475	FDRABR.VTSO006	.C1004900	27	ZCOBKUPD	16 Jun 96	0 U
		012517		23 Jun 1996		55	74
		015692		23 Jun 1996		56	163
		015646		23 Jun 1996		57	162
		016024		23 Jun 1996		58	177
59	016475	FDRABR.VESAD1A	.C1002800	28	ZCOBKUPD	16 Jun 96	0 U
		016024		23 Jun 1996		58	177
		015722		23 Jun 1996		59	167
		016048		23 Jun 1996		60	178
		012771		23 Jun 1996		61	90
60	016475	FDRABR.VVSAM00	.C1001300	29	ZCOBKUPD	16 Jun 96	0 U
		012771		23 Jun 1996		61	90
		010096		23 Jun 1996		62	16
		018862		23 Jun 1996		63	223
		018839		23 Jun 1996		64	222
		012914		23 Jun 1996		65	96
		016314		23 Jun 1996		66	190
		016296		23 Jun 1996		67	187
		016211		23 Jun 1996		68	184
61	016475	FDRABR.VSCLM02	.C1003500	30	ZCOBKUPD	16 Jun 96	0 U
		016211		23 Jun 1996		68	184
		016201		23 Jun 1996		69	183
		016127		23 Jun 1996		70	181
		016337		23 Jun 1996		71	193
		016322		23 Jun 1996		72	192
		015072		23 Jun 1996		73	148
62	016475	FDRABR.VVSAM12	.C1006800	31	ZCOBKUPD	16 Jun 96	0 U
		015072		23 Jun 1996		73	148
		015062		23 Jun 1996		74	147
		011488		23 Jun 1996		75	44
		011035		23 Jun 1996		76	31
		010970		23 Jun 1996		77	29
		010967		23 Jun 1996		78	28
63	017244	CA.BKUPMSTR	.G0502V00	1	CALDPBK	31 Dec 92	8,287 VB
		017244		07 Jun 1993		1	113
		013076		07 Jun 1993		2	110
		018149		07 Jun 1993		3	114

	015278	07 Jun 1993		4		112
	019308	07 Jun 1993		5		115
	013099	07 Jun 1993		6		111
64	018042	TS63.CATALOG.VWGL507.BAC	1	ZTSVSMB4	16 Jun 96	32,756 VBS
	018042	23 Jun 1996		1		9
	018137	23 Jun 1996		2		10
	013684	23 Jun 1996		3		7
	011951	23 Jun 1996		4		5
	011932	23 Jun 1996		5		61

NEW DIMENSION SOFTWARE, LTD. CONTROL-T REPORT (REL 2.1.0)

Datasets in Vault VAULT1

Pool Name: POOL-NUMBER-1

Row Id	First Volume	Dataset Name	DS Seq.	Create Job	Create Date	Record Length	Record Format
	Volser	Move Date		Vol Seq.		Slot No.	
64	018042	TS63.CATALOG.VWGL507.BAC	1	ZTSVSMB4	16 Jun 96	32,756	VBS
	011257	23 Jun 1996		6		40	
	010885	23 Jun 1996		7		25	
	010870	23 Jun 1996		8		24	
	017991	23 Jun 1996		9		209	
	017509	23 Jun 1996		10		203	
	014339	23 Jun 1996		11		136	
	015050	23 Jun 1996		12		145	
	015097	23 Jun 1996		13		149	
65	018228	TS63.CATALOG.VMVSCAT.BAC	1	ZTSVSMB1	16 Jun 96	32,756	VBS
	018228	23 Jun 1996		1		11	
66	018288	TS63.CATALOG.VWGL83A.BAC	1	ZTSVSMBA	16 Jun 96	32,756	VBS
	018288	23 Jun 1996		1		13	
	018280	23 Jun 1996		2		12	
67	018358	TS63.CATALOG.VWGL514.BAC	1	ZTSVSMB3	16 Jun 96	32,756	VBS
	018358	23 Jun 1996		1		14	

POOL-NUMBER-1 Datasets in VAULT1: 67

Examples: Extraction and Report Generation in Separate Runs

Extract Example

Extract all volumes located in VAULT1, which was created by job ZCOIBKUP in June 1996.

```
//EXTRACT EXEC CTTRPT
//SYSIN DD *
  EXTRACT PATH=VOLUME/ALLDS
  INCLUDE LOCATION=VAULT1,CREJBN=ZCOIBKUP
  EXCLUDE CREDIT<19960601
  EXCLUDE CREDIT>19960630
```

Report Formatting Example

```
//INVREP EXEC CTTRPT
//SYSIN DD *
  REPORT NAME=GENERAL, SUMMARY=YES, MARGINS=1-80, LINESIZE=80,
    TITLE='Inventory Report '
  FIELDS ROWID, VOLSER, VOLSTAT, MOVEDATE, VOLSEQ, SLOTNUM, VOLEXPDT,
    ID=P
  FIELDS DSNAME/C13, DSLABEL, CREJBN, CREDIT, BLOCKCT, DSEXPDT,
    ID=S
  SORTBY ACTIVE/BRK/RSP, POOL/BRK/RSP,
    LOCATION/BRK/RSR, VOLSER, DSLABEL
  BREAK FIELD=LOCATION, EDIT='ZZ9',
    SUMTITLE='Total number of volumes in %FIELD: %COUNTP'
  BREAK FIELD=ACTIVE, EDIT='ZZ,ZZ9',
    SUMTITLE='Number of %FIELD Volumes: %COUNTP'
  BREAK FIELD=POOL, EDIT='ZZ9',
    SUMTITLE='Total number of volumes in %FIELD: %COUNTP(%COUNT) '
  ATTR FIELD=ROWID, TITLE='Row/Id', EDIT='ZZ,ZZ9'
  ATTR FIELD=VOLSER, TITLE=' /Volser'
  ATTR FIELD=VOLSTAT, TITLE='Volume/Status'
  ATTR FIELD=MOVEDATE, TITLE='Move/Date', EDIT='DD MMM YYYY'
  ATTR FIELD=VOLSEQ, TITLE='Vol/Seq.', EDIT='Z9'
  ATTR FIELD=SLOTNUM, TITLE='Slot/No.', EDIT='Z,ZZ9'
  ATTR FIELD=VOLEXPDT, TITLE='Volume/Expiration', EDIT='DD MMM YYYY'
  ATTR FIELD=DSNAME, TITLE=' /Dataset Name', LENGTH=24
  ATTR FIELD=DSLABEL, TITLE='Ds/Seq.', EDIT='ZZ9'
  ATTR FIELD=CREJBN, TITLE='Create/Job'
  ATTR FIELD=CREDIT, TITLE='Create/Date', EDIT='DD MMM YY'
  ATTR FIELD=BLOCKCT, TITLE='Block/Count', EDIT='ZZ,ZZ9'
  ATTR FIELD=DSEXPDT, TITLE='Dataset/Expiration', EDIT='YYYY-MMM-DD'
  ATTR FIELD=ACTIVE, TITLE='Type', EDIT='ACTIVE/SCRATCH'
  ATTR FIELD=LOCATION, TITLE='Location'
  ATTR FIELD=POOL, TITLE='Pool/Name'
/*
```


The following is a sample output produced by the above example. Note the following about the output:

1. The page number, date, and time are not printed unless the LINESIZE specified in the REPORT statement is at least 127.
2. During printing, if there is insufficient room on the current page for secondary lines, the primary line is reprinted as part of the page header on the next page.
3. A field specified with option /BRK in a SORTBY statement is shown in the page header following the report title, and any slashes (/) specified in the TITLE definition are suppressed to blanks.

```

NEW DIMENSION SOFTWARE, LTD. CONTROL-T REPORT (REL 5.0.0)

                          Inventory Report
                          -----
Type:  ACTIVE      Pool Name: POOL-NUMBER-1      Location: VAULT1

Row          Volume      Move          Vol   Slot   Volume
Id           Volser      Status        Date   Seq.   No.    Expiration
          Dataset Name          DS   Create   Create   Block  Dataset
          Dataset Name          Seq. Job   Date     Count  Expiration
-----
1   010222   Vaulted      Jun 20-1996      2    427   21 Jun 1996
   FDRABR.VPROD02.C1005204      6  ZCOIBKUP  14 Jun 96      0  1996-Jun-21
   FDRABR.VPROD01.C1008804      5  ZCOIBKUP  14 Jun 96      0  1996-Jun-21
   FDRABR.VSAS001.C1006504      4  ZCOIBKUP  14 Jun 96      0  1996-Jun-21
   FDRABR.VTSO006.C1004804      3  ZCOIBKUP  14 Jun 96      0  1996-Jun-21
   FDRABR.VTSO003.C1015904      2  ZCOIBKUP  14 Jun 96     999  1996-Jun-21

2   010372   Vaulted      Jun 20-1996      1    433   21 Jun 1996
   FDRABR.VTSO003.C1015904      2  ZCOIBKUP  14 Jun 96     999  1996-Jun-21
   FDRABR.VTSO002.C1015904      1  ZCOIBKUP  14 Jun 96     999  1996-Jun-21

3   011938   Vaulted      Jun 20-1996     14    487   21 Jun 1996
   FDRABR.VWGLE00.C1035504     17  ZCOIBKUP  14 Jun 96      0  1996-Jun-21
   FDRABR.VWGLE1C.C1027404     16  ZCOIBKUP  14 Jun 96      0  1996-Jun-21
   FDRABR.VMVS89C.C1005304     15  ZCOIBKUP  14 Jun 96      0  1996-Jun-21
   FDRABR.VMVS89D.C1005304     14  ZCOIBKUP  14 Jun 96      0  1996-Jun-21
   FDRABR.VMVSRI1A.C1028204    13  ZCOIBKUP  14 Jun 96      0  1996-Jun-21
   FDRABR.VWGL800.C1006704     12  ZCOIBKUP  14 Jun 96      0  1996-Jun-21

4   011954   Vaulted      Jun 20-1996     15    492   21 Jun 1996
   FDRABR.VWGLE514.C1041704     20  ZCOIBKUP  14 Jun 96      0  1996-Jun-21
   FDRABR.VWGLE13.C1031304     19  ZCOIBKUP  14 Jun 96      0  1996-Jun-21
   FDRABR.VMVSRES.C1018704     18  ZCOIBKUP  14 Jun 96      0  1996-Jun-21
    
```

	FDRABR.VWGLE00.C1035504	17	ZCOIBKUP	14 Jun 96	0	1996-Jun-21	
5	012016 Vaulted			Jun 20-1996	13	506	21 Jun 1996
	FDRABR.VWGL800.C1006704	12	ZCOIBKUP	14 Jun 96	0	1996-Jun-21	
	FDRABR.VSCLM03.C1001304	11	ZCOIBKUP	14 Jun 96	0	1996-Jun-21	
6	012191 Vaulted			Jun 20-1996	3	529	21 Jun 1996
	FDRABR.VPROD04.C1016204	7	ZCOIBKUP	14 Jun 96	0	1996-Jun-21	
	FDRABR.VPROD02.C1005204	6	ZCOIBKUP	14 Jun 96	0	1996-Jun-21	
7	012271 Vaulted			Jun 20-1996	9	539	21 Jun 1996
	FDRABR.VSCLM03.C1001304	11	ZCOIBKUP	14 Jun 96	0	1996-Jun-21	
8	012313 Vaulted			Jun 20-1996	12	544	21 Jun 1996
	FDRABR.VSCLM03.C1001304	11	ZCOIBKUP	14 Jun 96	0	1996-Jun-21	
9	013315 Vaulted			Jun 20-1996	4	551	21 Jun 1996
	FDRABR.VPROD06.C1016204	9	ZCOIBKUP	14 Jun 96	0	1996-Jun-21	
	FDRABR.VPROD05.C1015904	8	ZCOIBKUP	14 Jun 96	0	1996-Jun-21	
	FDRABR.VPROD04.C1016204	7	ZCOIBKUP	14 Jun 96	0	1996-Jun-21	
10	014347 Vaulted			Jun 20-1996	11	568	21 Jun 1996
	FDRABR.VSCLM03.C1001304	11	ZCOIBKUP	14 Jun 96	0	1996-Jun-21	
11	014743 Vaulted			Jun 20-1996	10	572	21 Jun 1996
	FDRABR.VSCLM03.C1001304	11	ZCOIBKUP	14 Jun 96	0	1996-Jun-21	
12	016465 Vaulted			Jun 20-1996	5	583	21 Jun 1996
	FDRABR.VSCLM01.C1008904	10	ZCOIBKUP	14 Jun 96	0	1996-Jun-21	

NEW DIMENSION SOFTWARE, LTD. CONTROL-T REPORT (REL 5.0.0)

Inventory Report

Type: ACTIVE Pool Name: POOL-NUMBER-1 Location: VAULT1

Row Id	Volume Volser	Volume Status	Move Date	Vol Seq.	Slot No.	Volume Expiration
DataSet Name			Ds Create Seq. Job	Create Date	Block Count	Dataset Expiration
12	016465	Vaulted	Jun 20-1996	5	583	21 Jun 1996
FDRABR.VPROD06.C1016204			9 ZCOIBKUP	14 Jun 96		0 1996-Jun-21
13	017248	Vaulted	Jun 20-1996	6	587	21 Jun 1996
FDRABR.VSCLM01.C1008904			10 ZCOIBKUP	14 Jun 96		0 1996-Jun-21
14	018377	Vaulted	Jun 20-1996	8	596	21 Jun 1996
FDRABR.VSCLM03.C1001304			11 ZCOIBKUP	14 Jun 96		0 1996-Jun-21
FDRABR.VSCLM01.C1008904			10 ZCOIBKUP	14 Jun 96		0 1996-Jun-21
15	018393	Vaulted	Jun 20-1996	7	596	21 Jun 1996
FDRABR.VSCLM01.C1008904			10 ZCOIBKUP	14 Jun 96		0 1996-Jun-21

Total number of volumes in VAULT1 : 15

POOL-NUMBER-1 Volumes: 73 , POOL-NUMBER-1 Datasets: 581

Number of ACTIVE Volumes: 221

CTTRTM - Perform Retention Management

This utility performs batch retention management functions. Batch retention management is performed daily to check expiration criteria and determine which datasets should be marked scratch. When all the datasets on a volume have expired, the volume becomes a scratch volume. The utility should be run as part of the CONTROL-T New Day procedure (CTTDAY).

Utility CTTRTM supports a number of different types of retention. See "Retention Management" in Section 4 of the CONTROL-T User Manual for a list of the supported retention types.

Utility CTTRTM does not perform retention management for the following volumes:

- Vaulted volumes
- Volumes not in the currently active library (i.e., volumes which have a status of Act-Out)
- Volumes with a status of Pend-Vault
- Volumes which have been recalled from a vault
- Volumes with a status of Act-Use
- Deleted volumes
- EDM controlled volumes.

Note Retention information for these volumes is updated by utility CTTRTM if either *CONVERT* or *RECALC* is specified for parameter *TYPERET* (described later). However, these volumes (and the datasets they contain) are not expired by the utility (i.e., they are not marked as scratched).

Scratch Report

Utility CTTRTM can produce a Scratch report at the end of processing. The Scratch report contains a list of all volumes which have expired (i.e., become scratch) as a result of this utility run.

Note Full Scratch report *FULLSCR* is no longer generated by utility CTTRTM. Instead, this report can be generated as part of the *GENERAL* report by utility CTTRPT. For more information see utility CTTRPT earlier in this section.

Utility CTTRTM adds the following special pool names in the POOLNAME field of certain volume records in the Scratch report:

EXTERNAL	External volumes not deleted from the Media Database
EXT-DEL	External volumes deleted from the Media Database.
EXT-DUP	External volumes which are not deleted from the Media Database but whose SL-NAME and VOLSER are different.
DUPLICATE VOL	Internal volumes whose SL-NAME and VOLSER are different.

The special pool name is written over the previous name in field POOLNAME (if one existed).

Scratched volumes with different SL-NAMEs and VOLSERs (listed in the EXT-DUP and DUPLICATE VOL pools) should each be re-initialized (via utility CTTTPI) to set the SL-NAME to the value of the VOLSER.

Recovery From Abends

Since utility CTTRTM updates the Media Database in several stages, a CANCEL operation, job abend or system crash can lead to logical inconsistencies in the database.

CONTROL-T can recover from abends occurring during the run of utility CTTRTM. The type of recovery depends on the stage of the utility in which the abend occurred. Utility CTTRTM performs retention management in the following stages:

1. A physical file is produced which contains the information necessary for updating the relevant records in the Media Database.
2. The Media Database records are updated.
3. A Scratch report is produced.

If an abend occurred during Step 1, no updates were performed on the Media Database, and the utility can be rerun normally.

If an abend occurred while the Media Database was being updated (Step 2), the next time utility CTTRTM is run it will automatically enter RESTART mode and continue updating the Media Database from where the previous run terminated. (The utility uses the physical file created in Step 1, and a timestamp which indicates the point at which the previous run of utility CTTRTM was interrupted.)

If an abend occurred after all updates of the Media Database were performed but the Scratch report was not produced, this report can be generated by rerunning the utility with the following parameters:

```

TYPERUN      MODE=NORMAL
REPORT       NAME=SCRATCH
FIELDS       fields
SORTBY       fields
  
```

Where:

fields Same fields that were specified in the FIELDS and SORTBY statements in the original run of utility CTTRTM.

Note *No value should be specified for parameter TYPERET in this run of utility CTTRTM.*

Exits Used by Utility CTTRTM

The following exits can be used to influence a run of utility CTTRTM:

- CTTX006 Receives control immediately before updates are made to dataset/volume records in the Media Database. This exit can be used to accept or reject expiration of a dataset record, and/or to update a dataset/volume record before it is marked as scratched.
- CTTX008 Receives control when the Scratch report is used to update the information in the robotic tape library (if one exists at your site). This exit can be used to scan the scratch report and to determine which volume records are updated in the robot's database.

Relevant Installation Parameters

Installation parameters (specified in member CTT Parm) determine how utility CTTRTM operates. These parameters are described below.

RTNTYPE	Specifies when the datasets on the volume will be scratched.
VOL	Each dataset on the volume will be marked scratch only if all datasets on the volume have expired. This option is recommended for compatibility with CA-1 and CA-TLMS.
DSN	Each dataset record's expiration criteria in the Media Database is independent of other datasets on the volume.

Note *A multi-volume chain functions independently of parameter RTNTYPE. Only if all datasets on each volume have expired will all the datasets of the chain be marked scratch and all volumes of the chain become scratch.*

CYCLECNT	Defines how to calculate cyclic datasets. Valid values are:
DS1	Handles datasets with the same dataset name and creation date as a single cycle. Default.
DS	Handles datasets with the same dataset name as separate cycles. The creation date is ignored.
DSJ	Handles datasets with the same dataset name and job name as a single cycle. The creation date is ignored.
DSJ1	Handles datasets with the same dataset name, creation date and job name as a single cycle.

EXPDTYPE	Determines expiration logic depending on the type of retention criteria and value specified:
-----------------	--

- When DATE or DAYS retention criteria are assigned to a dataset:

CA1	CA-1 expiration logic applies. DATE or DAYS specifies the day on which the dataset is expired. Default.
------------	---

TLMS	CA-TLMS expiration logic applies. DATE or DAYS specifies the last day of retention. The dataset is expired the following day.
-------------	---

NONE	Expiration date in standard MVS format.
-------------	---

- When cyclic retention criteria are specified:

CA1	CA-1 expiration logic applies. Only primary datasets (datasets with LABEL=1) are considered in the expiration calculation. Default.
------------	---

Other	All cyclic datasets are considered regardless of label number.
--------------	--

CTLGWAIT	Defines the number of wait days before utility CTTRTM checks a catalog-controlled dataset (i.e., one for which DO RETENTION=MVS CATALOG is specified). In some cases, a dataset is created during a job step and cataloged at the end of a later job step. If utility CTTRTM is executed after the dataset is created but before it is cataloged, the utility expires the dataset. Parameter CTLGWAIT causes utility CTTRTM to wait a specified number of days before checking if the dataset is cataloged.
MODE	Determines whether or not to uncatalog the expired datasets in the MVS Catalog.
	TEST Datasets are not uncataloged.
	PROD or Datasets are uncataloged if no Test rules match the expired PHASE datasets (see parameter TESTRUL below).
SMSINTR	Determines whether or not the CONTROL-T to DF/SMS interface is active. If this interface is active, and expiration dates are defined by SMS managed classes, retention is dependent on expiration dates specified in SMS Management Classes.
TESTRUL	Determines whether or not Test rules are used in the production environment (global MODE=PROD or PHASED). If an expired dataset matches a Test rule, the dataset should not be uncataloged.
	YES Test rules are used. The utility loads and scans the CONTROL-T rules.
	NO Test rules are not used. The utility does not load and scan the CONTROL-T rules. Default.

For additional information refer to "Retention and Vault Parameters" in the IOA Installation Guide.

Activating the Utility

```
// EXEC      CTTRTM
//SYSIN      DD *
    TYPERUN  MODE=NORMAL|SIMULATION|[,DATE=date][,TRACE=YES|NO]
             [,DBGLEVEL=level][,RESTART=YES|NO][,LIST=YES|NO|FULL]
    TYPERET  MODE=REGULAR|RECALC|CALCONV|CONVERT[,UNCTLG=YES|NO]
             [,UPDVOL=YES|NO]
<INCLUDE/EXCLUDE statements>
REPORT NAME=SCRATCH,...
FIELDS ...
SORTBY ...
ATTR ...
BREAK ...
//
```

Parameters

Utility parameters are supplied via DD statement SYSIN:

Note While the *TYPERET* and *REPORT* parameters are each optional, at least one of them must be specified.

TYPERUN	Operation type. Mandatory.
MODE	Operation mode. Mandatory.
	NORMAL Normal operation is performed. The Media Database is updated as required by utility CTTRTM.
	SIMULATION Simulation is performed. No actual changes are made to the Media Database. In this mode, CONTROL-T keeps in memory a list of Media Database records that would have been updated if the utility ran in NORMAL mode.

Notes When utility CTTRTM is run in SIMULATION mode, it requires a much larger region than required in NORMAL mode. The actual region size necessary depends on the number of updated Media Database records (which this utility stores in memory).

Utility CTTRTM can be run in SIMULATION mode at the same time as a run of utility CTTRTM is run in NORMAL mode. However more than one run of utility CTTRTM in NORMAL mode is not permitted.

DATE	Operation date in 6- or 8-character format according to site standard. (For example, in USA standard the date can be specified mm/dd/yy or mm/dd/yyyy.) Optional.
TRACE	Specifies whether or not the Trace file will be used during CTTRTM run. Optional.
	YES The Trace file will be used. Default.
	NO The Trace file will not be used.
DBGLEVEL	Debug level. A numeric value from 1 to 255 can be specified.

Note This parameter should only be used if requested by your IOA representative.

RESTART	Specifies whether or not to enable Automatic Restart mode when the utility is run after an abend or cancellation of the previous run of the utility. Valid values are:
YES	Enable Automatic Restart mode. Default.
NO	Disable Automatic Restart mode.
LIST	Specifies whether or not specific diagnostic messages (i.e., CTT301I, CTT302I, CTT493I, and CTT494I) will be produced. The messages are sent to SYSPRINT. Valid values are:
NO	None of the above mentioned messages are produced.
YES	Messages CTT301I, CTT302I, and CTT493I are produced. Default.
FULL	All of the above mentioned diagnostic messages are produced.
TYPERET	Retention management type. Optional if parameter REPORT is specified. If TYPERET is not specified, retention processing is not performed, but the report produced by the last run of utility CTTRTM is regenerated.
MODE	Retention management mode.
REGULAR	Retention management is performed according to the retention criteria currently specified in dataset records in the Media Database.
CONVERT	This parameter is used after conversion from a different Tape Management System (e.g., CA-TLMS). Expiration date information is extracted from one of the following locations (depending on the specifications for CONTROL-T installation parameters): <ul style="list-style-type: none"> - CONTROL-T rules - JCL EXPDT parameters - SMS Management Classes - Default values defined in member CTTPARM <p>The extracted retention information is used to update dataset records in the Media Database. However, no actual expiration is performed. This enables the user to check that the conversion works as planned, before retention management is implemented.</p>

Note *TYPERET=CONVERT should only be specified in the following cases:*

- *This is the first run of utility CTTRTM since conversion from the previous Tape Management System, and retention information does not exist in the dataset records created as a result of this conversion (e.g., CA-TLMS).*
 - *Rules have been defined with new retention information that needs to be incorporated into the dataset records in the Media Database.*
-

RECALC Retention information in dataset records in the Media Database is updated, and expiration is performed. Running utility CTTRTM with TYPERET=RECALC specified has the same effect as running this utility twice; once with TYPERET=CONVERT specified, and a the second time with TYPERET=REGULAR specified.

CA1CONV This parameter is used only after CA-1 conversion. In this mode, no retention actions are taken. This mode is used to fill in the dataset name's prefix length (as defined in CONTROL-T rules) in the Media Database dataset records. This information is later used for retention management of cyclic datasets.

Note *If utility CTTRTM has already been run with TYPERET=CONVERT specified, it is not necessary to run it again with TYPERET=CA1CONV specified.*

UNCTLG Specifies whether or not to uncatalog datasets after scratching the dataset records from the Media Database. Optional.

YES Uncatalog the datasets. Default.

NO Do not uncatalog the datasets.

UPDVOL Specifies whether or not to copy the retention information from the dataset records to the volume where the datasets reside. (The highest retention period is stored in the volume record.) Applies only if the MODE specified in the TYPERET statement is RECALC or CONVERT. Valid values are:

YES Copy the retention information to the volume record.

NO Do not copy the retention information to the volume record. Default.

Specifying NO can significantly improve performance time when running the utility in CONVERT or RECALC mode. However, accurate retention information will only be available via inquiry of dataset records in the Media Database (i.e., an inquiry of the volume records may not yield accurate retention information).

INCLUDE/
EXCLUDE

The scope of the utility can be limited by one or more INCLUDE or EXCLUDE statements. INCLUDE/EXCLUDE statements are specified immediately after the TYPERET parameter and are applied to retention processing.

For additional information on INCLUDE and EXCLUDE statements, see "Record Selection Logic (INCLUDE/EXCLUDE Statements)" earlier in this section.

See also "Logical Field Names for the CONTROL-T Repository" in Appendix D.

Note As of CONTROL-T Release 2.1.0, INCLUDE/EXCLUDE statements cannot be specified after SORTBY parameters.

With few exceptions, the parameters and options of the REPORT, FIELDS, SORTBY, ATTR and BREAK statements are identical to those found in utility CTTRPT. For a complete description of these parameters and options, refer to utility CTTRPT earlier in this section. Differences between the parameters and options of utility CTTRTM and utility CTTRPT are described below.

REPORT Activates report utility. Optional if a TYPERET parameter is specified.

NAME Report name. Mandatory. Valid value is:

SCRATCH Scratch report listing volumes which expired based on the execution results of utility CTTRTM.

Note Full scratch report FULLSCR is no longer generated by utility CTTRTM. Instead, this report can be generated as part of the GENERAL report by utility CTTRPT. A sample Full Scratch report can be found in member REPFSCR in the CONTROL-T JCL library.

FIELDS Fields to print in the report. The fields are printed in the order they are specified. Available fields are:

MEDIA Media type.

POOL Pool name.

VOLSER Volume serial number.

SL-NAME Standard label name of the volume.

UNAME Unit name.

FIRSTVOL First volser in a multi-volume group.

DSNAME Name of the first dataset in the volume.

CREDS Creation date of the first dataset in the volume.

EXPDS Expiration date of the first dataset in the volume.

EXPDT Expiration date of the volume.

LACCDT Last access date of the volume.

LOCATION Location.

CHKINDT Check-in date.

IOERR Number of input/output errors.

LCLNDT Last clean date.

VOLSTAT Status(es) of the volume.

SORTBY Order in which the report is sorted. (Available fields are listed above under parameter FIELDS.) The /B option can be added as a suffix to any field to indicate that a page break should be inserted in the report when the field changes its value.

ATTR Override of default attributes defined for a field (see FIELDS above). Optional.

BREAK Attributes of the summary header line(s) printed when a field break occurs. Optional.

Return Codes

0 Operation performed successfully

8 Missing DD statement(s), or open error occurred

12 Control card error

16 Load of CONTROL-T routines failed

20 Sort error occurred

24 Media Database input/output error occurred

28 Internal error occurred

32 Error in report printing utility occurred

36 Load of rule, pool, or vault table failed

40 Authorization error (by security exit)

44 Media Database file in use
 48 Insufficient storage
 52 Logical error in Media Database

Examples

In the following example, retention management is performed according to the current retention specifications in the Media Database. Report SCRATCH is printed, listing volser and media type, and is sorted according to pool name. The list for each media type starts on a new page.

```
// EXEC      CTTRTM
//SYSIN      DD *
    TYPERUN  MODE=NORMAL
    TYPERET  MODE=REGULAR
    REPORT   NAME=SCRATCH
    FIELDS   POOL,VOLSER,MEDIA
    SORTBY   POOL,MEDIA/B,VOLSER
//
```

In the following example, retention information is updated for all datasets whose name begins with prefix BKP.

```
// EXEC      CTTRTM
//SYSIN      DD *
    TYPERUN  MODE=NORMAL
    TYPERET  MODE=CONVERT
    INCLUDE  DSNAME=BKP*
    REPORT   NAME=SCRATCH
    FIELDS   EXPDT,MEDIA,VOLSER,POOL
    SORTBY   EXPDT
//
```

In the following example, the utility is used to produce the scratch report from the last finished run of utility CTTRTM.

```
// EXEC      CTTRTM
//SYSIN      DD *
    TYPERUN  MODE=NORMAL
    REPORT   NAME=SCRATCH
    FIELDS   fields
    SORTBY   fields
//
```

CTTSPL - Split the Media Database

Utility CTTSPSPL extracts (i.e., moves or copies) Media Database records for transfer to a different Media Database. Before using this utility, it is important to review the information in "Split and Merge Processing".

Notes *There is no need to shut down CONTROL-T while running this utility.*

Utility CTTMER (used to merge information extracted via utility CTTSPSPL) automatically updates the Media Database index component. there is no need to run utility CTTBIX after the run of this utility.

Split and Merge Processing

The split and merge process is the method by which information is extracted (split) from one Media Database and inserted (merged) into another Media Database (usually at a different site). This process is especially useful when a decision is made to join two Media Databases, or to split a large Media Database in two or more smaller Media Databases.

The split and merge process is performed via utilities CTTSPSPL and CTTMER.

- Utility CTTSPSPL extracts records from the source Media Database according to specified selection criteria. This utility can be used with either of two functions.

MOVE Moves selected records to the extract file and deletes them from the source Media Database.

COPY Copies selected records from the source Media Database to the extract file. The source Media Database is not altered when this function is specified.

The output of utility CTTSPSPL (the extract file) is a predefined sequential file (olpreft.olvert.REPDATA) allocated as one of the CONTROL-T operations libraries at time of CONTROL-T installation.

- Utility CTTMER merges the extracted Media Database records into the Media Database in the target system. Utility CTTMER uses the extract file created by CTTSPSPL as input.

If utility CTTSPSPL is being used to move (not copy) Media Database records, the selected records are deleted from the source Media Database.

If the target Media Database is in a different site, the extract file (REPDATA) produced by utility CTTSPSPL must be transferred to the target site.

Data Integrity

Utilities CTTSPSPL and CTTMER do not check Media Database integrity.

It is highly recommended that utility CTTIDB be used to check the source Media Database integrity before execution of utility CTTSPSPL. To avoid transfer of data errors to the target Media Database, integrity errors detected by utility CTTIDB should be fixed in the source Media Database before execution of utility CTTSPSPL.

It is also recommended that utility CTTIDB be used after the split and merge process to check the integrity of the:

- Source Media Database after running utility CTTSPSPL (if the MOVE function was specified).
- Target Media Database after execution of utility CTTMER.

Simulation Mode

Utilities CTTSPL and CTTMER can be executed in simulation mode. In this mode, no Media Database updates are performed. In simulation mode, messages are sent to the utility log to record the updates that would be performed on the Media Database if the utility was running in regular mode.

If the last run of the utility abended, simulation mode is not allowed. If an attempt is made to run the utility in simulation mode, the utility is terminated and an appropriate message is displayed.

A special restart simulation mode is available for utility CTTMER. (See the description of the MODE parameter in the explanation of utility CTTMER for more information.)

Note Since the Media Database is not updated in simulation mode, it is possible that a run of one of these utilities in simulation mode will end successfully, while the corresponding run in normal mode could abend (e.g., due to a lack of free space in the Media Database).

Volume Status

Utilities CTTSPL and CTTMER do not check volume status. Volume records are split and merged regardless of volume status. For some volumes (e.g., those with a status of VAULTED or PENDING-VAULT), this can result in inaccurate slot definitions.

To correct this problem, it is necessary to run utility CTTVTM in SLOTBLD mode to rebuild the Slot Definition Records. Utility CTTVTM should be run against both the source Media Database and the target Media Database after completion of the split and merge process.

Recovery from Abends

CONTROL-T can recover from abends occurring during the run of utilities CTTSPL and CTTMER. If the utility has abended during update of the Media Database (deletion of records in utility CTTSPL, or addition of records in utility CTTMER), when the utility is run again it will automatically enter RESTART mode and recover from the abend.

The extract file (REPDATA) must not be altered (or deleted) prior to restart execution of utility CTTSPL or CTTMER. It is highly recommended that this file be backed up after each run of utility CTTSPL to facilitate recovery.

Undoing Splits and Merges

You can choose to undo the actions of the Split/Merge utilities as follows:

Splits If utility CTTSPL was used with the MOVE function. Utility CTTMER should be used to merge the extracted records from the extract (REPDATA) file back into the source Media Database (specify the source Media Database as the target Media Database).

Note If the extract file no longer exists but the extracted records have been merged into the target Media Database, they can be extracted (copied or moved) from the target Media Database into another extract file, and then merged back into the source Media Database.

If utility CTTSPL was used with the COPY function, the source Media Database is unchanged and there is no need to “undo” anything.

Merges To remove records that have been merged into the target Media Database, use utility CTTSPL with the MOVE function to extract the necessary Media Database records from the target Media Database. The selection statements passed to the utility must include only the volumes added previously by utility CTTMER.

Activating Utility CTTSPL

```
// EXEC CTTSPL
//SYSIN DD *
  EXTRACT PATH=ALL
  <INCLUDE/EXCLUDE statements>
/*
//DAIN DD *
  TYPERUN FUNCTION=COPY|MOVE[,MODE=NORMAL|SIMULATE][,DBGLEVEL=XXX]
/*
```

Parameters

EXTRACT EXTRACT statement specified via DD statement SYSIN. This is the only
PATH=ALL EXTRACT statement allowed with utility CTTSPL. Mandatory.

INCLUDE/ The scope of the utility can be limited by one or more INCLUDE and/or
EXCLUDE EXCLUDE statements.

INCLUDE/EXCLUDE statements are specified immediately after the EXTRACT statement in the SYSIN DD statement. For more information see “Record Selection Logic” earlier in this section.

TYPERUN The TYPERUN statement is specified via DD statement DAIN. The format for this statement is:

```
TYPERUN FUNCTION=COPY|MOVE[,MODE=NORMAL|SIMULATE][,DBGLEVEL=XXX]
```

Where:

FUNCTION The function to be performed. Mandatory. Valid values are:

COPY Records are to be copied (not deleted) from the source Media Database to the extract file (REpdata). Copied records are not deleted from the source Media Database.

MOVE Records are moved from the source Media Database to the extract file. Moved records are deleted from the source Media Database.

MODE	Operation mode. Optional.
NORMAL	Normal operation mode. Default.
SIMULATE	Simulation mode. No changes are made to the Media Database. Messages are sent to the utility log to show which changes would be made to the Media Database if the run was in NORMAL mode.

Note The result of running utility CTTSPLE in simulation mode is identical to running it with FUNCTION=COPY specified. The resulting extract file (REPDATA) can be used as input for a regular (or simulation mode) run of utility CTTMER.

DBGLEVEL	Debug level. A numeric value from 1 to 255 can be specified. Optional.
----------	--

Note This parameter should be used only if requested by your IOA representative.

Example 1

In this example, utility CTTSPLE extracts the records for all volumes beginning with VOL. The records for the volumes and their associated datasets are removed (deleted) from the source Media Database and placed in the extract file (REPDATA) to be used later as input for utility CTTMER.

```
// EXEC CTTSPLE
//SYSIN DD *
  EXTRACT PATH=ALL
  INCLUDE VOLSER=VOL*
/*
//DAIN DD *
  TYPERUN FUNCTION=MOVE
/*
```

Example 2

In this example, utility CTTSPPL extracts the records of all datasets beginning with MY.FILE. The records for these datasets and the associated volumes are copied (not deleted) from the source Media Database and placed in the extract file (REPDATA) to be used later as input for utility CTTMER.

```
// EXEC CTTSPPL
//SYSIN DD *
    EXTRACT PATH=ALL
    INCLUDE DSNAME=MY.FILE*
/*
//DAIN DD *
    TYPERUN FUNCTION=COPY
/*
```

Return Codes

0	Operation performed successfully.
8	Initialization problem
12	Extract failed
16	Split failed
20	Simulation mode is not allowed while restart is required.
Greater than 20	Severe error - the error is accompanied by a message describing the problem

CTTSTK - Build and Update Stacking Statistics File

This utility builds and updates the Stacking Statistics file. The utility should be run on a regular basis via job CTTDAY.

The utility accepts input from one of the following three types of datasets:

- System Management Facilities (SMF) accumulated type 15 records (not the actual SMF files)
- Trace file records
- Media Database records

Input records can be selected according to INCLUDE/EXCLUDE statements supplied to the utility. Record timestamps are registered in the dataset. Records from different utility runs with earlier timestamps are considered duplicate records and are dropped. Therefore, records with overlapping timestamps from different source computers should not be supplied to the utility in the same run.

Activating the Utility

```
// EXEC    CTTSTK
//SYSIN   DD *
          TYPERUN MODE=SMF|TRC|MDB
          <INCLUDE/EXCLUDE STATEMENTS>
//
```

Parameters

Utility parameters are supplied via DD statement SYSIN:

TYPERUN	Statement type.
MODE	Source of input records.
SMF	Input records come from accumulated SMF type 15 records. In this mode, parameter SMFIN (supplied to procedure CTTSTK) specifies the name of a file containing accumulated SMF records.
TRC	Input records come from the Trace file.
MDB	Input records come from the Media Database.

INCLUDE/EXCLUDE

The scope of the utility can be limited by one or more INCLUDE or EXCLUDE statements. The subparameters which can be specified in INCLUDE/EXCLUDE statements are listed in Table 3 in "Logical Field Names for the CONTROL-T Repository" in Appendix D.

For additional information on INCLUDE and EXCLUDE Statements, see "Record Selection Logic (INCLUDE/EXCLUDE Statements)" earlier in this section.

Return Codes

0	Operation performed successfully
4	Input statement(s) missing
8	DCB open error occurred
12	Input statement parsing error occurred
16	CONTROL-T Control Table (TCT) load failed
20	Sort error occurred
24	Media Database/ Stacking Statistics file /Trace file I/O error occurred

CTTTPI - Tape Initialization, Information Mapping, and Tape Erasure

The utility provides the following major functions:

- **Safe Tape Initialization (INITT)** Initializes tapes and cartridges. Checks that a volume is not active before initializing it, thereby ensuring that active volumes are not accidentally initialized.
- **Media Information Mapping (TAPEMAP)** Produces a detailed report about the logical and physical content of a volume including logical files, labels, tape marks, and unreadable areas (I/O errors).
- **Safe Tape Erasure (TAPERAS)** Physically erases the contents of a volume. Checks that a volume is not active before erasing it, thereby ensuring that active volumes are not accidentally erased.

These functions are described in detail.

Multiple jobs or started tasks can run utility CTTTPI in parallel on separate tape drives for distinct ranges of volume serial numbers.

(See "Preventing Unauthorized Access to the Tape Initialization and Information Mapping Utility (CTTTPI)" in the CONTROL-T Section of the IOA Security Guide.)

Safe Tape Initialization - Function INITT

Utility CTTTPI protects active volumes from initialization by checking the specified volume serial number and the mounted volume before creating tape labels. Volumes are protected regardless of whether or not they appear in the Media Database (MDB).

To initialize tapes, utility CTTTPI normally invokes IBM's tape initialization utility IEHINITT. A comparable utility can be substituted via parameter specification. Utility CTTTPI initializes magnetic media (tape-reels and cartridges) with standard labels (SL) or with ANSI standard labels (AL).

The initialization function is generally invoked as a started task via the command:

```
S CTTTPI,V=volser,[U=unitname|devnum,]F=INIT
```

This command invokes the utility with the single control statement:

```
INITT SER=volser,NUMTAPE=999
```

When invoked in this manner, the utility first checks that the specified target volume serial number is either not in the Media Database or is in the Media Database with scratch status.

If the target volume serial number passes the first check, the utility requests that the operator mount the physical volume. Once the volume is mounted, the utility reads it to verify that it is not an active volume in the Media Database.

If both the specified target volume and the physical mounted volume pass the checks, the desired label is written on the mounted volume.

If the specified volser is numeric, the utility requests successive volumes in ascending order until stopped or canceled by the operator.

The utility issues messages (e.g., CTT289A) and sometimes prompts the operator for a reply or decision.

If CONTROL-O or CONTROL-O/PC is installed at your site, responses to the messages can usually be automated.

