Cincom

SUPRA SERVER PDM

PDM and Directory Administration Guide
(OS/390 & VSE)

P26-2250-65

SUPRA® Server PDM and Directory Administration Guide (OS/390 & VSE)

Publication Number P26-2250-65

© 1991–1994, 1998, 1999, 2000, 2002 Cincom Systems, Inc.
All rights reserved

This document contains unpublished, confidential, and proprietary information of Cincom. No disclosure or use of any portion of the contents of these materials may be made without the express written consent of Cincom.

The following are trademarks, registered trademarks, or service marks of Cincom Systems, Inc.:

AD/Advantage®
C+A-RE™
CINCOM®
Cincom Encompass®
Cincom Smalltalk™
Cincom SupportWeb®
CINCOM SYSTEMS®
gOOi™
*i*D CinDoc™
*i*D CinDoc Web™
*i*D Consulting™
*i*D Correspondence™
*i*D Correspondence Express™
*i*D Environment™
*i*D Solutions™
intelligent Document Solutions™
Intermax™
MANTIS®
Socrates®
Socrates® XML
SPECTRA™
SUPRA®
SUPRA® Server
Visual Smalltalk®
VisualWorks®

UniSQL™ is a trademark of UniSQL, Inc.
ObjectStudio® is a registered trademark of CinMark Systems, Inc.

All other trademarks are trademarks or registered trademarks of their respective companies.

Cincom Systems, Inc.
55 Merchant Street
Cincinnati, Ohio 45246-3732
U.S.A.

PHONE: (513) 612-2300
FAX: (513) 612-2000
WORLD WIDE WEB: http://www.cincom.com

Attention:

Some Cincom products, programs, or services referred to in this publication may not be available in all countries in which Cincom does business. Additionally, some Cincom products, programs, or services may not be available for all operating systems or all product releases. Contact your Cincom representative to be certain the items are available to you.

Release information for this manual

The *SUPRA Server PDM and Directory Administration Guide (OS/390 & VSE),* P26-2250-65, is dated January 15, 2002. This document supports Release 2.7 of SUPRA Server PDM in IBM mainframe environments.

We welcome your comments

We encourage critiques concerning the technical content and organization of this manual. Please take the survey provided with the online documentation at your convenience.

Cincom Technical Support for SUPRA Server PDM

FAX: (513) 612-2000
Attn: SUPRA Server Support

E-mail: helpna@cincom.com

Phone: 1-800-727-3525

Mail: Cincom Systems, Inc.
Attn: SUPRA Server Support
55 Merchant Street
Cincinnati, OH 45246-3732
U.S.A.

Contents

About this book xi

Using this document xi
Document organization xiii
Revisions to this manual xiv
Conventions xv
SUPRA Server documentation series xviii

Introduction to the SUPRA Server Directory and Physical Data Manager (PDM) 21

What is the SUPRA Server Directory and PDM? 21
Overview of SUPRA Server components 22
Understanding the Directory 23
Types of Directory files 24
Data maintained by the Directory 25
Understanding the PDM 30
Types of PDM files 31
Versions of the PDM 34
Types of PDM tasks 35
PDM operating modes 36
Updating and maintaining the database 38
Using Directory Maintenance 38
Using Active Schema Maintenance 39
Controlling and monitoring the database 43
Using database utilities 43
Using Directory Maintenance utilities 44
Using Interactive Services 44
Using SUPRA Server logging 44
Managing file space 46
Managing record holding control blocks 47
Improving database performance 48
Controlling VSE interface polling 52

Installing TIS/XA subsystem (OS/390) 53

Installation 53
Initializing the subsystem at IPL 54
Initializing the subsystem by a START operator command 57

Initializing the PDM 63

Initialization tasks 63
Cold starting the PDM 64
Operating the PDM boot-driven or Directory-driven 65
Warm starting the PDM 67
Selecting initialization parameters 68
Coding initialization parameters 73
Example CSIPARM input 91
Creating a CSIPARM file 93
    Creating the file under OS/390 93
    Creating the file under VSE 94
Coding the Buffer Cache Facility input file 95
Defining tasks, threads, and interfaces 99

Defining the SUPRA Server database 103

Definition process 103
Creating and adding the Conceptual Schema 104
    Creating the Conceptual Schema 104
    Adding the Conceptual Schema 106
Creating and adding the Internal Schema 107
    Creating the Internal Schema 107
    Adding the Internal Schema 109
    Defining database files 110
Creating and adding the External Schema 114

Defining the operational environment 115

Definition process....115
Defining the Internal Schema....116
Adding primary or related files....117
Adding index files....132
Adding Task Log, System Log, or Statistics files....138
Adding Environment Descriptions....140
Relate the file to an Environment Description....144
Adding Buffer Pools....147
Adding Log Groups....149
Consistency check the Schema....151
Allocate the file....153
Format and open the file....154
Activate Secondary Key using the file utilities....157
Defining the Directory to the PDM....160
Modifying the bootstrap schema....163
Modifying the bootstrap Environment Description....189
Creating the bootstrap Validation Module....223
Establishing Directory security....229
Defining Security Groups....231
Relating Security Groups to users....235

Using PDM exits 249

Overview....249
Coding PDM interface exits....251
Coding the precommand exit....253
Coding the DATBASXT exit....255
Coding the post-command exit....258
Coding PDM exits....260
Coding the command initialization exit....264
Coding the open exit....268
Coding the read exit....269
Coding the write exit....270
Coding the check exit....272
Coding the close exit....273
Coding the logical write exit....274
Coding the new volume exit....277

Monitoring and controlling the PDM 279

Options 279
Using Interactive Services 280
Selecting Physical File Services 284
Selecting PDM Services 289
Selecting User Extension Services 302
Using batch task management 306
Using TIS/XA subsystem operator commands (OS/390) 309
DISPLAY command 310
DTISS command 315
SET command 317
STOP command 318
Terminating a multitask PDM 320
Uncontrolled termination 320
Controlled termination 321
Monitoring PDM memory usage 322

Expanding directory and PDM files 323

Expansion process 323
Expanding Directory files 324
Expanding PDM files 328

Using the TIS/XA Software Selection Facility 329

Overview 329
Executing the TIS/XA Software Selection Facility 331
Directly accessing menu options 333
Selecting menu options using alternate user ID and password 334
Changing current user ID and password 335
Using quick exits 335
Passing user data to MANTIS or CONTROL 336
Executing MANTIS programs 336
Executing menu options in batch mode 337
Using security exits 338
CSFS0000 SIGNON EXIT 339
CSFS0000 MENU EXIT 340
CSFS0000 VALUSER EXIT 342
CSFS0000 VALSELECT EXIT 344
CSFS0000 VALALT EXIT 346
CSFS0000 FAILSIGN EXIT 348

Generating Directory reports 349

Overview 349
Generating reports 350
Using trigger records 353
Using category selection 354
Using schema name selection 357
Using Conceptual Schema name selection 358
Using user name selection 359
Using maintenance date selection 360
Using sort option 361
Using text selection 362
Using file selection 363
Using Security Group selection 364
Using condense option 365
Using user sign-on 366
Using Directory reports 367
Using the DBA User Reference Report 368
Using the Directory Conceptual Schema Structure Report 370
Using the Directory External Fields Where Used Report 375
Using the Directory Full Definition Report 376
Using the Directory Full Text Description Report 399
Using the Directory Name Cross Reference Report 402
Using the Directory Name Report 404
Using the Directory Physical Structure Report 406
Using the Directory Schema Structure Report 409
Using the Directory Security Group Report 414
Using RDM views to create Directory reports 419
Example SPECTRA process 423

Routing console messages (OS/390) 427

Routing process 427

PDM Options Module 429

Replacement for optional fixes 429
Assembling the C$THOPTN macro 430
Available options 431
BBPSLOG 431
BDTDL 432
BEGDAT 433
BLDMTFILE 434
BLDSTFILE 435
ENDDAT 436
PSWRD 437
RDAH 438
RECST 439
USTATX 440
WARDAY 441

Using SUPRA Server sample procedures, macros, schemas, linkdecks, and sample source members 443

Samples provided 443
Using SUPRA Server sample procedures and JCL 444
Using SUPRA Server macros 445
Assembling the C$MDMLNK macro 446
Assembling the C$SAWORK macro 449
Assembling the C$XCDFLT macro 460
Assembling the CSTXOPRM macro 468
Assembling the CSIMCS macro 484
Using the SUPRA Server bootstrap and Directory schemas 487
Using the SUPRA Server bootstrap schema 487
Using SUPRA Server Directory schemas 489
Using SUPRA Server linkdecks 490
Using SUPRA Server sample source members 490

Using the trace and debug facilities 491

Trace and debug facilities 491

Index 493

About this book

Using this document

The first half of the *SUPRA Server PDM and Directory Administration Guide (OS/390 & VSE)*, P26-2250, reflects the order in which you perform the tasks on the SUPRA database. These tasks are as follows:

- OS/390 Installing the subsystem
- Initializing the PDM
- Defining the database
- Defining the operational environment

Once your database is up and running, you must perform a number of administrative tasks on the Directory and/or the PDM as the information they contain changes and expands. The second half of this manual discusses the following administrative tasks:

- Using PDM exits
- Controlling and monitoring the PDM
- Terminating the PDM
- Expanding Directory and PDM files
- Using the TIS/XA Software Selection Facility
- Generating Directory reports
- OS/390 Routing console messages

NOTE

Although SUPRA Server Release 2.7 supports VSE, OS/390 terminology is used throughout this manual. The OS/390 terms in the following table correspond to the indicated VSE terms or suffixes in the VSE library.

OS/390 term	VSE term or suffix
MACLIB	.A
LINKLIB	VSE library
PROCLIB	.PROC
JCLLIB	.A
INTERFLM	multitask interface library
address space	partition/address space

Document organization

The information in this manual is organized as follows:

Chapter 1—Introduction to the SUPRA Server Directory and Physical Data Manager (PDM)
Introduces the Directory, the PDM, and discusses updating, maintaining, monitoring, and improving database performance.

Chapter 2—Installing the TIS/XA subsystem (OS/390)
Describes how to initialize the subsystem at IPL or by a START operator command.

Chapter 3—Initializing the PDM
Describes how to start the PDM and code initialization parameters.

Chapter 4—Defining the SUPRA Server database
Describes the steps to creating and adding the conceptual schema, the internal schema, and the external schema.

Chapter 5—Defining the operational environment
Provides information on how to define the operational environment to run and control your PDM and Directory.

Chapter 6—Using PDM exits
Describes how to code PDM exits.

Chapter 7—Monitoring and controlling the PDM
Describes how to monitor and control the PDM using interactive services and TIS/XA subsystem operator commands, how to terminate a multitask PDM, and monitor PDM memory usage.

Chapter 8—Expanding directory and PDM files
Describes how to expand Directory and PDM files.

Chapter 9—Using the TIS/XA Software Selection Facility
Provides information on using the TIS/XA Software Selection Facility functions and using security exits.

Chapter 10—Generating Directory reports
Describes how to generate reports in OS/390 and VSE.

Chapter 11—Routing console messages (OS/390)
Provides a table showing components that allow you to tailor console message destination.

Chapter 12—PDM Options module
Discusses how to assemble the C$HTOPTN macro and lists available options.

Appendix A—Using SUPRA Server sample procedures, macros, schemas, linkdecks, and sample source members
Describes the samples provided with SUPRA.

Appendix B—Using the trace and debug facilities
Provides a brief description of how to use the debug and trace facilities.

Index

Revisions to this manual

The following changes have been made for this release:

- Change all sentences that read “Register 2 FTBLUPM parm address list” to “Register 2 FTBLUPM parm address list (CICS Interface only)”
- Added the description of the Read Ahead Buffering Facility under the section on “Improving database performance” on page 51.
- Added NOTE regarding the MAXUSER parameter on page 55.
- Changed the syntax in the EXEC PGM statement in SYS1.PROCLIB to add the ecsa-code parameter. See page 60.
- Added new CSIPARM parameters to tables for OS/390 on page 69 and VSE on page 71, and added the descriptions in the section titled “Coding initialization parameters” on page 73.
- Added syntax descriptions and new parameters: BEGDAT, BLDMTFILE, BLDSTFILE, ENDDAT, PSWRD, RDAH, and WARDAY in the PDM Options Module chapter, under “Available options” starting on page 431.
- Added the C$THOPTN macro to the list under “Using SUPRA Server macros” on page 445.
- The section on Assembling the CSTXOPRM macro has been removed from this manual. It has been moved to the *CICS Connector Systems Programming Guide*, P26-7452.

Conventions

The following table describes the conventions used in this document series:

Convention	Description	Example

`Constant width type`	Represents screen images and segments of code.	`PUT 'customer.dat'`
		`GET 'miller\customer.dat'`
		`PUT '\DEV\RMT0'`

Slashed b (ƀ)	Indicates a blank space. The example indicates that four blank spaces appear between the keywords.	`BEGNƀƀƀƀSERIAL`
Brackets [ ]	Indicate optional selection of parameters. (Do not attempt to enter brackets or to stack parameters.) Brackets indicate one of the following situations:	
	A single item enclosed by brackets indicates that the item is optional and can be omitted. The example indicates that you can optionally enter a WHERE clause.	`[WHERE` *`search-condition`*`]`
	Stacked items enclosed by brackets represent optional alternatives, one of which can be selected. The example indicates that you can optionally enter either WAIT or NOWAIT. (WAIT is underlined to signify that it is the default.)	[**(WAIT)** / **(NOWAIT)**]

Convention	Description	Example
Braces { }	Indicate selection of parameters. (Do not attempt to enter braces or to stack parameters.) Braces surrounding stacked items represent alternatives, one of which you must select. The example indicates that you must enter ON or OFF when using the MONITOR statement.	**MONITOR** { **ON** | **OFF** }
Underlining (In syntax)	Indicates the default value supplied when you omit a parameter. The example indicates that if you do not choose a parameter, the system defaults to WAIT.	[**(WAIT)** | **(NOWAIT)**]
	Underlining also indicates an allowable abbreviation or the shortest truncation allowed. The example indicates that you can enter either STAT or STATISTICS.	`STATISTICS`
Ellipsis points...	Indicate that the preceding item can be repeated. The example indicates that you can enter multiple host variables and associated indicator variables.	`INTO :`*`host-variable`* `[:`*`ind-variable`*`],...`

Convention	Description	Example
UPPERCASE lowercase	In most operating environments, keywords are not case-sensitive, and they are represented in uppercase. You can enter them in either uppercase or lowercase.	`COPY MY_DATA.SEQ` `HOLD_DATA.SEQ`
Italics	Indicate variables you replace with a value, a column name, a file name, and so on. The example indicates that you must substitute the name of a table.	`FROM` *`table-name`*
Punctuation marks	Indicate required syntax that you must code exactly as presented. `(  )` parentheses `.` period `,` comma `:` colon `' '` single quotation marks	(*`user-id`*, *`password`*, *`db-name`*) `INFILE 'Cust.Memo' CONTROL LEN4`
SMALL CAPS	Represent a required keystroke. Multiple keystrokes are hyphenated.	ALT-TAB
OS/390 **VSE**	Information specific to a certain operating system is flagged by a symbol in a shadowed box (**OS/390**) indicating which operating system is being discussed. Skip any information that does not pertain to your environment.	**OS/390** See the SUPRA Server procedure library member TIS$RDM for a list of RDM procedures. **VSE** See the SUPRA Server RDM sublibrary member TXJ$INDX for a list of JCL.