Depending on specified parameters, the Media Database can be updated by the utility.

Media Information Mapping - Function TAPEMAP

Utility CTTTPI can produce a detailed report about the logical and physical content of a volume including logical files, labels, tape marks, and unreadable areas (I/O errors).

This function is especially useful when recovering files and/or data which have been partially-overwritten on magnetic media (e.g., a longer file partially overwritten by a shorter file, or multi-file volumes in which the first few files have been overwritten).

The tape mapping function is generally invoked as a started task via the command:

```
S CTTTPI, [V=volser,] [U=unitname|devnum,] F=MAP
```

This command invokes utility CTTTPI with the single control statement:

```
TAPEMAP SER=volser
```

The TAPEMAP function produces a report which contains a single line for each of the following elements:

Element	Description
VOL1	Volume label.
HDR1	First part (and sometimes the only part) of a header label which describes a standard label dataset.
HDR2	Second part of a dataset label.
DATA BLOCKS	Actual data portion of a file on the media.
EOF1	First part of a dataset trailer label.
EOF2	Second part of a dataset trailer label.
EOV1	First part of an End-Of-Volume tape label. This element appears when the preceding standard label dataset continues on another volume.
EOV2	Second part of an End-Of-Volume tape label.
Tape Mark	Physical indication on the media which separates one physical file from another and separates standard labels from data. Two consecutive tape marks commonly determine the logical end of the recorded area on the media.
I/O Error	Area which could not be read by CTTTPI.

The TAPEMAP function reports information extracted from the above elements such as the dataset name (i.e., the full dataset name if it does not exceed 17 characters, or the last 17 characters of a longer dataset name), creation date, creating job name, block size, physical and logical file sequence numbers.

Physical file sequence numbers are counted by tape marks. Each tape mark on the tape indicates an end-of-file. Logical file sequence numbers are extracted from HDR1. If there is no HDR1 on the tape (e.g., an NL tape), the logical file sequence number is blank.

If a map of the complete media is requested (parameter STOP=N, described later), datasets written beyond the present logical end of the recorded area can be examined.

Safe Tape Erasure - Function TAPERAS

Utility CTTTPI protects active volumes from being erased by comparing the specified volume serial number and the volser of the mounted volume, before erasing the data on the tape. Volumes are protected regardless of whether or not they appear in the Media Database.

Utility CTTTPI uses an MVS function ('Data Security Erase' EXCP), to erase tapes. This function writes a random pattern of data on the specified tape.

The erase function is generally invoked as a started task via the following command:

```
S CTTTPI,V=volser,[U=unitname|devnum],F=ERAS
```

where:

volser	Volume serial number of the tape to be erased.
unitname	Type of the unit in which the specified tape is loaded.
devnum	Number of the device in which the specified tape is loaded.

This command invokes the utility with the following single control statement:

```
TAPERAS SER=volser,NUMTAPE=999
```

When invoked in this manner, the utility first checks that the specified target volume serial number is either not in the Media Database or is in the Media Database with scratch status.

If the target volume serial number passes the first check, the utility requests that the operator mount the physical volume. Once the volume is mounted, the utility reads it to verify that it is not an active volume in the Media Database.

The mounted volume is erased only if the volser of the mounted volume matches the volser specified in the JCL for the utility.

If the specified volser is numeric, the utility requests successive volumes in ascending order until stopped or canceled by the operator.

The utility issues messages (e.g., CTT289A) and sometimes prompts the operator for a reply or decision. If CONTROL-O or CONTROL-O/PC is installed at your site, responses to these messages can usually be automated.

Depending on specified parameters, the Media Database can be updated by the utility.

CTTTPI and I/O Error Recovery

Since utility CTTTPI thoroughly examines magnetic tapes in unknown condition, it may encounter more I/O errors than ordinary programs might encounter. In particular, CTTTPI encounters I/O errors when reading a new tape to which data has never been written. While CTTTPI handles most of these errors successfully, the user may observe pertinent error messages originated by MVS. Normally these messages can be ignored. If you have automation procedures driven by these messages, they should possibly be adjusted to exclude tape initialization and media swapping jobs.

To minimize the overhead of I/O error recovery, CTTTPI turns off DDR/SWAP capability in the allocated device for a short duration while it reads the label of a mounted tape to be initialized. If CTTTPI is canceled during that short period, the device may be left in a state where it is not a candidate for DDR/SWAP.

The operator can increase control over DDR/SWAP via MVS commands G OFF and G ON.

Note *In some cases the I/O error encountered by CTTTPI is due to incompatible recording formats of different device types. To avoid the destruction of media originally created by another device, care should be exercised in handling the mounted volume, in examining the I/O error messages on the console, and in specifying the CTTTPI protection parameters.*

In particular, utility CTTTPI recognizes volumes which have never been accessed by the occurrence of an I/O error when it reads the label. If a 3480 device attempts to read a cartridge written by a 3490 device, an I/O error message is issued by MVS indicating format incompatibility, and the cartridge is treated by CTTTPI as a media to which data has never been written. In some cases the cartridges can be overwritten by a 3480 device. If your site has both media types and you want enhanced protection against the accidental mounting of cartridges created by a 3490 device and their reinitialization by a 3480 device, specify PROTECT=PROMPT.

Activating the Utility

The utility can be activated as a job, started task (STC), or program called from within another program.

Parameters can be passed to the utility in either of the following ways:

- Via the EXEC statement.
- Via the SYSIN DD statement.

However, do not use both methods at the same time.

Passing Parameters via the EXEC Statement

When passing parameters via the EXEC statement, the statement format is:

```
// EXEC CTTTPI,V=volser,[U=unitname|devnum,]F=INIT|MAP|ERAS
```


Where:

- V Six-character volume serial number. Valid for INIT, MAP, or ERAS functions. Optional.
- F Function to be performed using default control statements. Valid functions are:
- | | |
|------|---|
| INIT | Initialize the tape (reel or cartridge). Create a volume label. |
| MAP | Map the tape: Provide information about the contents of a reel or cartridge. |
| ERAS | Erase the data on the tape. Write a random pattern from the beginning until the physical end-of-tape. |
- U Tape drive to be allocated for use by the utility. Valid values are a generic or esoteric unit name or a specific device number.

Passing Parameters via DD Statement SYSIN

When passing parameters via DD statement SYSIN, the syntax is:

```
// EXEC CTTTPI
//SYSIN DD *
TYPE RUN [MODE=NORMAL|SIMULATION]
        [, PROTECT=ALL|MDB|PROMPT|NOREAD]
        [, MDBUPDAT=Y|N]
        [, ADD=N|Y|E|P]
        [, PROG=labeling-program]
INITT SER=volser
        [, DDNAME=ddname]
        [, OWNER=owner]
        [, NUMBTAPE=number]
        [, LABTYPE=AL]
        [, ACCESS=char]
        [, ERASE=Y|N]
TAPEMAP SER=volser
        [, DDNAME=ddname]
        [, NUMERR=number]
        [, STOP=Y|N]
        [, MODE=S|E]
TAPERAS SER=volser,
        [, DDNAME=ddname]
        [, NUMBTAPE=number]
        [, ERASELBL=Y|N]
```

Input Parameters - SYSIN DD File

Note Throughout the documentation, the term "specified volser" refers to the volume serial number specified in parameter *SER=volser*. The term "mounted volume" refers to the physical volume (and its label, if one exists) which was mounted with the intent of writing a new volume label on it.

TYPERUN Provides general information affecting the remainder of the run. Optional. Valid only for function INITT.

Only one TYPERUN statement can be specified in each run of CTTTPI. When specified, it must be the first statement.

MODE Controls the operation of function INITT. Optional. Valid values are:

NORMAL The utility creates labels on the media and performs other requested functions. Default.

SIMULATION The utility only verifies that the specified target volume either does not appear in the Media Database or it appears in the Media Database in scratch status. The physical mounted volume, however, is not read and not initialized.

When this mode is used, mounting the target volume is not required.

PROTECT Controls the protection of the physical mounted volume. Optional. Valid only for function INITT. Valid values are:

ALL Default. Protects all mounted volumes, as follows:

- Mounted volumes which are active in the Media Database are rejected without a prompt to the operator.
- Mounted volumes which have never been written to are accepted without a prompt to the operator.

Note Media created on a 3490 may be treated as empty by CTTTPI. See "CTTTPI and I/O Error Recovery" above for more information.

- Mounted SL volumes which are not in the Media Database result in a prompt to the operator.
- NL volumes which contain data result in a prompt to the operator.
- A mounted SL scratch volume whose volume serial number is the same as the specified target volser is accepted without a prompt to the operator (reinitializing the existing volume).
- A mounted SL volume whose volume serial number exists in the Media Database in scratch status but whose volume serial number is not the same as the specified target volser results in a prompt to the operator.

MDB Protects mounted volumes only if they are in the Media Database. The mounted volume is protected in the same manner as in PROTECT=ALL, except that SL volumes which are not in the Media Database are accepted without a prompt.

NOREAD Mounted volumes are not read or otherwise protected before writing a new label on them.

Note Verification of the specified target volume serial number is still performed subject to other parameters.

PROMPT Mounted volumes are read and the operator is requested to approve the initialization of the mounted volume.

MDBUPDAT Controls updates to the Media Database record of the specified target volume. Optional. Valid for function INITT only. Cannot be used in SIMULATION mode. Valid values are:

Y (Yes) Default. If the volume is already in the Media Database, the information in its record is reset, including volume statistics and information regarding scratch datasets which have resided on the volume.

- The value of the last job to access the volume is changed to *CTTTPI*.
- If the volume is not in the Media Database and ADD=Y is specified, the volume record is added to the Media Database.

N (No) No changes are made to the Media Database as a result of the initialization process.

If MDBUPDAT=N is specified, parameter ADD is ignored.

ADD Controls the handling of specified volume serial numbers which are not in the Media Database. Optional. Valid for function INITT only. Cannot be used in SIMULATION mode. Valid values are:

N (No) A volume record is not added to the Media Database. Default.

Y (Yes) A volume record is added to the Media Database.

E (External) A volume record is added to the Media Database and marked External.

P (Prompt) The operator is prompted to decide whether or not to add a record to the Media Database.

PROG Labeling program name. Allows users to replace IBM's IEHINITT with another program which physically writes labels on volumes, or allows program IEHINITT to be used under a different name. The program name can have a maximum of 8 alphanumeric characters. The default is IEHINITT. Optional. Valid for function INITT only.

INITT Initialization function. Controls initialization (or initialization simulation) of magnetic media volumes. Optional. If this control statement is used, at least one optional subparameter must be specified:

SER Volume serial number to be initialized (up to 6 alphanumeric characters). Mandatory.

This parameter is identical to parameter SER= in IBM utility IEHINITT.

Note A volume serial number specified in // EXEC CTTTPI,V=volser or passed to the program as PARM='CTTTPIV=volser' overrides this value.

- NUMBTAPE** Maximum number of tapes to be initialized. Optional.
- After the first specified volume is processed, the numeric suffix of the volume serial number (if one exists) is repeatedly incremented by one to generate additional serial numbers, until the number of tapes initialized equals the value of NUMBTAPE or until a volume is initialized with a volser whose suffix is all 9's. NUMBTAPE must be an integer between 1 and 9999. The default is 1.
- OWNER** Owner identification to be recorded on the volume label. Maximum: 10 alphanumeric characters. (This parameter is identical to parameter OWNER= in IBM utility IEHINITT.) Optional.
- LABTYPE** Type of label to create. Optional. (This parameter is identical to parameter LABTYPE= in IBM utility IEHINITT.) If this parameter is omitted, an IBM standard label is created. Otherwise, the following value is valid:
- AL Create an ANSI standard label.
- ACCESS** Protection character for ANSI standard label. (This parameter is identical to parameter ACCESS= in IBM utility IEHINITT.) Optional.
- DDNAME** DD name for the tape drive to be used by CTTTPI. The ddname must start with the characters CTTTPI. The default value is CTTTPITP. Optional.
- MEDIA** Volume media name to be recorded in the Media Database for volumes being added to the Media Database as part of the initialization process. The media must be defined in member CTTPARM. If no value is specified, the media name is derived from the device type of the assigned tape drive (e.g., MEDIA=3480). Optional.
- VENDOR** Media vendor name to be recorded in the Media Database for volumes being added to the Media Database as part of the initialization process. Optional.
- ERASE** Indicates whether or not to erase all data on the tape from after the label until the physical end-of-tape.
- Y (Yes) Erase the data on the tape.
- N (No) Do not erase the data on the tape.

Note Since the ERAS function physically writes a random pattern of data until the physical end of the tape, a run of utility CTTTPI in which ERASE=Y is specified can be a time-consuming process.

TAPERAS

Erase function. Controls erase (or erase simulation) of magnetic media volumes. Optional. The following subparameters are available for this function:

SER Volume serial number to be erased (up to 6 alphanumeric characters). Mandatory.

Note A volume serial number can also be specified as follows:

// EXEC CTTTPI,V=volser

or it can be passed to the program as follows:

PARM='CTTTPIV=volser'

which overrides the volser in the EXEC statement. Both these statements override the value specified via parameter SER.

NUMBTAPE Maximum number of tapes to be erased. Optional.

After the first specified volume is processed, the numeric suffix of the volume serial number (if one exists) is repeatedly incremented by one to generate additional serial numbers, until the number of tapes erased equals the value of NUMBTAPE or until a volume is erased with a volser whose suffix is all 9's. NUMBTAPE must be an integer between 1 and 9999. The default is 1.

DDNAME DD name for the tape drive to be used by utility CTTTPI. The ddname must start with the characters CTTTPI. The default value is CTTTPITP. Optional.

ERASELBL Indicates whether or not to erase the label of the volume. Optional.

Y (Yes) ERASE the label.

N (No) Keep the label. Default.

TAPEMAP

Tape mapping function. Optional. If this parameter is used, at least one optional subparameter must be specified:

SER Volume serial number to appear in the mount message. This volume serial number does not have to match the volume serial number recorded on the media. Optional.

DDNAME Name of the DD statement identifying the tape drive to be used by CTTTPI. The ddname must start with CTTTPI. The default value is CTTTPITP. Optional.

NUMERR Number of I/O errors (between 1 and 999) which will cause utility CTTTPI to terminate. Default is 1. Optional.

STOP Controls termination of the TAPEMAP function. Optional.

Y (Yes) Default. The TAPEMAP function ends when one of the following is encountered:

- Logical End-of-Volume as indicated by:
 - Two consecutive tape marks, or
 - End-of-Volume label followed by a tape mark.
- The maximum number of I/O errors permitted (specified in parameter NUMERR=) has occurred.

N (No) Function TAPEMAP ends only when canceled by the operator or when it encounters an error it cannot ignore.

***Note** Processing a volume with STOP=N may cause unexpected operational problems, such as a tape running off a reel, or unusual I/O errors.*

It is recommended that STOP=N be used only in a test environment where unexpected problems will not disrupt production.

MODE Controls both the extent of the report produced by the TAPEMAP function, and the way in which this function reads the tape.

E (Extended) Produces a report on all datasets that exist on the volume including datasets that exist after the logical end of data (see value N for subparameter STOP above). When operating in this mode, utility CTTTPI physically reads every byte on the volume. This mode is therefore time-consuming. Default.

S (Standard) Produces a report on all datasets that exist on the volume up to the logical end of data. The utility scans the datasets on the volume, it does not physically read the data. This mode is therefore quicker than Extended mode.

When MODE=S is specified, subparameter STOP is ignored and field LARGEST BLOCK produced by utility CTTTPI is set to zero.

Invoking CTTTPI From Another Program

When calling CTTTPI as a program from within another program:

- Input parameters for CTTTPI should be passed only via DD statement SYSIN. Do not specify input parameters in the JCL EXEC statement.
- Linkage and parameter-passing conventions for utility CTTTPI are the same as for IBM utilities (see the IBM Utilities Manual).
- Utility CTTTPI supports substitution of DD names for its SYSIN and SYSPRINT statements via the same parameters used by IBM utilities (see the IBM Utilities Manual).

When constructing jobs which invoke CTTTPI, copy the DD statements from the CTTTPI JCL procedure as follows:

SYSIN	Input Stream.
SYSPRINT	Output stream for CTTTPI and for IEHINITT (or specified substitute). This DD statement must point to a SYSOUT file or the results can be unpredictable.
DAMDB	Media Database Data component.
DAMDI	Media Database Index component.
DALOG	IOA Log file.
DATRC	CONTROL-T Trace file.
CTTTPITP	Tape drive for tape initialization.
CTTTPICT	Intermediate file for transferring control statements to IEHINITT. The name of this DD statement cannot be overridden in the utility invocation.

If program CTTTPI is called from within another program:

- CTTTPI passes control statements to IEHINITT (or specified substitute) via DD statement CTTTPICT.
- PARM= parameters can be passed to IEHINITT (or specified substitute) by passing the parameters to CTTTPI.
- One PARM= parameter, CTTTPIV=volser, is utilized only by CTTTPI and is not passed to IEHINITT (or specified substitute).

If specified, the volume serial number must be the first PARM= parameter (i.e., it must precede any parameters passed to IEHINITT) and must be exactly 6 characters in length (e.g., PARM='CTTTPIV=VOL001,LINECNT=60'). This volume serial number will replace the volume serial number in the first occurrence of parameter SER= in an INITT or TAPEMAP statement in CTTTPI's input stream.

Return Codes

Upon completion, CTTTPI returns one of the following return codes (in register 15):

0	Successful completion
4	Operator requested termination by typing CANCEL in response to a program prompt
8	The specified target volume is a non-scratch volume in the Media Database
12	The Media Database could not be updated MDBUPDAT=Y was specified, but the specified target volume is not in the Media Database and the volume could not be added (i.e., either ADD=N was specified or the operator responded negatively to the ADD prompt)
16	Syntax error in control statements or parameters
20	IEHINITT (or the specified substitute) failed or was stopped by the operator
24	Error while opening the IOA Log file
28	Error while loading IOA modules or tables
32	Unable to open DD file CTTTPIxx for accessing a tape drive
36	Unable to open DD file SYSPRINT or its replacement
40	Unable to open DD file SYSIN or its replacement
44	Error while accessing the Media Database

Example

Below is a sample of TAPEMAP output with STOP=N.

```

15.33.04 CTF704I CONTROL-T UTILITY CTTTPI  STARTED
15.33.04 CTF700I  TAPEMAP SER=CTTMAP,STOP=N
VOL1          V=V00001
HDR1          DSN=...NOPOOL.F.G0028V00  FIRST=V00100  VOLSEQ=0002
FILESEQ=0002  CREDIT=94346
HDR2          CREAT-JOB=N77MUL01    BLKSZ=00080  LRECL=00080
RECFM=F
TAPE-MARK  -----
DATA BLOCKS: COUNT=00000005  LARGEST-BLKSZ=00080  LOGICAL-FILE
-SEQ=0002  PHYSICAL-FILE-SEQ=00002
TAPE-MARK  -----
EOF1          DSN=...NOPOOL.F.G0028V00
EOF2
TAPE-MARK  -----
TAPE-MARK  -----  **** LOGICAL END OF RECORDED AREA ****
IO-ERROR    *****
DATA BLOCKS: COUNT=00000051  LARGEST-BLKSZ=03120  LOGICAL-FILE
-SEQ=PHYSICAL-FILE-SEQ=00005
TAPE-MARK  -----
EOF1          DSN=NNN.TSC1
EOF2
TAPE-MARK  -----
TAPE-MARK  -----
. . .

```

CTTTRB - Synchronize Trace File and Media Database Backup

This utility marks records in the Trace file that are no longer needed, so that they can be overwritten. This utility should be run immediately before and after each backup of the Media Database and the Trace file.

The Trace file is used primarily to trace actions performed on the Media Database. When the Trace file becomes full, all CONTROL-T functioning is halted and an appropriate message is issued. This situation can be avoided by running utility CTTTRB on a regular basis. This utility is called by the CONTROL-T New Day procedure before and after the daily backup of the Media Database and the Trace file.

Activating the Utility

```
// EXEC CTTTRB,PARM='BKP=START|END'
```

Parameters

Utility parameters are supplied via JCL PARM:

BKP	Indicates if the Backup procedure is beginning or ending.
START	Backup procedure of Media Database has started.
END	Backup procedure of Media Database has ended.

The steps for implementing utility CTTTRB are as follows:

1. Run the utility before backup of the Media Database and the Trace file. In this run BKP=START must be specified. This run of the utility marks the most recently filled record in the Trace file.
2. Perform the necessary backups. Any local site backup procedures can be used to perform this task.
3. After the backup has been successfully completed, run utility CTTTRB with BKP=END specified. This run marks all Trace file records before the record marked in the previous run of the utility (Step 1 above).

Return Codes

0	Operation performed successfully
8	DCB open error occurred, TCT load failed, or PARM error
12	I/O error on Trace file occurred

Example

```
// EXEC CTTTRB,PARM='BKP=START'
// EXEC PGM=IEBGENER
...
// EXEC CTTTRB,PARM='BKP=END'
```

CTTUDB - Update the Media Database From Transaction File

Utility CTTUDB modifies the Media Database (MDB) according to a transaction file. The transaction file can be created either by utility CTTIDB or by manually editing a sequential file. For each transaction record, the utility verifies the data being changed, and issues a message specifying the type and the content of the changed field before and after the change.

Activating the Utility

Note The **SYSIN** statement and the **MTU** parameter are optional.

```
// EXEC CTTUDB,MTU='transaction-file'
//SYSIN DD *
TYPERUN MODE={NORMAL|SIMULATION}
```

Parameters

The following parameter is specified in the EXEC statement:

MTU Name of the transaction file to be used by utility CTTUDB. Optional; if not specified, an installation-defined default is used.

The following parameters are supplied via DD statement SYSIN:

TYPERUN	Statement type.
MODE	Operation mode. Mandatory
NORMAL	Normal operation is performed.
SIMULATION	Simulation is performed. No actual changes are made to the Media Database.

Return Codes

0	Operation performed successfully.
4	Verification of one or more data items in the Media Database failed - Execution continues
8	Invalid input from transaction file - Execution continues
12	Invalid control parameter given - Execution terminates
16	Error opening DCB - Execution terminates
Greater than 16	Severe error - the error is accompanied by an appropriate message indicating the problem - Execution terminates

CTTVTM - Perform Vault Management

This utility performs vault management functions. This utility should be run on a regular basis via job CTTDAY.

The utility checks each volume record's vault specifications in the Media Database and determines which volumes should be moved to a new location according to their vault patterns. The current volume location in the volume record is updated.

This utility can produce a distribution report at the end of processing, which lists all volumes which should be moved to a new location or assigned a new slot number.

Note Inventory report *INVENT* is no longer produced by utility *CTTVTM*. Instead, this report can be produced as part of the *GENERAL* report by utility *CTTRPT*. For more information see the description of utility *CTTRPT* in Section 5 of the *CONTROL-T User Manual*.

Since the utility handles a large number of Media Database records, a CANCEL operation, job abend or system crash can lead to logical inconsistencies in the database. After any of these situations, the next execution of the utility will automatically operate in Restart mode. In Restart mode, the utility will complete execution of the failed process, thereby eliminating the logical errors from the Media Database. The utility does not move any additional volumes during Restart execution.

Certain installation parameters (specified in member CTTPARM) determine how utility CTTVTM operates. These parameters are described below.

VLBYDS1 Specifies which dataset may determine how the vaulting pattern of a volume is chosen. Valid values are:

Y (Yes)	Only the first dataset of the volume determines the vaulting pattern for the volume. This improves performance of CTTVTM (and is compatible with other Tape Management Systems vaulting methods). Default.
N (No)	Any dataset can determine the vaulting pattern for the volume.

CYCLECNT Defines how to handle cyclic datasets.

For additional information refer to "Retention and Vault Parameters" in the IOA Installation Guide.

Activating the Utility

```
// EXEC      CTTVTM
//SYSIN     DD *
  TYPERUN  MODE=NORMAL|SIMULATION[,DATE=date][,TRACE=YES|NO][BOXLIM=num]
  TYPEVLT  MODE=REGULAR|RECALC|CONVERT|SLOTBLD|BOXBLD[,RESLOT=YES|NO]
  <INCLUDE/EXCLUDE STATEMENTS>
  REPORT   NAME=DISTRIB,...
  FIELDS...
  SORTBY...
  ATTR    ...
  BREAK   ...
//
```

Parameters

Utility parameters are supplied via DD statement SYSIN:

TYPERRUN	Operation type.
MODE	Operation mode. Mandatory.
	NORMAL Normal operation is performed.
	SIMULATION Simulation is performed. No actual changes are made to the Media Database
DATE	Operation date in 6- or 8-character format according to site standard with separating slashes. (For example, in USA standard the date can be specified mm/dd/yy or mm/dd/yyyy.) Optional. The utility assumes that the specified date is the current date, and performs vaulting accordingly. This parameter can be used together with MODE=SIMULATE (described above) to simulate vaulting to be done on a specific date in the future.
TRACE	Specifies whether or not the Trace file will be used during CTTVTM run. Optional.
	YES The Trace file will be used. Default.
	NO The Trace file will not be used. If NO is specified, recovery of the lost updates to the Media Database will be impossible. TRACE=NO is specified for the run of this utility during conversion from another product, but is not otherwise recommended.
DBGLEVEL	Debug level. A numeric value from 1 to 255 can be specified. Optional.
RESTART	Specifies whether or not to enable Automatic Restart mode when the utility is run after an abend or cancellation of the previous run of the utility. Valid values are:
	YES Enable Automatic Restart mode. Default.
	NO Disable Automatic Restart mode. It is highly recommended not to specify RESTART=NO, as this can result in integrity errors in the Media Database and loss of important information.
BOXLIM	Minimum number of volumes a box can contain. If the number of volumes is below the minimum, the volumes are placed in a slot instead of a box. Default is 1.

TYPEVLT	Vault management type. Optional if parameter REPORT is specified. If TYPEVLT is not specified, vault processing is not performed but reports are produced.	
MODE	Vault management mode.	
	REGULAR	Vault management is performed according to the vault patterns currently specified for the volumes (in Media Database volume records).
	RECALC	Rule definitions are reloaded and the vault pattern of each volume is recalculated.
	CONVERT	This parameter is used following conversion from another tape management system. Rule definitions are reloaded, the vault pattern for each volume is calculated, and the current location of each volume is reserved within the vault pattern.
	SLOTBLD	The slot map is built in the Media Database. This mode cannot be specified if RESLOT=YES.
	BOXBLD	Box records are built in the Media Database from the vault definition and volume records. This mode cannot be specified if RESLOT=YES.
	RESLOT	Specifies whether or not to reassign slot numbers to vaulted volumes. Optional.
	YES	New slot numbers will be assigned.
	NO	Existing slot numbers will be preserved. Default.

INCLUDE/
EXCLUDE

The scope of the utility can be limited by one or more INCLUDE or EXCLUDE statements. INCLUDE/EXCLUDE statements are specified immediately after the TYPEVLT parameter and are applied to vault processing. The subparameters which can be specified in INCLUDE/EXCLUDE statements are listed in Tables 1 and 2 in "Logical Field Names for the CONTROL-T Repository" in Appendix D.

Note As of CONTROL-T Release 2.1.0, INCLUDE/ EXCLUDE statements cannot be specified after SORTBY parameters.

For additional information on INCLUDE and EXCLUDE statements, see "Record Selection Logic (INCLUDE/EXCLUDE Statements)" earlier in this section.

Note *With few exceptions, the parameters and options of the REPORT, FIELDS, SORTBY, ATTR and BREAK statements are identical to those found in utility CTTRPT. For a complete description of these parameters and options, refer to utility CTTRPT in Section 5 of the CONTROL-T User Manual. Differences between utility CTTVTM and utility CTTRPT, in those parameters and options, are described below.*

REPORT

Activates report utility. Optional if parameter TYPEVLT is specified.

NAME Report name. Mandatory. Valid value is:

DISTRIB Distribution Report listing volumes which need to change location as a result of this utility run.

Notes *Inventory report INVENT is no longer produced by utility CTTVTM. Instead, this report can be produced as part of the GENERAL report by utility CTTRPT. A sample Inventory report can be found in member REPINV in the CONTROL-T JCL library.*

See the REPORT statement in utility CTTRPT (earlier in this section of the Administrator Guide) for a description of the other parameters and options of the REPORT statement.

FIELDS

Fields to print in the report. The fields are printed in the order they are specified.

The following fields can be printed on the Distribution report:

MEDIA	Media type
POOL	Pool name
VOLSTAT	Status(es) of the volume
VOLSER	Volume serial number
SL-NAME	Standard label name of the volume
VOLCNT	Number of volumes in a multi-volume chain
FIRSTVOL	First volser in a multi-volume chain
FROMLOC	"From" vault location of the volume
TOLOC	"To" vault location of the volume
FROMSLOT	"From" slot number in the vault

TOSLOT	"To" slot number in the vault
NEXTLOC	Next location in the vault pattern
NEXTLCDT	Date the volume should be moved to the next location in the vault pattern
BOXID	Name of the box in which the volume resides.
EXPDT	Expiration date
LBLTYP	Label type
DSNAME	Dataset name
VOLSEQ	Volume sequence number
LOCSEQ	Sequence number of the location in the vault pattern.
CREJBN	Creating job's name
CREDIT	Creation date
CRETM	Creation time

Note See the *FIELDS* statement in utility *CTTRPT* (earlier in this section of the *IOA Administrator Guide*) for a description of the options of the *FIELDS* statement.

SORTBY Sort order of the output fields (see *FIELDS* above), with the options that can be applied to each field. Optional.

Note See the *SORTBY* statement in utility *CTTRPT* (earlier in this section) for a description of the parameters and options of the *SORTBY* statement.

ATTR Override of default attributes defined for a field (see *FIELDS* above). Optional.

Note See the *ATTR* statement in utility *CTTRPT* (earlier in this section) for a description of the parameters of the *ATTR* statement.

BREAK Attributes of the summary header line(s) printed when a field break occurs. Optional.

Note See the *BREAK* statement in utility *CTTRPT* (earlier in this section) for a description of the parameters of the *BREAK* statement.

Return Codes

0	Operation performed successfully
4	Missing input statement(s)
8	Missing DD statement(s), or open error occurred
12	Control card error
16	Load of CONTROL-T programs failed
20	Sort error occurred
24	Media Database input/output error occurred
28	Internal error occurred
32	Error in report printing utility occurred
other	Operation is terminated

Example 1

Vault management is performed according to the current vault pattern specifications in the Media Database. Report DISTRIB is printed, listing the MEDIA, VOLSER, VOLSEQ, FROMLOC and TOLOC fields.

```
//      EXEC CTTVTM
//SYSIN DD *
        TYPERUN  MODE=NORMAL
        TYPEVLT  MODE=REGULAR
        REPORT   NAME=DISTRIB
        FIELDS  MEDIA, VOLSER, VOLSEQ, FROMLOC, TOLOC
        SORTBY  FROMLOC/B, TOLOC, VOLSER
```

```
//
```

The report is sorted according to the location from which each volume is being moved (FROMLOC). The report is then sorted according to the destination (TOLOC) and the volser name (VOLSER). A page break is inserted after each FROMLOC location listed in the report (i.e., each group of volumes from a specific location will start on a new page).

Example 2

The location of all volumes whose volume serial numbers start with DIV1 is modified. The volumes are assigned new slot numbers. Rule definitions are reloaded to recalculate the vaulting patterns.

```
// EXEC      CTTVTM
//SYSIN DD*
        TYPERUN  MODE=NORMAL
        TYPEVLT  MODE=RECALC, RESLOT=YES
        INCLUDE  VOLSER=DIV1*
        REPORT   NAME=DISTRIB
        FIELDS  MEDIA, VOLSER, POOL, FROMLOC, TOLOC
        SORTBY  FROMLOC/B, VOLSER, POOL
```

```
//
```

TOSLOT	"To" slot number in the vault
NEXTLOC	Next location in the vault pattern
NEXTLCDT	Date the volume should be moved to the next location in the vault pattern
BOXID	Name of the box in which the volume resides.
EXPDT	Expiration date
LBLTYP	Label type
DSNAME	Dataset name
VOLSEQ	Volume sequence number
LOCSEQ	Sequence number of the location in the vault pattern.
CREJBN	Creating job's name
CREDT	Creation date
CRETM	Creation time

Note See the *FIELDS* statement in utility *CTTRPT* (earlier in this section of the *IOA Administrator Guide*) for a description of the options of the *FIELDS* statement.

SORTBY Sort order of the output fields (see *FIELDS* above), with the options that can be applied to each field. Optional.

Note See the *SORTBY* statement in utility *CTTRPT* (earlier in this section) for a description of the parameters and options of the *SORTBY* statement.

ATTR Override of default attributes defined for a field (see *FIELDS* above). Optional.

Note See the *ATTR* statement in utility *CTTRPT* (earlier in this section) for a description of the parameters of the *ATTR* statement.

BREAK Attributes of the summary header line(s) printed when a field break occurs. Optional.

Note See the *BREAK* statement in utility *CTTRPT* (earlier in this section) for a description of the parameters of the *BREAK* statement.

Return Codes

0	Operation performed successfully
4	Missing input statement(s)
8	Missing DD statement(s), or open error occurred
12	Control card error
16	Load of CONTROL-T programs failed
20	Sort error occurred
24	Media Database input/output error occurred
28	Internal error occurred
32	Error in report printing utility occurred
other	Operation is terminated

Example 1

Vault management is performed according to the current vault pattern specifications in the Media Database. Report DISTRIB is printed, listing the MEDIA, VOLSER, VOLSEQ, FROMLOC and TOLOC fields.

```
//      EXEC CTTVTM
//SYSIN DD *
        TYPERUN MODE=NORMAL
        TYPEVLT MODE=REGULAR
        REPORT NAME=DISTRIB
        FIELDS MEDIA, VOLSER, VOLSEQ, FROMLOC, TOLOC
        SORTBY FROMLOC/B, TOLOC, VOLSER
//
```

The report is sorted according to the location from which each volume is being moved (FROMLOC). The report is then sorted according to the destination (TOLOC) and the volsr name (VOLSER). A page break is inserted after each FROMLOC location listed in the report (i.e., each group of volumes from a specific location will start on a new page).

Example 2

The location of all volumes whose volume serial numbers start with DIV1 is modified. The volumes are assigned new slot numbers. Rule definitions are reloaded to recalculate the vaulting patterns.

```
// EXEC      CTTVTM
//SYSIN DD*
        TYPERUN MODE=NORMAL
        TYPEVLT MODE=RECALC, RESLOT=YES
        INCLUDE VOLSER=DIV1*
        REPORT NAME=DISTRIB
        FIELDS MEDIA, VOLSER, POOL, FROMLOC, TOLOC
        SORTBY FROMLOC/B, VOLSER, POOL
//
```

Section 10: Exits

General	10-1
IOA Exits	10-2
CONTROL-M Exits	10-5
CONTROL-R Exit	10-8
CONTROL-D/V Exits	10-9
Replacing CONTROL-D User Exits	10-14
Tailoring CONTROL-D Banner Exit	10-14
Banner Pages.....	10-15
CONTROL-B Exits	10-21
CONTROL-T Exits	10-22
CONTROL-O Exits	10-23



General

Various user exits are available which can be used to modify the operation of IOA products to accommodate needs. The exits reside in the IOA SECUDATA library, and start with either IOAXnnn or CTxXnnn. These exits are installed using SMP/E USERMOD. To install an exit, locate the exit's associated USERMOD member in the IOA SECUDATA library (members starting with UMxXnnn, where x is the CONTROL product, and Xnnn refers to the exit number).

***Note** Special exits may exist in some cases. The exits can be identified by the additional character added to the member name (e.g., IOAX006D). For this purpose, a special USERMOD is available for each of these exits. For example, for user exit CTDX015V, the corresponding USERMOD to be used is UMDX015V.*

A full explanation of how to use the exits is available in members starting with IOAIXnnn or DOCIXnnn. For example, the explanation for exit IOAX006 can be found in member DOCIX006 in the SECUDATA library.

Before submitting the USERMOD jobs, follow the steps.

1. Verify that a backup of the LOAD library exists.
2. Make all necessary updates to the USERMOD:
 - A. Update the PRE operand of the ++VER MCS of the usermod to include the exit SRC entry RMID and UMIDs. If the exit has no UMIDs and the RMID is equal to the FMID, the PRE operand should be omitted.
 - B. Update the JCLIN (linkedit input cards) if required.
 - C. Update the source exit name in the SMP/D ++SRC statement if necessary.
 - D. Update the library and member names of the source exit in the IOA SECUDATA library on the unnamed DD statement following the ++SRC statement.
3. Submit the job. It will perform SMP/E RECEIVE and APPLY commands, selecting the SMP/E usermod just created. Do not run ACCEPT on this usermod.

IOA Exits

The following IOA exits are available:

IOAX006: This exit controls the use of the IOA Online facility.

IOAX006 and IOAX009 are twin exits. IOAX006 is invoked by the Online facility before the entry panel is displayed. IOAX009 is invoked only when signing on to the Online facility through the online monitor (i.e., when signing on through the online monitor, exit IOAX009 is invoked first, followed by exit IOAX006). When the Online facility is used without the online monitor, only exit 6 is invoked). These exits can be used to display a sign-on window for users to enter their user ID and password.

These sign-on modules determine and build the user's identity for all subsequent actions. They both have a similar structure of parameters, return codes, and functionality. However, they work in different address spaces. The home address space is the primary address space requesting and receiving services from the online monitor address space using the cross-memory facilities. Exit IOAX006 is invoked in the online monitor address space (when signing on through the online monitor), while exit IOAX009 is invoked in the home address space that is the VTAM monitor, CICS, IMS, CA-ROSCOE, etc. The control block that represents the user's identity accompanies the user during the entire session with the IOA Online services facility.

In the online monitor environment, the ACEE control block is stored in the user's TCB (Task Control Block) and the OCT (Online Control Table). MVS recognizes the ACEE as a standard control block to be used for authorization checks, so that task level security feature is achieved. If the ACEE is not stored in the TCB, either because module IOASE06 is not implemented or because the security package does not build an ACEE (e.g., ACF2 in native mode), then all authorization checks for file access are performed using the identify of the online monitor address space. All authorization checks are performed using the correct user's ACEE. If the ACEE is not built, it is quite likely that the security interface will not perform the authorization checks correctly.

MVS checks authorization for actions such as opening files by first checking if there is an ACEE in the current TCB. If it is found, authorization checks are performed using the TCB's ACEE. If it is not found, MVS continues to search for the appropriate TCB until the ACEE associated with the address space is found.

In a non-secured environment where security is not implemented, exits IOAX006 and IOAX009 can set the OCTUSER parameter, which is the reference parameter for all programs used as the identity of the current working user ID.

When using the Online facility under CA-ROSCOE, the IOA security interface receives the CA-ROSCOE started task procedure and not the user's own user ID. This means that the user needs to sign on twice, once under ROSCOE, and a second time under IOA. To avoid this process, a special routine is provided (in the IOA SECUDATA library), called IOARROT, which retrieves the user's private ID from the corresponding CA-ROSCOE control block and places it in IOA, so that it can be used for additional security authorizations. The IOARROT routine should be called from within the IOA Online environment, prior to the standard IOA security modules. This call should ideally be placed in one of the IOA user modules that are invoked before either IOASE06 or IOASE09 modules are called. The user modules that support CA-ROSCOE for the IOA Online facility communication are IOAX006T and IOAX009. These modules reside in the IOA SECUDATA library.

User exit IOAX006T should be called when all IOA functions are performed under native CA-ROSCOE address space.

See members DOCIX006 and DOCIX009 in the IOA SECUDATA library for more details on these exits.

Note When two environments run simultaneously, both user modules are placed in the IOA LOAD library. User exit IOAX006T is invoked in the online monitor address space as well, but cannot locate the required control blocks, as there are no CA-ROSCOE control blocks in the online monitor address space. The started session abends. It is recommended that these two modules be placed in the special IOA LOAD library which is concatenated to the STEPLIB in the CA-ROSCOE started task procedure before the IOA LOAD library.

IOAX007 This exit can be used to control update of the Conditions/Resources file. For further details, refer to the IOA Security Guide.

See member DOCIX007 in the IOA SECUDATA library for more details on this exit.

IOAX009 This exit controls entry to the IOA Online facility when working with the Online monitor (under CICS, VTAM, IMS/DC, TSO/ROSCOE cross memory options, COM-PLETE, and IDMS/DC).

User exit IOAX009T should be called when CA-ROSCOE communicates with IOA through cross-memory services. All IOA functions are performed in the separate address space, which is the IOA Online monitor address space (OMON1).

In a non-secured environment where security is not implemented, exits IOAX006 and IOAX009 can set the OCTUSER parameter, which is the reference parameter for all programs used as the identity of the current working user ID.

See member DOCIX009 in the IOA SECUDATA library and the explanation on exit IOAX006 earlier in this section.

Section 10: Exits

- IOAX012** This exit can be used to control or modify the operator commands issued by the CTMOPR, CTDOPR, or IOAOPR utilities, or CONTROL-O, or by the CONTROL-D/V CTDNDAY (New Day) procedure or the CONTROL-M CONTDAY (New Day) procedure.
- See member DOCIX012 in the IOA SECUDATA library for more details on this exit.
- IOAX028** IOA definition screen exit. This exit is invoked each time the user presses the <ENTER> or PF3 key to exit an IOA definition screen. This exit is now relevant for all IOA definition screens (screens 2, 8, R, M, TR, TV, TP, BM, BR, OR, and C).
- See member DOCIX028 in the IOA SECUDATA library for more details on this exit.
- IOAX029** IOA Online Sysout/Report Viewing exit. This exit is invoked for each sysout/report line which is to be displayed on the user terminal. For example, this exit allows translation of unprintable characters. It is activated as part of the Online Viewing facility. This exit also provides additional features (e.g., making special data invisible on the report).
- See member DOCIX029 in the IOA SECUDATA library for more details on this exit.
- IOAX031** IOA Log exit. This exit is invoked before a message is written to the IOA Log file. The exit may be used to document the message on other files (SMF, etc.), or to accumulate statistics for certain messages.
- See member DOCIX031 in the IOA SECUDATA library for more details on this exit.
- IOAX032** This exit is invoked in all IOA panels each time an IOA user attempts to perform an operation on a PDS library or member in a PDS library. This exit may deny operations such as accessing a library, listing its directory and browsing, editing or saving a member. The exit is invoked when a user requests to edit JCL members or documentation data through Enterprise Controlstation. The exit checks authorization, and either grants or denies the Edit (or Save) request prior to the performance of these operations by the MVS Gateway for Enterprise Controlstation (ECSGATE).
- IOAX034** Receives control for every message issued by the IOA Shout facility. It can modify the message text, change its destination, or suppress it. This exit can be activated by either the CONTROL-O monitor, or an IOA Functional monitor (used with CONTROL-T).
- See member DOCIX034 in the IOA SECUDATA library for more details on this exit.

- IOAX035** IOA Account Information Extraction exit. This exit is invoked by an IOA product whenever account field information is required. Exit IOAX035 can extract the requested information from different parts of the account field depending on site customization. The exit may be invoked for example by CDAM when writing directly to a CDAM file, by CONTROL-D when decollating from a spool, or by CONTROL-T when a job's account field is copied to the Media Database.
- See member DOCIX035 in the IOA SECUDATA library for more details on this exit.
- IOAX036** IOA Access Method (IOAAM) exit. This exit is invoked whenever program IOADBF is executed, both as an independent utility program and when called internally by IOA Access Method I/O routines. This exit can check for which function it is being called and either grant or deny the request.
- See member DOCIX036 in the IOA SECUDATA library for more information on this exit.
- IOAX037** IOA translation exit. This exit contains four 256-byte translation tables for the IOA online routines. This exit may be used to translate the screens to any language supported at the site in addition to upper casing, special characters sets, and other capabilities.
- See member DOCIX037 in the IOA SECUDATA library for more information on this exit.
- IOAX039** IOA Page Separating Exit. This exit is invoked during the creation of a CDAM file by a CONTROL-D decollating mission, or during direct writing to a CDAM file by a job. The exit can control page separation in a CDAM file being created.
- See member DOCIX039 in the IOA SECUDATA library for more details on this exit.

CONTROL-M Exits

The following CONTROL-M exits are available:

- CTMX001** This exit is invoked for every job order which should be placed in the Active Jobs file. The exit is usually used to modify the job production parameters. This exit has an associated security module CTMSE01. For further details, refer to the IOA Security Guide.
- See member DOCMX0001 in the IOA SECUDATA library for more details on this exit
- CTMX002** CONTROL-M submission exit. Every line which is submitted by CONTROL-M may be modified/deleted/replaced, etc. The exit can be used for adding /*PRIORITY 15 for every production job, etc. This exit has an associated security module CTMSE02. For further details, refer to the IOA Security Guide.
- See member DOCMX002 in the IOA SECUDATA library for more details on this exit.

Section 10: Exits

CTMX003 CONTROL-M sysout scan exit. After the job finishes executing, every line of the job's SYSDATA1 is passed to the exit. The main use of this exit is to detect user generated constants in the sysout (such as program DISPLAY messages), and to affect the execution results of the current job accordingly.

See member DOCMX003 in the IOA SECUDATA library for more details on this exit.

CTMX004 This exit allows the user to change the defaults of the CONTROL-M scheduling algorithm by assigning weights to quantitative resources. In this way, CONTROL-M can fine-tuned to achieve maximum throughput.

See member DOCMX004 in the IOA SECUDATA library for more details on this exit.

CTMX005 This exit is activated under utility CTMJSA which accumulates job statistical data from the IOA Log.

Exit CTMX005 is an integral part of the statistical data accumulation process. It is invoked by utility CTMJSA. The exit is used for the following reasons:

- By default the machine ID (one-character) is automatically derived from the SYSID (four-character) and from information contained in CTMPARM, before exit CTMX005 is invoked. Exit CTMX005 can override the machine ID.
- It enables the user to add to the job execution statistics, additional statistical data which can come from resource other than the IOA Log (e.g., performance monitors).

See member DOCMX005 in the IOA SECUDATA library for more details on this exit.

CTMX008 This exit is used to control access to the Active Jobs file. This exit has an associated security module CTMSE08. For further details, refer to the IOA Security Guide.

Exit CTMX008 and security module CTMSE08 are also invoked under the MVS Gateway for Enterprise Controlstation.

See member DOCMX008 in the IOA SECUDATA library for more details on this exit.

¹ SYSDATA is an IOA term used to designate the data in the following three job SYSOUT datasets:

1. The job log (console messages)
2. The expanded JCL
3. The system output messages

- CTMX010 This exit is used to control job submission via the Quick Submit facility. For example, it can force the use of CONTROL-M for all, or specified production job submissions under TSO.
- See member DOCMX010 in the IOA SECUDATA library for more details on this exit.
- CTMX011 This exit receives control for every message issued by the CONTROL-M Shout facility. It can modify the message text, change its destination, or suppress it. It may also be used for special purposes, such as an interface to InfoMan or to Electronic Mail Systems.
- See member DOCMX011 in the IOA SECUDATA library for more details on this exit.
- CTMX013 This exit receives control each time a line is displayed by the CONTROL-M Online facility in the Job Statistics screen (3.S). The exit may modify the line to be displayed.
- See member DOCMX013 in the IOA SECUDATA library for more details on this exit.
- CTMX014 This exit is invoked when a request is made to ISPF-edit a JCL member under screen 2 and screen 3 of the Online facility.
- See member DOCMX014 in the IOA SECUDATA library for more details on this exit.
- CTMX015 This exit is invoked when a job has finished executing, and has been assigned a CONTROL-M finishing status.
- See member DOCMX015 in the IOA SECUDATA library for more details on this exit.
- CTMX016 This exit is invoked when a job has been submitted and needs to be tracked. When the dual monitor facility is used, The exit can be used to direct tracking of the job to a specific monitor.
- See member DOCMX016 in the IOA SECUDATA library for more details on this exit.
- CTMX017 The exit receives control each time the Tape Pull list utility (CTMTAPUL) attempts to resolve volumes for each DD statement of the job.
- See member DOCMX017 in the IOA SECUDATA library for more details on this exit.
- CTMX018 This exit receives control at certain points during the process of determining the anticipated elapse time of a job.
- See member DOCMX018 in the IOA SECUDATA library for more details on this exit.

Section 10: Exits

CTMX019 This exit receives control during the CONTROL-M external writer initialization phase.

See member DOCMX019 in the IOA SECUDATA library for more details on this exit.

CMEM Exits

The CONTROL-M Event Manager (CMEM) can also use the following CONTROL-O exits:

CTOX001 Receives control under the Online interface whenever a rule ORDER/FORCE is performed.

See member DOCOX001 in the IOA SECUDATA library for more details on this exit.

CTOX002 Receives control for every rule before it is loaded to the active environment. This exit can be used to modify rule parameters prior to rule loading.

See member DOCOX002 in the IOA SECUDATA library for more details on this exit.

CONTROL-R Exit

The following CONTROL-R exit is available:

CTRX001 This exit can be activated as part of the CONTROLR step. It allows better control of the CONTROLR step execution, and can be used to interface with other products (e.g., checking non-standard datasets).

This exit can be used as an interface to a tape management system in order to extend CONTROL-R's processing.

When used as an interface with CONTROL-T, this exit receives dataset DELETE requests and changes the CONTROL-T controlled dataset status to Pending-Scratch. This enables expiration of the specified dataset by the next run of CONTROL-T Retention Management utility CTTRTM. Via customization of this exit, you can cause CONTROL-T controlled datasets to be scratched immediately (i.e., without running utility CTTRTM).

For more information see "CONTROL-R Interface to CONTROL-T" in Section 3 of this guide, and "CONTROL-R and Tape Management Systems" in the CONTROL-R User Manual.

See member DOCRX001 in the IOA SECUDATA library for more details on this exit.

CONTROL-D/V Exits

The following CONTROL-D/V exits are available:

- CTDX001** Receives control for every mission which should be placed on the Active Missions file. This exit is usually used to modify mission production parameters. See member DOCDX001 in the IOA SECUDATA library and review the IOA Security Guide for further details.
- CTDX002** Printer command exit. This exit receives control at the end of any chunk of lines which is sent by CONTROL-D to the spool. This exit can issue operator commands to set the printer for the coming chunk (Consult your IOA representative before using this exit.) See member DOCDX002 in the IOA SECUDATA library for more details about this exit.
- CTDX003** CONTROL-D banner exit. This exit receives control at the beginning and end of every print mission or the beginning of a user in the bundle or for every report in the bundle, or for every chunk in the bundle.
- This exit produces banner pages at any level and in any desired format. It can also, optionally, produce an index of the reports in a bundle.
- A detailed explanation of how to tailor and use the banner exit appears later in this section.
- See member DOCDX003 in the IOA SECUDATA library for more details about this exit.
- CTDX004** This exit controls access to the User Reports list files. When an online user specifies selection criteria for reports, he/she receives a list of reports which he/she is allowed to see. A user can perform Online viewing and other operations only for reports which are authorized by this exit. (The user will see only the decollated portion of any report which he/she is allowed to see.)
- When CONTROL-D is installed, a default dummy exit is active. To facilitate installation and product testing, this exit does not enforce any security standards. When no security standards are enforced, all of the users may see all reports under Option U. Once CONTROL-D becomes operational, and access to its Online facility is given to many users, it is highly recommended to use either the supplied security module CTDSE04, or sample user exit CTDX004A.
- Sample exit CTDX004A retrieves security definitions from the Recipient Tree. The administrator can identify a recipient (or several recipients) in the Recipient Tree with a TSO logon (or logon/CICS/VTAM/etc.) user ID. These authorizations enable users to see reports under Option U. The user IDs to be authorized should be filled in the AUTHORIZE field in the definition of each recipient which is authorized for them. For more information, refer to the Recipient Definition screen in Section 2 of the CONTROL-D User Manual. When a user is authorized in the Recipient Tree, it means that the TSO (or CICS, IMS/DC, etc.) user can view all the reports of that recipient, and the reports of his descendants in the Recipient Tree. The same TSO user ID (or other environment's sign-on ID) can be defined for more than one recipient in the Recipient Tree.

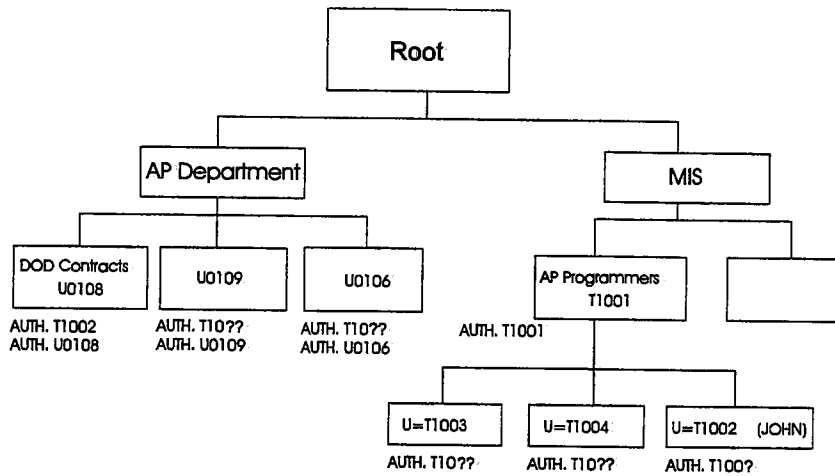
User ID identification in the AUTHORIZE field is performed according to the following rules:

- The specified user ID is treated as a prefix of the user ID (if optional wish WD2564 is set to YES in member IOADFLTC in the IOA SECUDATA library).
- The specified user ID can contain a number of ? characters. This wildcard character indicates any single character.

Example

AUTHORIZE A??X will match AIOX01, but will not match A1X01.

Consider the following tree:



Programmers with TSO user IDs beginning with T10 are responsible for reports assigned to the AP department. JOHN is the only programmer who writes programs also for the DOD contracts section which is classified. JOHN and every programmer whose user ID starts with T10 can view reports of all the members in the programming departments and all reports of the AP department. There is one exception: except for JOHN (T1002), no-one can look at the DOD contracts section and at JOHN's reports. His boss (TSO user T1001) can look at JOHN's test reports, but not at the DOD production reports. Every user in the AP department can look at his/her own reports only.

The Active User Reports list file also contains entries which describe the entire original report. The entries are referenced by a specially reserved user name - \$SYSDATA. Authorized users can perform Online Viewing of the original report (which is now compressed). The authorization for accessing \$SYSDATA entries is also controlled by this exit. A user who is authorized in the Recipient Definition screen may also be allowed to view \$SYSDATA entries if the SYSDATA field (which corresponds to the AUTHORIZE field) contains a value Y (Yes). For more information, refer to Recipient Definition screen in Section 2 of the CONTROL-D User Manual.

It is strongly recommended to authorize the use of the SYSDATA option for a restricted number of operations personnel, because this option allows viewing of original reports.

The exit also receives control for every function performed on an entry in the user list (e.g., update, print, restore, view). It is possible to control who is allowed to request a restore, to determine the maximum number of pages to be printed to a remote printer, etc. The functionality described above can also be performed by the security interface module (CTDSE04). If CTDSE04 is installed, the use of sample exit CTDX004A is not necessary.

See member DOCDX004 in the IOA SECUDATA library for more details.

CTDX005 This exit receives every line to be printed by the CONTROL-D Printers Control monitor before actual printing takes place. The exit can prevent the line from being printed, change its contents, or add more lines to the printed output. Two examples of its use:

Print page sequence numbers from the beginning of the printing mission (these numbers can be printed in the bundle index as well).

The actual printing is suppressed, and the bundle is saved to a sequential file (for file transfer to RJE stations, etc.). See "Printing to a File" in Section 4 of this guide.

See member DOCDX005 in the IOA SECUDATA library for more details about this exit.

CTDX006 SMF exit. This exit receives control before each CONTROL-D SMF record is written to the SMF datasets. It may suppress the record, or change it. The SMF record written by CONTROL-D is created for each combination of User/Report/Job name. Therefore, it allows accounting on each level – including accounting by report recipient. The number of pages used is accurate (unlike SMF type 6 records which use approximate numbers).

Note SMF type 6 records will be created by JES for the CONTROL-D Printers Control monitor when it prints bundles. Therefore, it is necessary not to count the same page twice. This is one of the reasons why the CONTROL-D SMF record is not of type 6. The SMF record number is determined by installation parameter SMF. For more details on the format of the record, see the sample exit in the IOA SECUDATA library.

See member DOCDX006 in the IOA SECUDATA library for more details about this exit.

CTDX007 This exit handles the problem of decollating reports which cannot be shown on the screen. Many installations use special character sets in reports which are not screen displayable, but which are readable when printed. It is possible under CONTROL-D to decollate such reports by specifying the name of a translation table before the string in the WHEN statement. Under this method, the user types what he/she can see, and decollation takes place using the actual non-standard spool representation. See member DOCDX007 in the IOA SECUDATA library for more details.

Section 10: Exits

- CTDX008** This exit is used to control the update of the Active Missions file. See member DOCDX008 in the IOA SECUDATA library and review the IOA Security Guide.
- CTDX009** Printing job tailoring exit. This exit can modify the contents of the printing job prepared by the CONTROL-D Print Mission. See Section 4 of this guide for a description of Printing Mission workflow.
- See also member DOCDX009 in the IOA SECUDATA library for more details about this exit.
- CTDX010** Backup job tailoring exit. This exit can modify the contents of the backup job prepared by the CONTROL-D Backup Mission. Refer to Section 4 in this guide for a description of Backup Mission workflow.
- See also member DOCDX010 in the IOA SECUDATA library for more details about this exit.
- CTDX011** Restore job tailoring exit. This exit can modify the contents of the restore job prepared by the CONTROL-D Restore Mission. Refer to Section 4 in this guide for a description of the Restore Mission workflow.
- See also member DOCDX011 in the IOA SECUDATA library for more details about this exit.
- CTDX012** Activated by utility CTDCA2P. This exit allows the user to change the contents of the records which are copied from the Active User Report list file to the Permanent User Report list file. The exit can also suppress the copying of any record.
- See also member DOCDX012 in the IOA SECUDATA library for more details about this exit.
- CTDX013** Activated by utility CTDCP2A. This exit allows the user to change the contents of the records which are copied from the Permanent User Report list file to the Active User Report list file. The exit can also suppress the copying of any record.
- See also member DOCDX013 in the IOA SECUDATA library for more details about this exit.
- CTDX014** Immediate print request banner exit. Used to print banners on immediate print request from the Online Viewing facility. The exit is similar in function to exit CTDX003 and is described in further detail in this guide.
- See also member DOCDX014 in the IOA SECUDATA library for more details about this exit.
- CTDX015** This exit receives every line to be printed by an immediate print request (under the Online Viewing facility). The exit can suppress the line, change its contents, etc. It is similar in function to exit CTDX005.
- See also member DOCDX015 in the IOA SECUDATA library for more details about this exit.

- CTDX016 This exit receives control during report decollation when a search is made in the Recipient Tree for a synonym which is identical to the recipient name found in the report (i.e., when trying to identify the recipient of the page). The exit can determine whether they are equal or not. It is mainly used to define synonym ranges. For example: User BR129 should receive all postal codes from 10200 to 10399. Refer to Section 7 of the CONTROL-D User Manual for a specific example of the use of this exit.
- See also member DOCDX016 in the IOA SECUDATA library for more details about this exit.
- CTDX017 This exit receives control for every message issued by the CONTROL-D Shout facility. It can modify the message text, change its destination, or suppress it. It may also be used for special purposes such as an interfacing Electronic Mail systems or InfoMan.
- See also member DOCDX017 in the IOA SECUDATA library for more details about this exit.
- CTDX018 This CDAM user exit is activated when CDAM parameter EXIT=YES is specified. The exit receives control for every line written to the CDAM file. A description of the exit can be found in Section 5 of the CONTROL-D User Manual.
- See also member DOCDX018 in the IOA SECUDATA library for more details about this exit.
- CTDX019 This exit is used to control the update of the CONTROL-D/PC Active Transfer file. Refer to member DOCDX019 in the IOA SECUDATA library and to the IOA Security Guide.
- See also member DOCDX019 in the IOA SECUDATA library for more details about this exit.
- CTDX020 This exit is invoked when the Print Plan file is built. It can modify the contents of print plan records before they are written to the print plan file.
- See also member DOCDX020 in the IOA SECUDATA library for more details about this exit.
- CTDX021 This exit is activated before the Recipient Tree is displayed (Screen T). Based on the library name and Recipient Tree member name, the exit determines if the Recipient Tree is to be edited or browsed.
- See also member DOCDX021 in the IOA SECUDATA library for more details about this exit.
- CTDX022 This exit is invoked during decollation. It can be used to access default records in the Active or Permanent User files, to create user and SYSDATA records in the Active User file, or to access each page of a report that is being decollated.
- See also member DOCDX022 in the IOA SECUDATA library for more details about this exit.

Replacing CONTROL-D User Exits

The CONTROL-D monitor supports several user exits: CTDX007, CTDX010, CTDX011, and CTDX022.

To load a new copy of an exit without shutting down the CONTROL-D monitor, simply enter the exit name using a modify command. For example:

```
F CONTROLD,CTDX007
```

After a few seconds, the following message is displayed on the operator console from which the modify command was issued.

```
CTD126I NEW EXIT CTDX007 LOADED
```

In case of an error while loading the new copy of an exit, an appropriate message appears on the operator console and the original copy of the exit remains active.

Tailoring CONTROL-D Banner Exit

General

There are two Banner exits under CONTROL-D:

- | | |
|---------|--|
| CTDX003 | The deferred print (printing mission) Banner exit. |
| CTDX014 | Immediate Print request Banner exit. Used for printing banners on direct print requests of Online Viewing users. |

The CONTROL-D Banner exits (CTDX003 and CTDX014) can be tailored for most installation requirements without actually coding Assembler instructions (described later in this section).

The following items are discussed in this section:

- Banner Pages
- Format of Banner Pages
- Printing User Address in Bundle Banner
- Index Printing Control
- AFP (APA) Support
- XEROX (DJDE) Support
- Initiating Page Marks
- Global Control of Printing Characteristics

Banner Pages

The CONTROL-D supplied Banner exit prints different types of banner pages. The Banner Pages are defined in the library allocated to DD statement DABANNER of the CONTROL-D Printers Control monitor, the Online monitor and of each Online user. A sample IOA Banners library is supplied as part of the IOA installation. The following banners can be found in the library:

xxBNDLST	Bundle Open banner
xxBNDLEN	Bundle Close banner
xxUSERST	Beginning of a User (Recipient) Reports banner
xxUSEREN	End of a User (Recipient) Reports banner
xxREPSTA	Start of Report banner
xxREPEND	End of Report banner
xxUINDXH	Header of Reports index
xxUINDXV	Format of each index line
\$\$ONLSTA	Start Banner of immediate print requests
\$\$ONLEND	End banner of immediate print requests

The prefix of the banner members beginning with xx is determined by the type of printer used to print the report (Xerox, APA, Siemens, etc.), as follows:

Type	Banner Prefix
REG	\$\$
LAS	\$\$
APA	\$1
XER	\$2
FOB	\$3

If a banner member of a specific printer type is not found, the default banner member is used (e.g., if \$1BNDLST is not found, default member \$\$BNDLST is used.)

The type of the printer specified in the CONTROL-D installation parameters (member CTDPARM in the IOA INSTCTD library).

Format of Banner Pages

The contents of each banner page is determined by the text which is found in the banner member. The following rules apply to printing banner pages:

The first character of each line can be:

- Any valid ASA code (1, +, -, etc.).
- X'5A' - for AFP (Advanced Function Printing).

Section 10: Exits

- A line descriptor:

Blank	Regular size letters
B	Big (large) size letters
M	Medium size letters
S	Small size letters
E	End of banner page
C	Comment line

It is possible to use special variables within the banner page definition. These variables will automatically be replaced for each banner printed. The variables are:

%MISSION% ²	Printing mission name
%USER%	User (recipient) name
%FATHER%	Father of the recipient in the tree
%REPORT%	Report name
%CATEGORY%*	Printing Mission category
%JOB NAME%	Name of the job which produced the report
%JOBID%	Job ID (in JES) which created the report
%DATE%	Current date (at printing time)
%TIME%	Current time
%ODATE%	Original scheduling date of the report
%PAGES%	Number of pages (for each type of banner and for the index)
%LINES%	Number of lines (for each type of banner and for the index)
%DEST%	Printing (JES) destination
%LASTUSER%	Last user (used in END banners)
%COPIES%	Number of copies to be printed (for report banner and for the index)
%CURRCOP%	Current printing copy (of the copies) (for report banner only)
%FROMPAGE%	From page printed
%TOPAGE%	Until page printed

² For a Deferred Print Banner only

%OUSER%³	Name of the user who requested the print
%REMARK%	Decollation remark (in Active User list file)
%RULER%	Name of the ruler being used to print the report. (If no ruler is being used, no RULER is printed.)
%GROUP%	Printing GROUP (from the printing mission definition)
%GLOBALn%	The nth line of member \$\$GLOBAL (where n can be from 1 to 10) The CONTROL-D Banners library contains the member \$\$GLOBAL. This member can contain 1-10 lines of data. When parameter %GLOBALn% is specified in the report, the nth line from this member is placed in the banner page. This parameter can be used to distribute NEWS to all report recipients.
%DATAn%	The nth line of a member whose name is identical to the user name (not to the synonym) in the Recipient Tree. Valid values for n are from 1 to 10. The CONTROL-D Banners library can contain a member for each report recipient. This member may contain 1-10 lines of data. When the parameter %DATAn% is specified in the report, the nth line from the member of the current user is placed in the banner page. CONTROL-D does not expect to find such a member for each user. A possible use of this parameter is for special distribution instructions for that user. See the sample members BMIAMI and UNIDENT.
%ADDRn%	The nth address line of the current user (recipient) in the tree. Valid values for n are from 1 to 10. The CONTROL-D Recipient Tree can contain address lines which can be inserted in the banner page of this user. Any text can be entered in these address lines. A maximum of ten address lines can be specified per user. CONTROL-D does not expect to find such lines for each user. A common use of this parameter is for the user address (to which the bundle should be delivered).
<hr/> <p>Note For performance reasons, it is recommended to use the %ADDRn% variable whenever possible (instead of the %DATAn% variable). The %ADDRn% data is always located in memory and not in a library.</p> <hr/>	
%LASTADDRn%	Used in the End user banner to print the address of the last user (which has already printed). This variable is identical to %ADDRn% and is supported for backward compatibility.

³ For immediate print banner only

Section 10: Exits

Printing User Address in Bundle Banner

CONTROL-D can group the reports of several recipients into one bundle. In such cases, it may be desirable to have a specific address assigned to the entire bundle, and to print this address on the bundle banner. CONTROL-D does this in the following way:

The contents of the GROUP field of the printing mission is compared with the Recipient Tree. If the GROUP field contains a user name from the tree (not a synonym), the use of the variable %ADDRn% in the bundle banner produces the address of this user.

Eliminate Banners

If you want to suppress the printing of a certain banner type, simply rename the banner member in the Banners library. You can suppress the printing of User and/or Report Banners for all reports of a particular recipient using the Recipient Tree definition. See Section 2 of the CONTROL-D User Manual for more details.

You can also suppress the printing of all Banners by a particular Print Mission, specify in the Print Mission category field the special name NOBANNER. (See Section 4 of the CONTROL-D User Manual for more details.)

See also parameter BANSEQ described in "Banner Printing Option" later in this section.

INDEX Printing Control

The title of the index is retrieved from member \$\$UINDXH in the BANNERS library. The format of each line of index is retrieved from member \$\$UINDXV in the BANNERS library.

If variable %COPIES% is used in the index, the printed value is the value before the actual print. If the copies count is changed during printing (e.g., it was suppressed by a user exit), the correction will not be shown on the index.

A number of different parameters can be specified in the banner exits (CTDX003 and CTDX014) to control the printing of indexes. These parameters are described as follows:

GINDEX	Whether or not an index of the bundle contents (global index) should be printed at the beginning of a printing mission (after the bundle banner page). Valid values are:
ON	The global index is printed.
OFF	The global index is not printed.
INDEX	Optionally, it is possible to create a separate index for each user in the bundle (a bundle can contain reports for more than one user). Valid values are:
ON	Each user will receive an index at the beginning of each user bundle. However, if parameter INDEX in the Recipient Tree is equal to N (or blank), an index will not be printed for this user.
OFF	An index will not be produced for each user. However, if the INDEX parameter in the Recipient Tree is equal to Y, an index will be printed for this user.

- LINECT** Specifies the maximum number of index records to be printed on a page. If LINECT=0 is specified there is no limit (i.e., any number of records may be printed on a page).
- EINDEX** Whether or not an End User Index should be printed. An End User Index is an index printed at the end of a specific user's report bundle that indicates which reports were actually printed (i.e., it does not include reports that were not printed) for that user. Valid values are:
- YES End User Indexes are to be printed. However, if parameter INDEX in the Recipient Tree is equal to N (or blank), an index will not be printed for this user.
- NO/blank End User Indexes are not printed.
- GEINDEX** Whether or not a Global End Index should be printed. The Global End Index contains an index of all reports that were actually printed in a report bundle. Valid values are:
- YES Print a Global End Index.
- NO/blank Do not print a Global End Index.

Banner Printing Options

Three parameters in the banner exits (CTDX003 and CTDX014) are used to determine where, when and how bundle banners are printed:

- BUNSEP** Where bundle banners should be printed. Valid values are:
- YES The bundle start banner, bundle index, and bundle end banner are printed to the destination specified in the print mission. (See Section 2 of the CONTROL-D User Manual, parameter DEST.)
- All other banners, indexes, and reports are printed to the destination specified in the User Reports screen. (See Section 2 of the CONTROL-D User Manual, parameter DEST.)
- NO All banners, indexes, and reports are printed to the destination specified in the User Reports screen. (See parameter DEST in Section 2 of the CONTROL-D User Manual.)
- BANSEQ** Indicates whether or not CONTROL-D should print banners when printing to a sequential file instead of the JES SPOOL. Valid values are:
- YES Print banners when printing to a sequential file.
- NO Do not print banners when printing to a sequential file.
- BANAFP** Indicates whether or not to implement PAGEDEF, FORMDEF, and OUTPUT statements when printing banners for the report. These statements can be included either with the PRINT/CDAM parameters for this report, or in the OUTPARMS library. The OUTPARMS library is described later in this section.

Section 10: Exits

YES	Implement these statements when printing banners for the relevant report.
NO/blank	Do not implement these statements when printing banners for the relevant report. Default.

OUTPARM Options

The OUTPARM options provide the ability to override the default printing characteristics (specified at decollation time) for all (or some) of the reports and banners to be printed by CONTROL-D. To activate this feature add the following parameters to the banner exits (CTDX003 and CTDX014):

OUTPARM	Whether all the members in the CONTROL-D OUTPARMS library refer to job names or to user IDs. Valid values are:
JOB	The members in the OUTPARMS library refer to job names.
USER	The members in the OUTPARMS library refer to user names.
BANNER	Whether or not banners should be printed with the characteristics specified for reports in the OUTPARMS library. Valid values are:
YES	Banners will be printed with these characteristics.
NO	Banners will not be printed with these characteristics.

For more information see "Using OUTPARM for Global Control of Printing Characteristics" in Section 4 of this guide.

AFP (APA) Support

Section 3 of the CONTROL-D User Manual and Section 4 in this guide explain how CONTROL-D supports AFP (APA) printers. When you are using AFP (APA) printers, the banner exit invokes a special routine called CTDAPA.

XEROX (DJDE) Support

See Section 4 in this guide for information about how CONTROL-D supports XEROX (DJDE) printers. When you are using XEROX printers, the banner exit invokes a special routine called CTDDJDE. A source of this routine can be found in the IOA SECUDATA library, should you require additional local adaptations.

Summary

By using various line types and special variables, it is possible to tailor the format of banner pages for most data centers without any modifying the banner exit itself. However, it is also possible to tailor the banner exit. If the supplied banner exit does not fit your requirements, you can modify the exit and adapt it for your special requirements. If you do so, let us know what changes you made and why. Your modifications may be implemented as a standard in a future release.

Submit the job, and save the member.

CONTROL-B Exits

The following CONTROL-B exits are available:

- CTBX001** Receives control for every mission order which should be placed in the Active Balancing file. For further details, review the IOA Security Guide.
- See member DOCBX001 in the IOA SECUDATA library for more details about this exit.
- CTBX003** CONTROL-B database exit. Controls all access (for read/write or update) to CONTROL-B database files (group file, basic variable definitions files, and variable generation files).
- See member DOCBX003 in the IOA SECUDATA library for more details about this exit.
- CTBX004** CONTROL-B Rule Activity file selection exit. This exit controls which users will be allowed to see what CONTROL-B invocations on the Rule Activity Display screen.
- See member DOCBX004 in the IOA SECUDATA library for more details about this exit.
- CTBX008** This exit is used to control access to the Active Balancing file. For example, the exit checks if a user is authorized to delete a mission using the Active Balancing Environment screen. For further details, see the IOA Security Guide.
- See member DOCBX008 in the IOA SECUDATA library for more details about this exit.
- CTBX009** This exit receives control for every message issued by the CONTROL-B Shout facility. It can modify the message text, change its destination, or suppress it. It can also be used for special purposes, such as an interface to InfoMan or to Electronic Mail Systems.
- See member DOCBX009 in the IOA SECUDATA library for more details about this exit.
- CTBX010** Extract user-defined processing exit. This exit is used to process extracted information according to user-defined specifications during processing of a DO EXTRACT statement. For more information on extract processing, see the description of the DO EXTRACT statement in Section 5 of the CONTROL-B user manual, and members CTBX010 and DOCBX010 in the IOA SECUDATA library.
- See member DOCBX010 in the IOA SECUDATA library for more details about this exit.

CONTROL-T Exits

The following CONTROL-T exits are available:

CTTX001 Receives control for every CONTROL-T rule which is loaded during initialization. This exit can be used to modify rule parameters. Security module CTTSE01 is associated with this exit. See the IOA Security Guide for more information on this security module.

See member DOCTX001 in the IOA SECUDATA library for more details about this exit.

CTTX002 Dynamic stacking exit. Receives control when utility CTTSTK is executed, when dynamic stacking for a dataset should be performed, and when a new dataset is added to the Media Database.

See member DOCTX002 in the IOA SECUDATA library for more details about this exit.

CTTX003 SVC operation decision. Receives control at dataset open and can determine the process of this open request. The exit has an associated security module CTTSE03.

See member DOCTX003 in the IOA SECUDATA library for more details about this exit.

CTTX004 Dynamic definition exit. Receives control when a dataset or volume is to be dynamically defined to the Media Database. The exit has an associated security module CTTSE04.

See member DOCTX004 in the IOA SECUDATA library for more details about this exit.

CTTX005 Abend exit. When CONTROL-T is about to abend a job, this exit receives control and determines whether to allow the job to abend or to bypass CONTROL-T.

If you decide to implement this exit, do so with extreme caution. If you bypass CONTROL-T, tape protection is disabled for this job.

See member DOCTX005 in the IOA SECUDATA library for more details about this exit.

CTTX006 Media Database update validation. Receives control when a request to update the Media Database is received from the SVC, Online environment, or utilities CTTRTM, CTTVTM and CTTEXP. Security module CTTSE06 is associated with this exit.

See member DOCTX006 in the IOA SECUDATA library for more details about this exit.

- CTTX007** Cycle processing exit. Receives control whenever utility CTTRTM or CTTVTM processes a cyclic dataset. The exit can determine the processing parameters.
- See member DOCTX007 in the IOA SECUDATA library for more details about this exit.
- CTTX008** Robotic Tape library interface confirmation exit. Receives control before a request is made to the Robotic Tape library routine to scratch or eject a volume. The exit can allow or reject the request.
- See member DOCTX008 in the IOA SECUDATA library for more details about this exit.
- CTTX009** External label printing. Receives control either from CONTROL-T's SVC (via statement DO LABEL) during dataset creation, or from the Online environment (screen TI or TC) by specific request. It can be used to format label structure and contents, and/or specify whether or not to print labels. Security module CTTSE09 is associated with this exit.
- See member DOCTX009 in the IOA SECUDATA library for more details about this exit.
- CTTX010** Receives control when the Dynamic Dataset Stacking facility scans the Media Database for a stackable volume. This exit can be used to change search decisions (i.e., to Reject or Accept a volume), or to supply a new volume to be used for stacking.
- See member DOCTX010 in the IOA SECUDATA library for more details about this exit.
- IOAX038** Determination of execution environment for IOA functions. This exit can be used to specify which Functional monitor is to be used to perform which IOA functions.
- See member DOCIX038 in the IOA SECUDATA library for more details about this exit.

CONTROL-O Exits

The following CONTROL-O exits are available:

- CTOX001** Receives control under the Online interface whenever a rule ORDER/FORCE is performed.
- See member DOCOX001 in the IOA SECUDATA library for more details about this exit.
- CTOX002** Receives control for every rule before it is loaded to the active environment. This exit can be used to modify rule parameters prior to rule loading.
- See member DOCOX002 in the IOA SECUDATA library for more details about this exit.

Section 10: Exits

CTOX003: Receives control before execution of each DO KSL/TSO statement. This exit can be used to modify specific requests.

See member DOCOX003 in the IOA SECUDATA library for more details about this exit.

CTOX004 Receives control each time a user enters an Automation Options screen or attempts to perform an action in an Automation Options screen.

See member DOCOX004 in the IOA SECUDATA library for more details about this exit.

CTOX008 Receives control each time a user enters the Rule Status screen or attempts to perform operations on the Rule Status screen.

See member DOCOX008 in the IOA SECUDATA library for more details about this exit.

Appendices

Appendix A: IOA Application Program Names

Appendix B: Dataset Formatting Utilities for IOA Products

Appendix C: Modifying IOA Online Facility Commands

Appendix D: Logical Field Names for the CONTROL-T Repository

Appendix E: IOA Online Options Cross-Reference

Index



Appendix A: IOA Application Program Names

The following application program names can be found on IOA product tapes:

CTBIUTI	CONTROL-B Utilities screen
CTBTABS	CONTROL-B Active Balancing Environment screen
CTBTBAL	CONTROL-B Balancing Mission Definition screen
CTBTDAT	CONTROL-B Database Variables Definition screen
CTBTJBL	CONTROL-B Rule Activity screen
CTBTRLB	CONTROL-B Rule Definition screen
CTDIUTI	CONTROL-D Utilities screen
CTDTAMS	CONTROL-D Active Missions screen
CTDTATF	CONTROL-D/PC File Transfer Status screen
CTDTPC	CONTROL-D file Transfer screen
CTDTMIS	CONTROL-D/V Migration/Printing/Backup/Restore Mission Definition screen
CTDTREP	CONTROL-D Report Decollating/Indexing Definition screen
CTDTUSR	CONTROL-D User Reports screen (Online Report Viewing)
CTDTUTR	CONTROL-D Recipient Tree screen
CTMTDTM	IOA Calendar screen
CTMTNRS	IOA Manual Conditions screen
CTMTRES	IOA Condition Resources screen
CTMTSCH	CONTROL-M Online Scheduling
CTMTRCM	CMEM Definition facility
CTMTSTA	CONTROL-M Status screen
CTMTTSO	IOA TSO Command Processor screen
CTOTARF	CONTROL-O Rule Status screen
CTOTMSC	CONTROL-O Message Statistics screen
CTOTOMP	CONTROL-O Rule Definition screen
CTOTALO	CONTROL-O Automation Log
CTOTAOP	CONTROL-O Automation Options
CTTTRLD	CONTROL-T Rule Definition screen
CTTTPLD	CONTROL-T Pool Definition screen
CTTTVLD	CONTROL-T Vault Definition screen
CTTTINQ	CONTROL-T Inquire/Update screen
CTTTKIN	CONTROL-T External Volume Check-In screen
IOATLOG	IOA Log screen
IOATMNU	IOA Primary Option menu



Appendix B: Dataset Formatting Utilities for IOA Products

The following JCL members can be used to create and format the IOA and CONTROL products' datasets:

1. IOA (in IOA INSTALL Library)

FORMDRES	IOA Mirror (Dual) Conditions/Resources file
FORMIOA	Formats all IOA files
FORMLOG	IOA Log
FORMNRS	IOA Manual Conditions file
FORMRES	IOA Conditions/Resources file

2. CONTROL-M (in IOA INSTCTM Library)

FORMCKP	CONTROL-M Active Jobs file and its backup file
FORMCTM	Formats all CONTROL-M files
FORMDCKP	CONTROL-M Mirror (Dual) Active Jobs file
FORMG2M	CONTROL-M Gateway Communications file
FORMGRF	CONTROL-M Dependencies file
FORMSUB1	CONTROL-M CMEM file
FORMSUB2	CONTROL-M CMEM file
FORMSTT	CONTROL-M Job Execution Statistics file

3. CONTROL-D (in IOA INSTCTD Library)

FORMAMF	CONTROL-D Active Missions file and its backup file
FORMAS4	CONTROL-D MF/AS400 Transfer file
FORMATF	CONTROL-D Active Transfer file for CONTROL-D/PC
FORMBTR	CONTROL-D Bundle Tracking file
FORMCAT	Creates the IOA user catalog for the CDAM files
FORMCOM	CONTROL-D Communication file
FORMCTD	Formats all CONTROL-D files
FORMPGC	The file that saves the number of pages prepared for all print missions
FORMUF1	Allocates and formats all CONTROL-D User files
FORMUF2	Reformats all CONTROL-D User files without reallocating

Appendix B: Dataset Formatting Utilities for IOA Products

4. CONTROL-V (in IOA INSTCTV library)

FORMCTV Allocates and formats CONTROL-V Migration file
FORMUF3 Allocates and formats CONTROL-V Migrated User Reports file
FORMUF4 Formats CONTROL-V Migrated User Reports file without reallocating

5. CONTROL-O (in the IOA INSTCTO library)

DEFGLOB Allocates and formats the Global Variables library
DEFSTAT Allocates and formats the Message Statistics file
DEFALO Allocates and formats the Automation Log file

6. CONTROL-T (in IOA INSTCTT library)

CTTCMDB Allocates and formats CONTROL-T Media Database (MDB) file
CTTCSTK Allocates and formats CONTROL-T Stacking Statistics file
CTTCTRC Allocates and formats CONTROL-T Trace file

7. Enterprise Controlstation (in IOA INSTECS library)

FORMM2G Allocates and formats CONTROL-M/ECSGATE Communications file

Appendix C: Modifying IOA Online Facility Commands

Every screen of the IOA Online facility has a set of valid commands which are supported under it. It is possible to change the names of these commands or to create synonyms. The commands reside in the IOA PARM library in members with the following naming conventions:

TxxxCMD1	The active commands member.
TxxxCMDD	Debugging aid member - contains additional commands for problem analysis.

xxx is the screen identifier. For a detailed description of how to modify a command member, refer to the Section 2 of this guide.