SUPRA Server documentation series

SUPRA Server is the advanced relational database management system for high-volume, update-oriented production processing. A number of tools are available with SUPRA Server including DBA Functions, DBAID, precompilers, SPECTRA, and MANTIS. The following list shows the manuals and tools used to fulfill the data management and retrieval requirements for various tasks. Some of these tools are optional. Therefore, you may not have all of the manuals listed. For a brief synopsis of each manual, refer to the *SUPRA Server PDM Digest (OS/390 & VSE)*, P26-9062.

Overview

- *SUPRA Server PDM Digest (OS/390 & VSE)*, P26-9062

Getting started

- *SUPRA Server PDM Migration Guide (OS/390 & VSE)*, P26-0550*
- *SUPRA Server PDM CICS Connector Systems Programming Guide (OS/390 & VSE)*, P26-7452

General use

- *SUPRA Server PDM Glossary*, P26-0675
- *SUPRA Server PDM Messages and Codes Reference Manual (RDM/PDM Support for OS/390 & VSE)*, P26-0126

Database administration tasks

- *SUPRA Server PDM and Directory Administration Guide (OS/390 & VSE)*, P26-2250
- *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260
- *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261
- *SUPRA Server PDM DBA Utilities User's Guide (OS/390 & VSE)*, P26-6260
- *SUPRA Server PDM Logging and Recovery (OS/390 & VSE)*, P26-2223
- *SUPRA Server PDM Tuning Guide (OS/390 & VSE)*, P26-0225
- *SUPRA Server PDM RDM Administration Guide (OS/390 & VSE),* P26-8220
- *SUPRA Server PDM RDM PDM Support Supplement (OS/390 & VSE),* P26-8221
- *SUPRA Server PDM RDM VSAM Support Supplement (OS/390 & VSE),* P26-8222
- *SUPRA Server PDM Migration Guide (OS/390 & VSE)*, P26-0550*
- *SUPRA Server PDM Windows Client Support User's Guide*, P26-7500*
- *SPECTRA Administrator's Guide,* P26-9220

Application programming tasks

- *SUPRA Server PDM DML Programming Guide (OS/390 & VSE)*, P26-4340
- *SUPRA Server PDM RDM COBOL Programming Guide (OS/390 & VSE)*, P26-8330
- *SUPRA Server PDM RDM PL/1 Programming Guide (OS/390 & VSE)*, P26-8331
- *SUPRA Server PDM Migration Guide (OS/390 & VSE)*, P26-0550*
- *SUPRA Server PDM Windows Client Support User's Guide*, P26-7500*

Report tasks

- SPECTRA User's Guide, P26-9561

NOTE

Manuals marked with an asterisk (*) are listed more than once because you use them for multiple tasks.

NOTE

Educational material is available from your regional Cincom education department.

1

Introduction to the SUPRA Server Directory and Physical Data Manager (PDM)

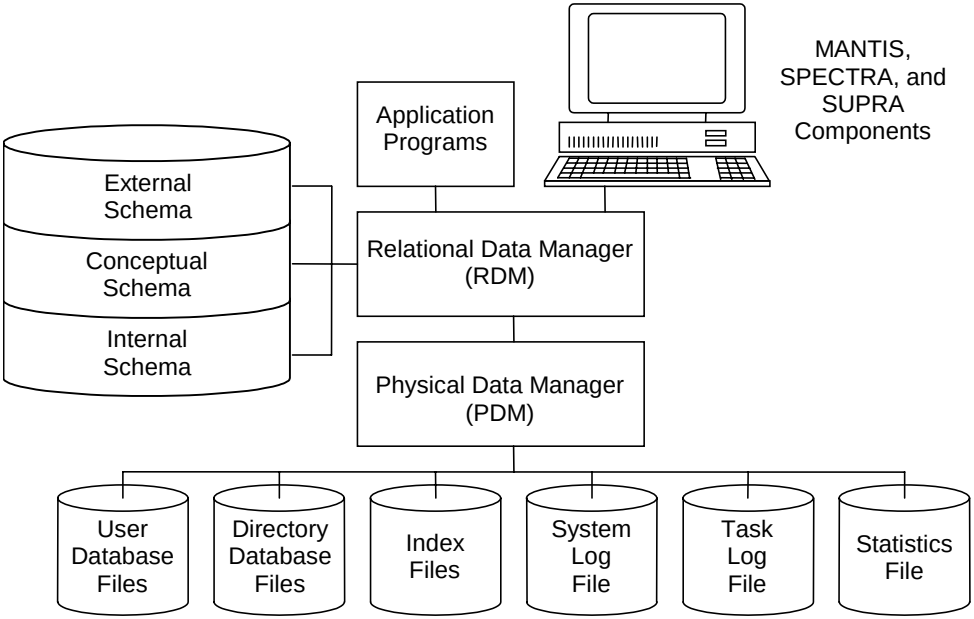
What is the SUPRA Server Directory and PDM?

SUPRA Server is an advanced relational database management system designed for high-volume, update-oriented production processing. It is based on a three-schema architecture. Central to the system are the Directory and the Physical Data Manager (PDM). The Directory is a catalog that stores metadata describing the database. It includes the definition of the internal, conceptual, and external schemas and their relationships to each other. The PDM manages the data structures of the physical files that store the database.

The operations of the Directory and the PDM are closely related to other SUPRA Server components, such as the Relational Data Manager (RDM). This manual discusses only how these components relate to the Directory and the PDM. For a more detailed discussion of any of these components, please refer to the appropriate manuals listed in the front material.

Overview of SUPRA Server components

The following figure shows the basic relationship of other SUPRA Server components to the Directory and the PDM.



Understanding the Directory

The SUPRA Server Directory contains all of the information about your database. It does so within the general framework of a three-schema architecture. This relational database model insulates the user from changes made in the physical storage of the data.

The three-schema architecture consists of the Internal, Conceptual, and External schemas. After installing the subsystem (see "Installing TIS/XA subsystem (OS/390)" on page 53), you must create and define your company's data to the three-schemas, indicate the relationships between the schema data, and indicate how SUPRA Server is to store the information.

The three schemas may be defined as follows:

- **Internal Schema.** Contains the files that contain your data. It defines the most efficient way to access your data. The PDM uses this information to maintain data.
- **Conceptual Schema.** Contains the definition of your company's data model. It defines your company's unique, business-integrity rules.
- **External Schema.** Contains "views" (subsets) of the database for a particular application. These views are derived from the Conceptual Schema.

For more information on creating and defining your database, see "Defining the SUPRA Server database" on page 103.

The Directory contains user environments consisting of the database files that define your company's data (the Conceptual and External schemas). The user environment contains a user schema (a part of the internal schema), which is made up of Directory database files that contain file descriptions and Environment Descriptions. These files are opened by the PDM during initialization.

For initialization, SUPRA Server provides one or more bootstrap environments, which contain the information the PDM requires to open and read the Directory database files contained in the user environment. The bootstrap environment defines the size of the database files, how they are accessed, and how they are allocated. See "Defining the operational environment" on page 115 for information on defining the Directory to the PDM using the tailored bootstrap environment and instructions for modifying the tailored environment to suit your data requirements.

Types of Directory files

The Directory consists of the following files:

- **Name primary file (named C$-N).** Cross-reference file of entity names and definition numbers.
- **Definition Number primary file (named C$-#).** Contains entity descriptions and links to the related files.
- **Structure related file (named C$-S).** Contains detail and header records that describe all of the relationships among the Directory entities.
- **Data related file (named C$-D).** Contains attribute data for entities with variable-length descriptions. The following categories can have data stored on the data file:
  - Access set
  - Buffer pool
  - Log group
  - Maintenance restriction
  - Procedure
  - Table
  - User
- **Text related file (named C$-T).** Contains long and short descriptive text records. Short text and long text are associated with entities in the Directory. You use either the LONG EDIT or SHORT EDIT command to add descriptive text to the Directory.

Data maintained by the Directory

The Directory maintains data by grouping it into five classifications: conceptual schema, external schema, internal schema, system, and user.

Internal schema data

Internal schema data contains the physical representation of your database and its relationships. The PDM uses this information to maintain data. See "Defining the operational environment" on page 115 for information on adding these entities.

Internal schema data consists of the following entities:

- **Buffer Pool.** Describes the file I/O buffers used by the PDM.
- **Environment Description.** Contains execution-time information that describes the operating environment to the PDM.
- **File.** Describes the physical characteristics of your database files.
- **Internal Record.** Defines the physical record layout, including the base portion and the redefined portion of a record.
- **Key Code.** Defines a Physical Field or a combination of Physical Fields as a Secondary Key for an Internal Record.
- **Log Group.** Connects one or more system log data sets together to form a logical system log.
- **Physical Field.** Describes the physical characteristics of a database element, including storage and format information.
- **Schema.** Defines the global view of your database. This global view includes the logical (external) and physical (internal) definitions.
- **Secondary Key.** Provides indexed access to a database file. A Secondary Key identifies a key code or combination of key codes used as the key for indexing.

Conceptual schema data

Conceptual schema data contains relational definitions of the data you use in database design to create relational data models of your organization's functions.

Conceptual schema data consists of the following entities:

- **Attribute.** Defines the characteristics of associated domains and relations.
- **Conceptual Schema.** A collection of relations that define the database in general relational terms.
- **Domain.** A set representing an idea and containing all valid values of an attribute. It specifies the function of the entity and provides validation options.
- **Foreign Key.** Identifies the attribute or combination of attributes in one relation that are defined across the same domains as the attribute(s) that compose the primary key in the same or another relation.
- **Relation.** Defines a two-dimensional table represented by rows (tuples) and columns (attributes). A row is a collection of columns. Each row in a relation contains the same columns but different values. One column or set of columns in the relation is defined as the primary key and uniquely identifies each row.

External schema data

External schema data contains the logical representation of your database and its relationships. Application programs use this information.

External schema data consists of the following categories:

- **Access Set.** Defines navigational information the RDM and Comprehensive Retrieval need to access data.
- **External Field.** Describes the external appearance of a data element along with a physical field name on a database file for use with RDM and Comprehensive Retrieval.
- **Logical View.** Contains information on a particular view of the database and indicates whether the RDM or Comprehensive Retrieval can use this view.

System data

System data describes operational considerations. This information is independent of any schema data for function and remains unchanged during system processing. See "Defining the operational environment" on page 115 for further information on defining the environment.

System data consists of the following categories:

- **Directory Component Description.** Provides execution options for Directory Maintenance.
- **Edit Mask.** Contains the format mask used to edit the physical representation of data into a format suitable for displaying or printing.
- **Reserved Word.** Defines a word with a predefined meaning to the system. This word cannot be used in any other capacity.
- **Table.** Defines table values used to convert or validate field values.

User data

User data describes all system users, any procedures they have defined, and all security restrictions regarding access to data. For information on defining Security Groups and Maintenance Restrictions, see "Defining the operational environment" on page 115.

User data consists of the following categories:

- **Maintenance Restriction.** Defines a set of rules that permit or deny access to Directory entities.
- **Procedure.** Defines retrieval procedures used by the RDM and Comprehensive Retrieval.
- **Security Group.** Contains a collection of Maintenance Restrictions that enforce access rules for the related user.
- **User.** Contains information about each user of Directory Maintenance, SPECTRA, RDM, and Comprehensive Retrieval.

Relationships between Directory entities

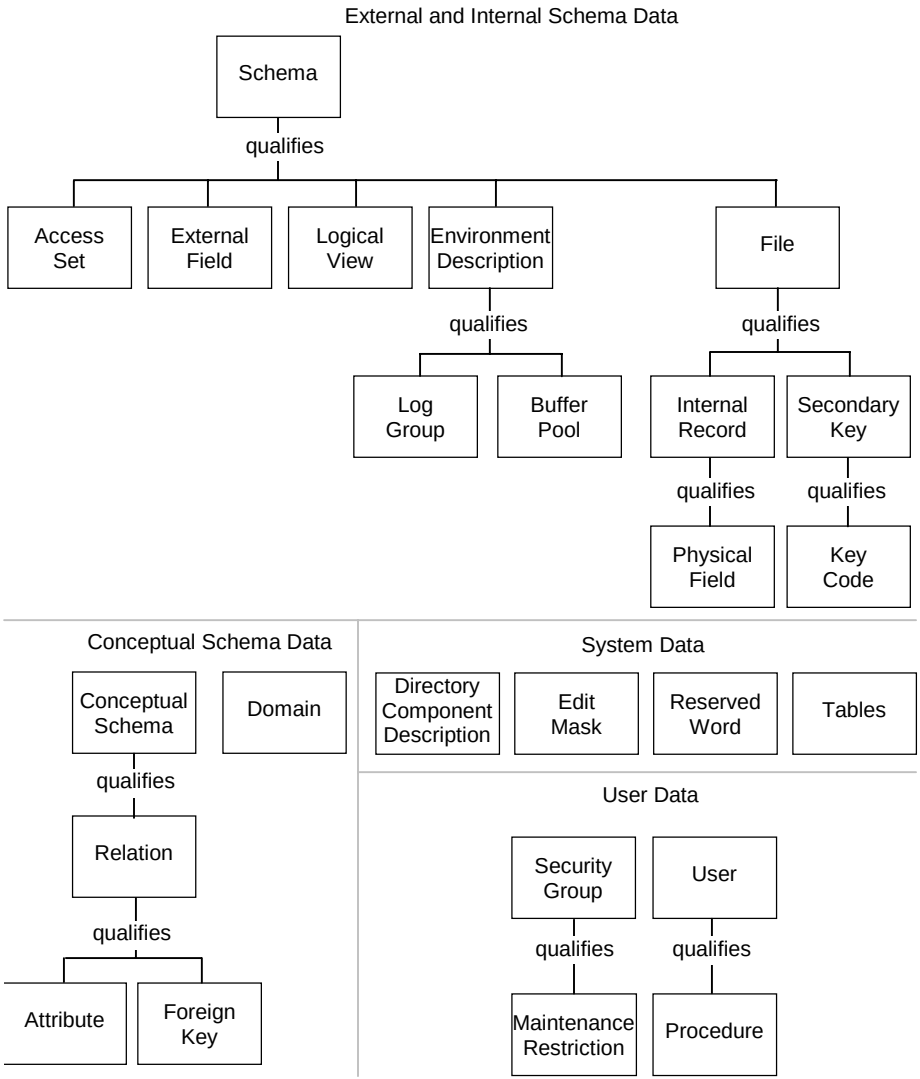
There are two types of relationships between entities on the Directory: hierarchical relationships, which are implicit, and user-defined relationships, which are explicit.

- **Hierarchical relationships.** Certain Directory categories exist in a hierarchical order. This implicit relationship, termed qualification, requires unique identification of entities within the same category. For example, the Schema category qualifies, among other things, the File category. No two files may have the same name if they are qualified by the same schema. The figure at the end of this section shows the qualification, if any, between Directory entities.

  You cannot perform entity maintenance until you have established all qualifiers. This means that you cannot add a file to the Directory unless the schema to which it belongs already exists. Likewise, you cannot add an Internal Record until you have added the file to which the Internal Record belongs.

- **User-defined relationships.** User-defined relationships are explicit relationships between entities that are established and maintained by the SUPRA DBA using the RELATE, RELATIONSHIP CHANGE, RELATIONSHIP DISPLAY, and REMOVE commands.

For more information on the relationship commands, refer to the *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260, or the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261.



Understanding the PDM

The SUPRA PDM maintains and controls the database and environment information stored on the Directory as the Internal Schema. The PDM maintains the internal structure and manages the physical data definitions which control your database.

When requests for data are processed through Directory Maintenance, RDM, or SPECTRA, the PDM accesses the Directory database files. The PDM opens these files during SUPRA Server initialization (if specified by the active schema Environment Description) and closes them during SUPRA Server termination.

After you create the logical representation of your data defined in the Conceptual Schema, you must create and manage the physical environment defined in the Internal Schema. In other words, you need to actually allocate the physical files represented by the Internal Schema definitions on the Directory. The PDM manages and uses the following data contained in the Internal Schema:

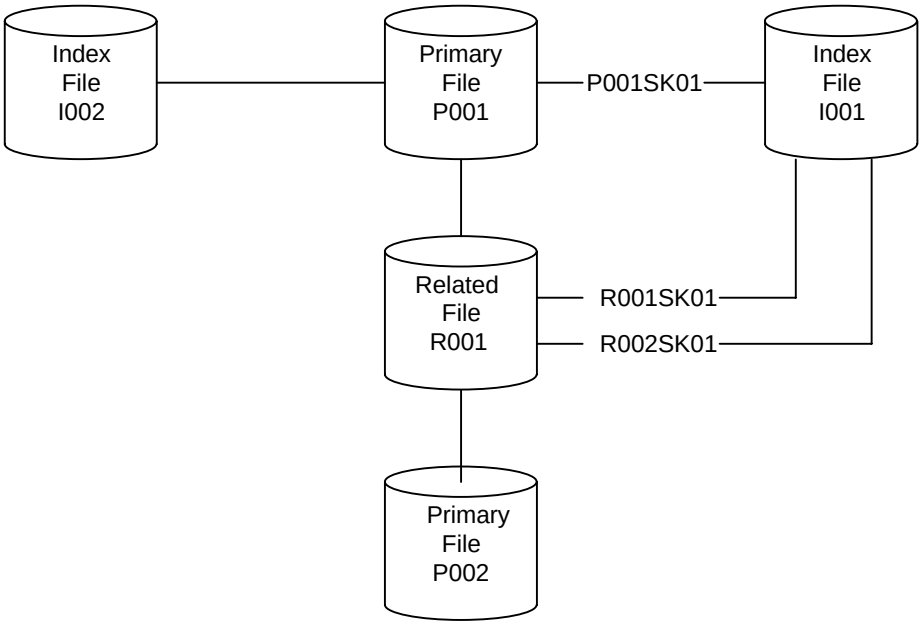
- **Schemas.** Global views of the entire database
- **Environment Descriptions.** Execution-time information describing the operating environment of the PDM
- **Files.** Physical characteristics of the database files
- **Buffer Pools.** I/O buffers used by the PDM
- **Log Groups.** Characteristics of the System Log groups
- **Internal Records.** Physical record layouts
- **Physical Field.** Physical characteristics of a database element
- **Secondary Key.** Key code(s) used as the key for indexing
- **Key Code.** Physical field(s) constituting a Secondary Key

Types of PDM files

The PDM contains the following types of files:

- **Primary file.** Contains data records, each of which includes a unique key, sufficient to access the record.
- **Related file.** Contains data records that can be accessed with pointers (links) in a primary file's data record.
- **Index file.** The index file does not contain data records; it contains definitions and indices for secondary keys. A Secondary Key consists of specified data from the data records of a primary or a related file, and it provides an alternate method of accessing those records.
- **Statistics Log file.** Stores execution statistics.
- **Task Log file.** Keeps track of all updates and relevant information on a task-and-transaction basis. Permits recovery or resetting of tasks to the beginning of each task's current transaction.
- **System Log file.** Specifies your system's logging options in your Environment Descriptions by name in the CSIPARM file.

The following figure illustrates the relationship between PDM files. For information on adding PDM files, see “Defining the SUPRA Server database” on page 103.



You must define the PDM files on the Directory in order to access them using RDM. To define the files, use either Batch or Online Directory Maintenance.

Relationship of the Relational Data Manager to the PDM

The Relational Data Manager (RDM) is a logical interface for your RDML application programs (written in MANTIS, COBOL, PL/1) and SPECTRA. With RDM, you can access database information as you view it without concern for its physical location or structure. RDM handles data navigation, security, referential integrity, and data validation for application programmers and end users; they work at the logical level using defined views or derived relations stored on the Directory as views. See “Overview of SUPRA Server components” on page 22 for an illustration of how the RDM relates to the Directory and the PDM.

The RDM processes requests for data, and the PDM then performs the input and output operations on the required PDM files. The RDM accesses data through the derived views. Derived views are defined by the SUPRA DBA, using DBAID or Directory Maintenance, and then stored on the Directory.

RDM requests are broken down by the RDM and subsequently mapped to the PDM physical files defined in the Internal Schema. This process ensures that all of your data conforms to your business integrity rules and redundancy controls, as specified in the Conceptual Schema.

For more information on using RDM, refer to the *SUPRA Server PDM RDM Administration Guide (OS/390 & VSE)*, P26-8220.

Versions of the PDM

The PDM has two versions: (1) the single-task PDM, which communicates with only one task and is initiated and terminated with that task, (2) and the multitask PDM, which exists independently and communicates with one or more tasks.

In a single-task PDM:

- A single-task PDM operates in single-task mode and exists in the same address space as the application task that is communicating with it. Only one application task is active, and this task has exclusive access to the single-task PDM through the STST interface.
- A single-task PDM exists only while the task exists.
- You initialize the PDM by issuing a SINON command. Refer to the *SUPRA Server PDM DML Programming Guide (OS/390 & VSE)*, P26-4340, for information about the SINON DML command.

In a multitask PDM:

- A multitask PDM exists independently from the applications communicating with it and executes and initializes in its own address space.
- Zero, one or many Batch applications or CICS systems, each with one or more tasks, can access the same copy of the multitask PDM through the interface.
- You must bring up the multitask PDM before you start any applications and before you connect CICS with the PDM.

Types of PDM tasks

Three types of tasks communicate with the PDM. Each of these tasks communicates in different ways with a single or multitask PDM.

- **Batch Tasks.** Batch tasks can communicate with a single-task or a multitask PDM. When communicating with a single-task PDM, the batch task uses the STST (single-task/single-task) interface. Using the STST interface, the batch task and the PDM execute in the same region. Any files opened for update by a single-task PDM cannot be opened for update by any other PDM.

  To communicate with a multitask PDM, the batch task uses the STMT (single-task/multitask) interface. For this mode, a PDM file opened for update may be updated by any tasks communicating with the same PDM. Any files opened by a multitask PDM cannot be opened for update by any other PDM.

- **CICS Tasks.** CICS tasks use the CSMT (CICS/Multitask Interface) to communicate with a multitask PDM.

- **SUPRA SQL DRDM Tasks.** SUPRA SQL tasks communicate with the DRDM using SQL for PDM requests. The DRDM contains a PDM multitask interface (DCMT).

PDM operating modes

The PDM can run in any of the following operating modes:

- **Single-Task Operating Mode.** In single-task operating mode, a single-task PDM exists in the same address space as the application task that is communicating with it. Only one application task is active and this task has exclusive access to the single-task PDM through the STST interface.

- **Central Operating Mode.**

  - OS/390 OS/390 central mode—Under OS/390, a central PDM uses the TIS/XA subsystem to control communication with PDM interfaces in other address spaces. Using the TIS/XA subsystem serves the following purposes:

    - Provides an anchor for cross-memory communication

    - Monitors the system

    - Limits authorized service requirements

    - Facilitates data movement

    The following restrictions apply:

    - Only one PDM interface is allowed in each address space.

    - If a job starts a PDM in the central or attached central mode, the *same job* cannot start another PDM in either of these modes.

  - VSE VSE central mode—Under VSE, the central PDM uses protect key zero to move data across partitions and XPCC to move data across address spaces. The central PDM and its interfaces in the same address space and in static (nondynamic) partitions communicate with each other by cross-partition posting of XECBs.

    If your applications reside in different address spaces, you need to code the PATH parameter in the CSIPARM file.

- **Attached Central Operating Mode.** In attached central mode, the multitask PDM executes and initializes in the same address space or partition as the initial application task. Each additional application task resides in a different address space from the one containing the multitask PDM.

  All cross-address space communications between the multitask PDM and an application task are accomplished through an operating-system and mode-dependent mechanism. Only intra-address space communication occurs between the PDM and DDI or between the PDM and the initial application task.

  You can attach the PDM by one of the following methods:

  - OS/390 or VSE with CICS, you can specify the ATTACH parameter in the CSIPARM file. The PDM is automatically attached and initialized when the internal CONNECT command is issued.

  - OS/390 only, you can use the Attacher program.

OS/390

If a job starts a PDM in the central or attached central mode, the same job cannot start another PDM in either of these modes.

- **Attached Operating Mode.** In the attached operating mode, the application task and a multitask PDM reside in the same address space with the PDM running as a subtask of the application and dependent on the application.

  Intra-address space communication occurs on an STMT interface between the PDM and the DDI and on an STMT interface between the PDM and the task.

  There are two ways to run in attached mode:

  - OS/390 Use the Attacher program to attach a PDM and an application as subtasks.

  - OS/390 or VSE Run CICS with the ATTACH parameter in the CSIPARM file.

Updating and maintaining the database

Once you have defined your user schemas, you can use Directory Maintenance to change the data on the schema currently being used by the PDM (the active schema). Within Directory Maintenance, you can enable Active Schema Maintenance to perform maintenance functions on the active schema.

Using Directory Maintenance

You can use Batch or Online Directory Maintenance to update the Directory. The operation of both facilities is similar; both ensure that names are unique and your input is valid. To use Online Directory Maintenance, you process a series of screens for each entity you want to update. To use Batch Directory Maintenance, you code a series of input statements for each entity you want to update. You can use Batch Directory Maintenance to do the following:

- Process large volumes of input that would be too time consuming to enter using the Online Directory Maintenance facility or that should be scheduled during off hours.
- Keep an audit trail of changes made to the Directory.
- Use the same Directory Maintenance input multiple times on the same or different Directories.
- Process Directory Maintenance statements created by other programs, such as the Inter-Directory Copy utility or the DDL convert.

Directory Maintenance consists of several load modules. Because all of the modules do not have to be in memory at the same time, your memory requirements are somewhat flexible. For more information on tuning and memory requirements, refer to the *SUPRA Server PDM Tuning Guide (OS/390 & VSE)*, P26-0225.

For reference information on Directory Maintenance, refer to the *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260, and the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261.

Using Active Schema Maintenance

You can enable Active Schema Maintenance to dynamically allocate files and perform certain maintenance functions without shutting down database operations. You can perform Active Schema Maintenance with Batch or Online Directory Maintenance. When using Active Schema Maintenance, you can modify only the active schema. You cannot use update functions against any other schema until you disable active schema maintenance. However, you can perform nonupdate functions (DISPLAY, SHORT EDIT, LONG EDIT, etc.) against a nonactive schema.

Enabling Active Schema Maintenance

To enable online Active Schema Maintenance, follow these three steps (see the figure on the next page for the corresponding screens):

1. Select SC (Schema) from the category menu screen or any command menu screen.

2. Select SF (Special Function) from the schema command menu and specify the active schema name. On the SPECIAL FUNCTION screen, the Maintain Active Schema field is N (the default).