A command member is composed of one header line, and any number of command lines. The number at the left of the header line is the total number of command lines in the member. It must be updated when lines from the command member are added or deleted.

The structure of the command line is as follows:

Column	Description
1-8	The command name.
25-28	Reserved hexadecimal (unprintable) value. Do not change it.
29-68	A description of the command.
69-72	Reserved hexadecimal (unprintable) value - the internal command member. Do not change it.

It is possible to change the command name and/or the description. To add a synonym, duplicate the command line, modify the command name on the duplicated line, and modify the command line counter at the left side of the header line.

Modifying IOA Online Facility PFKey Definitions

Every screen of the IOA Online facility has a set of PFKeys with pre-assigned commands. It is possible to change these PFKey definitions. The PFKey assignments reside in the IOA PARMCMD library in members with the following naming conventions:

TxxxPF1	xxx is the screen identifier.
---------	-------------------------------

A PFKey member is composed of one header line, and any number of PFKey definition lines.

The structure of the PKFey definition line is as follows:

Column	Description
1-8	The PFKey or <ENTER>.
9-22	The command assigned to the PFKey.

At least one PFKey/Enter key must be defined as the <ENTER> command.



Appendix D: Logical Field Names for the CONTROL-T Repository

The tables in the following pages list fields that can be:

- Specified in utility INCLUDE/EXCLUDE statements to determine which records are processed when the utility runs.
- For a complete explanation of INCLUDE/EXCLUDE statements, see "Record Selection Logic (INCLUDE/EXCLUDE Statements)" at the beginning of the CONTROL-T utility descriptions in Section 9 of this guide.
- Manually updated in the Media Database via utility CTTMUP.
- Included in utility-generated reports.

The following pages contain a series of tables which specify field names according to record type. Each table contains the following data for each field:

External Name	Field name as specified by the user.				
Valid Values	Either of two types of data: <table> <tr> <td>Keyword value</td> <td>Specific valid values. Shown in uppercase (e.g., CYCLE, DATE, JCL). Non-trivial values are accompanied by a description in the Description column of the table.</td> </tr> <tr> <td>Value type</td> <td>Shown in lowercase (e.g., char, integer, date). These are described below. If relevant, the maximum length of the value is displayed in parentheses next to the value type.</td> </tr> </table>	Keyword value	Specific valid values. Shown in uppercase (e.g., CYCLE, DATE, JCL). Non-trivial values are accompanied by a description in the Description column of the table.	Value type	Shown in lowercase (e.g., char, integer, date). These are described below. If relevant, the maximum length of the value is displayed in parentheses next to the value type.
Keyword value	Specific valid values. Shown in uppercase (e.g., CYCLE, DATE, JCL). Non-trivial values are accompanied by a description in the Description column of the table.				
Value type	Shown in lowercase (e.g., char, integer, date). These are described below. If relevant, the maximum length of the value is displayed in parentheses next to the value type.				
Description	Brief description of the field. This column also contains a description of any specific values specified in the Valid Values column.				
Internal Name	Name used in the macro to reference the field.				

Below is a brief description of the value types that can be specified in the Valid Values column:

char	Free character format. Maximum length indicated in parentheses.
hex	Hexadecimal character format (valid characters are 0-9, A-F). Maximum length indicated in parentheses.
date	Valid formats are: yymmdd or yyyyymmdd. The following keywords are also valid:
%CDAY	Current day (today).
%PDAY	Previous day (yesterday).
%NDAY	Next day (tomorrow).
%PWEEK	Previous week.
%NWEEK	Next week.
%PMONTH	Previous month (30 days).
%NMONTH	Next month (30 days).

Appendix D: Logical Field Names for the CONTROL-T Repository

time Valid format is hhmmss.

julian date Valid formats are yyddd or yyyyddd. Special "dates" (e.g., 98000, 99000) are also supported.

integer Numeric value.

date/integer Either a date value or an integer value can be specified (as described above). Date values should be prefixed by D (e.g., format Dyyyymmdd). Integer values should be prefixed by I (e.g., I35). By default (i.e., if no prefix is specified), the value is assumed to be a date.

Table 1: Keywords to Be Used With Dataset Type Records

EXTERNAL NAME	VALID VALUES	DESCRIPTION	INTERNAL NAME
ACCOUNT	char (50)	Job's accounting info.	DDSACCT
BLKSIZE	integer	Dataset's block size	DDSBLK
BLOCKCT	integer	Block count	DDSBLKC
CRECC	char (4)	Condition code of creating step	DDSCCC
CRECPU	char (4)	CPU in which dataset was created	DDSCCPU
CREDDN	char (8)	Creating DD name	DDSCDDN
CREDT	date	Creation date	DDSCDT
CREJBN	char (8)	Creating job name	DDSCJBN
CREJOBID	char (8)	Creating job ID (format JOBxxxxx)	DDSCJID
CREPGM	char (8)	Creating program name	DDSCPGM
CRESTEP	char (8)	Creating step name	DDSCSTP
CRETM	time	Creation time	DDSCTM
CREUAD	hex (4)	Unit address on which the dataset was created	DDSCUAD
CREUSER	char (8)	User ID which created the dataset	DDSCUSER
DCHANGED	date	Last change date	DDSMdT
DCHANGET	time	Last change time	DDSMtM
DCHANGEU	char (8)	Last user ID that changed the entry	DDSLUSER

Appendix D: Logical Field Names for the CONTROL-T Repository

EXTERNAL NAME	VALID VALUES	DESCRIPTION	INTERNAL NAME
DENSITY	BPI200 BPI556 BPI800 BPI1600 BPI6250 BPI38000 BPI38KE BPI3590	Density of the data on the volume	DDSDEN
DSENFET	integer	Dataset ending footage on the volume	DDSEFTG
DSEXCP	integer	EXCP count since dataset was created	DDSEXCP#
DSEXPDT	date/integer	Dataset's expiration date, number of cycles or number of days according to DSEXPTYP	DDSEXPDT
DSEXPTYP	CYCLE DATE EDM LACCS CATALOG PERM VLTRET SMS	Dataset's expiration type: Number of cycles Actual date EDM controlled Number of days since last accessed MVS Catalog Control Permanent dataset Controlled by vaulting Controlled by SMS Management Class	DDSEXPT1
DSFEET	integer	Footage occupied by the dataset	DDSFEEET
DSFLAGS	DISPMOD DYNDEF RECREATE MANUPDAT JCLRETPD DELRTM DSRSTRT	Dataset flag byte: Dataset opened with DISP=MOD Dataset dynamically defined Dataset recreated Dataset manually updated (Online/Utility CTTMUP) RETPD specified in the JCL for the dataset Dataset marked as to be deleted by CTTRTM Dataset processed under MVS RESTART	DDDFLG1

Appendix D: Logical Field Names for the CONTROL-T Repository

EXTERNAL NAME	VALID VALUES	DESCRIPTION	INTERNAL NAME
DSLABEL	integer	Dataset label number	DDSLBLNM
DSNAME	char (44)	Dataset name	DDSDSN
DSSTAT	DACTIVE DINUSE DPENDSCR DSCRATCH DSTACKED DEDM DABEND DNSTK	Dataset's status: Active dataset In-use dataset Pending scratch dataset Scrapped dataset Stacked dataset EDM controlled dataset Closed under abend dataset Unstackable dataset	DDSSTAT
DSUSECT	integer	Dataset's usage count	DDSUSED#
DSVOLSER	char (6)	Dataset's first volume	DDSVOLSR
DUSRDATA	char (20)	User data	DDSUSER2
JCLEXPDT	Julian date	JCL EXPDT value	DDSJEXPD
LASTACCS	C,R,W	Type of last access (create, read, write)	DDSLACS
LRECL	integer	Dataset's logical record length	DDSRECL
READCC	char (44)	Condition code of the last step which read the dataset	DDSRCC
READCPU	char (4)	CPU in which dataset was last read	DDSRCPU
READDDN	char (8)	Last read DD name	DDSRDDN
READDT	date	Last read date	DDSRDT
READJBN	char (8)	Name of the last job which read the dataset	DDSRJBN
READPGM	char (8)	Name of last program which read the dataset	DDSRPGM
READSTEP	char (8)	Name of the last step which read the dataset	DDSRSTP
READTM	time	Last read time	DDSRTM
READUAD	hex (4)	Unit address on which the dataset was last read	DDSRUAD

Appendix D: Logical Field Names for the CONTROL-T Repository

EXTERNAL NAME	VALID VALUES	DESCRIPTION	INTERNAL NAME
RECFM	U, F, FB, FBA, FBM, FBS, V, VB, VS, VBA, VBM, VBS, D	Dataset's record format	DDSRECFM
RTNFROM	DEFAULT JCL RULE SMS SMSDFLT SMSLIMIT SMSJCL	Dataset Retention origin: From CTPARM From JCL From CONTROL-T rules From SMS Management Class From SMS default From SMS limit From SMS JCL	DDSRTOG
SMSMC	char (8)	SMS Management Class	DDSSMCL
TRTCH	COMP, NOCOMP	Indication of whether or not the data is compressed (e.g. IDRC)	DDSTRTCH
VOLSNUM	integer	Number of volumes on which the dataset resides (multi-volume dataset)	DDSVOLS#
WRITECCC	char (4)	Condition code of the last step which wrote to the dataset	DDSWCC
WRITECPU	char (4)	CPU in which the dataset was last written	DDSWCPU
WRITEDDN	char (8)	Last write DD name	DDSWDDN
WRITEDT	date	Last write date	DDSWDT
WRITEJBN	char (8)	Name of the last job to write to the dataset	DDSWJBN
WRITEPGM	char (8)	Name of last program which wrote to the dataset	DDSWPGM
WRITESTP	char (8)	Name of the last step which wrote to the dataset	DDSWSTP
WRITETM	time	Last write time	DDSWTM
WRITEUAD	hex (4)	Unit address on which the dataset was last written	DDSWUAD

Table 2: Keywords to Be Used With Volume Type Records

EXTERNAL NAME	VALID VALUES	DESCRIPTION	INTERNAL NAME
ACTIVEDS	integer	Number of active datasets	DVLACTD#
BOXID	char (6)	Box ID	DVLBOXID
CHKINDT	date	Check-in date	DVLFCDT
CLNCOUNT	integer	Volume's Clean Count	DVLCLN#
EXPDSNUM	integer	Expiration dataset number	DVLEXPDS
EXPRTRN	date	Expected return date from out location	DVLRTRN
FIRSTVOL	char (6)	First volser for multi-volume chains	DVLFIRST
IOERPRM	integer	Read permanent errors count	DVLRPER
IOERPRMC	integer	Read permanent errors count since last clean	DVLRPERC
IOERTMP	integer	Read temporary errors count	DVLRTER
IOERTMPC	integer	Read temporary errors count since last clean	DVLRTERC
IOEWPRM	integer	Write permanent errors count	DVLWPER
IOEWPRMC	integer	Write permanent errors count since last clean	DVLWPERC
IOEWTMP	integer	Write temporary errors count	DVLWTER
IOEWTMPC	integer	Write temporary errors count since last clean	DVLWTERC
LACCDT	date	Date of last volume access	DVLADT
LACCJBN	char (8)	Name of last job which accessed the volume	DVLJBN
LACCTM	time	Time of last volume access	DVLATM
LBLNUM	integer	Last label number on the volume	DVLLBLNM
LBLTYP	SL, NL, NSL, SUL, AL, AUL, BLP	Label type	DVLLTYP
LCLNDT	date	Last clean date	DVLCLN
LDSFEET	integer	Last dataset on volume footage	DVLLDFTG
LOCATION	char (8)	Current location of the volume	DVLCODE

Appendix D: Logical Field Names for the CONTROL-T Repository

EXTERNAL NAME	VALID VALUES	DESCRIPTION	INTERNAL NAME
LOCSEQ	integer	Vault sequence number	DVLVTSQ
MEDIA	char (8)	Volume's media type	DVLMEDIA
MOVEDATE	date	Last move date	DVLMVDT
NEXTVOL	char (6)	Next volser for multi-volume chains	DVLNEXT
PREVVOL	char (6)	Previous volser for multi-volume chains	DVLPREV
RECFROM	char (8)	Vault name from which the volume recalled	DVLVLTRC
RECVLNUM	integer	Vault number (in pattern) from which the volume recalled	DVLVLTII
REMFEET	integer	Available footage of the volume	DVLRFEET
RETVLTDT	date	Return to vault date	DVLVRTRN
SCRDT	date	Last volume scratch date	DVLSCRDT
SLNAME	char (6)	Volume's SL-NAME	DVLSLNAME
SLOTNUM	integer	Vault slot number	DVLVSLOT
UNITNAME	char (8)	Volume's unit name	DVLUNAM
USERDATA	char (20)	User data	DVLUSER2
VAULT	char (8)	Vault name in vault info	DVLVLTNM
VCHANGED	date	Last change date	DVLMDT
VCHANGET	time	Last change time	DVLMTM
VCHANGEU	char (8)	Last user ID that changed the entry	DVLUSER
VDENSITY	BPI200 BPI556 BPI800 BPI1600 BPI6250 BPI38000 BPI38KE BPI3590	Density of the data on the volume	DVLDEN
VENDOR	char (8)	Name of vendor who manufactured the volume	DVLVNDOR
VLTDSDNUM	integer	Vaulting dataset number	DVLVLTDS
VLTDSDT	date	Entry date to the vault	DVLVENDT
VLTDSDNM	integer	Number of vault entries for the volume	DVLVLTNT

Appendix D: Logical Field Names for the CONTROL-T Repository

EXTERNAL NAME	VALID VALUES	DESCRIPTION	INTERNAL NAME
VLTEXPDT	date/integer	Vault expiration date, number of cycles or number of days	DVLVXPD1
VLTEXTYP	VCYC VDATE VLACC VCAT VPERM VEXP VDAYS	Vault expiration type: Number of cycles Actual date (in yyyyymmdd format) Number of days since last accessed MVS Catalog Control Permanent Until volume expiration Number of days in vault	DVLVXPT1
VLTPREFL	integer	Vaulting dataset name prefix length	DVLPRFLN
VOLEDMID	char (4)	EDM-ID for EDM controlled volumes	DVLEDMID
VOLEXCP	integer	EXCP count since last scratched	DVLEXCP#
VOLEXPDT	date	Expiration date	DVLEXPDT
VOLEXPTY	DATE, PERM, UNKNOWN	Expiration date type	DVLEXPT
VOLFEET	integer	Volume's used footage	DVLFEET
VOLFLAGS	RETURNVL DYNDEF MANUPDAT VLRSTRT EXTDEL MANVLT SINGLVOL INROBOT	Additional volume status flags: Volume returned from vault Volume dynamically defined Volume manually updated (Online/Utility CTTMUP) Volume processed under MVS RESTART Volume will be deleted when expired Volume manually moved to vault Single volume (out of multi-volume group) in VAULT/RECALL Volume resides in a robot	DVLFLG1

Appendix D: Logical Field Names for the CONTROL-T Repository

EXTERNAL NAME	VALID VALUES	DESCRIPTION	INTERNAL NAME
VOLFLAG2	VLTRYBOX	Additional status flag: Volume can be vaulted by boxes	DVLFLG2
VOLIND	PVLT PENDVLT RECALL HOLD NOSTACK INUSE	Additional volume status indications: Potential vault Pending vault Recalled back from a vault Hold before return to vault Stacked datasets not accepted Volume is in-use	DVLSTA2
VOLODESC	char (20)	Volume's description	DVLDESC
VOLOWNER	char (8)	Volume's owner	DVLOWNR
VOLSEQ	integer	Volume sequence number	DVLVSEQ
VOLSER	char (6)	Volume's serial number	DVLVOLSR
VOLSTAT	ACTIVE SCRATCH PENDSCR OUT VAULTED EXTERNAL EDM DELETED	Status of volume: Active Scratch Pending scratch Out of library (but not vaulted) Vaulted External volume EDM controlled volume Deleted volume	DVLSTAT
VOLUSECT	integer	Volume usage count from last scratch	DVLUSED#
VRTTCH	COMP NOCOMP	Indicates whether or not data is compressed (e.g., IDRC)	DVLTRTCH

Table 3: Keywords to Be Used With Stacking Type Records

EXTERNAL NAME	VALID VALUES	DESCRIPTION	INTERNAL NAME
SDENSITY	BPI200 BPI556 BPI800 BPI1600 BPI6250 BPI38000 BPI38KE BPI3590	Density of the data on the volume	STUDDEN
STKBLKCT	integer	Dataset's block count	STUDBLC
STKBLKSZ	integer	Dataset's block size	STUDBLKS
STKCOMPF	integer	Stacking compression factor	STUDCMPF
STKDATE	date	Creation date	STUDDTE
STKDSN	char (44)	Dataset name for which stacking statistics are kept	STUDDSN
STKIND	NOSTK	Indication whether or not the dataset can be stacked	STUFLAG1
STKJBN	char (8)	Job name by which stacking statistics are kept for the dataset	STUDJOB
STKPGM	char (8)	Name of the program which created the stacked dataset	STUDPGM
STKTIME	time	Creation time	STUDTME
STKUNIT	hex (4)	Unit name used to create the dataset	STUDUNIT
STKVOL1	char (6)	First volume in the group of volumes	STUDVOL
STRTCH	COMP NOCOMP	Indicates whether or not data is compressed (e.g. IDRC)	STUDRTCH

Table 4: Keywords to Be Used With Trace Type Records

EXTERNAL NAME	VALID VALUES	DESCRIPTION	INTERNAL NAME
CPUID	char (4)	CPU-ID	ARCCPUID
DATE	date	Date	ARCDATE
ENV	DLD5 EDM MUP ONL ONLC RTM STK SVCT UTL VTM \$RCV	CTTDLD utility External Data Manager Manual update utility (CTTMUP) Online facility External Volume Check-In (TC Screen) Retention management utility (CTTRTM). Dynamic Stacking Facility Real-time environment Volume expiration utility Vault management utility (CTTVTM) Utility CTTRCV Recover Media Database using Trace file	ARCIDENT
FUNCTION	ADD, DELETE, CHANGE	Activity type	ARCTYPE
JOBID	char (8)	Job ID (format JOBxxxxxx)	ARCJOBID
TIME	time	Time	ARCTIME
USER ID	char (8)	User ID/Job name	ARCUSRID

Table 5: Keywords to Be Used With Trace Data

EXTERNAL NAME	VALID VALUES	DESCRIPTION	INTERNAL NAME
RECTYPE	V, D	MDB Record Type	DVLRTYPE, DDSRTYPE
VOLSER	char (6)	Volume serial number	DVLVOLSR
DSNAME	char (44)	Dataset name	DDSDSN
RBA	hex (6)	Record's RBA	DVLRBA, DDSRBA

Appendix E: IOA Online Options Cross-Reference

IOA

Option	Screen Title	Module/ CSECT	Screen Member	Format Member	PFKey Member	Command Member	Help Member
1	IOA Primary Option Menu	IOATMNU		\$\$MNU	TMNUPF3	TMNUCMD3	CTMHMNU*
4	IOA Conditions/ Resources	CTMTRES	CTMSRES		TRES PF1	TMNUCMD3	CTMHMNU*
5	IOA LOG file	IOATLOG		\$\$LOG	TLOG PF1	TLOGCMD1	CTMHLOG
6	TSO - TSO command	CTMTTSO	CTMSTSO		TTSO PF1	TTSOCMD1	CTMHTSO
6 *	ISPF Utilities	IOAIUTI					
7	IOA Manual Conditions	CTMTNRS	CTMSNRS		TNRSPF1	TNRSCMD1	CTMHNRS
8	IOA Calendar Facility (Entry Panel)	CTMTDTM	CTMSDTM		TDTMPF1	TDTMCMD1	CTMHDTM
8.D	List of Calendars	CTMTDTM	CTMSDTD		TDTMPF1	TDTMCMD1	CTMHDT2
8.D	List of Years	CTMTDTM	CTMSDTD		TDTMPF3	TDTMCMD1	CTMHDT3
8.Y	Calendar Definition	CTMTYER	CTMSYER		TDTMPF4	TDTMCMD1	CTMHYER

* A separate ISPF panel cross-reference appears later in this section.

CONTROL-M

Option	Screen Title	Module/ CSECT	Screen Member	Format Member	PFKey Member	Command Member	Help Member
2	Scheduling Definition Entry Panel	CTMTSCH	CTMSSCH		TSCHPF1	TSCHCMD1	CTMHSC1
2	Table List	CTMTSCH	CTMSDIR		TSCHPF3	TSCHCMD1	CTMHSC2
2	Job List	CTMTSCH	CTMSDIR		TSCHPF3	TSCHCMD1	CTMHSC3
2.0	Job Order Messages	CTMTJOB	CTMSJOB		TJOBPF1	TJOBCMD1	CTMHJOB
2.G	Graphic Job Flow	CTMRFLW	CTMSFLW		TGRFPF1	TGRFCMD1	CTMHGRF
2.P	Job Scheduling Plan	CTMTRPL		\$\$RPL	TRPLPF1	TRPLCMD1	CTMHRPL
2	Job Definition Plan	CTMTGZM	CTMSGZM		TZUMCMD1	TZUMCMD1	CTMHZUM
2	Group Definition Screen	CTMTGZM	CTMSGZM		TGZMPF1	TGZMCMD1	CTMHGZM
3	Active Environment	CTMTSTA	CTMSSTA		TSTAPF1	TSTACMD1	CTMHSTA
3.Z	Job Zoom Screen	CTMTSZM	CTMSSZM		TSZMPF1	TSZMCMD1	CTMHSZM
3.N	Job's NET Display	CTMTNET		\$\$NET	TNETPF1	TNETCMD1	CTMHNET
3.?	Job's WHY Screen	CTMTNET		\$\$NET	TNETPF1	TNETCMD1	CTMHNET
3.S	Job's Statistics	CTMTVST	CTMTVST		TVSTPF1	TVSTCMD1	CTMHVST
3.V	Job's View SYSOUT	CTMTSYV		\$\$AJH \$\$AJA	TAJHPF1	TAJHCMD1	CTMHAJH
3.V.S	Job's SYSOUT Viewing	IOATOLV	CTMSSYV		TSYVPF1	TSYVCMD1	CTMHSYV
3.VIEW	Job's GROUP DISPLAY	CTMTVEW	CTMSVEW		TVEWPF1	TVEWCMD1	CTMHVEW
C	CMEM Entry Panel	CTMTRCM	CTMSRCM		TRCMPF1	TRCMCMD1	CTMHRCM
C	Tables/Rules of Library	CTMTRCM	CTMSRDR		TRCMPF3	TRCMCMD1	CTMHCM2
C	Rules of Tables in Library	CTMTRCM	CTMSRDR		TRCMPF3	TRCMCMD1	CTMHCM3
C	Rule Definition Screen	CTOTRUL	CTOSRUL		TRULPF1	TRULCMD1	CTMHRUL

CONTROL-D

Option	Screen Title	Module/ CSECT	Screen Member	Format Member	PFKey Member	Command Member	Help Member
A	Mission Status Screen	CTDTAMS	CTDSAMS		TAMSPF1	TAMSCMD1	CTDHAMS
A.Z	Zoom Decollating Mission Definition Screen	CTDTDEC	CTDSDEC		TDECPF1	TDECCMD1	CTDHDEC
A.Z	Zoom Printing Mission Definition Screen	CTDTAPR	CTDSAPR		TAPRPF1	TAPRCMD1	CTDHAPR
A.Z	Zoom Restore Mission Definition Screen	CTDTARS	CTDSARS		TARSPF1	TARSCMD1	CTDHARS
A.Z	Zoom Backup Mission Definition Screen	CTDTABK	CTDSABK		TABKPF1	TABKCMD1	CTDHABK
A.L	LOG messages	IOATLOG		\$\$LOG	TLOGPF1	TLOGCMD1	CTMHLOG
A.P	Print Plan Screen	CTDTPMM	CTDSPMM		TPMMPF1	TPMMCMD1	CTDHPMM
A.?	WHY Screen	CTDTDWY	CTDSDWY		TDWYPF1	TDWYCMD1	CTDHDWY
R	Report Decollating Mission Definition Entry Panel	CTDTREP	CTDSREP		TREPPF1	TREPCMD1	CTDHREP
R	Job List	CTDTREP	CTDSDIR		TREPPF3	TREPCMD1	CTDHRE2
R	Categories of Library	CTDTREP	CTDSDIR		TREPPF3	TREPCMD1	CTDHRE3
R	Report Decollation Mission Definition Screen	CTDTCAT	CTDSCAT		TCATPF1	TCATCMD1	CTDHCAT
R.O	Order Decollation Mission Screen	CTMTJOB	CTDSCAM		TCAMPF1	TCAMCMD1	CTDHCAM
M	Mission Definition Entry Panel	CTDTMIS	CTDSMIS		TMISPF1	TMISCMD1	CTDHMIS
M	Mission List	CTDTMIS	CTDSMSD		TMISPF3	TMISCMD1	CTDHMI2
M	Category List	CTDTMIS	CTDSMSD		TMISPF3	TMISCMD1	CTDHMI3

Appendix E: IOA Online Options Cross-Reference

Option	Screen Title	Module/ CSECT	Screen Member	Format Member	PFKey Member	Command Member	Help Member
M	Print Mission Definition Screen	CTDTPRT	CTDSPRT		TPRTPF1	TPRTCMD1	CTDHPRT
M	Backup Mission Definition Screen	CTDTBKP	CTDSBKP		TBKPPF1	TBKPCMD1	CTDHBKP
M	Restore Mission Definition Screen	CTDTRST	CTDSRST		TRSTPF1	TRSTCMD1	CTDHRST
M.O	Mission Order	CTMTJOB	CTDSMIM		TMIMPF1	TMIMCMD1	CTDHMIM
F	PC Packet Status	CTDTATF	CTDSATF		TATFPF1	TATFCMD1	CTDHATF
F.Z	Packet Zoom Screen	CTDTZTF	CTDSZTF		TDECPF1	TDECCMD1	CTDHZTF
F.I	Packet Index Screen	CTDTITF	CTDSITF		TITFPF1	TITFCMD1	CTDHITF
U	User Reports Entry Panel	CTDTUSR	CTDSABK	\$\$USR	TUSRPF1	TUSRCMD1	CTDHUSR
U	User Report Lists	CTDTUSR		\$\$FRM	TUSRPF2	TUSRCMD2	CTDHUS2
U.N	General Notepad Screen	CTDTNTP	CTDSNTP				
U.P	Print Report	CTDTUSR		\$\$FRM			CTDHUSI
U.V	Report Viewing	IOATOLV	CTDSOLV		TOLVPF1	TOLVCMD1	CTDHOLV
U.V	Tag Notepad Screen	CTDTNTP	CTDSNTP				
U.E	Report Editing Screen	CTDTEXT	CTDSEXT		TOLVPF2	TOLVCMD1	CTDHEXT
U.E.1	Edit Report Lines and Columns	CTDTLNF	CTDSLNF		TOLVPF3	TOLVCMD1	CTDHLNF
U.E.1.C	Edit Report Columns	CTDTCLF	CTDSCLF		TOLVPF5	TOLVCMD1	CTDHCLF
U.E.2	Include Lines Based on Strings	CTDTINC	CTDSINC		TOLVPF4	TOLVCMD1	CTDHINC
U.E.3	Exclude Lines Based on Strings	CTDTEXC	CTDSEXC		TOLVPF4	TOLVCMD1	CTDHINC
U.E.4	Color Lines Based on Strings	CTDTCOL	CTDSCLR		TOLVPF4	TOLVCMD1	CTDHCLR

Appendix E: IOA Online Options Cross-Reference

Option	Screen Title	Module/ CSECT	Screen Member	Format Member	PFKey Member	Command Member	Help Member
T	Recipient Tree Entry Panel	CTDTUTR	CTDSUTR		TUTRPF1	TUTRCMD1	CTDHUTR
T	CONTROL-D Recipient Tree	CTDTUTR	CTDSUTR		TUTRPF2	TUTRCMD1	CTDHUT2
T	CONTROL-D CHECKTREE Messages	CTMTJOB	CTDSTRM		TTRMPF1	TTRMCMD1	CTDHTRM
T.S	CONTROL-D Recipient Definition	CTDTRCP	CTDSRCP		TRCPPF1	TRCPCMD1	CTDHRCP

CONTROL-V

Option	Screen Title	Module/ CSECT	Screen Member	Format Member	PFKey Member	Command Member	Help Member
A	Mission Status	CTDTAMS	CTDSAMS		TAMSPF1	TAMSCMD1	CTVHAMS
A.Z	Zoom Decollating Mission Definition	CTDTDEC	CTDSDEC		TDECPF1	TDECCMD1	CTVHDEC
A.Z	Zoom Printing Mission Definition	CTDTAPR	CTDSAPR		TAPRPF1	TAPRCMD1	CTVHAPR
A.Z	Zoom Restore Mission Definition	CTDTARS	CTDSARS		TARSPF1	TARSCMD1	CTVHARS
A.Z	Zoom Backup Mission Definition	CTDTABK	CTDSABK		TABKPF1	TABKCMD1	CTVHABK
A.Z	Zoom Migration Mission Definition	CTDTABK	CTDSABK		TABKPF1	TABKCMD1	CTVHAMG
A.L	LOG Messages	IOATLOG		\$\$LOG	TLOGPF1	TLOGCMD1	CTMHLOG
A.P	Print Plan Screen	CTDTPMM	CTDSPMM		TPMMPF1	TPMMCMD1	CTVHPMM
A.?	WHY Screen	CTDWDWY	CTDSDWY		TDWYPF1	TDWYCMD1	CTVHDWY
R	Report Decollating Mission Definition Entry Panel	CTDTREP	CTDSREP		TREPPF1	TREPCMD1	CTVHREP
R	Job List	CTDTREP	CTDSDIR		TREPPF3	TREPCMD1	CTVHRE2

Appendix E: IOA Online Options Cross-Reference

Option	Screen Title	Module/ CSECT	Screen Member	Format Member	PFKey Member	Command Member	Help Member
R	Categories of Library	CTDTREP	CTDSDIR		TREPPF3	TREPCMD1	CTVHRE3
R	Report Decollation Mission Definition Screen	CTDTCAT	CTDSCAT		TCATPF1	TCATCMD1	CTVHCAT
R.O	Order Decollation Mission	CTMTJOB	CTDSCAM		TCAMPF1	TCAMCMD1	CTVHCAM
M	Mission Definition Entry Panel	CTDTMIS	CTDSMIS		TMISPF1	TMISCMD1	CTVHMIS
M	Mission List	CTDTMIS	CTDSMSD		TMISPF3	TMISCMD1	CTVHMI2
M	Category List	CTDTMIS	CTDSMSD		TMISPF3	TMISCMD1	CTVHMI3
M	Print Mission Definition Screen	CTDTPRT	CTDSPRT		TPRTPF1	TPRTCMD1	CTVHPRT
M	Backup mission Definition Screen	CTDTBKP	CTDSBKP		TBKPPF1	TBKPCMD1	CTVHBKP
M	Migration mission Definition Screen	CTDTBKP	CTDSBKP		TBKPPF1	TBKPCMD1	CTVHMIG
M	Restore Mission Definition Screen	CTDTRST	CTDSRST		TRSTPF1	TRSTCMD1	CTVHRST
M.O	Mission Order	CTMTJOB	CTDSMIM		TMIMPF1	TMIMCMD1	CTVHMIM
F	PC Packet Status	CTDTATF	CTDSATF		TATFPF1	TATFCMD1	CTVHATF
F.Z	Packet Zoom Screen	CTDTZTF	CTDSZTF		TDECPF1	TDECCMD1	CTVHZTF
F.I	Packet Index Screen	CTDTITF	CTDSITF		TITFPF1	TITFCMD1	CTVHITF
U	User Reports Entry Panel	CTDTUSR	CTDSABK	\$\$USR	TUSRPF1	TUSRCMD1	CTVHUSR
U	User Report Lists	CTDTUSR		\$\$FRM	TUSRPF2	TUSRCMD2	CTVHUS2
U.N	General Notepad Screen	CTDTNTP	CTDSNTP				
U.P	Print Report	CTDTUSR		\$\$FRM			CTVHUSI
U.V	Report Viewing	IOATOLV	CTDSOLV		TOLVPF1	TOLVCMD1	CTVHOLV
U.V	Tag Notepad Screen	CTDTNTP	CTDSNTP				

Appendix E: IOA Online Options Cross-Reference

Option	Screen Title	Module/ CSECT	Screen Member	Format Member	PFKey Member	Command Member	Help Member
U.E	Report Editing Screen	CTDTEXT	CTDSEXT		TOLVPF2	TOLVCMD1	CTVHEXT
U.E.1	Edit Report Lines and Columns	CTDTLNF	CTDSLNF		TOLVPF3	TOLVCMD1	CTVHLNF
U.E.1.C	Edit Report Columns	CTDTCLF	CTDSCLF		TOLVPF5	TOLVCMD1	CTVHCLF
U.E.2	Include Lines Based on Strings	CTDTINC	CTDSINC		TOLVPF4	TOLVCMD1	CTVHINC
U.E.3	Exclude Lines Based on Strings	CTDTEXC	CTDSEXC		TOLVPF4	TOLVCMD1	CTVHINC
U.E.4	Color Lines Based on Strings	CTDTCOL	CTDSCLR		TOLVPF4	TOLVCMD1	CTVHCLR
U.Q	Quick Access Panel	CTVTQAC		\$\$QAC	TQACPF1	TQACCMD1	CTVHQAC
U.Q.X	Values of Index	CTVTVOI		\$\$VOI	TVOIPF1	TVOICMD1	CTVHVOI
U.S.X	Values of Index	CTVTVOI		\$\$VOI	TVOIPF1	TVOICMD1	CTVHVOI
T	Recipient Tree Entry Panel	CTDTUTR	CTDSUTR		TUTRPF1	TUTRCMD1	CTVHUTR
T	CONTROL-D /V Recipient Tree	CTDTUTR	CTDSUTR		TUTRPF2	TUTRCMD1	CTVHUT2
T	CONTROL-D /V CHECK TREE Messages	CTMTJOB	CTDSTRM		TTRMPF1	TTRMCMD1	CTVHTRM
T.S	CONTROL-D /V Recipient Definition	CTDTRCP	CTDSRCP		TRCPPF1	TRCPCMD1	CTVHRCP

CONTROL-O

Option	Screen Title	Module/ GSECT	Screen Member	Format Member	PFKey Member	Command Member	Help Member
OR	Rule Definition Entry Panel	CTOTOMP	CTOSOMP		TOMPPF1	TOMPCMD1	CTOHOMP
OR	Tables/Rules of Library	CTOTOMP	CTOSDIR		TOMPPF1	TOMPCMD1	CTOHOM2
OR	Order/Force Messages	CTOTOMP	CTOSOMM		TOMPPF1	TOMPCMD1	CTOHOM3
OR	Rule Definition Screen	CTOTRUL	CTOSRUL		TRULPF1	TRULCMD1	CTOHRUL
OM	MSG Statistics	CTOTMSC	CTOSMSC		TMSCPF1	TMSCCMD1	CTOHMSC
OS	Rule Status	CTOTARF	CTOSARF	\$\$ARF	TARFPF1	TARFCMD1	CCTOHAR
OS	Error Messages	CTOTARM	CTOSARM		TARMPF1	TARMCMD1	CTOHARM
OS	Why (?)	CTOTOWY	CTOSOWY		TOWYPF1	TOWYCMD1	CTOHOWY
OL	Automation Log	CTOTALO		\$\$ALO	TALOPF1	TALOCMD1	IOAHALO
OA	Automation Options	CTOTAOP		\$\$AOP ##AOP*	TAOPPF1	TAOPCMD1	CTOHAOP
	COMMAND	CTOTMCS		\$\$AOPCMD	TAOPPF1	TAOPCMD1	CTOHAOP
	CONSOLE	CTOTCNS		\$\$AOPCNS	TAOPPF1	TAOPCMD1	CTOHAOP
	ENQINFO	CTOTGES		\$\$AOPGES	TAOPPF1	TAOPCMD1	CTOHAOP
	GLOBALS	CTOTGLB		\$\$AOPGLB	TAOPPF1	TAOPCMD1	CTOHAOP
	SUBSYS	CTOTSBS		\$\$AOPSBS	TAOPPF1	TAOPCMD1	CTOHAOP
	SERVERS	CTOTSRV		\$\$AOPSRV	TAOPPF1	TAOPCMD1	CTOHAOP
OA	OPERATOR Menu	CTOTAMN		\$\$AOPOPR ##OPER*	TAOPPF1	TAOPCMD1	CTOHAOP
OA	SAMPLE Menu	CTOTAMN		\$\$AOP ##SAMPLE*	TAOPPF1	TAOPCMD1	CTOHAOP
OA	SLIP Menu	CTOTAMN		\$\$AOPSLP ##SLIP*	TAOPPF1	TAOPCMD1	CTOHAOP
OA	Operator Command	CTOTMCS		\$\$AOPCMD	TAOPPF1	TAOPCMD1	CTOHAOP

* Data for the Format member.