3. Change the Maintain Active Schema field to Y. Directory Maintenance then returns the schema command menu and a message indicating that active schema maintenance is now permitted.

Once you have enabled active schema maintenance, you can use it to perform all commands for all categories and entities qualified by the schema. Some of the changes you make to certain categories during active schema maintenance may affect the PDM or RDM. You must consider the following when enabling Active Schema Maintenance:

- Changes you make to Environment Descriptions or Log Groups are not implemented until the next cold start of the PDM.

- To pass any changes that you make to Buffer Pools, Files, Internal Records, Key Codes, Physical Fields, and Secondary Keys to the PDM, use the File utilities or wait until the next cold start of the PDM.

- Changing Physical or External Fields which make up file definitions may affect RDM users. Directory Maintenance displays a warning message when this situation occurs.

- RDM views that use the modified Physical or External Fields require the following actions:

  - If RDM views are bound, rebind them using the DBAID utility or the changes will not be used.

  - If RDM views are global, reopen them by bringing RDM down and then back up again. Reopen all views in use.

1

```
ENTER CATEGORY:  SC         TIS/XA DIRECTORY MAINTENANCE

AS  ACCESS SET                                LV  LOGICAL VIEW
AT  ATTRIBUTE                                 MR  MAINTENANCE RESTRICTION
BP  BUFFER POOL                               PF  PHYSICAL FIELD
CS  CONCEPTUAL SCHEMA                         PR  PROCEDURE
```

2

```
SCHEMA:  COMMANDS      TS/XA DIRECTORY MAINTENANCE

AD  ADD                                       RD  REL DISPLAY
CG  CHANGE                                    RL  RELATE
CK  CHECK                                     RM  REMOVE
CO  COPY                                      RN  RENAME
DE  DELETE                                    SD  STRU DSPLY
DI  DISPLAY                                   SE  SHORT EDIT
LE  LONG EDIT                                 ST  SHORT EDIT
LT  LONG TEXT                                 UT  UTILITIES

ENTER SELECTION CODE: SF           SUBCATEGORY CODE:

ENTER NAMING DATA:
SCHEMA:            BURRYSCH
```

3

```
SCHEMA:  SPEC FUNC    TIS/XA DIRECTORY MAINTENANCE    1 OF 1
SCHEMA:  BURRYSCH

LAST UPDATE        19.50.51      01/06/93  V:   0002   USER: CSI-DBA
MAINTAIN ACTIVE SCHEMA: Y
```

4

```
SCHEMA:  COMMANDS    TIS/XA DIRECTORY MAINTENANCE

AD  ADD                                       RD  REL DISPLAY
CG  CHANGE                                    RL  RELATE
CK  CHECK                                     RM  REMOVE
CO  COPY                                      RN  RENAME
DE  DELETE                                    SD  STRU DSPLY
DI  DISPLAY                                   SE  SHORT EDIT
LE  LONG EDIT                                 ST  SHORT EDIT
LT  LONG TEXT                                 UT  UTILITIES

ENTER SELECTION CODE: SF           SUBCATEGORY CODE:

ENTER NAMING DATA:
SCHEMA:             BURRYSCH

ACTIVE SC: BURRYSCH   ACTIVE ED: BURRY ENV MAINTENANCE PERMITTED:  YES

CSMM6221:  ACTIVE SCHEMA MAINTENANCE IS NOW PERMITTED
```

To disable Active Schema maintenance, follow these three steps:

1. Select SC (Schema) from the category menu screen or any command menu screen.

2. Select SF (Special Function) from the schema command menu. On the SPECIAL FUNCTION screen, the Maintain Active Schema field is N (the default).

3. Directory Maintenance returns the schema command menu and a message indicating that active schema maintenance is denied. You can now perform maintenance to all schemas except the active schema.

Controlling and monitoring the database

SUPRA Server provides the following utilities that allow you to organize and maintain data in an efficient, flexible manner:

- Database utilities
- Directory Maintenance utilities
- Interactive Services
- SUPRA logging
- File space management

Using database utilities

To execute the SUPRA DBA utility functions, code Utility Command Language (UCL) programs. For information on the utilities and how to code them, refer to the *SUPRA Server PDM DBA Utilities User's Guide (OS/390 & VSE)*, P26-6260.

Two additional DBA utility functions do not use UCL:

- **PDM Termination Utility.** Allows you to terminate a central PDM.
- **Execution Statistics Utility.** Prints the contents of the Statistics File. The Statistics File is produced when the PDM calculates and accumulates specific statistics for the Read Statistics (RSTAT) command or when the PDM is initialized or terminated.

Several file compatibility options are available to you. If you use any file-compatibility options, refer to the *SUPRA Server PDM DBA Utilities User's Guide (OS/390 & VSE)*, P26-6260, for considerations on using the DBA utilities with these options.

Using Directory Maintenance utilities

Directory Maintenance utilities include:

- **DDL Convert Utility.** Converts Series 80 TOTAL DDL statements to SUPRA Server Batch Directory Maintenance (BDM) statements. Refer to the *SUPRA Server PDM Migration Guide (OS/390 & VSE)*, P26-0550, for information on using this utility.

- **Inter-Directory Copy Utility.** Copies all or part of the entities from one SUPRA Server Directory to another. Refer to the *SUPRA Server PDM DBA Utilities User's Guide (OS/390 & VSE)*, P26-6260, for information on using this utility.

Using Interactive Services

Interactive Services, which you access through the TIS/XA Software Selection Facility Menu, has programs to help you maintain the PDM. Activities you can perform through Interactive Services include:

- Display statistics information online

- Display environment information online

- Terminate tasks that fail to restart

- Add your own utilities to the Interactive Services Menu

See "Monitoring and controlling the PDM" on page 279 for more information on using Interactive Services.

Using SUPRA Server logging

SUPRA Server logging protects data stored in user files by recording changes to the contents of SUPRA Server files or to the status of SUPRA Server files or tasks. You use log files to back off changes, reapply changes, restart tasks, and analyze the logged activity.

Two types of logging are available: task logging and system logging.

NOTE

Cincom recommends for the SUPRA PDM that you specify task logging and system logging. This provides the maximum recovery options for PDM database files and for restart of PDM tasks.

Task logging

Task logging ensures that the PDM saves enough information for all tasks and transactions to perform Task Level Recovery (TLR). This information is stored on the PDM Task Log File (TLF).

If you wish to use task logging, you must specify task logging both in your bootstrap Directory Environment Description and your user Environment Description. If you consider changing the supplied TLF, refer to the *SUPRA Server PDM Logging and Recovery Guide (OS/390 & VSE)*, P26-2223, for recommendations.

System logging

System logging allows you to recover the database if the TLF is unreadable or unused, or if an updated data file is unreadable and must be reloaded. When you specify system logging, the PDM saves chronological audit information to the System Log File (SLF).

You can partially control the type of information saved on the SLF in your active Environment Description (before images, after images, PDM commands, sign-ons, and sign-offs). Your Directory must contain appropriate definitions for system logging to run.

For more information on logging facilities, refer to the *SUPRA Server PDM Logging and Recovery Guide (OS/390 & VSE)*, P26-2223.

Suppressing logging for database files

You can select which database files are logged under Task logging or System logging in the Environment-Description-to-File relationship. You can select individual files to have logging suppressed. (The default is to log the files.)

Logging of information for directory files cannot be suppressed, but you can suppress logging of MANTIS or SPECTRA slide files.

Cincom recommends that you log all files; however, you may find it useful to suppress logging on some work files or noncritical files in a test environment.

To view the screens you use when you create the Environment-Description-to-File relationship and to see the logging options available for individual files, see “Adding Environment Descriptions” on page 140.

For more information on suppressing logging under SUPRA, refer to the *SUPRA Server PDM Logging and Recovery Guide (OS/390 & VSE)*, P26-2223.

Managing file space

This section provides considerations for managing file space using the File Experience table option. You can select this option either expressly or by default, using the PRIMAP and RELMAP parameters in the CSIPARM file. This option is advantageous when you expect to add records to densely filled data files. See "Initializing the PDM" on page 63 for an explanation of these parameters.

When selected, the file experience table locates free space records when adding records to primary and/or related files (other than KSDS VSAM files). The file experience table is:

- Allocated when the file is opened
- Updated as the file is searched during an execution of the PDM
- Freed when the file is closed

Refer to the *SUPRA Server PDM Tuning Guide (OS/390 & VSE)*, P26-0225, for more information on the file experience table.

Calculating space for index files

In managing file space, the amount you estimate for your index file is very important. There are two ways you can use the index file space calculation:

- As an estimation tool to determine the index file block size.
- To estimate, for a given block size, how many blocks the Secondary Key will need on the index file.

Examples of index file space calculations can be found in the *SUPRA Server PDM Tuning Guide (OS/390 & VSE)*, P26-0225.

Monitoring memory

A memory monitor facility is now available in SUPRA. It reports the available memory, current usage, and highest usage at the time the console message containing the information displays. (This information is also available with the PDML SHOWX command. For more information on the SHOWX command, refer to the *SUPRA Server PDM DML Programming Guide (OS/390 & VSE)*, P26-4340.)

The memory monitor facility also allows for memory relief when the percentage of memory used is greater than the user-specified threshold. At the time of a COMIT, a task will release any unessential memory if the usage is greater than the threshold. You specify the memory Cleanup Threshold in the Environment Description. See "Adding Environment Descriptions" on page 140 for more information on adding an Environment Description. For more information on the Cleanup Threshold, refer to the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261, or the *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260.

Managing record holding control blocks

The SUPRA DBA now has more control over creating some record holding control blocks. In prior releases of SUPRA, record monitoring control blocks were generated automatically, which wasted PDM memory, as there were few applications that used this feature.

When you add an Environment Description under Directory Maintenance, you are asked to specify the maximum number of monitored records. This field is used for the number of shared records, even though it refers to monitored records. You should specify a nonzero value in this field to create shared record-holding entries.

See "Adding Environment Descriptions" on page 140 to view the Online Directory Maintenance screens when adding an Environment Description. For more information on adding an Environment Description, refer to the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261, or the *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260.

Improving database performance

Optimizing database performance involves tuning, or choosing the best options, for the following performance-related areas in the PDM:

- **Database statistics.** Database statistics monitor the execution of the database system and help you tune your own database. Execution statistics are stored on the Statistics file, and you can generate a report of the statistics by running the Execution Statistics utility. For more information on using this utility, refer to the *SUPRA Server PDM DBA Utilities User's Guide (OS/390 & VSE)*, P26-6260.

  Execution statistics records are added to the Statistics file when the PDM initializes, when you issue an RSTAT command, or when the PDM terminates. In addition, these statistics are available to user-written programs through the SHOWX command. For a description of the SHOWX command, refer to the *SUPRA Server PDM DML Programming Guide (OS/390 & VSE)*, P26-4340.

NOTE

Statistics from the optional Buffer Cache Facility are not stored in the Statistics file. They are provided through console messages.

- **Memory allocation.** The PDM's memory size is specified in the active Environment Description and under OS/390 in the CSIPARM file. The PDM uses this memory specification for control blocks, buffers, and other work areas. Specifying too much memory can interfere with applications which share an address space with an attached PDM or an attached central PDM. Specifying too little memory can degrade the PDM's performance or cause it to fail.

  The optional PDM Extended Storage support (SUPRA Server Release 2.4) relieves storage constraints in the PDM address space below the 16 MB line by placing selected PDM control blocks above the 16 MB line.

OS/390

PDM extended storage (memory allocation) is supported only under OS/390.

The OS/390 PDM executes in AMODE 31, except when requirements dictate AMODE 24, and allocates the following control blocks to space above the 16 MB line:

- File descriptors
- Record holding table
- Experience tables (PRIMAP and RELMAP)
- ACTV lists, excluding register save areas

NOTE

Both in OS/390 and VSE, the PDM interfaces operate in AMODE-31, and the CICS interface is resident in extended storage.

- **Held Records.** Held records, or the maximum number of records the PDM can hold, lock and/or monitor simultaneously, are specified in the active Environment Description. Specifying too many held records wastes memory, while specifying too few held records can degrade the PDM's performance of update tasks.

- **Buffer Pools.** Buffer pools, or the areas where information is temporarily held until it is needed by a particular program, are also defined in the active Environment Description. Buffer pools give you great flexibility and provide one of the best opportunities for improving performance. In the active Environment Description, you can specify how many buffers are in each pool and which files are used by which pool.

- **Buffer Cache Facility.** The optional Buffer Cache Facility (SUPRA Server Release 2.7) improves PDM performance by using user-defined pools of cache blocks to buffer PDM BDAM files above the 16 MB line reducing physical reads to files. It also allows you to free the PDM region from certain memory restrictions for users with large numbers of files by reclaiming much of the buffer memory below the 16 MB line. You cannot cache VSAM and BSAM files.

  The Buffer Cache Facility is defined as a user exit in the user Environment Description (see “Adding Environment Descriptions” on page 140 and “Using PDM exits” on page 249). The facility reads an input file (see “Coding the Buffer Cache Facility input file” on page 95) containing control information and allocates storage for the cache. During read processing, if the PDM determines that a file block does not exist in a buffer, the PDM looks next to the cache. If found, the block is moved to a buffer. If it is not found, a standard I/O request is issued. Once a physical read is completed, the facility moves a copy of the block from the buffer to the cache. During PDM write processing, the facility updates data in the cache.

- **Read Ahead Buffering Facility.** New with Supra Release 2.7, the Read Ahead Buffering Facility allows more than one record at a time to be returned to requesting STMT interfaces. The facility has been added for the RDNXT, READV, READR, and READX DML functions.

  The buffering facility is aimed at those applications that read many records in a file before finding the record desired. Its purpose is to reduce the overhead time required to transfer those requests to the central PDM by eliminating the duplicate subsystems requests in OS/390 and XPCC requests in VSE.

  The facility allows for three stages of buffering. The first stage returns 1 record at a time as it always has. Stage two returns as many records as will fit in the buffer specified in the CSIPARM RDBUFFER parameter without doing any additional physical reads. Stage three will fill the buffer even if additional I/O is needed.