CONTROL-B

Option	Screen Title	Module/ GSECT	Screen Member	Format Member	PFKey Member	Command Member	Help Member
BB	Balancing Status	CTBTABS	CTBSABS	\$\$ABS	TABSPF1	TABSCMD1	CTBHABS
BB.L	LOG messages	IOATLOG		\$\$LOG	TLOGPF1	TLOGCMD1	CTMHLOG
BB.?	WHY screen displayed for a rule.	CTBTBWY	CTBSBWY		TBWYPF1	TBWYCMD1	CTHBWY
BM	Mission Definition	CTBTBAL	CTBSBAM		TBALPF1	TBALCMD1	CTHBAL
BM	Mission List	CTBTBAL	CTBSBAD		TBALPF3	TBALCMD1	CTHBBA2
BM	Category List	CTBTBAL	CTBSBAD		TBALPF3	TBALCMD3	CTHBBA3
B.M.S	Balancing Mission Definition Screen	CTBTBMD	CTBSBMD		TBMDFPF1	TBMDCMD1	CTHBMD
BV	Database Variables Definition Facility	CTBTDAT	CTBSCH		TDATPF1	TDATCMD1	CTBHDTAT
BV	Database Facility - Group List	CTBTDAT	CTBSDAR		TDATPF3	TDATCMD1	CTBHDA2
BV	Database Facility - Variables List	CTBTDAT	CTBSDAR		TDATPF3	TDATCMD1	CTBHDA3
BV.G	Display Variable by Graph	IOATGRU		\$\$BVG	TGRPPF1	TGRPCMD1	IOAHGRU
BV.V	Display And Update Variables In Active File	CTBTDBV	CTBSDBV	\$\$DBV	TDBVPF1	TDBVCMD1	CTBHDBV
BV.S	CONTROL-B Variable Definition	CTBTMOD	CTBSMOD		TMODPF1	TMODCMD1	CTBHMOD
BR	Rule Definition - Entry Panel	CTBTRLB	CTBSRLB		TRLBPF1	TRLBCMD1	CTBHRLB
BR	Rule List	CTBTRLB	CTBSDIR		TRLBPF3	TRLBCMD1	CTBHRL2
BR.S	Rule Definition Screen	CTBTRUM	CTBSRUM		TRUMPF1	TRUMCMD1	CTBHRUM
BR.S	Rule Definition Compiler Messages Screen	CTMTJOB	CTBSJOB		TJOBPF1	TJOBCMD1	CTBHJOB
BA	Rule Activity Entry Panel	CTBTJBL	CTBSJBL		TJBLPF1	TJBLCMD1	CTBHJBL

Appendix E: IOA Online Options Cross-Reference

Option	Screen Title	Module/ CSECT	Screen Member	Format Member	PFKey Member	Command Member	Help Member
BA	Rule Activity	CTBTJBL	CTBSJAC	\$\$JAF	TJBLPF2	TJBLCMD1	CTBHJB2
BA	CONTROL-B List-Editing- Format Error Messages	CTMTJOB	CTBSJER		TJERPF1	TJERCMD1	CTBHJER
BA	CONTROL-B Report Viewing	CTMTJOB	CTBSJRP		TJRPPF1	TJRPCMD1	CTBHJRP
BA.G	Display Variable by Graph	IOATGRU		\$\$BJG	TGRPPF1	TGRPCMD1	IOAHGRU

CONTROL-T

Option	Screen Title	Module/ CSECT	Screen Member	Format Member	PFKey Member	Command Member	Help Member
TR	Rule Definition Entry Panel	CTTRLD	CTTSRLD		TRLDPF1	TRLDCMD1	CTTHRLD
TR	Rule Definition List of Tables	CTTRLD	CTTSDIR		TRLDPF3	TRLDCMD1	CTTHRL2
TR	Rule Definition List of Rules	CTTRLD	CTTSDIR		TRLDPF3	TRLDCMD1	CTTHRL3
TR	Rule Definition Screen	CTTRLM	CTTSRLM		TRLMPF1	TRLMCMD1	CTTHRLM
TP	Pool Definition Entry Panel	CTTPLD	CTTSPLD		TPLDPF1	TPLDCMD1	CTTHPLD
TP	Pool Definition List of Tables	CTTPLD	CTTSPDR		TPLDPF3	TPLDCMD1	CTTHPL2
TP	Pool Definition List of Pools	CTTPLD	CTTSPDR		TPLDPF3	TPLDCMD1	CTTHPL3
TP	Pool Definition Screen	CTTPLM	CTTSPLM		TPLMPF1	TPLMCMD1	CTTHPLM
TV	Vault Definition Entry Panel	CTTVLD	CTTSVLD		TVLDPF1	TVLDCMD1	CTTHVLD
TV	Vault Definition Table List	CTTVLD	CTTSVDR		TVLDPF3	TVLDCMD1	CTTHVL2
TV	Vault Definition Vaults List	CTTVLD	CTTSVDR		TVLDPF3	TVLDCMD1	CTTHVL3

Appendix E: IOA Online Options Cross-Reference

Option	Screen Title	Module/ CSECT	Screen Member	Format Member	PFKey Member	Command Member	Help Member
TV	Vault Definition Screen	CTTVLM	CTTSVLM		TVLMPF1	TVLMCMD1	CTTHVLM
TI	Inquire/ Update MDB Entry Panel	CTTTINQ		\$\$INQ	TINQPF1	TINQCMD1	CTTHINQ
TI	Inquire/ Update MDB Variables List	CTTTINQ		\$\$INS	TINQPF3	TINQCMD1	CTTHINS
TI	Inquire/ Update MDB Volume Update Screen	CTTTUPR		\$\$UPV	TUPRPF1	TUPRCMD1	CTTHUPR
TI	Inquire/ Update MDB Dataset Update Screen	CTTTUPR		\$\$UPD	TUPRPF1	TUPRCMD1	CTTHUPR
TC	Check In External Volume	CTTKIN		\$\$KIN	TKINPF1	TKINCMD1	CTTHKIN

Appendix E: IOA Online Options Cross-Reference

ISPF

Option	Screen Title	Module/ CSECT	Screen Member	Panel With Docutext/ Jobscan Member	Help Member
6	ISPF - UTILITIES	IOAIUTI			
	M		CTMPUTI	CTMPUTIJ	
	D		CTDPUTI		
	T		CTTPUTI		CTTPTH
	M+D		CTMPUTID	CTMPUTIX	
	D+T		CTTPUTID		CTTPTH
	M+T		CTTPUTIM	CTTPUTIN	CTTPTH
	M+D+T		CTTPUTIS	CTTPUTIX	CTTPTH
	M+B		CTMPUTI	CTMPUTIJ	
	D+B		CTDPUTI		
	B+T		CTTPUTI		CTTPTH
	M+D+B		CTMPUTID	CTMPUTIX	
	D+B+T		CTTPUTID		CTTPTH
	M+B+T		CTTPUTIM	CTTPUTIN	CTTPTH
	M+D+B+T		CTTPUTIS	CTTPUTIX	CTTPTH
D1	Decolating	CTDMISRQ			
D2	Printing	CTDMISRQ			
D3	Backup/Migration	CTDMISRQ			
D4	Restore	CTDMISRQ			
I1	Prerequisite Conditions	IOACCND			
M1	Condition Job Order Issue	CTMJOBRQ			
M2	AutoEdit	CTMCAES	CTMPROMP		
M3	Simulation	CTMCSIM			
M4	Simulation/Tape pull Parameter Prompting				
M5	Quick Schedule	CTMQUICK			
M6	User Interface	CTMJBINT			
T1	CONTROL-T Simulation	CTTCRSS			

Index

—#—

OF DAYS TO KEEP
Parameter
Backup Missions, 4-56

OF GENERATIONS TO
KEEP Parameter
Backup Missions, 4-56

~ Operator
INCLUDE/EXCLUDE
Statements, 9-102

—\$—

\$\$BANCHR Member
Banner Printing, 4-48

\$\$COMPST Member
Global Library
Compression, 5-14

\$\$FRM Member
Display Format Member
(CTD), 2-20

\$\$POOL Member
PARM Library (CTT), 7-5

\$\$USR Member
Display Format Member
(CTD), 2-20

\$\$VAULT Member
PARM Library (CTT), 7-5

\$FTEXT Exit
CA-ASM2 Support, 7-36

\$GLOBAL
Global Variable Member,
5-12

\$INDEX Records
CTVDELI Utility, 9-94

\$NTEXT CA-ASM2 Exit,
7-36

\$PROFFLD Member
Global Profile, 2-32

\$PROFMOD Member
Global Profile, 2-32

\$PROFVAR Member
Global Profile, 2-32

\$TAPPOOL CA-ASM2
Parameter, 7-36

—%—

%%\$AUTOLOG System
Variable, 5-11

%%\$STATID Reserved
Variable
Message ID (CTO), 5-10

%%STATUS Variable
Global Library
Compression, 5-14

%BKPUTIL% Parameter
Restore Skeleton, 4-65

%CATEGORY Parameter
Printing Mission, 4-33

%COM#% Parameter
Printing Mission, 4-33

%COND% Parameter
Backup Missions, 4-55
Migration Skeleton, 4-58

%COUNTP Parameter
CTTRPT Utility, 9-151

%DEST% Parameter
Printing Mission, 4-33

%DSNS% Parameter
Backup Mission, 4-55
Migration Skeleton, 4-58
Restore Skeleton, 4-64

%ENDREPEAT%
Parameter
Restore Skeleton, 4-64

%FIELD Parameter
CTTRPT Utility, 9-151

%GROUP% Parameter
Printing Mission, 4-33

%LEVEL% Parameter
Migration Missions, 4-59

%MISSION% Parameter
Printing Mission, 4-33

%MISSNAME% Parameter
Backup Mission, 4-55
Migration Skeleton, 4-58
Restore Skeleton, 4-64

%OWNER% Parameter
Printing Mission, 4-33

%PRTY% Parameter
Printing Mission, 4-33

%REPEAT% Parameter
Restore Skeleton, 4-64

%TIMESTMP% Parameter
Backup Mission, 4-55
Migration Skeleton, 4-58
Restore Skeleton, 4-64

%VOLUMES% Parameter
Restore Skeleton, 4-64

—&—

&ACSENVIR Parameter
ACS Routines, 7-38
DF/SMS Interface, 7-37

&TAPEMGT Parameter
CA-ASM2 Support, 7-36

—=—

= Operator
INCLUDE/EXCLUDE
Statements, 9-102

—@—

@DLM Line Type
Display Format Member,
2-24

@END Line Type
Display Format Member,
2-24

@FIELD Line Type
Color Parameters, 2-24
Display Format Member,
2-22

@HEADER Line Type
Color Parameters, 2-24
Display Format Member,
2-21

@LINE Line Type
Color Parameters, 2-24
Display Format Member,
2-22

@STYLE Line Type
Color Parameters, 2-24
Display Format Member,
2-20

@VAL Line Type
Display Format Member,
2-23

Index

—3—

3480/90/90E Tape
Cartridge

Stopping/Starting a
Device, 4-70

3490 Tape Cartridge

Utility CTTTPI, 9-182

—A—

Abend. *See also* Error

Backup Missions, 4-56

Migration Missions, 4-62

ABENDACT Parameter

Modification (CTT), 7-4

ABFBKP File

CTBABI Utility, 9-50

CTBFRM, 9-62

Access Method. *See* IOA

Access Method

ACCESS Parameter

CTTTPI Utility, 9-187

ACCOUNT Keyword

Dataset Record, D-2

ACCOUNT Parameter

FileTek Media, 4-68

ACIF Interface, 4-43

Parameter Member

Format, 4-45

ACIFPARM Library

ACIF Interface, 4-45

ACS Routines

Adjusting, 7-38

DF/SMS Interface, 7-37

ACT Parameter

CTTDBID Utility, 9-119

Utility CTBDBVIG, 9-61

Activating

CDAM, 4-2

CMEM Facility, 3-2

CONTROL-D Monitor,
4-1

CTB Missions, 6-5

Dynamic Dataset

Stacking, 7-41

Generic Processing CTD,
4-1

IOA Online Monitor, 2-5

VTAM Monitor, 2-7

Active Balancing File, 6-5

Formatting, 9-24

Initialization, 9-50

Reformatting, 6-2

Searching, 6-9

Utilities, 9-4

Active Jobs File

Changing Size, 9-47

Cleaning, 9-47

Compressing, 9-47

CONTROL-M, 3-4

Dual Checkpoint Mode,
3-26

Expanding (CTM), 3-35

Restoration, 3-26

Active Library (CTT)

Adding Volumes, 9-122

Active Mission Screen

Profile Variables, 2-36

Active Missions File

Changing Size, 9-68

Cleaning, 4-7

Compressing, 9-68

Expanding, 4-81

New Day Procedure

(CTD), 4-17

Active Report List

Backup Mission, 4-54

Copy from Permanent,
9-77

Copying, 9-72

CTDDELRP Utility, 4-78

Deleting Reports, 9-79

Description, 4-76

Dynamic Sorting, 4-83

Migration Mission, 4-57

Active Transfer File

Changing Size, 9-70

Compressing, 9-70

Expanding, 4-81

Active User File

Backup Utility, 9-87

Active User Report List

Copy to Permanent, 4-76

Copy to/from History,
4-78

Copy Utility, 4-78

Permanent User Report
List, 4-75

Sorting, 4-82

ACTIVEDS Keyword

Volume Record, D-6

ACTIVEDS Parameter

CTTMUP Utility, 9-137

ADBPFUNC Variable

API Error Handling, 7-71

ADBPIOPR Variable

CTTIOS Function, 7-71

ADBPIRC Variable

API Error Handling, 7-71

ADBPIRSN Variable

API Reason Code, 7-71

ADBPRC Variable

API Return Code, 7-70

ADD Parameter

CTTTPI Utility, 9-186

Adding

MDB Volume, 9-132

Volumes to MDB, 9-122

ADDMODE Statement

IOALDNRS Utility, 9-36

Address Space Vector

Table. *See* ASVT

ADDVOL Function

High Level API, 7-68,
7-69

Ad-hoc Maintenance

IOA Products, 8-4

Advanced Function

Printing. *See* AFP

AFP Printing

Commands, 4-41

Page Markers, 4-40

Page Mode Output, 4-39

Printing Missions, 4-38

AFPDS

Description, 4-43

Allocating. *See also*

Creating, and Formatting

CTT Files, 9-111

Trace Files (CTT), 9-114

Allocation Members

Customization, 2-11

ALREC# Parameter

Automation Log Size,
5-11

Alternate Allocation, 2-12

AMFSIZE Parameter

Active Missions File,
4-81, 9-68

- ANALYZE Step
 Migration Mission, 4-58
 Migration Skeletons, 4-60
 Restore Missions, 4-65
- AOP. *See* Automation Options
- APA Printing. *See also* AFP Printing
- APA Technology
 Laser Printing, 4-38
- APAPARM Library
 AFP Parameters, 4-40
 Format, 4-41
 Sample Member, 4-41
- APF Authorization
 CTTGVL Utility, 9-125
 High Level API, 7-68
- API. *See also* High Level API. *See also* Base Level API
- Base Level/High Level, 7-46
 CONTROL-T Interface, 7-44
- API (CTT)
 High Level, 7-68
- Application Program
 Invoking CONTROL-B, 6-7
- Application Programming Interface. *See* API
- APPLTYPE Parameter
 IOAONL CLIST, 2-2
- APPLY Parameter
 CTTMUP Utility, 9-132
- ARCCPUID Keyword
 Trace Record, D-11
- ARCDATE Keyword
 Trace Record, D-11
- Archive Server, 4-66
 DD Statements, 4-73
 Deactivating, 4-5
 Problem Determination, 4-73
- Archiving
 External Resources (ACIF), 4-44
- ARCIDENT Keyword
 Trace Record, D-11
- ARCJOBID Keyword
 Trace Record, D-11
- ARCS
 Backup Utility, 4-55
- ARCTIME Keyword
 Trace Record, D-11
- ARCTVEXT Exit
 HSM Tape Volume Exit, 7-35
- ARCTYPE Keyword
 Trace Record, D-11
- ARCUSRID Keyword
 Trace Record, D-11
- AS2FTEXT Member
 CA-ASM2 Support, 7-36
- ASM2
 Backup Utility, 4-55
- ASM3495 Member
 Interface to IBM Robot, 7-26
- ASMCTLEX Member
 DMS/OS Support, 7-35
- ASMMMRX Member
 MEMOREX TELEX Support, 7-32
- ASMRSTK Member
 StorageTek Robot Support, 7-29
- ASMTVEXT Member
 HSM Support (CTT), 7-35
- ASMUXCUA Member
 IBM Robot Support, 7-26
- ASMUXEJC Member
 IBM Robot Support, 7-27
- ASMUXVNL Member
 IBM Robot Support, 7-27
- ASVT
 MVS Table, 2-6; 4-4
- ATFSIZE Parameter
 Active Missions File, 4-81
 CTDFRATF Procedure, 9-70
 CTDPARM Member, 9-70
- ATTR Statement
 CTTRPT Utility, 9-148
 CTTRTM Utility, 9-170
 CTTVTM Utility, 9-198
 Authorization. *See* Security
- AutoEdit Facility
 Description, 1-12
 Global Variables (CTO), 5-12
- AUTOLOG Command (CTO), 5-11
- Automatic Class Selection
 DF/SMS Interface, 7-37
- Automation Log
 Backup Utility, 9-100
 Changing Size, 9-100
 Description, 5-10
 Search Limit, 2-42
 Size, 5-11
 Unnecessary Messages, 5-11
- Automation Options
 Client Programs, 5-24
 Format Members, 5-23
 Menus, 5-19
 Overview, 5-18
- AUTOMLOG Parameter (CTO), 5-10
- B—
- Backup
 Automation Log, 5-11, 9-100
 MDB and Trace File, 9-192
 Media Database, 7-9
 Retention and Volser (CTD), 9-87
- Backup and Recovery
 Repository (CTT), 7-23
- Backup File (CTB)
 Formatting, 9-24
- BACKUP IN PROCESS Status
 Backup Mission, 4-56
- Backup Mission
 Abend, 4-56
 Advanced Scheduling, 4-52
 Decollation Missions, 4-52
 Default Missions, 4-15
 Exception Handling, 4-56
 Overview, 4-51
 Profile Variable, 2-36

Index

- Retention Period, 4-56
- Scheduling, 4-13
- Workflow, 4-54
- Balance Workload
 - IOA Online Monitors, 2-5
- Balancing Activities
 - Invoking, 6-7
- Balancing Mission (CTB)
 - Invoking, 6-9
 - Scheduling, 6-2
- Banner
 - Printing Characteristics, 4-48
- Bar-code
 - AFPDS Printing, 4-43
- Base Level API (CTT)
 - Examples, 7-53
 - Functions, 7-47
 - High Level API, 7-46
 - Macro CTTIOS, 7-47
 - Media Database Access, 7-51
 - Read Entire MDB, 7-53
 - Record Access, 7-47
- Basic Tape Library Support. *See* BTLS
- BATCH Parameter
 - Printing Mission, 4-32
- BKP Parameter
 - CTTTRB Utility, 9-192
- BKPLIST Member
 - Backup Missions, 4-14
 - Migration Missions, 4-14
- BKPMIS Parameter
 - Utility CTDUPBKP, 9-88
- BKPPORDER KSL Utility
 - Backup Missions, 4-16
- BKPRESET Job
 - Backup Mission Rerun, 4-56
- BKPTO Parameter
 - Utility CTDUPBKP, 9-88
- BLK Parameter
 - CTTCRDF Utility, 9-112
- BLKSIZE Keyword
 - Dataset Record, D-2
- BLKSIZE Parameter
 - Exit CTDX005, 4-51
 - IOADBF Utility, 9-21
- BLOCKCT Keyword
 - Dataset Record, D-2
- BLPDEF Parameter
 - Modification (CTT), 7-4
- Bosch
 - Tape Library Support, 7-32
- BOXID Keyword
 - Volume Record, D-6
- BOXLIM Parameter
 - CTTVTM Utility, 9-195
- BREAK Statement
 - CTTRPT Utility, 9-150
 - CTTRTM Utility, 9-170
 - CTTVTM Utility, 9-198
- Browse Mode
 - Profile Variable, 2-40
- BTLS Support, 7-27
- BTLSDIGS Parameter
 - IBM BTLS Support, 7-27
- BTLSIN DD Statement
 - BTLS Support, 7-28
- BTLROUT DD Statement
 - BTLS Support, 7-28
- BTLUNIT Parameter
 - BTLS Library Unit, 7-27
- Buffer Address
 - High Level API, 7-69
- BUFL Parameter
 - IOADBF Utility, 9-22
- Building
 - Manual Conditions File, 9-35
 - Stacking Statistics File, 9-177
- Bundle Printing (CTD), 4-34
- C—
- C Parameter
 - Utility CTBJAFIG, 9-67
- CA-1
 - Compatibility, 9-164
 - Conversion of Retention Information, 9-168
 - Conversion Utility, 9-123
- CA-7 (UCC7) Interface, 4-28
- CA-ASM2 Support, 7-36
- Pool Definition, 7-36
- Calendar. *See also* Scheduling Facility
 - Building, 9-1
 - Tables, 1-9
- CALL Command
 - CTTGVL Utility, 9-125
- CA-ROSCOE. *See* ROSCOE
- CART Media
 - Media Definition, 4-67
 - Migration Target, 4-60
 - Stopping/Starting a Device, 4-70
- Cartridge Message Display
 - Customization, 7-33
- Cartridge to Disk
 - Migration Skeleton, 4-60
- Cataloging
 - Datasets (Utility CTTRCV), 9-141
- CATEGORY 5
 - AFP Printing, 4-39
- Category Field
 - Job Scheduling, 4-26
- CA-TLMS
 - Compatibility, 9-164
 - Conversion of Retention Information, 9-168
 - Conversion Utility, 9-123
- CBRUXCUA User Exit
 - IBM Robot, 7-26
- CBRUXEJC User Exit
 - IBM Robot, 7-26
- CBRUXENT User Exit
 - IBM Robot, 7-27
- CBRUXVNL User Exit
 - IBM Robot, 7-27
- CDAM
 - Activating, 4-2
 - AFP Printing, 4-39
 - Deactivating, 4-5
 - Deleting Files, 9-74
 - Parameters, 4-39
- CDS Backups
 - HSM Control, 7-35
- CHANGETO Parameters
 - CTDUPBKP Utility, 9-89
- CHECK Mode
 - CTTINIT Procedure, 7-4

- CHECK Parameter
 - CTTMUP Utility, 9-132
- CHECKACT Parameter
 - CTTRCV Utility, 9-141
- Checking. *See* Verification
- CHKINDT Keyword
 - Volume Record, D-6
- Chunk Printing. *See*
 - One-Chunk Method or Multi-Chunk Method
- CHUNKSIZE Parameter
 - Bundle Printing, 4-35
 - Bundle Printing (CTD), 4-35
- CICS
 - IOA Online Support, 2-1
 - Memory Requirement, 2-4
- CKPSIZE Parameter
 - Active Jobs File Size, 9-47
- CLASS JES Parameter
 - Multi-Chunk Printing, 4-35
- CLASS Parameter
 - @STYLE Display Format Line, 2-21
- CLASSLIKE Parameter
 - @STYLE Display Format Line, 2-21
- Cleaning
 - Active Jobs File, 9-47
 - Conditions/Resources File, 9-7
 - History User Report List, 9-76
 - Migrated Report List, 9-93
- CLEANUP Function
 - CTMCAJF Utility, 9-45
- Client Programs
 - Automation Options, 5-24
- CLIST
 - Entry to Online Facility, 2-2
 - Mission Scheduling (CTD/V), 4-16
- CLNCOUNT Keyword
 - Volume Record, D-6
- Clock
 - Changing Time, 3-23
- CLOSE Function
 - Base Level API (CTT), 7-47
- CMD Parameter
 - IOATEST Utility, 9-39
- CMEM Facility
 - Activation, 3-2
 - CONTROL-O Installed, 5-1
 - CTO Installed, 5-25
 - Deactivating a Rule Table, 3-28
 - Debugging, 3-30
 - Description, 3-26
 - Display Active Rules, 3-29
 - Multiple Rule Tables, 5-7
 - Profile Variable, 2-43
 - Replacing a Monitor, 3-3
 - Replacing Rule Tables, 3-28
 - Resource Utilization, 3-32
 - Rule Loading, 3-27, 5-3
 - Rule Operation Mode, 3-29
 - Security Cache, 3-30
 - Shutting Down, 3-3
 - Sleeping Interval, 3-30
- CMORDER Parameter
 - CONTROL-O Procedure, 5-3
 - Rule Lists (CMEM), 5-7
- CNGMSGID Parameter
 - Modification (CTT), 7-4
- CODE Parameter
 - CTMMMSG Macro, 2-47
- COLOR Parameter
 - Display Format Member, 2-25
- Color Support
 - COM-PLETE, 2-26
 - Customization, 2-26
 - Display Format Member, 2-24
 - Extended, 2-25
 - IDMS/DC, 2-25
 - IMS/DC, 2-25
 - IRMA, 2-26
 - ISPF, 2-25
 - Profile Variables, 2-39
- COLORA Parameter
 - Display Format Member, 2-25
- COLUMN Parameter
 - CTTRPT Utility, 9-151
- Command Members
 - Format, 2-13
 - Modification, 2-13
 - Online Facility, E-1
- Commands
 - Customization, C-1
 - Online Facility, 2-13
- COMMNDnn Member, 5-6
- Communication Monitors (CTM), 3-35
- Communications File
 - Inter-System Communications (CTM), 3-32
- COMPAREX Robots
 - CONTROL-T Support, 7-25
- Compiling
 - Rule Definitions (CTB), 9-51
- COM-PLETE
 - Color Support, 2-26
 - IOA Online Support, 2-1
- COMPRESS Function
 - CTMCAJF Utility, 9-45
- Compressed Dataset Access Method. *See* CDAM
- Compressing
 - Active Jobs File, 9-47
 - Active Missions File, 9-68
 - Active Transfer File, 9-70
 - Global AutoEdit Library, 5-14
- COMPRS Parameter
 - IOADBF Utility, 9-22
- COMPTB Parameter
 - IOADBF Utility, 9-22
- Condition. *See* Prerequisite Condition
- Conditions Selection List (CTM), 3-33
- Conditions/Resources File
 - Backup Missions, 4-55
 - Cleaning Utility, 9-7
 - Copying, 9-14

Index

- Description, 1-8
- Expanding, 2-49
- IOACND Utility, 9-10
- Conditions/Resources Screen
 - Profile Variable, 2-38
- CONID Parameter
 - IOATEST Utility, 9-39
- Constant Blocks
 - IOA Screen Definition, 2-18
- Constants
 - IOA Screens, 2-17
- CONTDAY Procedure
 - Customization (CTM), 3-10
 - DAREPMIS DD Statement, 4-24
 - New Day Processing (CTM), 3-6
- CONTROL D/PC
 - AFP Viewing Component, 4-44
- CONTROL Products. *See* IOA or specific product
- Control Resources
 - Adding/Deleting, 9-10
 - Description, 1-8
- Control Statements
 - AFP, 4-40
 - CTTMUP Utility, 9-132
- Control Table (CTT), 7-45
 - Loading, 7-47, 7-54
- Control Table (TCT)
 - CONTROL-T
 - Initialization, 7-2
- CONTROL-B
 - Activating Missions, 6-5
 - Compile Rule Definitions, 9-51
 - Database Integrity, 9-61
 - Deleting Variables, 9-59
 - Initialize Active Balancing File, 9-50
 - Invocation Utilities, 9-4
 - Invoking via
 - CONTROL-D, 6-8
 - CONTROL-M, 6-8
 - IOA Access Method, 2-29
 - Job Step Invocation, 6-8
 - Online Facility Members, E-9
 - Rebuild Index, 9-52
 - Report Utilities, 9-4
 - Repository, 1-11
 - Rule Activity File Integrity, 9-67
 - Runtime Environment, 6-6
 - Utility Summary, 9-4
 - Variable Generations, 9-54
 - Variable Group Utilities, 9-4
- CONTROL-D
 - Bundle Printing, 4-34
 - Chunks on the Spool, 4-38
 - CONTROL-M, 4-23
 - Daily Maintenance, 4-7
 - Date Control Record, 4-7
 - Generic Processing, 4-1
 - Initialization, 4-1
 - Invoking CONTROL-B, 6-8
 - IOA Access Method, 2-29
 - Monitor Activation, 4-1
 - New Day Processing, 4-10
 - Number of Users, 2-5
 - Online Facility Members, E-3
 - Online Viewing, 2-5
 - Repository, 1-10
 - Sleeping Interval, 4-2
 - SMF Accounting, 4-84
 - User Daily, 4-7
 - User Groups, 4-22
 - Utilities, 9-68
 - Utility Summary, 9-2
- CONTROLD OUTPUT Statement
 - AFP Printing, 4-39
- CONTROLD Statement
 - IOACLRES Utility, 9-8
 - IOALDNRS Utility, 9-36
- CONTROLF OUTPUT Statement
 - AFP Printing (CTD), 4-39
- CONTROL-M
 - Active Jobs File, 3-4, 9-45
 - Administration, 3-1
 - CONTROL-D, 4-23
 - CONTROL-O, 5-25
 - Dual Checkpoint Mode, 3-26
 - File Expansion, 3-35
 - Initialization, 3-1
 - Inter System Communications, 3-32
 - Invoking CONTROL-B, 6-8
 - Job Ordering, 3-4
 - Maintenance Jobs, 3-6
 - Monitor, 3-1
 - Monitor Shutdown, 3-1
 - New Day
 - Implementation, 3-10
 - New Day Processing, 3-4
 - Online Facility Members, E-2
 - Repository, 1-9
 - Rerun Backup Mission (CTD), 4-56
 - Rerun Migration Mission (CTV), 4-62
 - Rerun Restore Mission (CTD), 4-65
 - Scheduling Library, 4-24
 - Sleeping Interval, 3-2
 - User Daily Job, 3-4
 - Utility Summary, 9-2
- CONTROL-M Event Manager. *See* CMEM Facility
- CONTROLM Statement
 - IOALDNRS Utility, 9-36
- CONTROL-O
 - Administration, 5-2
 - Automation Log, 5-10
 - CMEM, 5-25
 - CMEM Control, 5-1
 - CONTROL-M, 5-25
 - Deactivating a Rule Table, 5-5
 - Debugging, 5-16
 - Internal Data Areas, 5-17
 - IPL Automation, 5-5
 - IPL Process, 5-8
 - Message Statistics, 5-9
 - Multiple Rule Tables, 5-7

- Online Facility Members, E-8
- Replacing a Monitor, 5-2
- Repository, 1-10
- Resource Utilization, 5-18
- Security Cache, 5-16
- Shutting Down, 5-1
- Sleeping Interval, 5-15
- Starting, 5-1
- Utilities, 9-98
- Utility Summary, 9-3
- CONTROL-R
 - CTT Interface, 7-43
- CONTROL-T
 - API Interface, 7-44
 - Backup and Recovery, 7-23
 - Control Table, 7-2, 7-45
 - CTR Interface, 7-43
 - Database Printing, 9-121
 - Dormant Mode, 7-4
 - EDM Support, 7-35
 - Initialization, 7-2, 7-3
 - Media Database Integrity, 7-19
 - MVS Interface Parameters (CTT), 7-4
 - MVS Interfaces, 7-2
 - New Day Procedure, 7-7
 - Normal Mode, 7-5
 - Online Facility Members, E-10
 - Operating Status, 7-4
 - Operation Mode, 7-8
 - Overview, 7-1
 - Real-time Environment, 7-1
 - Repository, 1-11
 - Repository Field Names, D-1
 - Stacking Facility, 7-39
 - Suspended Mode, 7-5
 - SVC Number, 7-2
 - Termination, 7-3, 7-6
 - Utilities, 9-102
 - Utility Summary, 9-5
- CONTROL-V. *See also*
- CONTROL-D
 - Customization, 4-1
 - Initialization, 4-1
 - Migration Mission, 4-56
 - Online Facility Members, E-5
 - Reports, 4-77
- CONVABF Job
 - IOADBF Utility, 9-23
- CONVC4MD Job
 - CTTDLD Utility, 9-123
- CONVC5MD Job
 - CTTDLD Utility, 9-123
- Converting
 - Volume Records, 9-122
- CONVGRP Job
 - IOADBF Utility, 9-23
- CONVJAF Job
 - IOADBF Utility, 9-23
- CONVLCTL Job
 - CTTDLD Utility, 9-123
- CONVMOD Job
 - IOADBF Utility, 9-23
- CONVREP Job
 - IOADBF Utility, 9-24
- CONVRMDB Job
 - CTTDLD Utility, 9-123
- CONVTMDB Job
 - CTTDLD Utility, 9-122
- Copy Count
 - Active User Report List, 4-76
 - Report List Files, 4-75
- COPY Function
 - CTMCAJF Utility, 9-45
- COPY Parameter
 - CTBDBVCG Utility, 9-54
 - CTBDBVCP Utility, 9-56
- COPY Statement
 - CTTACP Utility, 9-106
- Copying
 - Active Report List, 9-72
 - Conditions/Resources File, 9-14
 - Database Records, 9-172
 - IOA Log File, 9-16
 - Message Statistics File, 9-98
 - Permanent User Report List (CTD), 4-76
 - Rule Activity File, 9-63
 - Trace File (CTT), 9-106
- Variable Files (CTB), 9-56
- COPYTOLOG Parameter
 - Utility IOACPLOG, 9-17
- COPYTOSEQ Parameter
 - Utility IOACPLOG, 9-16
- COUNT Parameter
 - CTTRPT Utility, 9-151
- CPUID Keyword
 - Trace Record, D-11
- Creating
 - Trace Files, 9-114
- CRECC Keyword
 - Dataset Record, D-2
- CRECPU Keyword
 - Dataset Record, D-2
- CREDDN Keyword
 - Dataset Record, D-2
- CREDIT Keyword
 - Dataset Record, D-2
- CREFABF Job
 - IOADBF Utility, 9-23
 - Utility CTBABI, 9-50
- CREFBKP Job
 - IOADBF Utility, 9-23
 - Utility CTBABI, 9-50
- CREFDBM Job
 - IOADBF Utility, 9-23
- CREFDBV Job
 - IOADBF Utility, 9-23
- CREFGRP Job
 - IOADBF Utility, 9-23
- CREFJAF Job
 - IOADBF Utility, 9-23
- REFREP Job
 - IOADBF Utility, 9-23
- CREJBN Keyword
 - Dataset Record, D-2
- CREJOBID Keyword
 - Dataset Record, D-2
- CRESTEP Keyword
 - Dataset Record, D-2
- CRETM Keyword
 - Dataset Record, D-2
- CREUAD Keyword
 - Dataset Record, D-2
- CREUSER Keyword
 - Dataset Record, D-2

Index

- Cross Memory Interfaces
 - Online Facility, 2-1
- Cross-Product Interfaces
 - CTM and CTD, 4-27
 - CTT and CTR, 7-43
- CRPEGM Keyword
 - Dataset Record, D-2
- CSC Support
 - StorageTek Robot, 7-29
- CSECT List
 - Online Facility, E-1
- CTBABI Utility
 - Active Balancing File, 9-50
- CTBBAO Program
 - New Day Procedure (CTB), 6-1
- CTBCHK Program
 - New Day Procedure (CTB), 6-1
- CTBCMP Utility
 - Compile Rules, 9-51
- CTBDBIB Utility
 - Rebuild CTB Index, 9-52
- CTBDBVCG Utility
 - Variable Generations, 9-54
- CTBDBVCP Utility
 - Copy CTB Variables, 9-56
- CTBDBVDL Utility
 - Deleting Variables, 9-59
- CTBDBVIG Utility
 - Variable File Integrity, 9-61
- CTBFRM Utility
 - Active Balancing File, 6-2
 - New Day Procedure (CTB), 6-1
- CTBJAFCP Utility
 - Rule Activity File, 9-63
- CTBJAFDL Utility
 - Rule Activity File, 9-65
- CTBJAFIG Utility
 - Rule Activity File Integrity, 9-67
- CTBNDAY
 - New Day Procedure, 6-1
- CTBPDA Program
 - New Day Procedure (CTB), 6-1
- CTD34F Program
 - CTD Daily, 4-12
- CTDAMUTI Utility, 9-2
- CTDDBRQ Program
 - CTD Daily, 4-11
- CTDCA2P Utility
 - Copy User Report List, 9-72
 - Description, 4-76
- CTDCAMF Utility
 - Compress Active Missions File, 9-68
- CTDCATF Utility
 - Compress Active Transfer File, 9-70
 - CTD Daily, 4-11
- CTDCCU Utility
 - Deleting CDAM Files, 9-74
- CTDCHK Program, 4-8
 - CTD Daily, 4-11
- CTDCLHIS Utility
 - Clean History Report List, 9-76
 - Description, 4-79
- CTDCP2A Utility
 - Copy User Report List, 9-77
 - Description, 4-76
- CTDDELRP Utility
 - Deleting Reports, 9-79
 - Description, 4-78
- CTDDIB Utility
 - Access Method Index, 9-86
- CTDFRAMF Procedure
 - Active Missions File, 9-68
- CTDFRATF Procedure
 - ATFSIZE Parameter, 9-70
- CTDFRM Program
 - CTD Daily, 4-11
- CTDGRQ Program
 - CTD Daily, 4-11
- CTDILY Program
 - DD Statements, 4-15
- CTDMISRQ CLIST
 - Mission Scheduling, 4-16
- CTDNDAY Procedure
 - Mission Scheduling, 4-13
 - New Day Processing, 4-6
- CTDNDAY Started Task
 - Abend, 4-5
- CTDPDA Program, 4-9
 - CTD Daily, 4-12
- CTDPRINT Procedure
 - Printers Control Monitor, 4-38
- CTDPRQ Program
 - CTD Daily, 4-11
- CTDRRQ Program
 - CTD Daily, 4-12
 - DD Statements, 4-28
 - Production Control System, 4-28
- CTDSMFRP Utility, 9-3
- CTDSRQ Program
 - CTD Daily, 4-11
- CTDUDJP Macro
 - DJDE Printing, 4-43
- CTDUPBKP Utility
 - Backup Retention and Volser Info, 9-87
 - Dual Migration, 4-59
- CTDX001 Exit
 - Mission Scheduling, 4-17
 - Production Control Systems, 4-28
 - Scheduling with CTM, 4-23
- CTDX005 Exit
 - Customization, 4-49
 - Printing to a File, 4-49
- CTDX009 Exit
 - Printing Missions, 4-33
- CTDX010 Exit
 - Backup Missions, 4-55
 - Migration Missions, 4-58
- CTDX011 Exit
 - Restore Missions, 4-65
- CTMAESIM Procedure, 3-25
- CTMAJO Routine
 - Printing Missions (CTD), 4-33
- CTMBGRP Utility
 - Group Scheduling Tables, 9-43
- CTMCAJF Utility
 - Active Jobs File, 9-45

- Restoring the Active Jobs File, 3-26
- CTMCHK Program, 3-17
 - New Day Processing, 3-16
- CTMCJOBS CLIST, 4-24
- CTMCLRES Utility, 3-6, 3-11
- CTMCMEM Operator Command, 3-3
- CTMDAILY Procedure, 3-16
 - DAREPMIS DD Statement, 4-24
- CTMDAS Program
 - New Day Processing, 3-17
- CTMEDA Program
 - New Day Processing, 3-17
- CTMFRCKP Utility
 - Active Jobs File Size, 9-47
 - Restoring the Active Jobs File, 3-26
- CTMFRM Program, 3-10
 - New Day Processing, 3-16
- CTMILU Program
 - New Day Processing, 3-16
- CTMILY Program
 - New Day Processing, 3-16
- CTMISCA Member, 3-34
- CTMISCB Member, 3-34
- CTMISR Routine
 - Inter-System Communications (CTM), 3-35
- CTMJOB Program
 - Daily Processing (CTM), 3-18
 - New Day Processing, 3-17
- CTMJSA Utility
 - Job Statistics Accumulation, 9-49
- CTMLDNRS Maintenance Job, 3-6
- CTMLEVEL Macro, 2-47
- CTMMMSG Macro, 2-47
- CTMPDA Program
 - Daily Processing (CTM), 3-18
 - New Day Processing, 3-17
- CTMSBLK Macro
 - IOA Screen Definition, 2-17
- CTMSEND Macro
 - IOA Screen Definition, 2-17
- CTMSFLD Macro
 - Format, 2-18
 - IOA Screen Definition, 2-17
- CTMSSTA Macro
 - IOA Screen Definition, 2-17
- CTMTAPUL Procedure, 3-25
- CTOALOCP Utility
 - Back up Automation Log, 5-11, 9-100
- CTOCSF Utility
 - Copy Statistics File, 9-98
 - Statistics File Size (CTO), 5-10
- CTOGSFJ Member
 - Message Statistics, 9-98
- CTORSTM Utility
 - Message Statistics Report, 9-99
- CTR001 Exit
 - Interface to CTT, 7-43
- CTT3495 Member
 - Interface to IBM Robot, 7-26
- CTTACCDB Macro
 - High Level API, 7-68
 - Return Codes, 7-70
 - Work Area, 7-69
- CTTACP Utility
 - Copy Trace File, 9-106
 - Trace File Size, 7-23
- CTTADBP Macro
 - Call From CTTACCDB, 7-69
 - Variables, 7-70
- CTTAFR Utility
 - Format/Initialize Trace File, 9-108
- CTTARC Macro
 - Trace File Mapping, 7-17
- CTTBIX Utility
 - Media, 7-20
 - Rebuild Indices (CTT), 7-24
 - Rebuild MDB Index, 9-109
- CTTCHKDB Macro
 - High Level API, 7-70
- CTTCMDB Job
 - Utility CTTCRDF, 9-113
- CTTCRDF Utility
 - Allocate/Format CTT Files, 9-111
- CTTCSTK Job
 - Utility CTTCRDF, 9-113
- CTTCTRC Utility
 - Allocate/Format Trace File, 9-114
- CTTDAY Procedure
 - New Day Procedure (CTT), 7-7
- CTTDBDLD Utility
 - Load MDB and STK Files, 9-115
- CTTDBDUL Utility
 - Loading Files, 9-115
 - Unload MDB and STK Files, 9-116
 - Unloading Files, 9-116
- CTTDBF Utility
 - Format an Existing File, 9-117
- CTTDBIB Utility
 - Rebuild Stacking Statistics Index, 9-118
- CTTDBID Utility
 - Check Data File Integrity, 9-119
 - Rebuild Indices (CTT), 7-24
- CTTDBII Utility
 - Index File Integrity, 9-120
- CTTDBPRT Utility
 - Print Database Records, 9-121
- CTTDBTP Macro
 - Call From CTTACCDB, 7-69
- CTTDDX Macro
 - Index Mapping, 7-13
- CTTDLD Utility
 - Add/Delete/Convert Volumes, 9-122
- CTTDLX Macro
 - Index Mapping, 7-13

Index

- CTTDVX Macro
 - Index Mapping, 7-13
- CTTFMDB Job
 - CTTDBF Utility, 9-117
- CTTFSTK Job
 - CTTDBF Utility, 9-117
- CTTGVL Utility
 - Stacking Interface, 9-125
- CTTIDB Utility
 - Common Integrity Errors, 7-20
 - MDB Integrity, 7-19, 9-127
 - New Day Procedure (CTT), 7-9
- CTTINIT Procedure
 - CTTPARM Modifications, 7-4
 - Description, 7-2
 - Parameters, 7-2
 - Starting CONTROL-T, 7-2
- CTTIOS Macro
 - API Access of MDB, 7-47
 - Format, 7-48
 - Invocation Form, 7-50
 - Reason Codes, 7-49, 7-71
 - Return Codes, 7-51, 7-71
- CTTMER Utility
 - Merge Records, 9-129
- CTTMMRX Member
 - MEMOREX TELEX Support, 7-32
- CTTMUP Utility, 9-131
 - Examples, 9-138
 - Field Names, D-1
 - Field/Value Specification, 9-136
 - Manual Update of the MDB, 7-21
 - Media Database Integrity, 7-20
 - Return Codes, 9-137
- CTTPARM Member
 - Modification via CTTINIT, 7-4
- CTTRBM Step
 - CTTRTM/CTTVTM Procedure, 7-28
- CTTRBXLC Report
 - CTTRPT Utility, 9-155
- CTTRCHK1 Report
 - CTTRPT Utility, 9-154
- CTTRCLN Report
 - CTTRPT Utility, 9-154
- CTTRCV Utility
 - Database Recovery, 7-24
 - Examples, 9-142
 - MDB Recovery, 9-139
 - Return Codes, 9-142
 - Selective Recovery, 7-25
- CTTRDCRE Report
 - CTTRPT Utility, 9-154
- CTTRDELV Report
 - CTTRPT Utility, 9-154
- CTTRDUPD Report
 - CTTRPT Utility, 9-155
- CTTREDM Report
 - CTTRPT Utility, 9-154
- CTTRFSCR Report
 - CTTRPT Utility, 9-154
- CTTRINV Report
 - CTTRPT Utility, 9-154
- CTTRINVT Report
 - CTTRPT Utility, 9-154
- CTTRLACV Report
 - CTTRPT Utility, 9-154
- CTTRLAVJ Report
 - CTTRPT Utility, 9-154
- CTTRLRTV Report
 - CTTRPT Utility, 9-154
- CTTRM10V Report
 - CTTRPT Utility, 9-154
- CTTRM25D Report
 - CTTRPT Utility, 9-155
- CTTRMACV Report
 - CTTRPT Utility, 9-154
- CTTRMOVW Report
 - CTTRPT Utility, 9-154
- CTTRNRTR Report
 - CTTRPT Utility, 9-155
- CTTROUTL Report
 - CTTRPT Utility, 9-154
- CTTRPT Utility
 - Report Generation, 9-143
 - Return Codes, 9-153
 - Sample Reports, 9-154
- CTTRRCLV Report
 - CTTRPT Utility, 9-154
- CTTRRTNP Report
 - CTTRPT Utility, 9-154
- CTTRSCRD Report
 - CTTRPT Utility, 9-154
- CTTRSTK Member
 - StorageTek Robot Support, 7-29
- CTTRTM Utility
 - Condition Codes, 9-170
 - Examples, 9-171
 - New Day Procedure (CTT), 7-8
 - Parameters, 9-166
 - Retention Management, 9-162
- CTTRUTIL Report
 - CTTRPT Utility, 9-154
- CTTRVBOX Report
 - CTTRPT Utility, 9-155
- CTTRVUPD Report
 - CTTRPT Utility, 9-155
- CTTRWDCP Report
 - CTTRPT Utility, 9-154
- CTTSPL Utility
 - CTTMER Utility, 9-129
 - Extract MDB Info, 9-172
- CTTSTK Utility
 - Build/Update STK File, 9-177
 - New Day Procedure (CTT), 7-8
 - Trace Mode, 7-41
- CTTTCT Macro
 - Control Table, 7-45
- CTTTLD Routine
 - Load the Control Table, 7-45
- CTTTPI Utility
 - Activation, 9-182
 - Examples, 9-191
 - Invoking, 9-190
 - Return Codes, 9-191
 - StorageTek Robot Support, 7-30
 - Tape Init/Erase/Mapping, 9-179
- CTTTPICT DD Statement
 - CTTTPI Utility, 9-190

- CTTTPITP DD Statement
 CTTTPI Utility, 9-190
- CTTTRB Utility
 Backup MDB and Trace File, 9-192
- CTTUDB Utility
 Update MDB, 9-193
- CTTVTM Utility
 Examples, 9-199
 New Day Procedure (CTT), 7-8
 Return Codes, 9-199
 Vault Management, 9-194
- CTTX002 Exit
 Stacking Facility, 7-40
- CTTX010 Exit
 Stacking Algorithm, 7-40
- CTVCLMIG Utility
 Clean Migrated Report List, 9-93
 Description, 4-77
- CTVDELI Utility
 Deleting DASD Index Files, 9-94
- Customization. *See also* Modification
 Allocation of Files, 2-11
 Color Support, 2-26
 Commands, C-1
 Display Format Member, 2-19
 Exit CTDX005, 4-49
 IOA Messages, 2-47
 IOA Profiles, 2-32
 IOA Screens, 2-17
 Limited Access to IOA Products, 2-8
 Online Facility, 2-8
 PFKeys, 2-13
 Primary Option Menu, 2-15
 Screen Definition, 2-34
- CUSTOMIZE Option
 IOA Installation Menu, 2-36
- CYCLECNT Parameter
 CTTTPARM, 9-194
 CTTTPARM Member, 9-164
- D—
- D Parameter
 IOADBF Utility, 9-18
- D37 Abend
 Global AutoEdit Library, 5-14
- DAALOCIN DD Statement
 Dynamic File Allocation, 4-29
- DAALPPRM DD Statement
 Utility CTOALOCP, 9-100
- DAALTALC DD Statement
 Allocation Customization, 2-12
- DAAMF DD Statement
 CTDRRQ Program, 4-28
- DAAPA DD Statement
 AFP Control Parameters, 4-40
- DABKPLST DD Statement
 Backup Missions, 4-14
 Migration Missions, 4-14
- DABRULE DD Statement
 CTB Rule Libraries, 6-4
- DACHK DD Statement, 3-13, 4-8
 CTBNDAY Procedure, 6-2
 CTMDAILY Procedure, 4-24
- DACOPPRM DD Statement
 CTDCAMF Utility, 9-68
 Utility CTMCAJF, 9-45
- DACRSIN DD Statement
 IOACLRES Utility, 9-7
- DACTMLST DD Statement
 CMEM Rule List, 5-25
 Rule Table List (CMEM), 3-27, 5-3
- DADJDE DD Statement
 Printers Control Monitor, 4-42
- DADJOB DD statement
 Backup Missions, 4-55
 Migration Library, 4-58
 Restore Mission, 4-65
- DADSKL DD Statement
 Backup Mission, 4-54
 Backup Missions, 4-55
- CONTROL-D Monitor, 4-32
 Migration Skeleton, 4-57
 Restore Mission, 4-64
- DADUMP DD Statement
 Archive Server Procedure, 4-73
 CMEM Procedure, 3-31
 CONTROL-O Procedure, 5-17
- DAFILE DD statement
 IOADBF Utility, 9-18
- DAFRMIN DD Statement, 3-11
 New Day Procedure (CTM), 4-29
- DAGENLST DD Statement
 Generic Decollating, 4-14
- DAGENUSR DD Statement
 CTDRRQ Program, 4-28
 Generic User List, 4-22
- DAGLBLST DD Statement
 Global Variables, 5-12
- Daily Checkpointing, 3-8
- Daily Processing
 Description, 1-6
- Daily Subsystem. *See* New Day Procedure
- DAILYPRD Job
 Sample User Daily, 3-6, 3-13
- DAILYSYS Job
 Sample User Daily, 3-6, 3-13
- DAISCNn DD Statement
 Inter-System Communications (CTM), 3-34
- DAJOB DD Statement, 3-13
- DALCPPRM DD Statement
 Utility IOACPLOG, 9-16
- DALNRIN DD Statement
 New Day Procedure (CTM & CTD)), 4-29
- DALOG DD Statement
 CTDRRQ Program, 4-28
 CTTTPI Utility, 9-190
 Utility CTMBGRP, 9-43
 Utility IOACPLOG, 9-16

Index

- DALOGCUR DD Statement
 - Utility IOACPLG, 9-16
- DALOGSEQ DD Statement
 - Copy IOA Log, 9-16
- DAMDB DD Statement
 - CTTTPI Utility, 9-190
 - Media Database, 7-9
- DAMDI DD Statement
 - CTTTPI Utility, 9-190
 - Media Database, 7-9
- DAOUT DD Statement
 - CTDRRQ Program, 4-28
- DAOVRALC DD Statement
 - Allocation Customization, 2-12
- DAPOOLS DD Statement
 - CTTINIT Procedure, 7-3, 7-5
- DAPROG DD Statement
 - New Day Processing (CTM), 3-16
- DAPRTLST DD Statement
 - Printing Missions, 4-14
- DAREPIN DD Statement
 - Utility CTORSTM, 9-99
- DAREPLST DD Statement
 - Decollating Missions, 4-14
- DAREPMIS DD Statement
 - Decollating Mission Library, 4-24
 - Decollating Missions Library, 4-23
- DARSTLST DD Statement
 - Restore Missions, 4-14
- DARULLST DD Statement
 - CTO Rule List, 5-25
 - CTTINIT Procedure, 7-3, 7-5
 - Loading Rules (CTO), 5-2
- DASD
 - Deleting Index Files, 9-94
 - Disaster Recovery, 7-24
- DASORTPR DD Statement
 - Active User Report
 - Sorting, 4-83
- Data Extraction
 - Media Database, 9-143
- Data File
 - Fields, 7-18
 - Media Database, 7-10
 - Printing (CTT), 9-121
 - Stacking Statistics File, 7-18
- Data File Parameters
 - CTTCRDF Utility, 9-111
 - CTTCTRC Utility, 9-114
- DATA Parameter
 - @FIELD Display Format Line, 2-22
 - @HEADER Display Format Line, 2-21
 - @VAL Display Format Line, 2-23
 - CTTIOS Macro, 7-49
- Database (CTT)
 - Printing, 9-121
- Database Utilities
 - CONTROL-T, 9-6
- Dataset
 - Multi-Volume, 7-15
- Dataset Allocation
 - Utility CTTGVL, 9-125
- Dataset Length
 - Stacking Facility, 7-40
- DATASET Parameter
 - CTTGVL Utility, 9-125
- Dataset Record
 - Index File, 7-13
 - Keywords, D-2
 - Media Database, 7-10
 - Volume Record, 7-13
- Date
 - Printing Mission Scheduling, 4-34
- Date Control Record
 - Checkpointing, 3-8
 - CONTROL-B, 6-2
 - Creation (CTM), 3-14
 - Decollating Scheduling (CTD), 4-14
 - Format (CTB), 6-3
 - Format (CTM), 3-14
 - Format CTD, 4-7
 - General Daily (CTD), 4-10
 - User Daily Job (CTM), 3-4, 3-17
- Date Format
 - INCLUDE/EXCLUDE Statements, D-1
 - Profile Variable, 2-40
- DATE Keyword
 - Trace Record, D-11
- DATE Parameter
 - CTDCCU Utility, 9-74
 - CTTRTM Utility, 9-166
 - CTTVTM Utility, 9-195
- Date Range
 - IOA Log, 9-16
 - Profile Variable, 2-42
- DATEREC
 - Date Control Record (CTM), 3-6
- DATERECU
 - Date Control Record (CTM), 3-6
- DATRC DD Statement
 - CTTTPI Utility, 9-190
- DAVLTS DD Statement
 - CTTINIT Procedure, 7-3, 7-5
- Daylight Savings Time
 - Clock Changing, 3-23
- DAYS Parameter
 - CTBJAFDL Utility, 9-65
- DAYS Statement
 - CTDDELRP Utility, 9-80
 - CTVDELI Utility, 9-95
- DAYTIME Parameter
 - Modification (CTT), 7-4
- DBABFB Parameter
 - CTBFRM Utility, 9-62
- DBBKPB Parameter
 - CTBFRM Utility, 9-62
- DBFILE Parameter
 - CTBDBIB Utility, 9-52
 - IOADBSR Utility, 9-25
 - IOADIG Utility, 9-29
 - IOADII Utility, 9-30
 - IOADPT Utility, 9-32
 - IOADUL Utility, 9-33
- DBFILEB Parameter
 - Utility CTBABI, 9-50
- DBFILEI Parameter
 - CTTDBDLD Utility, 9-115

- CTTDBDUL Utility, 9-116
- DBFILEO Parameter
 - CTTDBDL D Utility, 9-115
 - CTTDBDUL Utility, 9-116
- DBFILET Parameter
 - CTTCRDF Utility, 9-111
 - CTTDBF Utility, 9-117
 - CTTDBIB Utility, 9-118
 - CTTDBID Utility, 9-119
 - CTTDBII Utility, 9-120
 - CTTDBPRT Utility, 9-121
 - Media Database, 7-21
 - Trace File, 7-22
- DBGJOB Parameter
 - CTTINIT Procedure, 7-5
- DBGLEVEL Parameter
 - CTTINIT Procedure, 7-5
 - CTTMER Utility, 9-130
 - CTTRTM Utility, 9-166
 - CTTSPL Utility, 9-175
 - CTTVTM Utility, 9-195
- DBGLVL Parameter
 - CTTIOS Macro, 7-49
- DBJAFB Parameter
 - CTBJAFDL Utility, 9-65
- DBJAFI Parameter
 - CTBJAFCP Utility, 9-63
- DBJAFO Parameter
 - CTBJAFCP Utility, 9-63
- DBPREFA Parameter
 - IOADBSR Utility, 9-25
- DBPREFT Parameter
 - CTTCRDF Utility, 9-111
 - CTTCTRC Utility, 9-114
 - Media Database, 7-21
 - Trace File, 7-22
- DBREPB Parameter
 - CTBJAFDL Utility, 9-65
- DBREPI Parameter
 - CTBJAFCP Utility, 9-63
- DBREPO Parameter
 - CTBJAFCP Utility, 9-63
- DBSRTPRM DD Statement
 - Active User Report
 - Sorting, 4-83
- DBUNITT Parameter
 - CTTCRDF Utility, 9-111
 - CTTCTRC Utility, 9-114
- DBVERT Parameter
 - CTTCRDF Utility, 9-111
 - CTTCTRC Utility, 9-114
- DBVOLT Parameter
 - CTTCRDF Utility, 9-111
- DCF
 - IBM word processor, 4-40
- DCHANGED Keyword
 - Dataset Record, D-2
- DCHANGET Keyword
 - Dataset Record, D-2
- DCHANGEU Keyword
 - Dataset Record, D-2
- DD Statements
 - CTTINIT Procedure, 7-5
- DDNAME Parameter
 - CTTTPI Utility, 9-187, 9-188
- DDSACCT Keyword
 - Dataset Record, D-2
- DDSBLK Keyword
 - Dataset Record, D-2
- DDSBLKC Keyword
 - Dataset Record, D-2
- DDSCCC Keyword
 - Dataset Record, D-2
- DDSCCPU Keyword
 - Dataset Record, D-2
- DDSCDDN Keyword
 - Dataset Record, D-2
- DDSCDT Keyword
 - Dataset Record, D-2
- DDSCJBN Keyword
 - Dataset Record, D-2
- DDSCJID Keyword
 - Dataset Record, D-2
- DDSCPGM Keyword
 - Dataset Record, D-2
- DDSCSTP Keyword
 - Dataset Record, D-2
- DDSCTM Keyword
 - Dataset Record, D-2
- DDSCUAD Keyword
 - Dataset Record, D-2
- DDSCUSER Keyword
 - Dataset Record, D-2
- DDSDEN Keyword
 - Dataset Record, D-3
- DDSDSN Keyword
 - Dataset Record, D-4
 - Trace Data, D-12
- DDSEFTG Keyword
 - Dataset Record, D-3
- DDSEXCP# Keyword
 - Dataset Record, D-3
- DDSEXPDI Keyword
 - Dataset Record, D-3
- DDSEXPT1 Keyword
 - Dataset Record, D-3
- DDSFEEET Keyword
 - Dataset Record, D-3
- DDSFGL1 Keyword
 - Dataset Record, D-3
- DDSJEXPD Keyword
 - Dataset Record, D-4
- DDSLACS Keyword
 - Dataset Record, D-4
- DDSLBLNM Keyword
 - Dataset Record, D-4
- DDSLUSER Keyword
 - Dataset Record, D-2
- DDSMDDT Keyword
 - Dataset Record, D-2
- DDSMDDM Keyword
 - Dataset Record, D-2
- DDSRBA Keyword
 - CTTMUP Utility, 9-133
 - Trace Data, D-12
- DDSRCC Keyword
 - Dataset Record, D-4
- DDSRCPU Keyword
 - Dataset Record, D-4
- DDSRDDN Keyword
 - Dataset Record, D-4
- DDSRDDT Keyword
 - Dataset Record, D-4
- DDSRRCFM Keyword
 - Dataset Record, D-5
- DDSRRECL Keyword
 - Dataset Record, D-4
- DDSRJBN Keyword
 - Dataset Record, D-4
- DDSRPGM Keyword
 - Dataset Record, D-4

Index

- DDSRSTP Keyword
 - Dataset Record, D-4
- DDSRM Keyword
 - Dataset Record, D-4
- DDSRTOrg Keyword
 - Dataset Record, D-5
- DDSRType Keyword
 - Trace Data, D-12
- DDSRUAD Keyword
 - Dataset Record, D-4
- DDSSMCL Keyword
 - Dataset Record, D-5
- DDSSSTAT Keyword
 - Dataset Record, D-4
- DDSTRTCH Keyword
 - Dataset Record, D-5
- DDSUSED Keyword
 - Dataset Record, D-4
- DDSUSeR2 Keyword
 - Dataset Record, D-4
- DDSVOLS# Keyword
 - Dataset Record, D-5
- DDSVOLSR Keyword
 - Dataset Record, D-4
- DDSWCC Keyword
 - Dataset Record, D-5
- DDSWCPU Keyword
 - Dataset Record, D-5
- DDSWDDN Keyword
 - Dataset Record, D-5
- DDSWDT Keyword
 - Dataset Record, D-5
- DDSWJBN Keyword
 - Dataset Record, D-5
- DDSWPGM Keyword
 - Dataset Record, D-5
- DDSWSTP Keyword
 - Dataset Record, D-5
- DDSWTM Keyword
 - Dataset Record, D-5
- DDSWUAD Keyword
 - Dataset Record, D-5
- Deactivating
 - Archive Server (CTV), 4-5
 - CDAM, 4-5
 - CMEM Monitor, 3-3
 - CMEM Rule Table, 3-28
 - CONTROL-M Monitor, 3-1
- Generic Processing (CTD), 4-4
 - VTAM Monitor, 2-8
- Deactivation
 - CONTROL-O, 5-1
- DEBUG Command
 - CMEM, 3-31
 - CONTROL-O, 5-17
- Debugging
 - Archive Server, 4-73
 - CMEM Rules, 3-30
 - CONTROL-O, 5-16
 - Job Name (CTT), 7-5
- DECFROM Parameter
 - Utility CTDUPBKP, 9-88
- DECOLLATED Status
 - Active User Report List, 4-77
- Decollating Mission
 - CONTROL-B Rule, 6-9
 - Generic, 4-19
 - Profile Variable, 2-36
 - Scheduling, 4-13
- Decollation Mission
 - Overview, 4-19
- DECTO Parameter
 - Utility CTDUPBKP, 9-88
- Dedicated Device Mode
 - CART Media, 4-67
- DEFABEND Parameter
 - Modification (CTT), 7-4
- Default. *See also* specific variable, parameter, or function
- DEFAULTEDIT Parameter
 - @FIELD Display Format Line, 2-23
- DEFDELAY Parameter
 - IOATEST Utility, 9-39
- DEFER Parameter
 - Exit CTDX005, 4-51
- DEFEXPDT Parameter
 - Modification (CTT), 7-4
- DEFGRP Member, 2-30
- DEFGRPI Member, 2-30
- DEFHST Member, 2-30
- DEFHSTI Member, 2-30
- DEFSTAT Member
 - Message Statistics, 9-98
- Deleting. *See also* Erasing.
See also Cleaning
 - CDAM Files, 9-74
 - CTB Variables, 9-59
 - DASD Index Files, 9-94
 - MDB Volume, 9-132
 - Prerequisite Conditions, 9-7
 - Reports (CTD), 9-79
 - Rule Activity File, 9-65
 - Rule Table (CTO), 5-5
 - Unneeded Jobs, 9-47
 - Volumes from MDB, 9-122
- Deletion Rules
 - Utility CTDDELRP, 9-82
- DELGR Parameter
 - CTBDBVDL Utility, 9-59
- DELVOL Function
 - High Level API, 7-68, 7-69
- DELYBLK Parameter
 - IOADBSR Utility, 9-26
- DEN Parameter
 - Exit CTDX005, 4-51
- DENSITY Keyword
 - Dataset Record, D-3
- Dependency
 - CTM Job and CTD Report, 4-25
- DESC Parameter
 - Automation Option Menu, 5-19
 - IOATEST Utility, 9-39
- DEST JES2 Parameter
 - Multi-Chunk Printing, 4-35
- Destination Table. *See* Dynamic Destination Table
- DETAIL Parameter
 - CTDUPBKP Utility, 9-90
- DEVADDR Parameter
 - CART Media, 4-67
 - Migration Media, 4-59
- Device Definition
 - CART Media, 4-67
- DEVICE Parameter
 - MEDIA Operator Command, 4-69

- DEVQTY Parameter
 - CART Media, 4-67
 - MEDIA Operator Command, 4-70
- DF/DSS
 - Backup Utility, 4-55
- DF/HSM
 - Backup Utility, 4-55
- DF/SMS Support, 7-37
- DFLT Parameter
 - @FIELD Display Format Line, 2-23
- DGRP Parameter
 - CTBDBVDL Utility, 9-59
- DIGRP Parameter
 - CTBDBVCG Utility, 9-54
 - CTBDBVCP Utility, 9-56
- DIMOD Parameter
 - CTBDBVCG Utility, 9-54
 - CTBDBVCP Utility, 9-56
- Disaster Recovery
 - Media Database, 7-24
- Disk
 - Migration Skeleton, 4-60
- Display. *See also* Screens
 - Media/Resource Information, 4-72
- DISPLAY Command
 - Active Users, 2-6
 - CMEM, 3-29
- Display Format Member
 - Customization, 2-19
 - Format, 2-20
- DISPLAY Parameter
 - IOAVERFY Utility, 9-41
- Display Type Editing. *See* Display Format Members
- Distribution Report
 - CTTVTM Utility, 9-194
- DIVAR Parameter
 - CTBDBVCG Utility, 9-54
 - CTBDBVCP Utility, 9-56
- DJDE Parameters
 - XEROX Printers, 4-42
- DJDE Printing. *See also* XEROX
- DJDEPARM Library
 - Format, 4-42
 - XEROX DJDE Parameters, 4-42
- DMOD Parameter
 - CTBDBVDL Utility, 9-59
- DMS/OS Support
 - Backup Utility, 4-55
 - DD Statements, 7-36
 - External Data Manager, 7-35
- DOCDPAGM Member
 - AFP Printing, 4-40
- DOGRP Parameter
 - CTBDBVCP Utility, 9-56
- DOLV. *See also* Online Viewing
- DOMOD Parameter
 - CTBDBVCP Utility, 9-56
- DORM Mode
 - CTTINIT Procedure, 7-4
- Dormant Mode
 - CONTROL-T, 7-4
- Double Byte Character Set
 - Profile Variable, 2-45
- DOVAR Parameter
 - CTBDBVCP Utility, 9-56
- Driver Exit
 - CONTROL-R, 7-43
- DSDELL Statement
 - CTTMUP Utility, 9-134
- DSENDFET Keyword
 - Dataset Record, D-3
- DSEXCP Keyword
 - Dataset Record, D-3
- DSEXPDT Keyword
 - Dataset Record, D-3
- DSEXTYP Keyword
 - Dataset Record, D-3
- DSFEET Keyword
 - Dataset Record, D-3
- DSFLAGS Keyword
 - Dataset Record, D-3
- DSLABEL Keyword
 - CTTMUP Utility, 9-133, 9-134
 - Dataset Record, D-4
- DSN Parameter
 - Exit CTDX005, 4-50
 - IOADBF Utility, 9-19
- DSNADD Statement
 - CTTMUP Utility, 9-134
- DSNAME Keyword
 - CTTMUP Utility, 9-134
 - Dataset Record, D-4
 - Trace Data, D-12
- DSNDEL Statement
 - CTTMUP Utility, 9-134
- DSNMCHK Parameter
 - Modification (CTT), 7-4
- DSNPREF Parameter
 - Dual Migration, 4-59
 - Migration Media, 4-59
- DSNUPD Statement
 - CTTMUP Utility, 9-133
- DSNUPDN Statement
 - CTTMUP Utility, 9-134
- DSSTAT Keyword
 - Dataset Record, D-4
- DSUSECT Keyword
 - Dataset Record, D-4
- DSVOLSER Keyword
 - CTTMUP Utility, 9-133, 9-134
 - Dataset Record, D-4
- D-type Index
 - Media Database, 7-13
- Dual Checkpoint Mode
 - CONTROL-M Monitor, 3-26
- Dual Migration (CTV), 4-59
- Dual Mirror Image File
 - Recovery, 9-27
 - Support, 2-30
- DUAL Parameter
 - IOADBF Utility, 9-20
- DUALM Parameter
 - IOA Access Method, 2-31
 - IOADBF Utility, 9-20
- DUALST Parameter
 - IOA Access Method, 2-31
 - IOADBF Utility, 9-21
- DUMMY Parameter
 - Exit CTDX005, 4-51
- DUSRDATA Keyword
 - Dataset Record, D-4
- DVAR Parameter
 - CTBDBVDL Utility, 9-59
- DVLACTD# Keyword
 - Volume Record, D-6

Index

- DVLADT Keyword
 - Volume Record, D-6
- DVLATM Keyword
 - Volume Record, D-6
- DVLBOXID Keyword
 - Volume Record, D-6
- DVLCCLN Keyword
 - Volume Record, D-6
- DVLCCLN# Keyword
 - Volume Record, D-6
- DVLCODE Keyword
 - Volume Record, D-6
- DVLDEN Keyword
 - Volume Record, D-7
- DVLDESC Keyword
 - Volume Record, D-9
- DVLEDMID Keyword
 - Volume Record, D-8
- DVLEXCP# Keyword
 - Volume Record, D-8
- DVLEXPD Keyword
 - Volume Record, D-8
- DVLEXPDS Keyword
 - Volume Record, D-6
- DVLEXPT Keyword
 - Volume Record, D-8
- DVLFCDT Keyword
 - Volume Record, D-6
- DVLFEET Keyword
 - Volume Record, D-8
- DVLFIRST Keyword
 - Volume Record, D-6
- DVLFLG1 Keyword
 - Volume Record, D-8
- DVLFLG2 Keyword
 - Volume Record, D-9
- DVLJBN Keyword
 - Volume Record, D-6
- DVLLBLNM Keyword
 - Volume Record, D-6
- DVLLDFTG Keyword
 - Volume Record, D-6
- DVLLTYP Keyword
 - Volume Record, D-6
- DVLMDT Keyword
 - Volume Record, D-7
- DVLMEDIA Keyword
 - Volume Record, D-7
- DVLMTM Keyword
 - Volume Record, D-7
- DVLMVDT Keyword
 - Volume Record, D-7
- DVLNEXT Keyword
 - Volume Record, D-7
- DVLOWNR Keyword
 - Volume Record, D-9
- DVLPREV Keyword
 - Volume Record, D-7
- DVLPRFLN Keyword
 - Volume Record, D-8
- DVLRBA Keyword
 - CTTMUP Utility, 9-132
 - Trace Data, D-12
- DVLRFEET Keyword
 - Volume Record, D-7
- DVLRPER Keyword
 - Volume Record, D-6
- DVLRPERC Keyword
 - Volume Record, D-6
- DVLRTER Keyword
 - Volume Record, D-6
- DVLRTERC Keyword
 - Volume Record, D-6
- DVLRTRN Keyword
 - Volume Record, D-6
- DVLRTYPE Keyword
 - Trace Data, D-12
- DVLSCRDT Keyword
 - Volume Record, D-7
- DVLSLNAME Keyword
 - Volume Record, D-7
- DVLSTA2 Keyword
 - Volume Record, D-9
- DVLSTAT Keyword
 - Volume Record, D-9
- DVLTRTCH Keyword
 - Volume Record, D-9
- DVLUNAM Keyword
 - Volume Record, D-7
- DVLUSED# Keyword
 - Volume Record, D-9
- DVLUSER Keyword
 - Volume Record, D-7
- DVLUSER2 Keyword
 - Volume Record, D-7
- DVLVENDT Keyword
 - Volume Record, D-7
- DVLVLT1 Keyword
 - Volume Record, D-7
- DVLVLTD5 Keyword
 - Volume Record, D-7
- DVLVLTNM Keyword
 - Volume Record, D-7
- DVLVLTRC Keyword
 - Volume Record, D-7
- DVLVLTT Keyword
 - Volume Record, D-7
- DVLVNDOR Keyword
 - Volume Record, D-7
- DVLVOLSR Keyword
 - Trace Data, D-12
 - Volume Record, D-9
- DVLVRTRN Keyword
 - Volume Record, D-7
- DVLVSEQ Keyword
 - Volume Record, D-9
- DVLVSLOT Keyword
 - Volume Record, D-7
- DVLVTSQ Keyword
 - Volume Record, D-7
- DVLVXPD1 Keyword
 - Volume Record, D-8
- DVLVXPT1 Keyword
 - Volume Record, D-8
- DVLWPER Keyword
 - Volume Record, D-6
- DVLWPERC Keyword
 - Volume Record, D-6
- DVLWTER Keyword
 - Volume Record, D-6
- DVLWTERC Keyword
 - Volume Record, D-6
- Dynamic Dataset Stacking.
 - See* Stacking Facility
- Dynamic Destination Table, 2-49
 - Description, 1-11
- Dynamic Device Mode
 - CART Media, 4-67
- Dynamic Sorting
 - Active Report List, 4-83
- DYNDEVRLSE Parameter
 - CART Media, 4-67
- DYNDS Parameter
 - Modification (CTT), 7-4

- DYNSTYP Parameter
Modification (CTT), 7-4
- DYNVOL Parameter
Modification (CTT), 7-4
- DYNWTO Parameter
HSC/CSC Scratch
Support, 7-31
- E—
- EDIMACNM Profile
Variable
Editing JCL, 2-46
- EDIT Parameter
@FIELD Display Format
Line, 2-23
CTTRPT Utility, 9-148
- EMASS
Tape Library Support,
7-32
- ENABLE Parameter
IOADBSR Utility, 9-25
- END Function
High Level API, 7-68,
7-69
- END Parameter
IOAVERIFY Utility, 9-42
- ENDDATE Parameter
Utility CTOALOCF,
9-100
- ENDED Status
Backup Mission, 4-56
Migration Mission, 4-62
- ENDTIME Parameter
Utility CTOALOCF,
9-100
- ENDUPD Statement
CTTMUP Utility, 9-135
- Enhanced Daily
Checkpointing
(CTD), 3-8
(CTM), 3-18
- Enlarging
Media Database, 7-21
Trace File, 7-22
- ENQ Mechanism, 4-33
- Enqueue Manager
CONTROL-O, 5-7
- ENV Keyword
Trace Record, D-11
- ENV Parameter
CTTIOS Macro, 7-49
- Environment. *See* Online
Facility
- IOA Online, 2-1
- Program List Member,
2-10
- ERASE Parameter
CTTTPI Utility, 9-187
- ERASELBL Parameter
CTTTPI Utility, 9-188
- Erasing
Information on a Tape,
9-181
- Error. *See also* Integrity.
See also Abend
- Media Database
Integrity, 7-19; 9-127
- Error Handling
High Level API (CTT),
7-70
- Error Message. *See*
Message or Messages
- Exception Handling
Backup Missions, 4-56
Migration Missions, 4-62
Restore Mission, 4-65
- EXCLUDE Statement. *See
also* INCLUDE/EXCLUDE
Statements
- EXCLUDE Statements
Format (CTT), 9-103
Record Selection (CTT),
9-102
- EXEC Statement
CTTTPI Utility, 9-182
- Execution Parameters
ACIF Interface, 4-44
- Exits. *See also* User Exit
- CTDX001, 4-17, 4-23,
4-28
- CTDX009, 4-33
- CTDX010, 4-55, 4-58
- CTDX011, 4-65
- CTR001, 7-43
- CTTX002, 7-40
- CTTX010, 7-40
- IGXMSGEX, 7-33
- IOAX007, 3-33
- ExLM
StorageTek Silo Support,
7-31
- EXLMJOB Member
ExLM JCL, 7-32
- EXLMSAMP Member
ExLM Control
Statements, 7-32
- Expanding
Conditions/Resources
File, 2-49
IOA LOG File, 2-50
- EXPDSNUM Keyword
Volume Record, D-6
- EXPDT Parameter
Exit CTDX005, 4-51
- EXPDTDDN
Modification (CTT), 7-4
- EXPDTYPE Parameter
CTTPARM Member,
9-164
Modification (CTT), 7-4
- Expert Library Manager.
See ExLM
- EXPRTRN Keyword
Volume Record, D-6
- EXTEND Parameter
IOADBF Utility, 9-21
- Extended Color Support,
2-25. *See also* Color
Support
Profile Variable, 2-45
- External Data Manager
CA-ASM2 Support, 7-36
CONTROL-T Support,
7-35
DF/SMS Support, 7-37
DMS/OS Support, 7-35
HSM Support, 7-35
- External Resources
Saving via ACIF, 4-44
- Extract File
CTTMER Utility, 9-129
- EXTRACT PATH
Statement
CTTRPT Utility, 9-144
- EXTRACT Statement
CTTSPL Utility, 9-174
- EXTRNVOL Parameter
Modification (CTT), 7-4

Index

—F—

- Fat-DASD
 - Migration Skeleton, 4-60, 4-61
- FDR
 - Backup Utility, 4-55
- Field Names
 - CTT Repository, D-1
 - Dataset Records, D-2
 - Stacking Records, D-10
 - Trace Data, D-12
 - Trace Records, D-11
 - Volume Records, D-6
- FIELD Parameter
 - CTTRPT Utility, 9-148
- FIELDS Statement
 - CTTRPT Utility, 9-146
 - CTTRTM Utility, 9-170
 - CTTVTM Utility, 9-197
- File
 - Allocation (CTT), 9-111
 - Creation (CTT), 9-111
 - Expanding (CTM), 3-35
 - Formatting (CTT), 9-111
- File Allocation
 - IOA Access Method, 2-28
- File Definition
 - IOA Access Method, 2-30, 9-19
- FILE Parameter
 - CTBJAFCP Utility, 9-63
 - Utility CTBJAFIG, 9-67
 - Utility CTDUPBKP, 9-88
- File Structure
 - IOA Access Method, 2-27
- FileTek Storage Machine
 - Logical Devices, 4-70
 - Media Definition, 4-68
 - Migration Skeleton, 4-60, 4-61
- FileTek Storage Machine Machine
 - Media Information, 4-73
- Finish Indicator Date (CTD), 4-9
- FIRST Parameter
 - CTDLD Utility, 9-123
- FIRSTVOL Keyword
 - Volume Record, D-6
- FOB Parameter
 - Exit CTDX005, 4-51
- Fonts
 - AFPDS Printing, 4-43
- FORCE Parameter
 - CONTROL-O Procedure, 5-2
 - CTBFRM Utility, 9-62
 - CTDDELRP Utility, 9-80
 - CTTRCV Utility, 9-141
 - Mission Scheduling, 4-15
- Form Definitions
 - AFPDS Printing, 4-43
- FORMAMF Member, 9-69
- FORMAT Parameter
 - Automation Option Menu, 5-23
- FORMATF Member
 - INSTCTD Library, 9-70
- Formatting
 - Access Method Variable File (CTB), 9-24
 - Active Balancing File, 9-24
 - Automation Options, 5-23
 - Backup File (CTB), 9-24
 - CTB Files, 9-24
 - CTT Files, 9-111, 9-117
 - Group File (CTB), 9-24
 - Online Facility, E-1
 - Report File (CTB), 9-24
 - Rule Activity File (CTB), 9-24
 - Trace File (CTT), 9-108, 9-114
- FORMCKP Member
 - Active Jobs File Size, 9-48
- FORMDEF Parameter
 - AFP Printing, 4-38
 - CDAM, 4-39
- FREE Parameter
 - Printing Mission, 4-32
- Free Space Chain
 - Rebuilding (CTT), 7-24
- Free Tape Pool
 - Utility CTDCLHIS, 4-79
- French
 - Language Support, 8-2
- FROM Parameter
 - CTMCAJF Utility, 9-46
 - CTTACP Utility, 9-106
 - CTTDBPRT Utility, 9-121
- FROMDATE Parameter
 - Utility CTMJSA, 9-49
 - Utility IOACPLOG, 9-16
- FROMDSN Parameter
 - Utility CTOALOCF, 9-100
- FROMGEN Parameter
 - CTBDBVCP Utility, 9-57
- FROMKEY Parameter
 - IOADUL Utility, 9-33
- FROMREC Parameter
 - CTTRCV Utility, 9-140
- FROMTYPE Parameter
 - Utility CTOALOCF, 9-100
- FSET Parameter
 - FileTek Migration, 4-61
- FUNC Parameter
 - IOADBF Utility, 9-18
 - IOADIG Utility, 9-29
 - IOADPT Utility, 9-32
- FUNCTION Keyword
 - Trace Data, D-11
- FUNCTION Parameter
 - CTTSPL Utility, 9-174
- Funtional Monitor
 - Trace File (CTT), 7-17

—G—

- GDDM
 - AFP Documents, 4-40
- GDG. *See* Generation Data Groups
- GENCLAS Parameter
 - Generic Decollating, 4-20
- GENER Parameter
 - CTDUPBKP Utility, 9-89
- General Daily. *See* New Day Procedure
 - Date Control Record (CTD), 4-10
- Generation Data Groups
 - Dataset Stacking, 7-42
- Generations
 - CTB Variables, 9-54

- Generic Decollating Mission. *See also* Decollating Missions
 - Overview, 4-19
 - Scheduling, 4-21
 - Generic Processing
 - Activating, 4-1
 - Generic User Name
 - Definition, 4-22
 - GENLIST Member
 - Generic Decollating, 4-14
 - German
 - Language Support, 8-2
 - GLBCOMP Parameter
 - Global Library Compression, 5-15
 - GLBPREF Parameter
 - Global Variables (CTO), 5-12
 - Global AutoEdit Variable Library
 - Automatic Compression, 5-14
 - Global Control
 - Printing Characteristics, 4-47
 - Global Member
 - Definition (CTO), 5-13
 - Global Profile, 2-32
 - GLOBAL SMP/E Zone, 8-1
 - Global Variables
 - Description (CTO), 5-12
 - Graphics
 - AFPDS Printing, 4-43
 - GRAU
 - Tape Library Support, 7-32
 - Group File (CTB)
 - Copying, 9-56
 - Formatting, 9-24
 - Rebuilding, 9-24
 - Group of Users
 - CONTROL-D, 4-22
 - GROUP Parameter
 - CTBDBVCG Utility, 9-54
 - CTBDBVCP Utility, 9-56
 - CTBDBVDL Utility, 9-59
 - FileTek Media, 4-68
 - IGNORE/SELECT Statements (CTM), 3-11
 - Invoking CONTROL-B, 6-7
 - Printing Mission
 - Definition, 4-38
 - Utility CTMBGRP, 9-43
 - Group Scheduling Table
 - Utility CTMBGRP, 9-43
 - GROUPPWD Parameter
 - FileTek Media, 4-68
 - GRPBIX Statement
 - CTTMUP Utility, 9-133, 9-136
 - GRPSCR Statement
 - CTTMUP Utility, 9-133
- H—**
- HDS Robots
 - CONTROL-T Support, 7-25
 - Held Output Class (CTM), 3-25
 - Help Members
 - Online Facility, E-1
 - Hexadecimal
 - AFP Parameters, 4-41
 - High Level API (CTT)
 - Base Level API, 7-46
 - Buffer Address, 7-69
 - Description, 7-68
 - Error Handling, 7-70
 - Functions, 7-68
 - Macro CTTACCDB, 7-68
 - Macro CTTCHKDB, 7-70
 - Sample JCL, 7-71
 - Highlight. *See also* Color Support
 - Display Format, 2-25
 - HILITE Parameter
 - Display Format Member, 2-25
 - HILITEA Parameter
 - Display Format Member, 2-25
 - History Report List
 - Cleaning, 9-76
 - CTDCCU Utility, 9-74
 - CTDDELRP Utility, 4-78, 9-79
 - Description, 4-78
 - Restore Missions, 4-64
 - History User File
 - Backup Utility, 9-87
 - History User Report List
 - Utility CTDCLHIS, 4-79
 - HLDECLAS Parameter
 - Held Output Class (CTM), 3-25
 - Housekeeping. *See* Maintenance
 - HSC Support
 - StorageTek Robot, 7-28
 - HSM Support, 7-35
- I—**
- IBM
 - 3480 Tape Cartridge Device, 4-70
 - SMS Managed Tape Library, 7-26
 - IBM 3390-9
 - Migration Skeleton, 4-60
 - IBM 3480 Tape. *See also* CART Media
 - IBM 3995
 - Migration Skeleton, 4-61
 - IBM 3995 Optical Library
 - Migration Skeleton, 4-60
 - IBM Tape Library
 - CONTROL-T Interface, 7-26
 - Tape Eject, 7-26
 - Tape Entry, 7-27
 - Use Attribute, 7-26
 - User Exits, 7-26
 - ICE
 - Description, 1-2
 - IOA Maintenance, 8-10
 - Profile Customization, 2-32
 - Profile Variables, 2-35
 - ICOLOR
 - Profile Variable, 2-45
 - ID Parameter
 - @STYLE Display Format Line, 2-20
 - CTTRPT Utility, 9-146
 - IDBCS
 - Profile Variable, 2-45