  The parameters governing the move from one stage of buffering to another stage of buffering is governed by the user specifying various CSIPARM parameters. RDTHRSH1=n indicates that the user wants to go to stage two after n DML requests with the same function, file and accompanying parameters. RDTHRSH2=m, likewise, indicates a move to stage 3 after m DML requests with the same parameters.

  It is the user's responsibility to tune these parameters for each application or determine a set of general values that best suit all applications.

Keep in mind the following general considerations when choosing options to optimize database performance:

- Log files cannot share buffers being used by database files or by the Statistics file.
- The PDM does not allocate any buffers for KSDS files. Their buffering is handled by the VSAM access method.
- Files that share Buffer Pools should have similar block sizes; otherwise, memory is wasted.
- In general, the more active the file, the more buffers should be in its pool and the fewer files should share its pool.

Controlling VSE interface polling

The VSE DBA now has control over the interface-polling rate for XECB communication, which allows you to control how soon a VSE PDM application can be resubmitted after it abends.

You specify the polling rate in the Environment Description. See "Adding Environment Descriptions" on page 140 for information on adding an Environment Description. For more information on adding an Environment Description, refer to the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261, or the *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260.

2

Installing TIS/XA subsystem (OS/390)

Installation

The TIS/XA subsystem provides an anchor for communications across address spaces and is necessary when you use a multitask PDM in central and attached central operating modes. See "Introduction to the SUPRA Server Directory and Physical Data Manager (PDM)" on page 21 for a discussion of PDM operating modes.

The subsystem also provides an interface to the OS/390 operating system, handling system events such as operator commands. Several operator commands exist to provide you with information concerning the status of the central PDM and any connected interfaces, and to stop the subsystem. For more information on the TIS/XA subsystem commands, see "Monitoring and controlling the PDM" on page 279.

You can initialize the TIS/XA subsystem in one of two ways:

- For permanent installation, Cincom recommends initializing the TIS/XA subsystem at IPL (initial program load) time.
- The subsystem or batch job can also be initialized with a START operator command. This method allows you to start and stop the subsystem as often as you wish.

Both methods require an IPL for the system changes to take effect.

NOTE

If you are migrating from an earlier release, you must define a new subsystem that is different from the one used by that earlier release. SUPRA Server subsystems are neither upward nor downward compatible. They are unique to their particular release of SUPRA. Ensure you are installing the current subsystem for your release.

Initializing the subsystem at IPL

To initialize the permanent subsystem at IPL, follow these six steps:

1. Define the TIS/XA subsystem in the subsystem table.

   a. Update IEFSSN*xx*:

   b. Specify several subsystem IDs (such as SUP1, SUP2, SUP3) in the IEFSSN*xx* SYS1.PARMLIB member. One subsystem is enough for any number of central PDMs, but you will need different subsystem names to run different versions of the subsystem concurrently.

   c. Update the IEFSSN*xx* member using the names in an 80-byte record format as follows:

   ssname*,CSTSINI,*command-prefix

   ssname

   Description Specifies the subsystem name.

   Format 1–4 alphanumeric characters (first character must be alphabetical)

   CSTSINI

   Description CSTSINI is the initial routine.

   command-prefix

   Description Identifies the TIS/XA subsystem operator commands.

   Default /

   Format A single character

   Consideration You must use a different command prefix for each different subsystem.

2. Define an APF-Authorized Library for the TIS/XA subsystem as follows:

   a. Allocate library of 1 cylinder to hold 20 members.

   b. Update LNKLST*xx*. Specify the library name and volume in the LNKLST*xx* S1.PARMLIB member. The library must be in the LNKLST concatenation.

   c. Update IEAAPF*xx*. Specify the library name and volume in the IEAAPF*xx* SYS1.PARMLIB member, making it APF authorized.

3. Increase the MAXUSER parameter in the IEASYS*xx* SYS1.PARMLIB member as follows:

   a. Update IEASYS*xx*:

   b. Increase the MAXUSER parameter in the IEASYS*xx* SYS1.PARMLIB member by 10 to 15.

NOTE

The value of MAXUSER should be increased because each time a job listed in an Entry Table (which provides space switch Program Call routines) terminates, OS/390 terminates the address space. OS/390 does not reassign the address space ID (ASID) that represents this address space until the next IPL. MAXUSER should be increased by the number of times all central PDMs terminate between IPLs. For more information about MAXUSER, refer to the *IBM OS/VS2 OS/390 System Programming Library: Initialization and Tuning Guide*, GC28-1029.

4. If you defined a separate library in Step 3, move the TIS/XA subsystem modules to the subsystem Library as follows:

   a. Move the TIS/XA subsystem modules CSTSAFMX, CSTSINI, CSTSPCS, CSTSSSI, and CSTSWTO in the APF-Authorized Library specified in Step 2. These modules are in the main LINKLIB on the installation tape.

   b. Update LPA resident modules. In order to bring up the TIS/XA subsystem at IPL time, the subsystem modules CSTSPCS, CSTSSSI, and CSTSWTO must be available from the LPA. Use one of two methods to place the modules in the LPA:

      i. Copy or move the CSTSPCS, CSTSSSI, and CSTSWTO modules to SYS1.LPALIB.

      ii. Specify the LINKLIB and the three modules in the IEALPA*xx* member in SYS1.PARMLIB. Refer to IBM documentation for more information on SYS1.PARMLIB.

5. Install Dump Format Exit. Dump exits format any TIS/XA subsystem abend dumps. For SUPRA PDM users, SYSUDUMP dumps will contain PDM subsystem control block data only if the Dump Format exit has been installed. Install dump format exits as follows:

   a. Linkedit, copy, or move the CSTSAFMX module into SYS1.LPALIB, SYS1.LINKLIB, or any library in the LNKLST*xx* SYS1.PARMLIB member.

   b. Add the CSTSAFMX name to the IEAVADFM table in IGC0805A.

   For more information, refer to the *IBM XA User Exits Manual,* GC28-1147, under SNAP, SYSABEND, and SYSUDUMP Dumps.

6. Initialize the TIS/XA subsystem at IPL as follows:

   - If the TIS/XA subsystem modules reside in the Link Pack Area (LPA) instead of the Modified Link Pack Area (MLPA), you need to IPL with the CLPA command.

   Refer to IBM documentation for instructions on using the MLPA.

NOTE

You need to specify the subsystem name in the TISSID parameter in your CSIPARM file used by the PDM, and in your CSIPARM files used by any tasks communicating with the PDM. Sample CSIPARM members are supplied on the JCLLIB on your release tape. For more information on CSIPARM files, see “Initializing the PDM” on page 63.

Initializing the subsystem by a START operator command

You can also initialize the subsystem by a START operator command. Using this method allows you to start and stop the subsystem as often as you wish. Initializing the subsystem with a START operator command requires an IPL for the changes to take effect.

1. Define the TIS/XA subsystem in the Subsystem Table.

   This step is the same as Step 1 for initializing the subsystem at IPL time (see "Initializing the subsystem at IPL" on page 54), except that you must omit CSTSINI and the command-prefix when listing the subsystem names in the IEFSSN*xx* SYS1.PARMLIB member.

   a. Update IEFSSN*xx* as follows. Specify several subsystem IDs (such as SUP1, SUP2, SUP3) in the IEFSSN*xx* SYS1.PARMLIB member. One subsystem is enough for any number of central PDMs, but you will need different subsystems to run different service levels of the TIS/XA subsystem concurrently.

   b. Update the IEFSSN*xx* member using the names in an 80-byte record format as follows:

   ssname

Description	Specifies the subsystem name.
Format	1–4 alphanumeric characters (first character must be alphabetical)

2. Define an APF-Authorized Library for the TIS/XA subsystem as follows:

   This step is the same as Step 2 for initializing the subsystem at IPL time ("Initializing the subsystem at IPL" on page 54), except for testing you may want more than one authorized library (one in the LNKLST concatenation and the others in candidate steplibs). This library is the steplib for the procedure you will create in Step 6 to invoke the driver program.

   a. Allocate a library of 1 cylinder to hold 20 members.

   b. Update LNKLST*xx*. Specify the library name and volume in the LNKLST*xx* SYS1.PARMLIB member. The library must be in the LNKLST concatenation.

   c. Update IEAAPF*xx*. Specify the library name and volume in the IEAAPF*xx* SYS1.PARMLIB member, making it APF authorized.

3. Increase the MAXUSER parameter in the IEASYS*xx* SYS1.PARMLIB member as follows:

   This step is the same as Step 3 for initializing the subsystem at IPL time (see "Initializing the subsystem at IPL" on page 54).

   Update IEASYS as follows. Increase the MAXUSER parameter in the IEASYS*xx* SYS1.PARMLIB member by 10 to 15. The value of MAXUSER should be increased because each time a job listed in an Entry Table (which provides space switch Program Call routines) terminates, OS/390 terminates the address space. OS/390 does not reassign the address space ID (ASID) that represents this address space until the next IPL. MAXUSER should be increased by the number of times all central PDMs terminate between IPLs. For more information about MAXUSER, refer to the *IBM OS/VS2 OS/390 System Programming Library: Initialization and Tuning Guide*, GC28-1029, under System Initialization.

4. Move the TIS/XA subsystem modules to the subsystem library as follows:

   a. The TIS/XA subsystem modules CSTSINI, CSTSISR, CSTSPCS, CSTSSDRV, CSTSSSI, and CSTSWTO need to be put in the APF-Authorized Library specified in Step 2. These modules are in the main TIS/XA linklib on the installation tape. This library is the steplib for the procedure you will create in Step 6 to invoke the driver program. The TIS/XA subsystem modules CSTSSDRV, CSTSISR, CSTSPCS, and CSTSSSI will be loaded from the steplib into the common storage area (CSA). The modules CSTSWTO and CSTSAFMX must be in the LNKLST concatenation, so you cannot have different versions of this module. By this means, each subsystem can have different levels of code (for testing and production), except for CSTSWTO and CSTSAFMX.

   b. Update IEALPA*xx*. For efficiency, Cincom recommends that you also put the highly used TIS/XA subsystem modules CSTSPCS, CSTSSSI, and CSTSWTO in the SYS1.LPALIB, and specify them in the IEALPA*xx* SYS1.PARMLIB member.

5. Install Dump Format Exit.

   This step is the same as Step 5 for initializing the subsystem at IPL time (see "Initializing the subsystem at IPL" on page 54). This exit formats any TISXA subsystem abend dumps for easier reading. For SUPRA PDM users, SYSUDUMP dumps contain subsystem control block data.

   Install the dump exits as follows:

   a. Linkedit the CSTSAFMX module into SYS1.LPALIB, SYS1.LINKLIB or any library in the LNKLST*xx* SYS1.PARMLIB member.

   b. Add the CSTSAFMX name to the IEAVADFM table in IGC0805A. For more information, refer to the *IBM XA User Exits Manual,* GC28-1147, under SNAP, SYSABEND, and SYSUDUMP dumps.

6. Initialize the TIS/XA subsystem as follows:

   - If the TIS/XA subsystem modules reside in the Link Pack Area (LPA) instead of the Modified Link Pack Area (MLPA), you must IPL with the CLPA command.

NOTE

A separate procedure can be written for each subsystem, or a single procedure can be written that accepts parameters.

Create a procedure in SYS1.PROCLIB to invoke the TIS/XA subsystem driver program as follows:

```
//TISSUB PROC
//name EXEC PGM=CSTSSDRV,PARM='ssname,
    command-prefix[,ecsa-code]'
//STEPLIB DD DSN=APF authorized library name
```

name

Description Specifies the name of the step.

CSTSSDRV

Description CSTSSDRV is the driver program.

ssname

Description Specifies the subsystem name.

Default SUP1 (primary TIS/XA subsystem)

Format 1–4 alphanumeric characters (first character must be alphabetical)

command-prefix

Description Identifies the TIS/XA subsystem operator commands.

Default /

Format A single character

Consideration You must use different command prefixes for different subsystems.

ecsa-code

Description *Optional*. Field to specify whether to load the subsystem modules above the line, in ECSA (Extended Common Storage Area), or to load below the line in CSA.

Default None. Omit this parm field to load the subsystem modules in CSA.

Format The single character E, to load the subsystem modules in ECSA.

Consideration If you choose to load the subsystem modules in ECSA, you must link all the applications using the subsystem with the AMODE 31 attribute.

APF *authorized library name*

Description

APF authorized library name is the library specified in Step 2, in the IEAAPF*xx* SYS1.PARMLIB member.

Issue the START procedure as follows:

Issue the START console command, and specify the name of the procedure in SYS1.PROCLIB which invokes the TIS/XA subsystem driver program (START TISSUB).

NOTE

To take the subsystem out of service, use a TIS/XA subsystem operator command. For more information on the TIS/XA subsystem operator commands available, see "Monitoring and controlling the PDM" on page 279.

NOTE

You must specify the subsystem name in the TISSID parameter in a CSIPARM file used by the central PDM. The TISSID is also used by any tasks communicating with the central PDM (Batch Directory Maintenance, CICS, Batch SPECTRA, etc.).

3

Initializing the PDM

Initialization tasks

To initialize the SUPRA PDM, you must perform several tasks:

- Cold start the PDM
- Operate the PDM boot-driven or Directory-driven
- Warm start the PDM
- Select initialization parameters
- Code initialization parameters
- Create the CSIPARM file
- Code the Buffer Cache input file
- Define tasks, threads, and interfaces

Each of these tasks is discussed in the sections that follow.

Cold starting the PDM

If you are cold starting the PDM, the following must be set up before PDM initialization:

- **TIS/XA subsystem (OS/390).** If you are planning to use the TIS/XA subsystem, it must be up before you initialize the PDM. See "Installing TIS/XA subsystem (OS/390)" on page 53 for subsystem installation requirements.

- **Bootstrap modules.** The bootstrap modules describe the Directory and task logging environments. For information on modifying the bootstrap environment, see "Defining the operational environment" on page 115.

- **Directory and PDM files.** Directory and PDM files must be defined to your Directory. See "Defining the operational environment" on page 115 for information on adding files.

- **Task Log File (TLF).** You must format the TLF before running a new release of SUPRA. Your installation tape includes a TLF tailored for you. Cincom suggests you use it without change. For information on adding a Task Log file using Directory Maintenance, see "Defining the operational environment" on page 115.

- **System Log File (SLF).** Specify your system logging options in your Environment Description, and you specify your Environment Description by name in your CSIPARM file. For information on adding a SLF using Directory Maintenance, see "Defining the SUPRA Server database" on page 103.

- **Defining initialization parameters.** Initialization parameters are defined in the CSIPARM file. The PDM reads the CSIPARM file at initialization to determine your operating environment, so you must define the CSIPARM file before you run the PDM. See "Selecting initialization parameters" on page 68, "Coding initialization parameters" on page 73, and "Example CSIPARM input" on page 91 for information on defining initialization parameters.

Operating the PDM boot-driven or Directory-driven

The PDM can be initialized boot-driven or Directory-driven. The PDM distinguishes Directory-driven mode and boot-driven mode at initialization time through the DIRECTORY and REALM parameters in the CSIPARM file. You must always supply the DIRECTORY parameter. To run Directory-driven, you must also supply the REALM parameter; to run boot-driven, you omit the REALM parameter.

Initializing the PDM using the Directory is the norm; however, sometimes it may be necessary to run boot-driven. You must run boot-driven to run a utility that affects the Directory files (including UNLOAD, RELOAD, and FORMAT).

If SUPRA Server does not initialize properly due to memory constraints, or because your environment was inappropriately defined, you can go back in and come up boot-driven and correct the information on the Directory.

With the DIRECTORY parameter, you name the bootstrap schema and the bootstrap Environment Description, which describe the Directory files used during initialization and while running. They also describe the TLF (if any) and the task logging option used in warm start recovery (if applicable) and while running. When running boot-driven, the bootstrap schema and bootstrap Environment Description are the Active Schema and Environment Description. The Modify Schema input consists of batch input and an existing schema. The input makes changes to the physical information about primary files and related files. It makes changes to no more than one SLF, no more than one TLF, and no more than one Statistics File.

When running Directory-driven, you supply the REALM parameter at initialization. With this parameter, you name the user schema and the user Environment Description. These are the active schema and Environment Description and describe the user database files, views, users, and so on, used while running. In the boot-driven mode, you have access to the Directory files and can modify anything in them, (you can add, delete, or change the definitions of user schemas, user Environment Descriptions, files, views, users, etc., using Directory Maintenance). You can also modify all of this when running Directory-driven, but in the boot-driven mode you cannot use any of these entities defined on the Directory. The boot environment, specified on the release tape, consists of:

- **Bootstrap schema module (CSTASCHM).** Contains the definition of the Directory files and the TLF.
- **Bootstrap Environment Description modules.** Contain execution parameters (the same information which is in a Directory Environment Description).
- **Bootstrap validation module.** Checks that the combination of a schema and Environment Description is consistent.

You can tailor any of these modules to your needs. See “Defining the operational environment” on page 115 for information on defining your Directory to the PDM.

Warm starting the PDM

Warm starting the PDM with Task Level Recovery (TLR) is almost completely automatic:

- Resubmit the job that originally brought up the PDM in the first place. You must not have changed the Environment Description, the schema, the PDM name, or the job name (if the PDM name defaults to the job name).
- A central mode PDM under OS/390 cannot be warm started in another step of the same job that brought it up before; it must be warm started as a separate job.

You can also warm connect to the PDM after being forcibly disconnected for any reason with any TP monitor whose tasks can sign on to a PDM with TLR. The PDM TLF keeps track of all signed-on tasks and treats them as restartable, if RESTART=NORMAL was specified in the CSIPARM file for their interface.

Refer to the *SUPRA Server PDM Logging and Recovery Guide (OS/390 & VSE)*, P26-2223, for more information about warm starting and special cases where the standard recovery procedures are inapplicable or insufficient.

Selecting initialization parameters

During initialization, each PDM and PDM interface reads its CSIPARM file to determine the environment. The CSIPARM file provides a means of controlling your system. You can have one or more CSIPARM files, depending on the operating mode and the type of environment (test or production) you are using, but each job uses only one CSIPARM file.

The initialization parameters you define in the CSIPARM file determine the following:

- User schema and Environment Description
- Bootstrap schema and Environment Description
- The PDM and interface to use

NOTE

Cincom recommends that you have one CSIPARM file for each different single-task PDM, one for each attached mode PDM, one for each central mode PDM (which its interfaces share), and up to two for each attached central mode PDM (one for the PDM and its attaching interface, and one for its central interfaces).

The following three tables summarize the CSIPARM file parameters for each operating system and operating mode. The footnotes refer to the considerations on the page following each table. Sample CSIPARM members are supplied in the CSIPARM library on the installation tape.

CSIPARM file parameters for OS/390

CSIPARM parameter	Single-task mode (STST) with single-task PDM	Central mode interface (STMT, CSMT, or MTMT)	Central mode PDM	Attached mode
ATTACH	Ignored	Not Allowed[5]	Ignored	Required[1], [8]
DBM	Ignored[6]	Required[7]	Optional	Optional[6]
DEBUG	Optional	Ignored	Optional	Optional
DIRECTORY	Optional	Ignored	Optional	Optional
INTERFACE	Optional	Optional[4]	Ignored	Optional[3]
MACHID	Ignored	Ignored	Ignored	Ignored
MAXIO	Ignored	Ignored	Ignored	Ignored
MAXARG	Ignored	Ignored	Ignored	Ignored
MAXPACKET	Ignored	Ignored	Ignored	Ignored
PAGESIZE	Optional	Ignored	Optional	Optional
PATH	Ignored	Ignored	Ignored	Ignored
PRIMAP	Optional	Ignored	Optional	Optional
RDBUFFER	Ignored	Optional	Optional	Optional
RDTHRSH1	Ignored	Optional	Optional	Optional
RDTHRSH2	Ignored	Optional	Optional	Optional
READAHEAD	Ignored	Optional	Optional	Optional
REALM	Optional	Ignored	Optional	Optional
RELMAP	Optional	Ignored	Optional	Optional
RESTART	Optional	Optional	Ignored	Ignored
TISSID	Ignored	Required	Required	Optional[2]
TRACESIZE	Optional	Ignored	Optional	Optional
XAMEM	Optional	Optional	Optional	Optional
Recommended CSIPARM Parameters	DIRECTORY, REALM	DBM, TISSID	DBM, TISSID, DIRECTORY, REALM	DBM, TISSID[2], DIRECTORY, REALM, ATTACH[1]

Notes to the preceding table:

1. If you use the Attacher program, do not code the ATTACH parameter. In addition, the ATTACH parameter is ignored by an STMT interface.

2. If you want to execute in attached central mode, the TISSID parameter is required.

3. In an attached operating mode, if you code the INTERFACE parameter, an error occurs in the Attacher program if the application task fails.

4. In a central operating mode, if you code the INTERFACE parameter and all interfaces use the same CSIPARM file, they have the same interface name. Therefore, the second interface to attempt to connect results in an error.

5. In a central operating mode, do not code the ATTACH parameter. Coding the ATTACH parameter instructs the task to attach a PDM with which to communicate. The application task will then be unable to communicate with a PDM in a different address space.

   In an attached central operating mode, the ATTACH parameter for central-mode interfaces is ignored. This assumes the attached central PDM is already initialized.

6. In single-task or attached operating modes, do not code the DBM parameter because PDMs with the same name are single-threaded.

7. In a central operating mode, the DBM parameter is required for the interfaces to communicate with the PDM. Use the DBM parameter to control which PDM all of the application tasks communicate with. You may have two CSIPARM files: one specifying DBM=TESTDBM and the other specifying DBM=PRODDBM. When you code your JCL stream, you could code one of the following depending on whether you are running in a test or a production environment:

   ```
   //CSIPARM DD DSN=A.B(TEST)
   ```

   or

   ```
   //CSIPARM DD DSN=A.B(PROD)
   ```

8. The ATTACH parameter is only valid for a CICS interface (CSMT) CSIPARM file. Using the ATTACH parameter in an STMT or an MTMT interface will cause unpredictable results.

CSIPARM file parameters for VSE/ESA

CSIPARM parameter	Single-task mode (STST) with single-task PDM	Central mode interface (STMT, CSMT, or MTMT)	Central mode PDM	Attached mode
ATTACH	Ignored	Not Allowed[4]	Ignored	Required
DBM	Ignored[6]	Required[7]	Optional	Optional[6]
DEBUG	Optional	Ignored	Optional	Optional
DIRECTORY	Optional	Ignored	Optional	Optional
INTERFACE	Optional	Optional[5]	Ignored	Optional[3]
MACHID	Ignored	Ignored	Ignored	Ignored
MAXIO	Ignored	Optional[2]	Optional	Optional
MAXARG	Ignored	Optional[2]	Optional	Optional
MAXPACKET	Ignored	Optional[2]	Optional	Optional
PAGESIZE	Optional	Ignored	Optional	Optional
PATH[3]	Ignored	Optional[2]	Optional	Optional
PRIMAP	Optional	Ignored	Optional	Optional
RDBUFFER	Ignored	Optional	Optional	Optional
RDTHRSH1	Ignored	Optional	Optional	Optional
RDTHRSH2	Ignored	Optional	Optional	Optional
READAHEAD	Ignored	Optional	Optional	Optional
REALM	Optional	Ignored	Optional	Optional
RELMAP	Optional	Ignored	Optional	Optional
RESTART	Optional	Optional	Ignored	Ignored
TISSID	Ignored	Ignored	Ignored	Ignored
TRACESIZE	Optional	Ignored	Optional	Optional
Recommended CSIPARM Parameters	DIRECTORY, REALM	DBM PATH[3]	DBM, DIRECTORY, REALM PATH[3]	DBM, DIRECTORY, REALM, ATTACH

Notes to the preceding table:

1. STMT attached is not available.

2. If the application is in a static partition, in the same address space as the PDM, this parameter is ignored.

3. If the application is in a different address space from the PDM, the PATH parameter is required.

4. In the central operating mode, if you code the ATTACH parameter, a CICS application task cannot communicate with a PDM in a different address space from its own (you have instructed the interface to attach a PDM with which it will communicate).

5. In a central operating mode, if you code the INTERFACE parameter and all interfaces use the same CSIPARM file, they have the same interface name. Therefore, the second interface to attempt to connect results in an error.

6. In single-task or attached operating modes, do not code the DBM parameter because PDMs with the same name are single-threaded.

7. In a central operating mode, the DBM parameter is required for the interfaces to communicate with the PDM. Use the DBM parameter to control which PDM the application tasks communicate with. You may have two CSIPARM files: one specifying DBM=TESTDBM and the other specifying DBM=PRODDBM. Then, when you code your JCL stream, you could code one of the following depending on whether you are running in a test or a production environment:

   ```
   // DLBL CSIPARM,'CSIPARM.TEST'
   // EXTENT SYSWK2
   ```

   or

   ```
   // DLBL CSIPARM,'CSIPARM.PROD'
   // EXTENT SYSWK2
   ```

Coding initialization parameters

Once you have selected the initialization parameters for your CSIPARM file, you need to code the file accordingly. You can code the CSIPARM parameters in any order (except for END.). Code the parameters in records of 80 bytes. The PDM reads only the first 72 bytes and ignores the last 8. You can code more than one parameter per record, but a parameter cannot be split between lines and the parameters must be separated with commas. END. must be the last parameter; anything following END. is ignored.

You can insert comments by putting an asterisk in column one. Putting an asterisk (*) in the first column causes the PDM to ignore the entire 80 bytes. You can also insert comments on the same line with a parameter if you put a space after the last parameter. The following three examples show various ways to code comments and parameters:

```
*comment
PAGESIZE=2KØcomment
ØØØPRIMAP=YESØcomment
```

In the third example, the PDM does not consider the spaces at the beginning of the line an indication that a comment follows. The space must occur after a parameter specification.

[ATTACH=*dbm-name*]

[DBM=*dbm-id*]

$$\left[\mathbf{DEBUG} = \left\{\frac{\mathbf{NO}}{\mathbf{YES}}\right\}\right]$$

$$\left[\mathbf{DIRECTORY} = \left\{\begin{matrix}(\mathbf{SCHEMA} = \boldsymbol{schema} - \boldsymbol{id}, \mathbf{ENVDESC} = \boldsymbol{env} = \boldsymbol{desc} - \boldsymbol{id}) \\ (\mathbf{ENVDESC} = \boldsymbol{env} = \boldsymbol{desc} - \boldsymbol{id}, \mathbf{SCHEMA} = \boldsymbol{schema} - \boldsymbol{id})\end{matrix}\right\},\right]$$

[INTERFACE=*interface-id*]

$$\left[\mathbf{MAXARG} = \left\{\frac{\mathbf{256}}{\boldsymbol{NNNNN}}\right\},\right]$$

$$\left[\mathbf{MAXPACKET} = \left\{\frac{\boldsymbol{nnnnn}}{\mathbf{15360}}\right\},\right]$$

$$\left[\mathbf{MAXIO} = \left\{\frac{\mathbf{2048}}{\boldsymbol{nnnnn}}\right\},\right]$$

$$\left[\mathbf{PAGESIZE} = \left\{\begin{matrix}\mathbf{8}\boldsymbol{K} \\ \mathbf{4}\boldsymbol{K} \\ \mathbf{2}\boldsymbol{K} \\ \mathbf{1}\boldsymbol{K} \\ \mathbf{512} \\ \mathbf{256} \\ \\ \end{matrix}\right\},\right]$$

$$\left[\mathbf{PATH} = (\mathbf{XPCC} = \left\{\frac{\mathbf{NO}}{\mathbf{YES}}\right\}),\right]$$

$$\left[\mathbf{PRIMAP} = \left\{\frac{\mathbf{NO}}{\mathbf{YES}}\right\},\right]$$

$$\left[\mathbf{RDBUFFER} = \left\{\frac{\mathbf{8000}}{\boldsymbol{nnnn}}\right\},\right]$$

$$\left[\mathbf{RDTHRSH1} = \left\{\frac{\mathbf{10}}{\boldsymbol{nnnn}}\right\},\right]$$

$$\left[\mathbf{RDTHRSH2} = \left\{\frac{\mathbf{100}}{\boldsymbol{nnnn}}\right\},\right]$$

$$\left[\mathbf{READAHEAD} = \left\{\frac{\mathbf{YES}}{\mathbf{NO}}\right\},\right]$$

$$\left[\mathbf{REALM} = \left\{\begin{matrix}(\mathbf{SCHEMA} = \boldsymbol{schema} - \boldsymbol{id}, \mathbf{ENVDESC} = \boldsymbol{env} = \boldsymbol{desc} - \boldsymbol{id}) \\ (\mathbf{ENVDESC} = \boldsymbol{env} = \boldsymbol{desc} - \boldsymbol{id}, \mathbf{SCHEMA} = \boldsymbol{schema} - \boldsymbol{id})\end{matrix}\right\},\right]$$

$$\left[\mathbf{RELMAP} = \left\{\begin{matrix}\mathbf{NO} \\ \underline{\mathbf{YES}}\end{matrix}\right\},\right]$$

$$\left[\mathbf{RESTART} = \left\{\begin{matrix}\mathbf{NONE} \\ \underline{\mathbf{NORMAL}}\end{matrix}\right\},\right]$$

[TISSID=*subsystem-id*]

$$\left[\mathbf{TRACESIZE} = \left\{\begin{matrix}\mathbf{1} \\ \mathbf{2} \\ \mathbf{4} \\ \underline{\mathbf{8}} \\ \mathbf{n}\end{matrix}\right\},\right]$$

[XAMEM=*nnnn*M,]

END.