Index

- IDLIKE Parameter
 - @STYLE Display Format Line, 2-21
- IDM=
 - AFP Command, 4-41
- IDMS
 - Color Support, 2-45
- IDMS/DC
 - Color Support, 2-25
 - IOA Online Support, 2-1
- IEASYSSX Member, 2-6; 4-4
- IEBGENER Utility
 - Media Database Backup, 7-9
- IEFSSNxx Member
 - Subsystem List, 5-5
- IFLTSTA
 - Profile Variable (CTM), 2-42
- IGNORE Statement
 - CTD Daily Processing, 4-12
 - IOACLRES Utility, 9-7
 - IOALDNRS Utility, 9-35
 - Program CTMFRM, 3-11
- IGNORE/SELECT Statements
 - CTM Daily Processing, 3-10
- IGXMSGEX Exit
 - Cartridge Message Display, 7-33
- IMM=
 - AFP Command, 4-41
- Immediate Printing
 - Archive Server, 4-66
- IMS
 - Color Support, 2-45
- IMS/DC
 - Color Support, 2-25
 - IOA Online Support, 2-1
- IN Parameter
 - Backup Missions, 4-52
 - Migration Missions, 4-57
 - Printing Missions, 4-34
- IN THE ROBOT Status Bit
 - IBM Robot Support, 7-26
- INCLUDE/EXCLUDE Statements
 - CTBJAFCP Utility, 9-63
 - CTBJAFDL Utility, 9-65
 - CTDCA2P Utility, 9-72
 - CTDCP2A Utility, 9-77
 - CTDDELRP Utility, 9-79, 9-81
 - CTORSTM Utility, 9-99
 - CTTACP Utility, 9-106
 - CTTRCV Utility, 9-141
 - CTTRPT Utility, 9-145
 - CTTRTM Utility, 9-169
 - CTTSPL Utility, 9-174
 - CTTSTK Utility, 9-177, 9-178
 - CTTVTM Utility, 9-196
 - CTVDELI Utility, 9-94, 9-95
 - Examples, 9-104
 - Field Names, D-1
 - Format (CTT), 9-102
 - Record Selection (CTT), 9-102
 - Record Selection Logic, 9-102
- Increasing. *See* Enlarging
- Index
 - CTD Reports, 4-37
- Index File
 - Deleting from DASD, 9-94
 - Integrity (CTT), 9-120
 - IOA Access Method, 2-27, 9-86
 - Media Database, 7-12; 9-109
 - Parameters (CTT), 9-112
 - Printing (CTT), 9-121
 - Rebuild Stacking Index, 9-118
 - Rebuilding (CTT), 7-24
 - Stacking Statistics File, 7-18
- Index Record
 - CTTIOS Macro, 7-49
 - Types, 7-13
- Indexing Reports
 - ACIF Interface, 4-44
- Information Message. *See* Message or Messages
- INIT Mode
 - CTTINIT Procedure, 7-3
- INITIAL USE Attribute
 - IBM Robot Support, 7-27
- Initialization
 - CONTROL-D, 4-1
 - CONTROL-M, 3-1
 - CONTROL-O, 5-1
 - CONTROL-T, 7-2
 - CONTROL-V, 4-1
 - Tape Initialization, 9-179
 - Trace File, 9-108
- INITT Function
 - CTTTPI Utility, 9-186
- Inline Resource Option
 - AFP Printing, 4-39
- INOUT Attribute
 - Global Variable Member (CTO), 5-13
- INP Parameter
 - CTTDLD Utility, 9-122; 9-123
- INPUT Attribute
 - Global Variable Member (CTO), 5-13
- Inquire/Update Screen (CTT)
 - Profile Variable, 2-41
- Installation and Customization Engine. *See* ICE
- Installation Working Date, 4-8
- IN-TABLE Parameter
 - Utility CTMBGRP, 9-43
- Integrity
 - Access Method Index, 9-30
 - Data Files (CTT), 9-119
 - Index Files (CTT), 9-120
 - IOA Access Method, 9-29
 - Media Database, 7-9; 7-19, 9-127, 9-132; 9-172
 - Rule Activity File (CTB), 9-67
 - Variable Files (CTB), 9-61
- Integrity Error
 - Correction, 7-20

- INTENS Parameter
 - Display Format Member, 2-25
- Inter System Communications (CTM), 3-32
- Interface
 - CONTROL-T to MVS, 7-2
 - CTM and CTD, 4-27
 - CTR to CTT, 7-43
- Interfaces
 - CA-7 (UCC7), 4-28
- Internal Data Areas
 - CONTROL-O, 5-17
- INTERVAL Command
 - CMEM, 3-30
 - CONTROL-D Monitor, 4-2
 - CONTROL-M Monitor, 3-2
 - CONTROL-O Monitor, 5-15
- INTERVAL Parameter
 - IOADBSR Utility, 9-26
- INTRDR DD Statement
 - CONTROL-M Monitor, 3-25
- Inventory Report
 - Vault Management, 9-194
- Invoking CONTROL-B, 6-6
 - Utilities, 9-4
- IOA
 - Ad hoc Maintenance, 8-4
 - Application Program Names, A-1
 - Color Support, 2-24, 2-25
 - Customizing Screens, 2-17
 - Dataset Formatting Utilities, B-1
 - Display Format Members, 2-20
 - Dynamic Destination Table, 2-49
 - Introduction, 1-1
 - Limited Access, 2-8
 - Online Monitor Activation, 2-5
 - Periodic Maintenance, 8-3
 - Profiles, 2-31
 - Utilities, 9-1
- IOA Access Method, 1-7
 - Dual Mirror Image Files, 2-30
 - File Allocation, 2-28, 9-18
 - File Definition, 2-30, 9-19
 - File Formatting, 9-18
 - File Recovery, 4-83
 - File Structure, 2-27
 - Files (CTB), 2-29
 - Files (CTD), 2-29
 - Index Integrity, 9-30, 9-86
 - Index of CTD Files, 9-86
 - Integrity, 9-29
 - Loading Files, 9-31
 - Naming Conventions, 2-28
 - Overview, 1-7, 2-27
 - Printing Files, 9-32
 - Recovery, 2-31, 9-27
 - Sorting Files, 4-83, 9-25
 - Unloading Files, 9-33
 - Utilities, 2-29
 - Utilities (CTB), 9-4
 - Utilities (CTD), 9-3
 - Variable File (CTB), 9-24
- IOA Core
 - Description, 1-7
- IOA Log
 - Copying, 9-16
- IOA LOG File
 - Expanding, 2-50
- IOA Menu. *See* Primary Option Menu
- IOA Online. *See* Online Facility
- IOA Resources
 - Mirror File, 3-26
- IOA Screen Definition
 - Constant Blocks, 2-18
- IOA5DZN SMP/E Zone, 8-1
- IOABLCAL Utility, 9-1
- IOACLRES Utility, 4-29
 - Clean Conditions/Resources File, 9-7
 - CONTROL-M Maintenance, 3-6
- IOACMEML Member
 - CMEM Rule List, 5-25
 - Rule Table List (CMEM), 3-27, 5-3
- IOACND Utility
 - Manage Conditions/Resources, 9-10
- IOACOPRS Utility
 - Copy/Resize Cond/Res File, 9-14
 - Restore Conditions/Resources File, 3-26
- IOACPLOG Utility
 - Copy Log File, 9-16
- IOADBF Utility, 9-50
 - Access Method Files, 9-18
 - IOA Access Method, 2-29
- IOADBSR Utility
 - Sort Access Method Files, 4-83, 9-25
- IOADCPY Utility
 - Access Method File Recovery, 9-27
 - CONTROL-D File Recovery, 4-83
 - IOA Access Method, 2-30
- IOADEST Member
 - Destination Table, 2-49
- IOADESTJ Member, 2-49
- IOADFLTC Member
 - Updating, 2-46
- IOADIG Utility
 - Access Method File Integrity, 9-29
 - IOA Access Method, 2-29
- IOADII Utility
 - Access Method Index Integrity, 9-30
 - IOA Access Method, 2-29
- IOADLD Utility
 - IOA Access Method, 2-30
 - Load Access Method Files, 9-31
- IOADPT Utility
 - IOA Access Method, 2-30
 - Print Access Method Files, 9-32
- IOADUL Utility
 - IOA Access Method, 2-30

Index

- Unload Access Method Files, 9-33
 - IOAEXSMP Member
 - SMS Managed Tape Library, 7-26
 - IOAFRRRES Utility
 - Restore Conditions/Resources File, 3-26
 - IOAINSJ Member
 - IOA Maintenance, 8-8
 - IOAIPLCM Member
 - CMEM Rule List, 5-1, 5-7
 - IOAISPF CLIST
 - Entry to Online Facility, 2-2
 - IOALDNRS Utility, 4-29
 - CONTROL-M Maintenance, 3-6
 - Manual Conditions File, 9-35
 - Manual Conditions List, 4-3
 - Multiple Runs (CTM), 3-15
 - IOANOTE Utility
 - Sending a Message, 9-38
 - IOAONL CLIST
 - Entry of Online Facility, 2-2
 - IOAPROPJ Member
 - IOA Maintenance, 8-7
 - IOASPARM Member
 - Migration Media Definition, 4-59
 - IOATEST Utility
 - Simulation, 9-39
 - IOATSO CLIST
 - Entry to Online Facility, 2-2
 - IOAVERFY Utility, 9-41
 - IOAVMON
 - Operator Command, 2-7
 - VTAM Monitor, 2-7
 - IOAX007 Exit
 - Customization, 3-35
 - Inter-System Communications (CTM), 3-33
 - IOAX009 Exit
 - Problem Determination, 2-7
 - IOAXTSO CLIST
 - Entry to Online Facility, 2-2
 - IOERPRM Keyword
 - Volume Record, D-6
 - IOERPRMC Keyword
 - Volume Record, D-6
 - IOERTMP Keyword
 - Volume Record, D-6
 - IOERTMPC Keyword
 - Volume Record, D-6
 - IOEWPRM Keyword
 - Volume Record, D-6
 - IOEWPRMC Keyword
 - Volume Record, D-6
 - IOEWTMP Keyword
 - Volume Record, D-6
 - IOEWTMPC Keyword
 - Volume Record, D-6
 - IPL
 - Automation (CTO), 5-5
 - CONTROL-O IPL Process, 5-8
 - Starting CONTROL-O, 5-1
 - IPLRULES Member
 - CONTROL-O Rule List, 5-7
 - IRMA
 - Color Support, 2-26
 - ISCALOC Member
 - Inter-System Communications, 3-33
 - ISCPARM Member (CTM), 3-33
 - ISPF
 - Color Support, 2-25
 - Editing JCL, 2-46
 - Entry to Online Facility, 2-2
 - Online Facility Members, E-12
 - ISPFBOTL
 - Profile Variable, 2-43
- J—
- Japanese
 - Language Support, 8-2
 - JCL
 - Editing via ISPF, 2-46
 - JCLEXPDT Keyword
 - Dataset Record, D-4
 - JDEPARMS Member
 - Sample DJDE Parameters, 4-43
 - JES
 - IPL Automation (CTO), 5-5
 - JES2
 - Multi-Chunk Printing, 4-35
 - JES2-DEST Printer Setting, 4-37
 - JES3
 - Multi-Chunk Printing, 4-36
 - JES3-CLASS Printer Setting, 4-37
 - Job
 - Dependency on CTD Report, 4-25
 - Job Ordering
 - CONTROL-M, 3-4
 - Job Rerun
 - Profile Variable (CTM), 2-37
 - Job Rerun Screen
 - Profile Variable (CTD), 2-37
 - Job Scheduling
 - Category Field, 4-26
 - Job Statistics
 - Utility CTMJSA, 9-49
 - Job Status Screen
 - Profile Variable (CTM), 2-38, 2-44
 - Job Step
 - Invoking CONTROL-B, 6-7
 - Job Step Invocation
 - CONTROL-B, 6-8
 - Job Submission
 - CTMAJO routines (CTD), 4-33

- JOBBBDM Job**
 Rebuild Access Method
 Variable File, 9-24
- JOBBDBM Job**
 CTBDBIB Utility, 9-53
- JOBBDBV Job**
 CTBDBIB Utility, 9-53
 Rebuild Access Method
 Variable File, 9-24
- JOBBGRP Job**
 CTBDBIB Utility, 9-53
 Rebuild Group File, 9-24
- JOBBJAF Job**
 CTBDBIB Utility, 9-53
 Rebuild Rule Activity
 File, 9-24
- JOBBREP Job**
 CTBDBIB Utility, 9-53
 Rebuild Report File, 9-24
- JOBFABF Job**
 Format Active Balancing
 File, 9-24
- JOFBKBP Job**
 Format Backup File, 9-24
- JOBFGRP Job**
 Format Group File, 9-24
- JOBFJAF Job**
 Format Rule Activity
 File, 9-24
- JOBFMOD Job**
 IOA Access Method
 Variable File, 9-24
- JOBFREP Job**
 Format Report File, 9-24
- JOBFVAR Job**
 Format IOA Access
 Method Variable File,
 9-24
- JOBID Keyword**
 Trace Record, D-11
- JOBNAME Parameter**
 CTTGVL Utility, 9-125
 IGNORE/SELECT
 Statements (CTM), 3-11
- Jobs Dependency Network
 File**
 Expanding (CTM), 3-36
- K—
- KANJI Parameter**
 @FIELD Display Format
 Line, 2-23
- KEEPDSN Parameter**
 CTDELRP Utility, 9-81
- Key. See Index**
- KEY Parameter**
 CTTIOS Macro, 7-49
- Keyed Access**
 API Interface, 7-52
- KEYFROM Parameter**
 IOADPT Utility, 9-32
- KEYLEN Parameter**
 CTTCRDF Utility, 9-112,
 9-117
 CTTDBF Utility, 9-117
 CTTIOS Macro, 7-50
 IOADBF Utility, 9-22
- KEYTO Parameter**
 IOADPT Utility, 9-32
- Key-Type**
 Media Database Index,
 7-12
- KEYTYPE Parameter**
 CTTIOS Macro, 7-50
- Keywords**
 Dataset Records, D-2
 Stacking Records, D-10
 Trace Data, D-12
 Trace Records, D-11
 Volume Records, D-6
- KEYxxx Members**
 PARM library, 9-118
- KODAK OPTISTAR
 Storage System. See OSS**
- KODAK/DataWare OSS**
 Updating, 4-67
- KSL Utility**
 Mission Scheduling
 (CTD), 4-16
- L—
- LABEL Parameter**
 CTTGVL Utility, 9-126
 Exit CTDX005, 4-50
- LABTYPE Parameter**
 CTTTPI Utility, 9-187
- LACCDT Keyword**
 Volume Record, D-6
- LACCJBN Keyword**
 Volume Record, D-6
- LACCTM Keyword**
 Volume Record, D-6
- Language Support**
 IOA Products, 8-2
- Laser Printers**
 AFP Printing, 4-38
- LAST Parameter**
 CTTDLI Utility, 9-123
- LASTACCS Keyword**
 Dataset Record, D-4
- LBLNUM Keyword**
 Volume Record, D-6
- LBLNUM Parameter**
 CTTMUP Utility, 9-137
- LBLROUTC Parameter**
 Modification (CTT), 7-4
- LBLTYP Keyword**
 Volume Record, D-6
- LCLNDT Keyword**
 Volume Record, D-6
- LDSFEET Keyword**
 Volume Record, D-6
- LENGTH Parameter**
 CTTTRPT Utility, 9-148
- Limited Access. See
 Security. See
 Customization**
- LINECMD Parameter**
 Automation Option
 Menu, 5-19
- LINESIZE Parameter**
 CTTACP Utility, 9-107
 CTTTRPT Utility, 9-145
- LINK Command**
 CTTGVL Utility, 9-125
- LIST Parameter**
 CTTTRM Utility, 9-167
 IOAVERIFY Utility, 9-42
- LOADGLOBAL Command**
 Global Variable Members
 (CTO), 5-14
- Loading. See also item or
 feature to be loaded**
 CMEM Rules, 3-27
 Global Variables (CTO),
 5-12

Index

- IOA Access Method Files, 9-31
- Media Database, 9-115
- Stacking Statistics File, 9-115
- LOADTREE Command
 - IOA Online Monitor, 4-3
 - Recipient Tree Loading (CTD), 4-3
- LOCATION Keyword
 - Volume Record, D-6
- LOCSEQ Keyword
 - Volume Record, D-7
- LODOSSDB Job
 - OSS Database Update, 4-68
- Log. *See* IOA Log. *See* Automation Log
- LOG Command
 - Automation Log (CTO), 5-12
 - CMEM, 3-29
- Log File
 - Copying the IOA Log, 9-16
- Logical Field Names
 - CTT Repository, D-1
- Logical Recovery
 - Description, 7-25
 - Media Database, 9-139
 - vs. Physical Recovery, 7-25
- LOGSIZE Parameter
 - IOA Log File, 2-50
- LONG Parameter
 - @LINE Display Format Line, 2-22
- LREC Parameter
 - CTTCRDF Utility, 9-112
- LRECL Keyword
 - Dataset Record, D-4
- LRECL Parameter
 - Exit CTDX005, 4-51
 - IOADBF Utility, 9-22
- LTH Parameter
 - @FIELD Display Format Line, 2-22
 - @STYLE Display Format Line, 2-21
- L-type Index
 - Media Database, 7-13

—M—

- M Parameter
 - IOADBF Utility, 9-18
 - IOADBSR Utility, 9-25
- Main Menu. *See* Primary Option Menu
- MAINDAY Scheduling Table
 - CTM, 3-6
 - Customization, 3-13
- Maintenance
 - Active Jobs File, 9-45
 - Ad-hoc, 8-4
 - CONTROL-D, 4-7
 - Periodic, 8-3
 - Report List Files, 4-79
 - User Reports List File, 4-82
- Maintenance Jobs
 - CONTROL-M, 3-6
- Management Class
 - Adjusting, 7-38
 - DF/SMS Support, 7-37
- Manual Conditions
 - Default Date, 2-42
 - Profile Variable, 2-38
 - Reloading (CTD), 4-3
- Manual Conditions List
 - Building, 9-35
 - Utility IOALDNRS, 4-29
- Manual Update
 - Media Database, 9-131
- Mapping
 - Media Information, 9-180
- MARGINS Parameter
 - CTTRPT Utility, 9-145
- Mask Characters
 - INCLUDE/EXCLUDE Statements, 9-102
- MAXCONN Parameter
 - FileTek Media, 4-68
- MAXUSER MVS Parameter, 2-6, 4-4
- MDBUPDAT Parameter
 - CTTTPI Utility, 9-185
- Media Capacity
 - CART Media, 4-60
- Media Control
 - Archive Server, 4-69

- Media Database
 - Access via Base Level API, 7-51
 - Adding Volumes, 9-122
 - API Access, 7-47
 - Backup and Recovery, 7-23
 - Backup Utility, 9-192
 - Checking Integrity (CTT), 7-9
 - Converting Volumes, 9-122
 - Creation, 9-111
 - Daily Backup, 7-9
 - Deleting Volumes, 9-122
 - Enlarging, 7-21
 - ExLM Access, 7-31
 - Formatting, 9-113, 9-117
 - Index Integrity, 9-120
 - Integrity, 7-19, 9-119, 9-127, 9-132
 - Keyed Access (API), 7-55
 - Loading, 9-115
 - Manual Update, 7-21
 - Merge Utility, 9-129
 - Non-Keyed Access, 7-58
 - Printing, 9-121
 - Rebuild Index, 9-109
 - Record Types, 7-10
 - Recovery, 7-20, 9-139
 - Selective Recovery, 7-25
 - Split/Merge Process, 9-172
 - Statistics, 9-127
 - Structure, 7-9
 - Unloading, 9-116
 - Update Utility, 9-131
 - Updates, 7-17
 - Utilities, 9-5, 9-6
- Media Definition
 - CART Media, 4-67
 - FileTek Storage Machine, 4-68
 - Migration Mission, 4-59
- Media Information
 - Display, 4-72
 - Tape Mapping, 9-180
- MEDIA Keyword
 - Volume Record, D-7
- Media Management
 - CART Media, 4-60

- MEDIA Parameter
 - CTDLD Utility, 9-123
 - CTTTPI Utility, 9-187
- MEMBER Parameter
 - IGNORE/SELECT Statements (CTM), 3-11
- MEMNAME Member
 - CTD Scheduling, 4-23
- MEMOREX TELEX Tape Library
 - CONTROL-T Interface, 7-32
- Memory Requirement
 - IOA Online, 2-4
- Menu. *See also* Primary Option Menu
- MENU Member, 2-15
- Menus
 - Automation Options (CTO), 5-19
- Merging
 - MDB Records, 9-129
- Message
 - Color of Display, 2-39
 - Modification, 2-47
- Message Display
 - Cartridge Tape Subsystems, 7-33
- MESSAGE Parameter
 - CTMMMSG Macro, 2-47
- Message Statistics Facility (CTO), 5-9
 - Copy Statistics File, 9-98
 - Profile Variable, 2-37
 - Report Utility, 9-99
- Message to Log File
 - IOANOTE Utility, 9-38
- Messages
 - CME123I, 3-30
 - CME150I, 3-32, 5-18
 - CME15DI, 3-32
 - CME251I, 3-30
 - CTD100I, 4-1
 - CTD107I, 4-4, 4-5
 - CTD113W, 4-6
 - CTD120I, 4-4
 - CTD123I, 4-2
 - CTD130I, 4-37
 - CTD136I, 4-4
 - CTD139I, 4-1
 - CTD140I, 4-5
 - CTD160I, 4-3
 - CTD271I, 4-5
 - CTD426W, 4-6
 - CTD427W, 4-6
 - CTD428W, 4-6
 - CTD533S, 9-68
 - CTD775I, 4-1
 - CTD779I, 4-4, 4-6
 - CTDF63W, 9-68
 - CTM100I, 3-1
 - CTM107I, 3-1
 - CTM120I, 3-1
 - CTM123I, 3-2
 - CTM227I, 4-2
 - CTM231I, 4-5
 - CTM645I, 2-6
 - CTM646I, 2-6
 - CTM717E, 2-40
 - CTM777I, 2-5
 - CTM778I, 2-5
 - CTM786I, 4-3
 - CTM7A0I, 2-8
 - CTM7A1I, 2-8
 - CTM7ACI, 2-8
 - CTM7B0I, 2-8
 - CTM7B1I, 2-8
 - CTO123I, 5-16
 - CTO147I, 5-8
 - CTO15DI, 5-18
 - CTO184I, 9-98
 - CTO189I, 9-98
 - CTO240W, 5-10
 - CTO241E, 5-10
 - CTO251I, 5-16
 - CTO297I, 5-11
 - CTT200S, 7-51, 7-71
 - IOA10BI, 4-6
 - IOA128I, 4-73
- MF Parameter
 - CTTIOS Macro, 7-50
- MIGCART Member
 - Migration Skeleton, 4-60
- MIGDASD Member
 - Migration Skeleton, 4-60
- MIGORDER KSL Utility
 - Migration Missions, 4-16
- MIGRAT Step
 - Migration Mission, 4-58
 - Migration Skeletons, 4-60
- MIGRATE IN PROCESS Status
 - Migration Mission, 4-62
- Migrated User Report List
 - Backup Utility, 9-87
 - Cleaning, 9-93
 - CTDDELRP Utility, 4-78
 - Description, 4-77
- Migration Mission
 - Abends, 4-62
 - Decollation Dependency, 4-57
 - Dual Migration, 4-59
 - Exception Handling, 4-62
 - Media Definition, 4-59
 - Overview, 4-56
 - Scheduling, 4-13, 4-56
 - Scheduling Dates, 4-57
 - Skeleton Job Parameters, 4-58
 - Workflow, 4-57
- Mirror Image File
 - CONTROL-M Monitor, 3-26
 - Recovery, 9-27
 - Support, 2-30
- Mission
 - Scheduling
 - Workflow, 4-17
 - Workflow, 4-17
- Mission List Members (CTD), 4-14
- Mission List Screen (CTB)
 - Profile Variable, 2-43
- MISSION Parameter
 - Invoking CONTROL-B, 6-7
- Mission Scheduling
 - CONTROL-D, 4-12
 - Manual, 4-16
 - Migration Missions, 4-57
 - Production Control Systems, 4-27
 - Restore Missions, 4-63
- MODE Installation Parameter
 - Modification (CTT), 7-4
- MODE Parameter
 - CMEM Rule, 3-29

Index

- CTDCLHIS Utility, 9-76
- CTDDELRP Utility, 9-80
- CTDLD Utility, 9-123
- CTTIDB Utility, 9-128
- CTTINIT Procedure, 7-3
- CTTMER Utility, 9-129
- CTTMUP Utility, 9-131
- CTTPARM Member, 9-165
- CTTRCV Utility, 9-140
- CTTRTM Utility, 9-166
- CTTSPL Utility, 9-175
- CTTSTK Utility, 9-177
- CTTTPI Utility, 9-184
- CTTUDB Utility, 9-193
- CTTVTM Utility, 9-195, 9-196
- CTVDELI Utility, 9-94
- Rule Operation (CTO), 5-12
- Modification. *See* Customization
- Monitor
 - CMEM, 3-3, 5-25
 - CONTROL-D, 4-1
 - CONTROL-M, 3-1
 - CONTROL-O, 5-25
 - IOA Online, 1-5, 2-3, 2-4, 2-5
 - IOA Products, 1-5
 - Replacing (CTO), 5-2
 - Sleeping Interval (CTD), 4-2
 - Sleeping Interval (CTM), 3-2
 - VTAM, 2-7
- Mount Messages
 - Modification, 7-2
- MOVEDATE Keyword
 - Volume Record, D-7
- Moving
 - Database Records, 9-172
- MARGIN Parameter
 - Utility CTTMER, 9-129
- MSGCLASS Parameter
 - Held Output Class (CTM), 3-25
- MSGFMT Parameter
 - Modification (CTT), 7-4
- MTU Parameter
 - CTTIDB Utility, 9-127
 - CTTUDB Utility, 9-193
- Multi Chunk Method
 - Bundle Printing, 4-35
 - JES2, 4-35
 - JES3, 4-36
- Multi-CPU Sites
 - Inter-System Communications (CTM), 3-32
- Multi-Volume Chain
 - Chain Link Error, 9-127
 - Manual Update, 7-21
 - RTNTYPE* Parameter, 9-164
 - Volume/Dataset Records, 7-16
- Multi-Volume Dataset, 7-15
- MVS
 - ASVT, 2-6, 4-4
 - Catalog Conversion, 9-123
 - Interface (CTT), 7-2, 7-4
 - IPL Automation (CTO), 5-5
 - MVS Gateway for Enterprise Controlstation. *See* ECSGATE
- N—
- NAME Parameter
 - @FIELD Display Format Line, 2-22
 - CTTRPT Utility, 9-145
 - Migration Media, 4-59
- Naming Conventions
 - IOA Access Method, 2-28
- NBLKS Parameter
 - CTTCRDF Utility, 9-111
- New Day Procedure
 - Changing Time, 3-24
 - CONTROL-B, 6-1
 - Mission Scheduling, 4-14
 - Scheduling Generic Missions, 4-21
 - Starting, 4-6
- New Day Processing
 - Conditions/Resources (CTM/D), 4-29
 - CONTROL-D, 4-6
 - CONTROL-M, 3-4, 4-23
 - CONTROL-T, 7-7
 - CTDCA2P, 4-76
 - Description, 1-6
 - Implementation (CTM), 3-9
 - Programs (CTD), 4-10
 - Programs (CTM), 3-16
 - Rule Refresh Option (CTT), 7-8
 - Sample Components (CTM), 3-5
- NEWAJF File Statement
 - Utility CTMCAJF, 9-47
- NEWCONLIST Command
 - CMEM, 3-28
- NEWDEST Command
 - Destination Table, 2-49
- NEWGLOB Job
 - Global Library Compression, 5-15
- NEWNAME Parameter
 - CTTMUP Utility, 9-134
- NEWRES Parameter
 - IOACOPRS Utility, 9-15
- NEWSECDEF Command
 - CMEM, 3-30
 - CONTROL-O Security Cache, 5-16
- NEWSNC Parameter
 - IOACOPRS Utility, 9-15
- NEXTVOL Keyword
 - Volume Record, D-7
- Night Plan Report
 - Production Control Systems, 4-28
- NLASKOP Parameter
 - Modification (CTT), 7-4
- NOCHECK Parameter
 - OSS Migration, 4-61
- NOFORCE Parameter
 - CONTROL-O Procedure, 5-2
- NOINDEX Parameter
 - CTDDELRP Utility, 9-81
- Non-Keyed Access
 - Base Level API (CTT), 7-53

- NOPRINT Parameter
 CTBJAFDL Utility, 9-65
 NOREPORT Parameter
 CTDDELRP Utility, 9-81
 Normal Mode
 CONTROL-T, 7-5
 NOSEP JES Parameter
 Multi-Chunk Printing,
 4-35
 NOTOK Status
 Backup Missions, 4-56
 Migration Mission, 4-62
 NOTRST Parameter
 CTDDELRP Utility, 9-80
 NOTWD Parameter
 CTDDELRP Utility, 9-80
 NRECS Parameter
 CTTCRDF Utility, 9-111,
 9-117
 CTTDBF Utility, 9-117
 Null Values
 CTTMUP Utility, 9-137
 NUMTAPE Parameter
 CTTTPI Utility, 9-187,
 9-188
 NUMERR Parameter
 CTTTPI Utility, 9-188
 NUMGEN Parameter
 CTBDBVCG Utility, 9-55
 CTBDBVCP Utility, 9-57
- O—
- ODATE
 CTDCA2P Utility, 4-76
 CTDCP2A Utility, 4-76
 Migrated Report List,
 4-77
 Mission Scheduling, 4-17
 Utility CTMJSA, 9-49
 OFFSET Parameter
 @FIELD Display Format
 Line, 2-23
 OLDAJF File Statement
 Utility CTMCAJF, 9-47
 OLDRES Parameter
 IOACOPRS Utility, 9-14
 OLDSNC Parameter
 IOACOPRS Utility, 9-15
- One-Chunk Method
 Bundle Printing (CTD),
 4-35
 One-Outgroup Method
 Bundle Printing, 4-36
 Online Facility
 Command Customization,
 C-1
 Command Modification,
 2-13
 Cross Memory Interfaces,
 2-1
 Cross-Reference of
 Options, E-1
 Customization, 2-8
 Entry, 2-2
 Environments, 2-1
 New Recipient Tree, 4-3
 PFKey Definition, 2-13
 Online Monitor
 Balance Workload, 2-5
 Deactivation, 2-6
 Description, 1-5, 2-3
 IOA, 2-4, 2-5
 Number of Users, 2-5
 Online Viewing
 Archive Server, 4-66
 OPEN Function
 Base Level API (CTT),
 7-47
 Opening. *See* Starting
 Operating Status
 Parameters (CTT), 7-4
 Operation Mode
 Checking (CTT), 7-8
 CMEM, 3-29
 CONTROL-O Rules, 5-12
 Dormant (CTT), 7-4
 Normal (CTT), 7-5
 Suspended (CTT), 7-5
 Operator Commands
 F CONTROLLO,
 C=library(table), 5-4
 F
 CONTROLLO,INTERVAL=
 , 5-15
 F CONTROLLO,O/F, 5-3
 F
 CTMCMEM,INTERVAL=
 3-30
- F IOASMON,DISPLAY
 MEDIA=, 4-72
 F IOASMON,SET
 DEBUG=, 4-73
 F IOASMON,START
 MEDIA, 4-69
 F IOASMON,STOP
 MEDIA, 4-69
 F
 CONTROLD,INTERVAL,
 4-2
 F
 CONTROLD,LOADTREE,
 4-3
 F
 CONTROLD,STARTGEN,
 4-1, 4-22
 F
 CONTROLD,STARTPRT,
 4-37
 F
 CONTROLD,STOPGEN,
 4-4, 4-22
 F CONTROLD,STOPPRT,
 4-37
 F
 CONTROLM,INTERVAL,
 3-2
 F
 CONTROLM,NEWCONLI
 ST, 3-28
 F CONTROLLO,
 LOADGLOBAL, 5-14
 F
 CONTROLLO,AUTOLOG,
 5-10
 F CONTROLLO,D=, 5-5
 F CONTROLLO,DEBUG=,
 5-17
 F CONTROLLO,LOG=,
 5-12
 F
 CONTROLLO,NEWSECDE
 F, 5-16
 F
 CONTROLLO,O=ALL,REB
 UILD, 5-9
 F
 CONTROLLO,RESETSTA
 T, 5-9
 F CONTROLLO,SNAP,
 5-17

Index

- F
CONTROLO,STARTSTAT
, 5-9
- F
CONTROLO,STARTSTAT
S, 9-98
- F
CONTROLO,STOPSTAT,
5-9
- F
CONTROLO,STOPSTATS
, 9-98
- F
CONTROLO,WATERMA
RKS, 5-18
- F
CONTROLO,WRITEGLO
BAL, 5-12
- F
CONTROL_x,NEWDEST,
2-49
- F CTMCMEM, C=, 3-27
- F CTMCMEM,D=, 3-28
- F CTMCMEM,DEBUG=,
3-31
- F CTMCMEM,DISPLAY,
3-29
- F CTMCMEM,LOG=,
3-29
- F
CTMCMEM,NEWSECDE
F, 3-30
- F CTMCMEM,SNAP,
3-31
- F CTMCMEM,STOP, 3-3
- F
CTMCMEM,WATERMAR
KS, 3-32
- F
IOAOMON_x,LOADTREE,
4-3
- F IOAVMON,D, 2-8
- F
monitor_name,DISPLAY,
2-6
- P CONTROLM, 3-1
- P CTMCMEM, 3-3
- P IOAOMON_x, 2-8
- P monitor_name, 2-6
- S IOASMON,DEBUG=,
4-73
- S CONTROLM, 3-1
- S CONTROLO,DEBUG=,
5-17
- S
CONTROLO,TYPE=CTO
CMEM, 5-26
- S CTMCMEM, 3-3; 3-27
- S CTMCMEM,DEBUG=,
3-31
- S CTTINIT, 7-2
- S IOAVMON, 2-7
- Operators
INCLUDE/EXCLUDE
Statements, 9-102
- Optical Disk
Migration Missions, 4-61
- Optical Library Dataserver
Migration Skeleton, 4-61
- Optical Storage System. *See*
OSS
- Option Code
Customization, 2-10
- OPTION Parameter
Automation Option
Menu, 5-19
- Options
Primary Option Menu,
2-16
- OPTISTAR Storage System.
See OSS
- OPTLIST Parameter
Automation Option
Menu, 5-23
- ORDER Parameter
CONTROL-O Procedure,
5-2
- Rule Lists (CTO), 5-7
- Rule Table Loading
(CTT), 7-6
- S CTMCMEM Command,
3-27
- OSS
Archive Server, 4-66
- Stopping/Starting a
Device, 4-69
- OSS Database
Adjustments to Migration
Skeleton, 4-60
- Migration of Extract DB,
4-61
- Updating, 4-67
- OSSDBEXT Step
OSS Migration, 4-60
- OSSDBVL Parameter
OSS Migration, 4-61
- OUT Parameter
Decollation Missions,
4-34, 4-52, 4-57
- Job Dependency, 4-25
- OUTDD Parameter
CTTRPT Utility, 9-146
- OUTGROUP Parameter
CONTROL-D Chunks,
4-38
- OUTGRP Parameter
Bundle Printing (CTD),
4-35
- OUTLIST Parameter
CONTROL-O
Initialization, 5-1
- OUTPARMS Library
Global Printing
Characteristics, 4-47
- OUTPUT CDAM Parameter
AFP Printing, 4-39
- OUTPUT Statement
AFP Printing, 4-38
- CTTACP Utility, 9-106
- Printers Control Monitor
(CTD), 4-38
- OUT-TABLE Parameter
Utility CTMBGRP, 9-43
- OVERJCL Parameter
Modification (CTT), 7-4
- Overlays
AFPDS Printing, 4-43
- Overriding
Allocation Member, 2-12
- OWNER Parameter
CTTTPI Utility, 9-187

—P—

- P Option
Active Missions Screen,
4-37
- Packaging
IOA, 8-1
- Page Break
AFP Printing, 4-39

- Page Markers
 - AFP Printing, 4-40
- Page Mode Output
 - AFP Printing, 4-39
- Page Segments
 - AFPPDS Printing, 4-43
- PAGEDEF Parameter
 - AFP Printing, 4-38
 - CDAM, 4-39
- PAGESIZE Parameter
 - CTTRPT Utility, 9-145
- Panel. *See* Screens
- Parameters
 - ACIF, 4-44
 - CDAM, 4-39
 - CTTINIT Procedure, 7-2
 - DJDE (XEROX Printers), 4-42
 - Migration Media, 4-59
 - Migration Skeleton, 4-58
 - Restore Skeleton Job, 4-64
- PARM Parameter
 - Automation Option Menu, 5-21
 - CTMMMSG Macro, 2-47
- PARMLIB Parameter
 - IOADBSR Utility, 9-25
- PASSWORD Parameter
 - FileTek Media, 4-68
- PDATELTH
 - Profile Variable, 2-40
- Periodic Maintenance
 - IOA Products, 8-3
- Permanent Report List
 - Copy from Active, 9-72
 - Copy to Active, 9-77
- Permanent User Report List
 - CTDCA2P Utility, 4-76
 - CTDCP2A Utility, 4-76
 - Description, 4-75
- PF03 Function Key
 - Profile Variable, 2-42
- PFKey Definition, C-1
 - Format, 2-13
 - Modification, 2-13
- PFKey Members
 - Online Facility, E-1
- PGM Parameter
 - Transaction Member, 2-9
- PGMCTx Member
 - Program List Member, 2-10
- Physical Recovery
 - Media Database, 7-24, 9-139
 - vs. Logical Recovery, 7-25
- Platter Prefix
 - Migration Missions, 4-61
- PNRSDTYP
 - Profile Variable, 2-42
- Pool Definition
 - CA-ASM2 Support, 7-36
 - DD Statement (CTT), 7-5
 - Initializing (CTT), 7-3
- Pool Definitions
 - HSC Scratch Pools, 7-31
- Post Processing Parameters
 - Backup Missions, 4-55
 - Migration Missions, 4-58
 - Restore Mission, 4-65
- Postscript Output
 - AFP Printing, 4-40
- PREF Parameter
 - @FIELD Display Format Line, 2-22
- PREFIX Parameter
 - CTDCCU Utility, 9-74
 - Exit CTDX005, 4-50
- Prerequisite Condition
 - Adding/Deleting/Verifying, 9-10
 - Backup Mission Dependency, 4-52
 - Date Related Erasure, 3-10
 - Deleting, 9-7
 - Description, 1-8
 - Inter-System Communications (CTM), 3-32
 - Job Dependency, 4-28
 - Manual Conditions, 4-3
 - Migration Mission, 4-57
 - Migration Skeleton, 4-58
 - Printing Mission Dependency, 4-34
- Presentation Mode
 - Profile Variables, 2-43
- PREVVOL Keyword
 - Volume Record, D-7
- Primary Extent Dataset
 - IOA Access Method, 2-28
- Primary Key
 - CTTRPT Utility, 9-143
- Primary Option Menu
 - Customization, 2-15
 - Description, 1-3
 - Format, 2-14, 2-16
- PRIMARY Parameter
 - CTDUPBKP Utility, 9-89
 - Exit CTDX005, 4-51
- Print Destination
 - Active User Report List, 4-76
 - Report List Files, 4-75
- PRINT Parameter
 - CTBJAFDL Utility, 9-65
- PRINTED Status
 - Active Report List, 4-77
- PRINTED-WAIT BKP Status
 - Active User Report List, 4-77
- Printer Control
 - Bundle Printing (CTD), 4-34
- PRINTER Parameter
 - Bundle Printing (CTD), 4-35
 - Opening/Closing Printers, 4-36
- Printers
 - AFP Printers, 4-38
 - Opening/Closing, 4-36
 - XEROX, 4-42
- Printers Control Monitor
 - Activation CTD, 4-1
 - New Day Processing, 4-6
 - OUTPUT Statement, 4-38
 - XEROX Printers, 4-43
- Printing. *See also* Printers Control Monitor
 - Data File (CTT), 9-121
 - Database Records (CTT), 9-121
 - Global Control, 4-47

Index

- Index File (CTT), 9-121
- IOA Access Method Files, 9-32
- Media Database, 9-121
- Stacking Stacking File, 9-121
- Printing Mission
 - ACIF Parameter Member, 4-45
 - CHUNKSIZE Parameter, 4-35
 - Decollation Missions, 4-34
 - Overview, 4-30
 - Profile Variable, 2-36
 - Scheduling, 4-13
 - Scheduling Dates, 4-34
 - Workflow, 4-31, 4-37
- Printing to a File
 - Exit CTDX005, 4-49
- PRIVATE Parameter
 - Exit CTDX005, 4-50
- Production Control System
 - CTDRRQ Program, 4-28
- PROF Parameter
 - Profile Attributes, 2-34
- Profile
 - Description, 2-31
 - Members, 2-32
 - Profile Member Format, 2-33
 - Saving a Profile Member, 2-35
- Profile Variables. *See also* specific variable
 - Color Support, 2-39
 - Modification, 2-35
- PROG Parameter
 - CTTTPI Utility, 9-186
- Program List Member
 - Default Members, 2-10
 - IOA Customization, 2-9
- Program Names
 - IOA Applications, A-1
- PROGRAM Parameter
 - Automation Option Menu, 5-20
- PROMPT Parameter
 - Automation Option Menu, 5-21
- PROTECT Parameter
 - CTTTPI Utility, 9-184
- PROTECTED Attribute
 - Global Variable Member (CTO), 5-14
- PRTDBG DD Statement
 - Archive Server Procedure, 4-73
 - CMEM Procedure, 3-31
 - CONTROL-O Procedure, 5-17
- PRTLST Member
 - Printing Missions, 4-14
- PRTMON# Parameter
 - Printing Missions, 4-33
- PRTORDER KSL Utility
 - Printing Missions, 4-16
- PSCHDTYP
 - Profile Variable (CTM), 2-40
- PTF Level
 - ExLM Support (CTT), 7-31
- PZUMFAPP
 - Profile Variable (CTM), 2-41
- PZUMFGRUP
 - Profile Variable (CTM), 2-41
- Q—**
- QNAME Parameter
 - CTTMUP Utility, 9-135
- Quantitative Resource
 - Adding/Deleting/Modifying, 9-10
 - Description, 1-8
- R—**
- RANGE Statement
 - CTTDLD Utility, 9-123
- RBA
 - Media Database, 7-10
- RBA Keyword
 - Trace Data, D-12
- RBA Parameter
 - CTTIOS Macro, 7-49
- RBAOFF Parameter
 - IOADPT Utility, 9-32
- IOADUL Utility, 9-33
- RBAOFS Parameter
 - IOADBSR Utility, 9-25
- RBTTYE Parameter
 - MEMOREX Tape Library Support, 7-32
 - Modification (CTT), 7-4
 - Robotic Library Support, 7-26
- RC Parameter
 - CTTIOS Macro, 7-49
- READ Function
 - Base Level API (CTT), 7-47, 7-51
- READCC Keyword
 - Dataset Record, D-4
- READCPU Keyword
 - Dataset Record, D-4
- READDDN Keyword
 - Dataset Record, D-4
- READDDT Keyword
 - Dataset Record, D-4
- Reading Files
 - API Interface (CTT), 7-51
- READJBN Keyword
 - Dataset Record, D-4
- READNEXT Function
 - Base Level API (CTT), 7-47, 7-52
- READPGM Keyword
 - Dataset Record, D-4
- READSTEP Keyword
 - Dataset Record, D-4
- READTM Keyword
 - Dataset Record, D-4
- READUAD Keyword
 - Dataset Record, D-4
- READVOL Function
 - High Level API, 7-68, 7-69
- Real-time Environment
 - CONTROL-T, 7-1, 7-4
 - Termination (CTT), 7-6
- Reason Codes
 - High Level API, 7-71
 - Macro CTTIOS, 7-49
- REBUILD Option
 - CMEM Facility, 3-28
 - Rule Loading (CTO), 5-4