ATTACH=*dbm-name*

Description *Optional*. Indicates to the interface that it should attach the PDM, and identifies the load module name of the PDM to be attached.

Considerations

- Code this parameter only if you want to run in an attached operating mode or an attached central operating mode with CICS.
- Code this parameter as the name of the PDM load module that is attached. During installation, the PDM load module is link edited with the name DBMS. If you change the load module name of the PDM, use that name instead of DBMS.

DBM=*dbm-id*

Description *Optional*. Specifies the name of the PDM.

Default The name of the job used to bring up the component (the PDM or the PDM interface).

Format 1–8 characters. The first character must be alphabetic, $, #, or @; the remaining characters must be alphanumeric, $, #, or @.

Consideration OS/390 or VSE Always code this parameter in a central or attached central operating mode so that the interfaces can communicate with the PDM.

[DEBUG = {NO | YES}]

Description *Optional.* Specifies whether to print additional console messages used for debugging if an error occurs.

Default No

Consideration Using this option may cause the PDM to send an excessive number of messages to the console. The *SUPRA Server PDM Messages and Codes Reference Manual (RDM/PDM Support for OS/390 & VSE)*, P26-0126, lists any messages this option produces.

$$\left[\mathbf{DIRECTORY} = \left\{\begin{matrix}(\mathbf{SCHEMA} = \boldsymbol{schema-id}, \mathbf{ENVDESC} = \boldsymbol{env=desc-id}) \\ (\mathbf{ENVDESC} = \boldsymbol{env=desc-id}, \mathbf{SCHEMA} = \boldsymbol{schema-id})\end{matrix}\right\},\right]$$

Restriction This parameter applies only to the PDM; it is ignored by the interface.

Description *Optional.* Names the bootstrap schema and bootstrap Environment Description that describe the Directory.

Default If you omit this parameter, the PDM operates as if the REALM parameter of the CSIPARM file defined the bootstrap schema and the bootstrap Environment Description.

Format

env-desc-id	Existing bootstrap Environment Description name.
schema-id	Existing bootstrap schema name.

Considerations

- The PDM cannot initialize if you omit both the REALM and DIRECTORY parameters on the CSIPARM file and the REALM parameter on the SINON command.
- Code the DIRECTORY parameter specifying a bootstrap schema and a bootstrap Environment Description when running the DBA utilities against user files and when running Recover, Restore, or the Version 2 utilities against Directory files. Do not code the DIRECTORY parameter when running the other DBA utilities against Directory files. For more information, see the REALM parameter. For more information about the utilities' CSIPARM file requirements, refer to the *SUPRA Server PDM DBA Utilities User's Guide (OS/390 & VSE)*, P26-6260.

INTERFACE=*interface-id*

Restriction This parameter applies only to the interface. It is ignored by the PDM.

Description *Optional*. Specifies an interface name so the PDM can identify the interface.

Default Job name of the application task, (CICS).

Format 1–8 characters. The first character must be alphabetic, $, #, or @; the remaining characters must be alphanumeric, $, #, or @.

Considerations

- If you have multiple interfaces accessing the same PDM, you would normally have only one CSIPARM file all interfaces can read. If you code this parameter in a central operating mode, all interfaces using this CSIPARM file have the same interface name. Therefore, the second interface to connect receives a status indicating that it is a duplicate and cannot connect.
- If you want each interface to have its own name (not its job name), create multiple copies of the CSIPARM file. However, this may cause a maintenance problem.

$$\left[\mathbf{MAXARG} = \left\{\frac{\mathbf{256}}{\boldsymbol{NNNNN}}\right\},\right]$$

Restriction Used only for central or attached central mode, and only under VSE/AF 2.1.

Description *Optional.* Specifies the maximum number of bytes reserved for the argument parameter of the FINDX physical DML command.

Default 256

Format 1–5 numeric characters

Considerations

- Specify the number of bytes for the argument if an application or component issues commands that require more than the default. If the argument parameter in the FINDX command requires more than 256 bytes, use the MAXARG parameter to reserve more space.
- Any values specified in the PDM's CSIPARM file for MAXARG, MAXIO, or MAXPACKET override the default values for all interfaces. Any values specified for these parameters in an interface's CSIPARM file override the PDM values for that interface only.
- Your MAXPACKET value must be larger than the sum of your MAXARG and MAXIO values.

$$\left[\mathbf{MAXIO} = \left\{\frac{\mathbf{2048}}{\boldsymbol{nnnnn}}\right\},\right]$$

Restriction This parameter is ignored under OS/390.

Description *Optional.* Specifies the maximum number of bytes reserved for the physical DML command's data area.

Default 2048 bytes

Format 1–5 numeric characters

Considerations

- You should specify the number of bytes for the data area if an application or component issues commands that require more than the default. You may need to enlarge the data area parameter in the following commands: ADD-M, ADDVA, ADDVB, ADDVC, ADDVR, DEL-M, DELVD, RDNXT, READD, READM, READR, READV, READX, RSTAT, WRITD, WRITM, WRITV.
- Any values specified in the PDM's CSIPARM file for MAXARG, MAXIO, or MAXPACKET override the default values for all interfaces. Any values specified for these parameters in an interface's CSIPARM file override the PDM values for that interface only.
- Your MAXPACKET value must be larger than the sum of your MAXARG and MAXIO values.

$$\left[\text{MAXPACKET} = \left\{\frac{\textit{nnnnn}}{\textbf{15360}}\right\},\right]$$

Restriction Ignored under OS/390 and VSE/AF 1.4. Ignored for operating modes other than central or attached central.

Description *Optional.* Specifies the maximum size, in bytes, of a communications packet. A packet contains all of the parameters of a single, physical DML command plus interface control information.

Default 15360 bytes

Format 1–5 numeric characters

Considerations

- You cannot specify a packet size larger than 32K bytes.
- Your MAXPACKET value must be larger than the total length of the parameters of any DML command you issue through this PDM interface. Consequently, your MAXPACKET value must be larger than the sum of your MAXARG and MAXIO values.

  Use the following rough numbers as rules of thumb for choosing a MAXPACKET value: Batch Directory Maintenance needs up to 15,000 bytes; most applications need only about 5,000 bytes. If your MAXPACKET value is too small, you will receive an error status on the request that failed.
- Any values specified in the PDM's CSIPARM file for MAXARG, MAXIO, or MAXPACKET override the default values for all interfaces. Any values specified for these parameters in an interface's CSIPARM file override the PDM values for that interface only.
- VSE/ESA The MAXPACKET value for a CICS interface should be compatible with the CICS memory block size to avoid wasting memory space. In other words, the CICS memory block size (63K for CICS 2.1 divided by the MAXPACKET value should leave a zero or small remainder.
- VSE/ESA The PDM's memory requirement for MAXPACKET is taken from the memory allocation in the Environment Description. For information on modifying the Environment Description, see "Defining the operational environment" on page 115.

  The memory required by a CICS PDM interface is taken from the MEMORY parameter of the PDM connect request (OPER CONNECT MEMORY=) or from the default value in the CSTXOTBL module. For more information on calculating the memory in CICS, refer to the *SUPRA Server PDM CICS Connector Systems Programming Guide (OS/390 & VSE)*, P26-7452.

$$\left[\text{PAGESIZE} = \left\{\begin{matrix} \mathbf{8K} \\ \mathbf{4K} \\ \mathbf{2K} \\ \mathbf{1K} \\ \mathbf{512} \\ \mathbf{256} \end{matrix}\right\},\right]$$

Description *Optional.* Indicates to the PDM the size of the pages into which the general memory pool is divided. Memory is allocated to tasks using this page size. The PDM can subdivide these pages as required for a task.

Default Operating system page size

VSE 2K

OS/390 4K

Consideration Allow this parameter to default to the operating system page size. Changing this parameter could worsen PDM memory fragmentation. Modify this parameter only on advice from your Cincom Support Center.

$$\left[\text{PATH} = (\text{XPCC} = \left\{\frac{\text{NO}}{\text{YES}}\right\}),\right]$$

Description *Optional.* Indicates whether you want to communicate across address spaces in VSE/ESA. The PDM uses Cross Partition Communications Control (XPCC) to communicate across address spaces. The PDM interface selects the most efficient method of communication available (an application does not use XPCC if the application is in the same address space as the PDM, unless operating in a dynamic partition).

Default No

Considerations

- Valid for Cross Partition Communications Control (XPCC) only.
- If you code XPCC=YES, you must also consider whether the defaults for the MAXIO, MAXARG, and MAXPACKET parameters are the size you need.

$$\left[\textbf{PRIMAP} = \left\{\frac{\textbf{NO}}{\textbf{YES}}\right\},\right]$$

Restriction This parameter applies only to the PDM; it is ignored by the interface in SUPRA Server 2.7.

Description *Optional.* Specifies whether to use a file experience table (a bit map) to find free space on primary files.

Default No

Considerations

- Use the File experience table if most of your primary files are very dense.
- If you use the File Experience table, the PDM allocates a small amount of additional memory when you open a primary file.
- Using the PRIMAP option may cause a false FULL condition if used with the RESET command. RESET backs out records that were added to a file, but this action is not reflected in the bit map. For more information about the RESET command, refer to the *SUPRA Server PDM DML Programming Guide (OS/390 & VSE)*, P26-4340.

  If you try to add records (ADD-M) to a file from which you previously backed out records, the bit map may show a FULL condition. To add records to that file, you must refresh the bit map by closing the file, committing the close, and then reopening the file. Then you can add records to the file.

$$\left[\textbf{RDBUFFER} = \left\{\frac{\textbf{8000}}{\textbf{\textit{nnnn}}}\right\},\right]$$

Restriction This parameter applies only to the PDM and its STMT interfaces; it is ignored by the other interfaces. It is applicable for SUPRA Server 2.7 and above.

Description *Optional.* Specifies the size of each buffer in the Read Ahead buffering facility for increased performance in batch jobs with repetitive read DMLs. Three buffers of the specified size will be generated per STMT interface.

Default 8000

Consideration Use the READAHEAD CSIPARM File parameters to tune the buffering facility for your applications. Performance can be increased or decreased based on how you tune these values.

$$\left[\textbf{RDTHRSH1} = \left\{\frac{\textbf{10}}{\textbf{\textit{nnnn}}}\right\},\right]$$

Restriction This parameter applies only to the PDM and its STMT interfaces; it is ignored by the other interfaces. It is applicable for SUPRA Server 2.7 and above.

Description *Optional.* Specifies at which repetition of a read DML to move from 1 record at a time to multiple records returned in stage two.

Default 10

Consideration Use the READAHEAD CSIPARM File parameters to tune the buffering facility for your applications. Performance can be increased or decreased based on how you tune these values.

$$\left[\textbf{RDTHRSH2} = \left\{\frac{\textbf{100}}{\textbf{\textit{nnnn}}}\right\},\right]$$

Restriction This parameter applies only to the PDM and its STMT interfaces; it is ignored by the other interfaces. It is applicable for SUPRA Server 2.7 and above.

Description *Optional.* Specifies at which repetition of a read DML to enter stage three of the buffering facility. The number of records returned in the buffer will not be limited to a single physical I/O, but will fill the buffer with records retrieved by multiple I/Os.

Default 100

Consideration Use the READAHEAD CSIPARM File parameters to tune the buffering facility for your applications. Performance can be increased or decreased based on how you tune these values.

$$\left[\textbf{READAHEAD} = \left\{\frac{\textbf{YES}}{\textbf{NO}}\right\},\right]$$

Restriction This parameter applies only to the PDM and its STMT interfaces; it is ignored by the other interfaces. It is applicable for SUPRA Server 2.7 and above.

Description *Optional.* Specifies whether the user does not want to use the buffering facility to increase performance in applications where a multitude of read requests are issued using the same DML function and the same parameters.

Default YES

Consideration Use the READAHEAD CSIPARM File parameters to tune the buffering facility for your applications. Performance can be increased or decreased based on how you tune these values.

$$\left[\text{REALM} = \left\{\begin{matrix}(\text{SCHEMA} = \textit{schema} - \textit{id}, \text{ENVDESC} = \textit{env} = \textit{desc} - \textit{id}) \\ (\text{ENVDESC} = \textit{env} = \textit{desc} - \textit{id}, \text{SCHEMA} = \textit{schema} - \textit{id})\end{matrix}\right\},\right]$$

Restriction This parameter applies only to the PDM; it is ignored by the interface.

Description *Optional.* Names the user schema and Environment Description defining the user's database as they exist on the Directory.

Default If you omit this parameter, the system operates with no user schema or Environment Description. All operation is done with the bootstrap Directory schema and Environment Description named in the DIRECTORY parameter of the CSIPARM file.