- Rebuilding
 - Access Method Variable File (CTB), 9-24
 - CTB Index, 9-52
 - Group File (CTB), 9-24
 - Report File (CTB), 9-24
 - Rule Activity File (CTB), 9-24
- REC Parameter
 - CTTIOS Macro, 7-49
- RECAT Parameter
 - CTTRCV Utility, 9-141
- RECDEL Statement
 - CTTMUP Utility, 9-134
- RECFM Parameter
 - APAPARM Library, 4-40
 - Dataset Record, D-5
 - Exit CTDX005, 4-51
- RECFROM Keyword
 - Volume Record, D-7
- Recipient Tree
 - Loading via IOA, 4-3
 - Reloading (CTD), 4-3
- Record Selection Logic
 - INCLUDE/EXCLUDE Statements
 - CTT Utilities, 9-102
- RECOVER Statement
 - CTTRCV Utility, 9-140
- Recovery
 - Access Method Files, 4-83
 - CONTROL-M, 3-19
 - CONTROL-T, 7-20
 - CONTROL-T Repository, 7-23
 - CTTRTM Utility, 9-163
 - CTTTPI Utility, 9-182
 - IOA Access Method, 2-31
 - MDB from Trace File, 9-139
 - Media Database, 7-24
 - Selective (CTT), 7-25
 - Split/Merge Process (CTT), 9-173
- RECREATE Parameter
 - Modification (CTT), 7-4
- RECTYPE Keyword
 - Trace Data, D-12
- RECUPD Statement
 - CTTMUP Utility, 9-135
- RECVLNUM Keyword
 - Volume Record, D-7
- Reformatting
 - Active Balancing File, 6-2
- Refreshing
 - CMEM Security Cache, 3-30
 - Rule Definitions (CTT), 7-8
- Relative Byte Address. *See* RBA
- RELOAD Mode
 - CTTINIT Procedure, 7-3
- Reloading
 - CONTROL-T Rules, 7-8
 - Recipient Tree, 4-3
 - User Exits (CTT), 7-3
- REMFEET Keyword
 - Volume Record, D-7
- RENAME Parameter
 - CTTDLD Utility, 9-123
- Repeating Step
 - Restore Mission, 4-64
- REPLACE Parameter
 - CTBDBVCP Utility, 9-56
- Replacing
 - CONTROL-O IPL Monitor, 5-8
- REPLIST Member
 - Decollating Missions, 4-14
- REORDER KSL Utility
 - Decollating Missions, 4-16
- Report
 - Deleting (CTD), 9-79
 - Dependency on CTM Job, 4-25
 - Printing to a File, 4-49
 - Scratch Report (CTT), 9-162
- Report Decollating. *See* Decollating Mission
- Report File (CTB)
 - Formatting, 9-24
 - Rebuilding, 9-24
- Report Generation (CTT)
 - CTTRPT Utility, 9-143
 - Sample Reports, 9-154
- Report List Files
 - Active, 4-76
 - CTDCP2A Utility, 4-76
 - History, 4-78
 - Management (CTD), 4-74
 - Permanent User Report, 4-75
 - Summary, 4-79
 - Workflow (Diagram), 4-80
- REPORT Statement
 - CTTRPT Utility, 9-145
 - CTTRTM Utility, 9-169
 - CTTVTM Utility, 9-197
- Report Status
 - Active User Report List, 4-76
- Report Utilities
 - CONTROL-B, 9-4
- Repository
 - Backup and Recovery (CTT), 7-23
 - Disaster Recovery, 7-24
 - IOA Products, 1-9
 - Overview (CTT), 7-9
 - Recovery Methods (CTT), 7-25
 - Selective Recovery (CTT), 7-25
- Rerun
 - Migration Missions, 4-62
- RESETSTAT Command
 - Message Statistics (CTO), 5-9
- RESIDENT Parameter
 - CTVDELI Utility, 9-95
- RESLOT Parameter
 - CTTVTM Utility, 9-196
- Resource Control
 - Archive Server, 4-69
- Resource Information
 - Display, 4-72
- Resource Utilization
 - CMEM, 3-32
 - CONTROL-O, 5-18
- RESTART Parameter
 - CTTRTM Utility, 9-167
 - CTTVTM Utility, 9-195

Index

- RESTORE IN PROCESS
 - Status
 - Restore Mission Abend, 4-65
 - Restore Mission
 - Default Missions, 4-15
 - Exception Handling, 4-65
 - History User Report List, 4-78
 - Overview, 4-62
 - Post Processing Parameters, 4-65
 - Profile Variable, 2-36
 - Scheduling, 4-13
 - Scheduling Dates, 4-63
 - Skeleton Job Parameters, 4-64
 - Workflow, 4-63
- RESTORED Parameter
 - CTDDELRP Utility, 9-80
- Restoring
 - Media Database, 7-24
- RESUME Mode
 - CTTINIT Procedure, 7-5
- RESWPF3
 - Profile Variable, 2-42
- Retention Criteria
 - CTTMUP Utility, 9-137
- Retention Management
 - CTR Interface to CTT, 7-43
 - CTTRTM Utility, 9-162
 - New Day Procedure (CTT), 7-8
- Retention Period
 - Backup Mission (CTD), 4-56
 - CTDCLHIS Utility, 9-76
 - CTDDELRP Utility, 9-80
 - CTVDELI Utility, 9-95
 - Migrated Report List, 9-93
- RETPD Parameter
 - CTDUPBKP Utility, 9-89
- RETRO Parameter
 - Decollating Scheduling, 4-14
- Return Codes. *See also* Specific Program or Utility
 - High Level API, 7-70, 7-71
- RETURNS Parameter
 - Automation Option Menu, 5-22
- RETVLTFT Keyword
 - Volume Record, D-7
- RLSE Parameter
 - Exit CTDX005, 4-51
- RMM
 - Conversion Utility, 9-123
- Robot Support
 - COMPAREX, 7-25
 - EMASS, 7-32
 - HDS 7450/7430, 7-25
 - IBM Robots, 7-26
 - MEMOREX TELEX, 7-32
 - StorageTek Silo, 7-28
- Robotic Support
 - CONTROL-T, 7-25
- Roll Back Recovery
 - Media Database, 7-25, 9-139
- Roll Forward Recovery
 - Media Database, 7-24, 9-139
- ROSCOE
 - Color Support, 2-45
 - Cross-Memory Interface, 2-1
- ROUT Parameter
 - IOATEST Utility, 9-39
- RPF
 - Mission Scheduling (CTD/V), 4-16
- RSTLIST Member
 - Restore Missions, 4-14
- RSTORDER KSL Utility
 - Restore Missions, 4-16
- RSVNONR MVS
 - Parameter, 2-6, 4-4
- RSVSTRM MVS Parameter,
 - 2-6, 4-4
- RTNFROM Keyword
 - Dataset Record, D-5
- RTNTYPE Parameter
 - CTTPARM Member, 9-164
- RTNUPD Parameter
 - Modification (CTT), 7-4
- Rule
 - Automatic Loading (CTO), 5-2
 - CTB Activation, 6-4
 - Deactivating (CTO), 5-5
 - Loading (CMEM), 3-27, 5-3
 - Loading (CTO), 5-3
 - Loading (CTT), 7-6
 - Operation Mode (CMEM), 3-29
 - Operation Mode (CTO), 5-12
 - Refreshing (CTT), 7-8
 - Scheduling (CTO), 5-8
- Rule Activity File (CTB)
 - Formatting, 9-24
 - Rebuilding, 9-24
- Rule List
 - DD Statement (CTT), 7-5
- Rule List Member
 - Multiple Lists (CMEM), 5-7
 - Multiple Lists (CTO), 5-7
- RULE Parameter
 - Invoking CONTROL-B, 6-7
- Rule Status Screen (CTO), 5-9
- Rule Table
 - Date, 7-6
 - Loading (CTT), 7-3, 7-6
- Rule Table List
 - Profile Variable, 2-43
- RULEINFO Parameter
 - CTTMUP Utility, 9-131
- RULELIST Member
 - CTO Rule List, 5-25
 - Format (CTO), 5-2
- RULER Parameter
 - Exit CTDX005, 4-51
- RULLIST Member
 - CTTINIT Procedure, 7-3
 - Format (CTT), 7-6
- RUNTCACH Parameter
 - Security Cache (CTO), 5-16
- Runtime Environment (CTB)
 - Invoking, 6-5

—S—

- SABKWND
 - Profile Variable (CTD), 2-36
- SAMAWND
 - Profile Variable (CTD), 2-37
- SAMDELI
 - Profile Variable (CTD), 2-37
- SAPRWND
 - Profile Variable, 2-36
- SARSWND
 - Profile Variable (CTD), 2-36
- Saving. *See* feature or value to be saved
- SBALTBO
 - Profile Variable (CTB), 2-43
- SCHEDULE TAG Field
 - Utility CTMBGRP, 9-43
- Scheduling
 - Balancing Missions, 6-2
 - CONTROL-M, 4-26
 - CONTROL-O Rules, 5-8
 - CTD and CTM Workflow, 4-23
 - CTD Missions, 4-13
- Scheduling Facility. *See also* Calendar Facility
 - Calendar-based Rules (CTT), 7-8
- Scheduling Tables
 - Regular to Group Conversion, 9-43
- SCOPE Parameter
 - CONTROL-B Missions, 6-10
- SCRATCH Parameter
 - CTTDLD Utility, 9-123
- Scratch Pools
 - HSC/CSC Support, 7-31
- Scratch Report
 - CTTRTM Utility, 9-162
 - Fields, 9-170
- Scratch Volumes
 - Daily Report, 7-8
- SCRDT Keyword
 - Volume Record, D-7
- Screen Definition
 - Constant Blocks, 2-18
 - Macros, 2-17
 - Profile Attribute, 2-34
- Screen Members
 - Online Facility, E-1
- Screen Modification
 - Recommended Stages, 2-19
- Screens
 - Customization, 2-17
 - Primary Option Menu, 2-14
- SCRPROT Parameter
 - Modification (CTT), 7-4
- SDECWND
 - Profile Variable (CTD), 2-36
- SDENSITY Keyword
 - Stacking Record, D-10
- SDWYCNA
 - Profile Variable (CTD), 2-37
- SEARCH Parameter
 - Decollating Missions, 4-76
- Secondary Extent Dataset
 - IOA Access Method, 2-28
- SECONDARY Parameter
 - CTDUPBKP Utility, 9-90
 - Exit CTDX005, 4-51
 - Migration Mission, 4-59
- SECPREF Parameter
 - Dual Migration, 4-59
- Security
 - CMEM, 3-30
 - Description, 1-12
- Security Cache
 - Refreshing (CTO), 5-16
- SELECT Statement
 - Program CTMFRM, 3-11
- Selection Criteria
 - INCLUDE/EXCLUDE Statements (CTT), 9-102
- Selection Group. *See* INCLUDE/EXCLUDE Statements
- Selective Recovery
 - Media Database, 7-25
- SEPDS JES2 Parameter
 - Multi-Chunk Printing, 4-35
- Sequential File
 - IOA Log, 9-16
 - Printing to, 4-49
- Sequential Read
 - Media Database, 7-53
- SER Parameter
 - CTTTPI Utility, 9-186, 9-188
- Server. *See* Monitor
- Shared DASD (CTM), 3-32
- Shared Spool (CTM), 3-32
- Shout Facility
 - Dynamic Destination Table, 1-11, 2-49
- Show Screen Filter
 - Profile Variable (CTM), 2-42
- SHOW Window Filters
 - Profile Customization, 2-34
- SHOWBLINE Parameter
 - @LINE Display Format Line, 2-22
- Shutting Down. *See* Deactivation
- SIMULATE Parameter
 - Utility CTDUPBKP, 9-89
- Simulation. *See also* MODE Parameter
 - CTDCCU Utility, 9-74
 - CTVDELI Utility, 9-94
 - IOATEST Utility, 9-39
- SIMULATION Parameter
 - CTDCCU Utility, 9-74
- SINSLMT
 - Profile Variable (CTT), 2-41
- Size
 - Active Missions File, 4-81, 9-68
 - Active Transfer File, 4-81, 9-70
 - Automation Log, 5-11, 9-100
 - Conditions/Resources File, 9-14
 - CONTROL-M Files, 3-35
 - Media Database, 7-21

Index

- Trace File (CTT), 7-22
- Skeleton Job
 - Backup Mission, 4-54
 - Migration Missions, 4-57
 - Printing Missions, 4-32
 - Restore Mission, 4-64
- SKELETON Parameter
 - Printing Mission, 4-32
- Sleeping Interval
 - CMEM, 3-30
 - CONTROL-D, 4-2
 - CONTROL-M, 3-2
 - CONTROL-O Monitor, 5-15
- SLNAME Keyword
 - Volume Record, D-7
- SLNFPAG
 - Profile Variable (CTD), 2-40
- Slot Numbers
 - Vaulted Volumes, 9-196
- SLOTNUM Keyword
 - Volume Record, D-7
- SLSUX01 User Exit
 - HSC/CSC Job Processing, 7-31
- SMF
 - Type 15 Records, 9-177
- SMF Accounting
 - CONTROL-D, 4-84
- SMISTBO
 - Profile Variable (CTD), 2-43
- SMNAME Parameter
 - FileTek Media, 4-68
- SMP/E
 - IOA Maintenance, 8-4
- SMP/E Modules
 - IOA Packaging, 8-1
- SMS Managed Tape Library
 - IBM Robot, 7-26
- SMSCWND
 - Profile Variable (CTO), 2-37
- SMSGERR
 - Profile Variable, 2-39
- SMSGINF
 - Profile Variable, 2-39
- SMSGSVR
 - Profile Variable, 2-39
- SMSGWRN
 - Profile Variable, 2-39
- SMSINTR Parameter
 - CTTPARM Member, 9-165
 - DF/SMS Support, 7-37
 - Modification (CTT), 7-4
- SMSMC Keyword
 - Dataset Record, D-5
- SNAP Command
 - CMEM, 3-31
 - CONTROL-O, 5-17
- Snapshot
 - CTT Database Files, 9-121
- SNRSCNE
 - Profile Variable (IOA), 2-38
- SNRSDRNG
 - Profile Variable, 2-42
- SOLVHFC
 - Profile Variable, 2-39
- SOLVOCL
 - Profile Variable, 2-40
- SOLVSCL
 - Profile Variable, 2-40
- SOMPTBO
 - Profile Variable (CTO), 2-43
- SORTBY Statement
 - CTTRPT Utility, 9-147
 - CTTRTM Utility, 9-170
 - CTTVTM Utility, 9-198
 - Utility CTORSTM, 9-99
- SORTIN Statement
 - OSS Migration, 4-61
- Sorting
 - Access Method Files, 9-25
 - Active Report List, 4-83
 - Active User Report List, 4-82
- SPACE Parameter
 - IOADBF Utility, 9-20
- SPACETY Parameter
 - Exit CTDX005, 4-51
- Split/Merge Process
 - Description (CTT), 9-172
 - Merge Utility, 9-129
- SQFILE Parameter
 - IOADLD Utility, 9-31
 - IOADUL Utility, 9-33
- SRCHLMT
 - Profile Variable, 2-42
- SRCMTBO
 - Profile Variable (CTM), 2-43
- SREPTBO
 - Profile Variable (CTD), 2-43
- SRESCND
 - Profile Variable (IOA), 2-38
- SRESDRNG
 - Profile Variable, 2-42
- SRLDATO
 - Profile Variable (CTT), 2-44
- SRVD Step
 - OSS Migration, 4-60
- SSCHBRO
 - Profile Variable, 2-40
- SSCHTBO
 - Profile Variable (CTM), 2-44
- SSTACNS
 - Profile Variable (CTM), 2-37
- SSTALTO
 - Profile Variable (CTM), 2-44
- SSTAWND
 - Profile Variable (CTM), 2-37
- SSTDEL1
 - Profile Variable (CTM), 2-38
- SSYVORD
 - Profile Variable (CTR), 2-40
- SSYVSCL
 - Profile Variable (CTR), 2-40
- SSZMWND
 - Profile Variable (CTM), 2-38
- Stack File. *See* Stacking
- Statistics File
- STACKID Parameter
 - CTTGVL Utility, 9-125

- Stacking Facility (CTT)
 - Activation, 7-41
 - GDGs, 7-42
 - Interface Utility, 9-125
 - Overview, 7-39
 - Starting, 7-5
 - Stopping, 7-5
 - Testing, 7-41
 - Utilities, 9-5
- Stacking Record
 - Keywords, D-10
- Stacking Statistics File, 7-41
 - Build/Update Utility, 9-177
 - Creation, 9-111
 - Daily Update, 7-8
 - Formatting, 9-113, 9-117
 - Index Integrity, 9-120
 - Integrity, 9-119
 - Loading, 9-115
 - Printing, 9-121
 - Rebuild Index, 9-118
 - Structure, 7-17
 - Unloading, 9-116
 - Utilities, 9-6
- START Function
 - High Level API, 7-68, 7-69
- STARTDATE Parameter
 - Utility CTOALOCF, 9-100
- Started Task. *See* Monitor
- STARTGEN Command
 - Generic Processing (CTD), 4-1, 4-22
- Starting. *See* Initialization.
See Activation
 - Printers, 4-36
- Starting a New Day. *See* New Day Procedure
- STARTPRT Command
 - Starting a Printer, 4-37
- STARTSTAT Command
 - Message Statistics (CTO), 5-9
- STARTSTATS Command
 - CTO Message Statistics, 9-98
- STARTSTK Mode
 - CTTINIT Procedure, 7-5
- STARTTIME Parameter
 - Utility CTOALOCF, 9-100
- START-WHEN-OVER Parameter
 - IOADBSR Utility, 9-26
- Statistics. *See* Stacking Statistics. *See* Message Statistics
 - CONTROL-M Jobs, 9-49
 - Media Database, 9-127
 - Message Statistics File, 9-98
- Statistics File (CTM)
 - Expanding, 3-36
- Statistics File (CTO), 5-9
 - Full Conditions, 5-10
 - Unnecessary Enlargement, 5-9
- STATS Rule Table (CTO), 5-10
- STATUS Parameter
 - CTMCAJF Utility, 9-46
 - IGNORE/SELECT Statements (CTM), 3-12
- STEPLIB DD Statement
 - CTDRRQ Program, 4-28
- STKBLKCT Keyword
 - Stacking Record, D-10
- STKBLKSZ Keyword
 - Stacking Record, D-10
- STKCOMPF Keyword
 - Stacking Record, D-10
- STKDATE Keyword
 - Stacking Record, D-10
- STKDEFFT Parameter
 - Default Dataset Length, 7-40
 - Modification (CTT), 7-4
- STKDSN Keyword
 - Stacking Record, D-10
- STKIND Keyword
 - Stacking Record, D-10
- STKJBN Keyword
 - Stacking Record, D-10
- STKKEY Field
 - Stacking Statistics File, 7-18
- STKOBBS Field
 - Stacking Statistics File, 7-18
- STKPGM Keyword
 - Stacking Record, D-10
- STKPRED Field
 - Stacking Statistics File, 7-18
- STKSRCHL Parameter
 - Modification (CTT), 7-4
 - Stacking Search, 7-40
- STKSTAMP Field
 - Stacking Statistics File, 7-18
- STKTEST Parameter
 - Modification (CTT), 7-4
- STKTIME Keyword
 - Stacking Record, D-10
- STKUNIT Keyword
 - Stacking Record, D-10
- STKVOL1 Keyword
 - Stacking Record, D-10
- STOP Parameter
 - CTTTPI Utility, 9-189
- STOPGEN Command
 - Decollation Deactivation, 4-5
 - Generic Processing (CTD), 4-22
- STOPPRT Command
 - Stopping a Printer, 4-37
- STOPSTAT Command
 - Message Statistics (CTO), 5-9
- STOPSTATS Command
 - CTO Message Statistics, 9-98
- STOPSTK Mode
 - CTTINIT Procedure, 7-5
- Storage Class
 - Adjusting, 7-38
 - DF/SMS Support, 7-37
- StorageTek Silo
 - CONTROL-T Interface, 7-28
 - ExLM Support, 7-31
 - HSC/CSC Scratch Support, 7-31
- STRTCH Keyword
 - Stacking Record, D-10

Index

- Structure
 - Media Database, 7-9
 - Stacking Statistics File, 7-17
 - Trace File (CTT), 7-17
 - STUDBLC Keyword
 - Stacking Record, D-10
 - STUDBLKS Keyword
 - Stacking Record, D-10
 - STUDCMPF Keyword
 - Stacking Record, D-10
 - STUDDEN Keyword
 - Stacking Record, D-10
 - STUDDSN Keyword
 - Stacking Record, D-10
 - STUDDTE Keyword
 - Stacking Record, D-10
 - STUDJOB Keyword
 - Stacking Record, D-10
 - STUDPGM Keyword
 - Stacking Record, D-10
 - STUDRTCH Keyword
 - Stacking Record, D-10
 - STUDTME Keyword
 - Stacking Record, D-10
 - STUDUNIT Keyword
 - Stacking Record, D-10
 - STUDVOL Keyword
 - Stacking Record, D-10
 - STUFLAG1 Keyword
 - Stacking Record, D-10
 - SUB Parameter
 - CONTROL-O
 - Initialization, 5-1
 - SUBSYS JCL Parameter
 - CDAM Files, 9-74
 - Utility CTDDLRP, 9-82
 - SUBSYS Parameter
 - FileTek Media, 4-68
 - IOA Subsystem, 5-5
 - Subsystem Chains
 - Problem Determination, 2-7
 - SUFFIX Parameter
 - Utility IOADCPY, 9-27
 - SUMMARY Parameter
 - CTTRPT Utility, 9-145, 9-146
 - SUMTITLE Parameter
 - CTTRPT Utility, 9-151
 - Supervisor Calls. *See* SVC
 - SUS2LMT
 - Profile Variable (CTD), 2-41
 - SUS2SRTI
 - Profile Variable (CTD), 2-41
 - SUSPEND Mode
 - CTTINIT Procedure, 7-5
 - Suspended Mode
 - CONTROL-T, 7-5
 - SUSRDELW
 - Profile Variable (CTD), 2-38
 - SUSRFTTR
 - Profile Variable (CTD), 2-44
 - SUSRHDR
 - Profile Variable (CTD), 2-44
 - SUSRVEW
 - Profile Variable (CTD), 2-44
 - SVC
 - Installation (CTT), 7-4
 - SVC Number, 7-2
 - SVEWFFI
 - Profile Variable, 2-41
 - SWHYCNA
 - Profile Variable (CTM), 2-38
 - Syntax
 - Rule Definitions (CTB), 9-51
 - SYSDATA (CTM), 3-25
 - SYSDATA Entry
 - CTDDLRP Utility, 9-81, 9-82
 - SYSIN DD Statement
 - CTTTPI Utility, 9-190
 - Utility CTMBGRP, 9-43
 - Utility CTTTPI, 9-183
 - SYSLIB Library
 - Ad-hoc Maintenance, 8-8
 - SYSRINT DD Statement
 - CTTTPI Utility, 9-190
 - Utility CTMBGRP, 9-43
- T—
- Table List Screen
 - Profile Variable (CTM), 2-44
 - Tape Cartridge Subsystem.
See CART Media
 - Tape Eject
 - IBM Robot User Exit, 7-26
 - Tape Entry
 - IBM Robot User Exit, 7-27
 - Tape Erasure, 9-181
 - Tape Initialization, 9-179
 - Tape Library. *See* Robot Support
 - Tape Mapping, 9-180
 - Tape Pool
 - CTDCLHIS Utility, 9-76
 - Tape to Disk
 - Migration Skeleton, 4-60
 - Tape/CART Migration
 - Skeleton Job
 - Adjustments, 4-61
 - TAPEMAP Function
 - CTTTPI Utility, 9-188
 - TAPEMS Parameter (CTR)
 - Interface to CTT, 7-43
 - TAPERAS Function
 - CTTTPI Utility, 9-188
 - Target Media
 - Migration Missions, 4-59
 - TBLT Parameter
 - CTTINIT Procedure, 7-3
 - TCOLOR
 - Profile Variable, 2-45
 - TCT. *See* Control Table (CTT)
 - TCT Parameter
 - CTTIOS Macro, 7-49
 - TDBCS
 - Profile Variable, 2-45
 - TEMPORARY Attribute
 - Global Variable Member (CTO), 5-14
 - TERM Mode
 - CTTINIT Procedure, 7-3
 - TERM Parameter
 - IOATEST Utility, 9-39

- Terminal Characteristics
 - Double Byte Character Set, 2-45
 - Termination
 - CONTROL-T, 7-6
 - TERMTYPE
 - Profile Variable, 2-46
 - Testing
 - Dynamic Dataset Stacking, 7-41
 - TESTRUL Parameter
 - CTTPARM Member, 9-165
 - Time
 - Time-Dependent Actions, 3-23
 - TIME Keyword
 - Trace Record, D-11
 - TIMEFROM Parameter
 - IOADBSR Utility, 9-25
 - TIMEOUT Parameter
 - Printing Mission, 4-32
 - Timestamp
 - CTTSTK Utility, 9-177
 - Migration Skeleton, 4-58
 - TIMETO Parameter
 - IOADBSR Utility, 9-25
 - TITLE Parameter
 - CTTRPT Utility, 9-145, 9-148
 - TMSCTLEX Exit
 - DMS Tape Volume Exit, 7-35
 - TO Parameter
 - CTMCAJF Utility, 9-46
 - CTTACP Utility, 9-106
 - CTTDBPRT Utility, 9-121
 - TODATE Parameter
 - Utility CTMJSA, 9-49
 - Utility IOACPLOG, 9-16
 - TODSN Parameter
 - Utility CTOALOCP, 9-100
 - TOGEN Parameter
 - CTBDBVCP Utility, 9-57
 - TOKEY Parameter
 - IOADUL Utility, 9-33
 - TORECNUM Parameter
 - Utility CTOALOCP, 9-101
 - TOTYPE Parameter
 - Utility CTOALOCP, 9-100
 - TOUNIT Parameter
 - Utility CTOALOCP, 9-101
 - TOVOLUME Parameter
 - Utility CTOALOCP, 9-100
 - Trace Data
 - Keywords, D-12
 - Trace File
 - Backup, 7-23
 - Backup Utility, 9-192
 - Copying, 9-106
 - CTTRTM Utility, 9-166
 - CTTVTM Utility, 9-195
 - Enlarging, 7-22
 - Formatting/Creating, 9-114
 - Formatting/Initializing, 9-108
 - MDB Recovery, 9-139
 - Structure (CTT), 7-17
 - Utilities, 9-5
 - Trace Modes
 - Rule Operation (CTO), 5-12
 - TRACE Parameter
 - CTTRTM Utility, 9-166
 - CTTVTM Utility, 9-195
 - Trace Record
 - Keywords, D-11
 - TRANID Parameter
 - IOAONL CLIST, 2-2
 - Transaction File
 - CTTIDB Utility, 9-128
 - Update MDB, 9-193
 - Transaction ID
 - IOA Online, 2-2
 - Transaction Member
 - IOA Customization, 2-8
 - Predefined (Reserved) Members, 2-9
 - TRCBLKS Parameter
 - CTTCTRC Utility, 9-114
 - TRCIN Parameter
 - CTTACP Utility, 9-106
 - TRCOUT Parameter
 - CTTACP Utility, 9-106
 - TRCSIZE Parameter
 - CTTPARM Member, 9-108
 - Trace File Size, 7-22
 - TRCVOL Parameter
 - CTTCTRC Utility, 9-114
 - Tree. *See* Recipient Tree
 - TRTCH Keyword
 - Dataset Record, D-5
 - TRTCH Parameter
 - Exit CTDX005, 4-51
 - TSO
 - Color Support, 2-45
 - Cross-Memory Interface, 2-2
 - Entry to Online Facility, 2-2
 - TYPE Parameter
 - @STYLE Display Format Line, 2-21
 - CONTROL-O Monitor, 5-8
 - CTTACP Utility, 9-106
 - CTTRCV Utility, 9-140
 - IOADBF Utility, 9-19
 - Migration Media, 4-59
 - TYPELIKE Parameter
 - @STYLE Display Format Line, 2-21
 - TYPERET Statement
 - CTTRTM Utility, 9-167
 - TYPERUN Statement
 - CTTBIX Utility, 9-109
 - CTTDLD Utility, 9-123
 - CTTIDB Utility, 9-128
 - CTTMER Utility, 9-129
 - CTTMUP Utility, 9-131
 - CTTRTM Utility, 9-166
 - CTTSPL Utility, 9-174
 - CTTSTK Utility, 9-177
 - CTTTPI Utility, 9-184
 - CTTUDB Utility, 9-193
 - CTTVTM Utility, 9-195
 - Utility CTTMUP, 7-21
 - TYPEVLT Statement
 - CTTVTM Utility, 9-196
- U—
- UNCTLG Parameter
 - CTTRTM Utility, 9-168

Index

- UNDERSCORE Attribute
 - Profile Variable, 2-45
- UNIT Parameter
 - Exit CTBX005, 4-50
 - IOADBF Utility, 9-19
 - Utility IOADCPY, 9-27
- UNITD Parameter
 - IOADBF Utility, 9-19
- UNITNAME Parameter
 - Migration Media, 4-59
 - Volume Record, D-7
- Unloading
 - IOA Access Method Files, 9-33
 - Media Database, 9-116
 - Stacking Statistics File, 9-116
- UNMATCH Parameter
 - CTDUPBKP Utility, 9-90
- Updating
 - Global Variables (CTO), 5-12
 - MDB Volume, 9-132
 - Media Database, 9-131, 9-193
 - Stacking Statistics File, 9-177
- UPDQNM Statement
 - CTTMUP Utility, 9-135
- UPDTVOL Function
 - High Level API, 7-68, 7-69
- UPDVOL Parameter
 - CTTRTM Utility, 9-169
- USCORE
 - Profile Variable, 2-45
- Use Attribute
 - IBM Robot User Exit, 7-26
- USEDAYS Parameter
 - CTTIDB Utility, 9-128
- User Daily
 - CONTROL-D, 4-7
 - CONTROL-M, 3-4
 - Date Control Record (CTD), 4-8
 - Scheduling (CTM), 3-13
 - Work Flow (CTM), 3-17
 - User Daily (CTD), 4-19
- User Exit. *See also* Exits
 - Reloading (CTT), 7-3
- User Groups
 - CONTROL-D, 4-22
- User Profile, 2-32. *See also* Profile
- User Report. *See* Report
- User Reports List File
 - Maintenance, 4-82
- USERDATA Keyword
 - Volume Record, D-7
- USERID Keyword
 - Trace Record, D-11
- Users
 - Multiple Users (IOA), 2-5
- Utilities. *See also* Specific
- Utility
 - CTBABI, 9-50
 - CTBCMP, 9-51
 - CTBDBIB, 9-52
 - CTBDBVCG, 9-54
 - CTBDBVCP, 9-56
 - CTBDBVDL, 9-59
 - CTBDBVIG, 9-61
 - CTBJAFCP, 9-63
 - CTBJAFDL, 9-65
 - CTBJAFIG, 9-67
 - CTDAMUTI, 9-2
 - CTDCA2P, 4-76, 9-72
 - CTDCAME, 9-68
 - CTDCATF, 9-70
 - CTDCCU, 9-74
 - CTDCLHIS, 4-79, 9-76
 - CTDCP2A, 9-77
 - CTDDELRP, 4-78, 9-79
 - CTDDIB, 9-86
 - CTDSMFRP, 9-3
 - CTDUPBKP, 9-87
 - CTMBGRP, 9-43
 - CTMCAJF, 9-45
 - CTMJSA, 9-49
 - CTOALCP, 5-11, 9-100
 - CTOCSF, 5-10, 9-98
 - CTORSTM, 9-99
 - CTTACP, 7-23, 9-106
 - CTTAFR, 9-108
 - CTTBIX, 7-20, 7-24, 9-109
 - CTTCRDF, 9-111
 - CTTCTRC, 9-114
 - CTTDBDL, 9-115
 - CTTDBDUL, 9-116
 - CTTDBF, 9-117
 - CTTDBIB, 9-118
 - CTTDBID, 7-24, 9-119
 - CTTDBII, 9-120
 - CTTDBPRT, 9-121
 - CTTDLD, 9-122
 - CTTGVL, 9-125
 - CTTIDB, 7-9, 7-20, 9-127
 - CTTMER, 9-129
 - CTTMUP, 7-21, 9-131
 - CTTRCV, 7-24, 9-139
 - CTTRPT, 9-143
 - CTTRTM, 7-8, 9-162
 - CTTSPL, 9-172
 - CTTSTK, 7-8, 7-41, 9-177
 - CTTTPI, 9-179
 - CTTTRB, 9-192
 - CTTUDB, 9-193
 - CTTVTM, 7-8, 9-194
 - CTVCLMIG, 9-93
 - CTVDELI, 9-94
 - Dataset Formatting, B-1
 - IEBGENER, 7-9
 - IOA Access Method, 2-29
 - IOABLCAL, 9-1
 - IOACLRES, 4-29, 9-7
 - IOACND, 4-28, 9-10
 - IOACOPRS, 9-14
 - IOACPLOG, 9-16
 - IOADBF, 9-18
 - IOADBSR, 4-83, 9-25
 - IOADCPY, 9-27
 - IOADIG, 9-29
 - IOADII, 9-30
 - IOADLD, 9-31
 - IOADPT, 9-32
 - IOADUL, 9-33
 - IOALDNRS, 4-3, 4-29, 9-35
 - IOANOTE, 9-38
 - IOATEST, 9-39
 - IOAVERFY, 9-41

—V—

- Validation. *See* Integrity
- Validity
 - Checking via CONTROL-B, 6-7
- Variable Generations
 - Utility CTBDBVCG, 9-54
- Variable Groups
 - CTB Utilities, 9-4
- VARIABLE Parameter
 - CTBDBVCG Utility, 9-55
 - CTBDBVCP Utility, 9-57
 - CTBDBVDL Utility, 9-59
- Variables. *See also* Profile Variables
 - Global Variables (CTO), 5-12
- Vault Definition
 - DD Statement (CTT), 7-5
 - Initializing (CTT), 7-3
- VAULT Keyword
 - Volume Record, D-7
- Vault Management
 - CTTVTM Utility, 9-194
 - New Day Procedure (CTT), 7-8
- VCHANGED Keyword
 - Volume Record, D-7
- VCHANGET Keyword
 - Volume Record, D-7
- VCHANGEU Keyword
 - Volume Record, D-7
- VDENSITY Keyword
 - Volume Record, D-7
- VENDOR Keyword
 - Volume Record, D-7
- VENDOR Parameter
 - CTTDLD Utility, 9-123
 - CTTTPI Utility, 9-187
- Verification
 - File Installation and Status, 9-41
- VERIFY Mode
 - CTTINIT Procedure, 7-4
- VERIFY Parameter
 - IOAVERIFY Utility, 9-41
- VERIFY-IN-HISTORY Parameter
 - CTDCCU Utility, 9-74
- VLTBYS1 Parameter
 - CTTPARM, 9-194
 - Modification (CTT), 7-4
- VLTDSDNUM Keyword
 - Volume Record, D-7
- VLTEXTDT Keyword
 - Volume Record, D-7
- VLTEXTNM Keyword
 - Volume Record, D-7
- VLTEXPDY Keyword
 - Volume Record, D-8
- VLTEXTYP Keyword
 - Volume Record, D-8
- VLTPREFL Keyword
 - Volume Record, D-8
- VOL Parameter
 - IOADBF Utility, 9-19
 - Utility IOADCPY, 9-27
- VOL1 Parameter
 - Exit CTDX005, 4-50
- VOL2 Parameter
 - Exit CTDX005, 4-50
- VOLADD Statement
 - CTTMUP Utility, 9-132
- VOLBIX Statement
 - CTTMUP Utility, 9-133, 9-136
- VOLCNT Parameter
 - Exit CTDX005, 4-50
- VOLD Parameter
 - IOADBF Utility, 9-19
- VOLDEL Statement
 - CTTMUP Utility, 9-132
- VOLEDMID Keyword
 - Volume Record, D-8
- VOLEXCP Keyword
 - Volume Record, D-8
- VOLEXPDT Keyword
 - Volume Record, D-8
- VOLEXPTY Keyword
 - Volume Record, D-8
- VOLFEET Keyword
 - Volume Record, D-8
- VOLFLAG2 Keyword
 - Volume Record, D-9
- VOLFLAGS Keyword
 - Volume Record, D-8
- VOLIND Keyword
 - Volume Record, D-9
- VOLODESC Keyword
 - Volume Record, D-9
- VOLOWNER Keyword
 - Volume Record, D-9
- VOLSCR Statement
 - CTTMUP Utility, 9-133
- VOLSCRF Statement
 - CTTMUP Utility, 9-135, 9-136
- VOLSEQ Keyword
 - Volume Record, D-9
- VOLSER Keyword
 - Trace Data, D-12
 - Volume Record, D-9
- VOLSER Parameter
 - CONTROL-O Files, 5-6
 - CTDUPBKP Utility, 9-89
 - CTTGVL Utility, 9-126
 - CTTMUP Utility, 9-132
 - Utility CTDUPBKP, 9-88, 9-89
- VOLSDNUM Keyword
 - Dataset Record, D-5
- VOLSTAT Keyword
 - Volume Record, D-9
- Volume Assignment
 - Utility CTTGVL, 9-125
- Volume Expiration
 - CTTRTM Utility, 9-162
- Volume Record
 - Dataset Record, 7-13
 - Index File, 7-12
 - Keywords, D-6
 - Media Database, 7-10
- VOLUPD Statement
 - CTTMUP Utility, 9-132, 9-136
- VOLUSECT Keyword
 - Volume Record, D-9
- VSET Parameter
 - FileTek Migration, 4-61
- VTAM
 - IOA Online Support, 2-1
 - Monitor Activation, 2-7
 - Monitor Deactivation, 2-8
- VTRTCH Keyword
 - Volume Record, D-9
- V-type Index
 - Media Database, 7-12

Index

—W—

WAIT DECOLLATION

Status

Active User Report List,
4-76

WAIT MIGRATE Status

Active User Report List,
4-77

WAIT Parameter

IOATEST Utility, 9-39

WAIT PRINT Status

Active User Report List,
4-77

WAITDEC Parameter

CTDDELRP Utility, 9-80

WAITING FOR BACKUP

Status

Utility CTDDELRP, 4-78

WAITING FOR

MIGRATION Status

Utility CTDDELRP, 4-78

WAITPRINT Parameter

CTDDELRP Utility, 9-80

Warning Message. *See*

Message or Messages

Decollation Inactive, 4-5

WATERMARKS Command

CMEM, 3-32

Resource Utilization
(CTO), 5-18

Why Screen

Profile Variable (CTD),
2-37

Profile Variable (CTM),
2-38

Windows Display

Profile Variables, 2-36

Work Modes

Profile Variables, 2-40

Workflow

Backup Mission, 4-54

CTD New Day Procedure,
4-7

CTM and CTD, 4-23

Restore Mission (CTD),
4-63

Write to Operator. *See*

WTO

Write to Operator With
Reply. *See* WTOR

WRITECCC Keyword

Dataset Record, D-5

WRITECPU Keyword

Dataset Record, D-5

WRITEDDN Keyword

Dataset Record, D-5

WRITEDT Keyword

Dataset Record, D-5

WRITEGLOBAL Command

Global Variables (CTO),
5-12

WRITEJBN Keyword

Dataset Record, D-5

WRITEPGM Keyword

Dataset Record, D-5

WRITESTP Keyword

Dataset Record, D-5

WRITETM Keyword

Dataset Record, D-5

WRITEUAD Keyword

Dataset Record, D-5

WTO

CONTROL-T Intercept,
7-2

WTO Parameter

IOATEST Utility, 9-39

WTOM Parameter

IOATEST Utility, 9-39

WTOR Parameter

IOATEST Utility, 9-39

—X—

X98ASKOP Parameter

Modification (CTT), 7-4

XEROX Printers

DJDE Parameters, 4-42

—Z—

Zoom Screen

Profile Variable (CTM),
2-38



New Dimension Software Inc.
One Park Plaza
11th Floor
Irvine, California USA 92614
Phone: 714-757-4300
Fax: 714-756-3900

New Dimension Software Ltd.
Atidim, Neve Sharet, Bldg. 7
P.O.B. 43227
Tel Aviv, Israel 61430
Phone: 972-3-645-1111
Fax: 972-3-645-1100

New Dimension Australasia Pty. Ltd.
390 St. Kilda Road
6th Floor
Melbourne, Victoria 3004 Australia
Phone: 61-3-820-8677
Fax: 61-3-820-8689

Visit our web site at: www.ndsoft.com