Format *env-desc-id* Existing bootstrap Environment Description name.

schema-id Existing bootstrap schema name.

Considerations

- REALM is normally required, but it may be omitted when you first install SUPRA Server and the user database is not defined on the Directory, or when you specify it on the DML SINON command (single-task mode only).
- If you run the DBA utilities, the utility input values for schema and Environment Description override the REALM value in the CSIPARM file.
- When running DBA utilities, code the REALM parameter only for the Expand Function and the Version 2 utilities, when running against:
  - User files—Specify a user schema and Environment Description.
  - Directory files—Specify a boot schema and Environment Description.

  For additional DBA utilities information, see the DIRECTORY parameter. For more information about the utilities' CSIPARM file requirements, refer to the *SUPRA Server PDM DBA Utilities User's Guide (OS/390 & VSE)*, P26-6260.

Warning: When running the Recover or Restore Utility, *DO NOT CODE* the REALM parameter.

- In single-task mode, the PDM cannot initialize if you omit both the REALM and DIRECTORY parameters on the CSIPARM file and the REALM parameter on the SINON command.
- In single-task mode, you may code the REALM on a DML SINON command. However, if the application task ever runs with a multitask PDM, the PDM generates an error status.
- REALM must be omitted to run the PDM boot-driven.

$$\left[\mathbf{RELMAP} = \left\{\begin{matrix}\mathbf{NO}\\ \underline{\mathbf{YES}}\end{matrix}\right\},\right]$$

Restriction This parameter applies only to the PDM. It is ignored by the interface.

Description *Optional.* Specifies whether to use a file experience table (a bit map) to find free space on related files.

Default Yes

Considerations

- Use the file experience table if the majority of your related files are very dense. By letting the related file's bit-mapped option default to YES, the PDM assumes that your related files are dense. If you use the file experience table, the PDM allocates a small amount of additional memory for each related file you open.
- Do not use the file experience table if most of your related files are not very dense or when adding records to an empty or almost empty related file.
- Using the RELMAP option may cause a false FULL condition if used with the RESET command. RESET backs out records that were added to a file, but this action is not reflected on the bit map. For more information about the RESET command, refer to the *SUPRA Server PDM DML Programming Guide (OS/390 & VSE)*, P26-4340.

  If you try to add records (ADDVx) to a file from which you previously backed out records, the bit map may show a FULL condition. To add records to that file, refresh the bit map by closing the file, committing the close, then reopening the file.

$$\left[\text{RESTART} = \left\{\begin{matrix}\text{NONE} \\ \underline{\text{NORMAL}}\end{matrix}\right\},\right]$$

Restrictions This parameter applies only to the interface. It is ignored by the PDM.

This parameter is ignored in a non-TLR environment.

Description *Optional.* Points to field indicating whether you want the interface's tasks to be restartable or not restartable.

Default NORMAL

Options

NORMAL The task is restartable. The PDM backs out the task's updates to the last commit point. If any files are open for exclusive update (EUPD) by the task, the PDM closes and unlocks them. When the task signs on again, it receives an RSTR status and can retrieve its committed data from the Task Log file.

NONE The task is not restartable. The PDM backs out its updates to the last commit point. If any files are open for exclusive update by the task, the PDM closes and unlocks them. The task information is removed from PDM memory.

Considerations

- If you code the RESTART mode in the SINON command, it overrides what you code here in the CSIPARM file.
- You cannot code RESTART=NONE for STST tasks. Either code RESTART=NORMAL or omit this parameter.

TISSID=*subsystem-id*

Restriction Ignored except under OS/390.

Description *Optional* for OS/390. Specifies the name of the TIS/XA subsystem to be used if you have a central PDM communicating with other address spaces. Indicates that the PDM should operate as a central PDM.

Format The name of an existing, installed PDM subsystem

Considerations

- The subsystem must be valid for your release of SUPRA. If the named subsystem is nonexistent or invalid, the PDM and interfaces cannot communicate across address spaces, and they may terminate abnormally. For information on how to install a TIS/XA subsystem, see “Installing TIS/XA subsystem (OS/390)” on page 53.
- The TISSID parameter is required under OS/390 for a central PDM, an attached central PDM, or an interface that is to communicate with such a PDM. In order to communicate, such a PDM and such an interface must specify the same subsystem name with the TISSID parameter.
- Do not code the TISSID parameter for an attached PDM. If coded, the TISSID parameter indicates an attached central PDM.
- In OS/390, if you specify TISSID and you are using one CSIPARM file for all interfaces, you must also specify a DBM parameter. OS/390 will not run two jobs with the same name simultaneously, so you cannot use the default (jobname) in the DBM parameter.

$$\left[\ \mathbf{TRACESIZE} = \left\{\begin{matrix}\mathbf{1}\\ \mathbf{2}\\ \mathbf{4}\\ \underline{\mathbf{8}}\\ \mathbf{n}\end{matrix}\right\},\ \right]$$

Restriction This parameter applies only to the PDM. It is ignored by the interface.

Description *Optional.* Specifies the size of the PDM Internal Trace Table in kilobytes (1024K). The Internal Trace Table is for use by Cincom Support Services.

Default 8

Consideration Cincom recommends that you allow the parameter to default to 8K. Override the trace size only in a testing environment or from Cincom Support Services recommendation.

XAMEM=*nnnn*M

Restriction Valid in SUPRA Server Release 2.4.

Description *Optional.* Under OS/390, specifies the amount of memory to use above the 16 MB line.

Format The amount of memory (*nnnn*) in megabytes (MB)

Consideration If you have not purchased the option to run with memory above the 16 MB line, this entry is ignored.

END.

Description *Required*. Terminates the CSIPARM file parameter list.

Example CSIPARM input

CSIPARM sample members are included on the SUPRA Server product library provided on your installation tape. The following provides descriptions and sample input for commonly used CSIPARM members.

- For running a Directory-driven attached central PDM with task logging against the Burrys database:

```
DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTATLOG),
REALM=(SCHEMA=BURRYSCH,ENVDESC=BURRYENV),
ATTACH=DBMS,TISSID=SUP1,END.
```

- For running a Directory-driven attached PDM with task logging against the Burrys database:

```
DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTATLOG),
REALM=(SCHEMA=BURRYSCH,ENVDESC=BURRYENV),
ATTACH=DBMS,END.
```

- For running against a central PDM:

```
TISSID=SUP1,DBM=DEMOTIS,END.
```

- For running DBA Utilities when all of the environment information is provided in the utility's input:

```
END.
```

- For running a non-Directory-driven, noncentral PDM without opening the Directory files at PDM initialization time:

```
DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTANONE),END.
```

- For running a non-Directory-driven PDM opening the Directory files for read-only access at PDM initialization time:

```
DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTAREAD),END.
```

- For running a Directory-driven attached PDM with task logging and with extra debugging error messages:

```
ATTACH=DBMS,
DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTATLOG),
REALM=(SCHEMA=CSISCH20,ENVDESC=CSIENVRG),
DEBUG=YES,END.
```

- For running a non-Directory-driven, noncentral PDM with the Directory files opened for update:

```
DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTASUPD),END.
```

- For running a Directory-driven central PDM with task logging against the Burrys database:

```
DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTATLOG),
REALM=(SCHEMA=BURRYSCH,ENVDESC=BURRYENV),
TISSID=SUP1,DBM=DEMOTIS,END.
```

- For running a Directory-driven, noncentral PDM with no task logging and with Directory files open for read only:

```
DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTAREAD),
REALM=(SCHEMA=CSISCH20,ENVDESC=CSIENVRN),END.
```

- For migrating the SPECTRA personal file data set LV00:

```
DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTASUPD),
REALM=(SCHEMA=BURRYSCH,ENVDESC=BURRYNLG),END.
```

- For running a non-Directory-driven, noncentral PDM with no task logging, nor Directory files open for update:

```
DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTASUPD),END.
```

- For running a non-Directory-driven attached PDM:

```
DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTAOLDM),
ATTACH=DBMS,END.
```

- For running a Directory-driven, noncentral PDM with no task logging, nor Directory files open for read only against the Burrys database:

```
DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTAREAD),
REALM=(SCHEMA=BURRYSCH,ENVDESC=CSIENVRN),END.
```

- For running a Directory-driven, noncentral PDM with no task logging, nor Directory files open for read only against the Burrys database:

```
DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTANONE),
REALM=(SCHEMA=BURRYSCH,ENVDESC=BURRYENN),END.
```

- For running a Directory-driven, noncentral PDM with task logging against the Burrys database:

```
DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTATLOG),
REALM=(SCHEMA=BURRYSCH,ENVDESC=BURRYENV),END.
```

Creating a CSIPARM file

You can create a CSIPARM file with an online editor like TSO and XEDIT or an IBM Utility like IEBGENER. Examples of creating the file in different operating systems follow. Sample CSIPARM members are distributed in the SUPRA Server CSIPARM library and are used by sample JCL streams. An index of sample members, TXP$INDX, is provided giving a cross reference to JCL samples.

Creating the file under OS/390

The following examples show two ways to create the file under OS/390:

- Execute the IBM utility IEBGENER. For example:

  ```
  // EXEC PGM=IEBGENER
  .
  .
  .
  .
  //SYSUT2 DD DSN=A.B.C(PARM1),DISP=OLD
  //SYSUT1 DD *
  ```

  The data set name A.B.C(PARM1) names the member of the partitioned data set that contains the CSIPARM file.

- Use an editor like TSO ISPF to create the file as a member of a partitioned data set.

  In a central operating mode, the disposition of the CSIPARM file must be SHR if the PDM and the interfaces all access it at the same time. For example:

  ```
  //CSIPARM  DD  DSN=A.B.C(PARM1),DISP=SHR
  ```

Creating the file under VSE

The CSIPARM file is sequential. To create a sequential file on disk, you can use any of several system utilities. When you create sequential files, be certain that the number of CSIPARM files does not become unmanageable. When using temporary CSIPARM files (zero retention date for the test environment), be careful to prevent more than one address space from updating the contents of a common CSIPARM file.

The following are examples of using IBM system utilities and program products to create a CSIPARM file:

- DITTO

```
// JOB DITTO
// DLBL CSIPARM,'CSI.PARM.ONE'
// EXTENT SYS020,,,,xxxxx,1
// ASSGN SYS020,DISK,VOL=SYSWK2,SHR1
// EXEC DITTO,SIZE=128K
$$DITTO CSQ FILEOUT=CSIPARM
DIRECTORY=(SCHEMA=.....),
REALM=(SCHEMA=.....),END.
/*
$$DITTO SPR FILEIN=CSIPARM
$$DITTO EOJ
/*
/&
```

- OBJMAINT

```
// JOB OBJMNT
// DLBL UOUT,'CSI.PARM.ONE'
// EXTENT SYS020,,,,xxxxx,1
// ASSGN SYS020,DISK,VOL=SYSWK2,SHR
// ASSGN SYS004,SYSIPT
// ASSGN SYS005,SYS020
// EXEC OBJMAINT,SIZE=AUTO
./ COPY
DIRECTORY=(SCHEMA=.....),
REALM=(SCHEMA=.....),END.
/*
/&
```

Coding the Buffer Cache Facility input file

If you are using the optional Buffer Cache Facility, the system first determines the user environment and then loads and initializes the facility. The facility reads an input file (PDMXIN) containing control information and uses the information to allocate storage for the cache above the 16 MB line.

The facility uses 80-byte statements to define the mode of execution, the allocation of the cache, and the files that will use the cache.

Use the following statements to perform those functions:

- **Mode statement.** The Mode statement defines the mode of execution. This statement must be the first statement in the PDMXIN input file.

  Positions 1–4 SYST

  Description *Required*. Identifies the statement as the Mode Statement.

  Positions 10–13 mode

  Description *Required*. Identifies the mode in which to operate.

  Options

TEST	The cache is maintained, but data is not used to satisfy reads. Data is checked at each I/O for correctness.
NONE	Does no processing.
PROD	Process from the cache.

- **Cache statement.** The Cache statement defines the size and structure of the cache. Each statement defines a cache pool of a particular blocksize. If the size of the pool is not a multiple of the blocksize, the pool is rounded down to the nearest multiple of a blocksize. The Cache Statement must follow the Mode Statement in the PDMXIN input file.

  Positions 1–4 BUFF

  Description *Required*. Identifies the statement as the Cache Statement.

  Positions 10–13 cache pool name

  Description *Required*. Identifies the cache pool.

  Format 4 alphanumeric characters

  Positions 15–23 cache pool size

  Description *Required*. Identifies the cache pool size in bytes.

  Format 9 numeric characters

  Positions 25–32 cache pool block size

  Description *Required*. Identifies the cache pool block size in bytes.

  Format 8 numeric characters

- **File statement.** The File statement defines the files cached in each pool. The block size of each file assigned must be less than or equal to the block size of the pool. You may cache only BDAM files. This statement must follow the Cache Statement in the PDMXIN input file.

  Positions 1–4 FILE

  Description *Required*. Identifies the statement as the File Statement.

  Positions 10–13 file name

  Description *Required*. Identifies a file to be cached.

  Format 4 alphanumeric characters

  Positions 15–18 cache name

  Description *Required*. Identifies a cache.

  Format 4 alphanumeric characters

  Consideration Only BDAM files (not VSAM or BSAM files) can be cached.

- **Comment statement.** The Comment statement contains clarifying information unnecessary to processing. You may place it anywhere in the input. You may also add comments to Mode, Cache, and File statements. Any data not in the defined positions is considered a comment.

  Positions 1–80 comment

  Description *Optional.* Adds clarifying information.

  Format 1–80 alphanumeric characters

Following is an example of the input to the Buffer Cache Facility. This example is distributed member CSTK6INP in the installed macro library.

```
SYST     PROD                            For production
*
*Define cache pools
*
BUFF     BUF1 000043000 00004300         Directory cache
BUFF     BUF2 000081920 00008192         TLOG cache
BUFF     BUF3 000042720 00004272         Cache for C$N1 file
BUFF     BUF4 000040960 00004096         Cache for E$BR file
*
*Assign files to a cache pool
*
FILE     C$-N BUF1
FILE     C$-# BUF1
FILE     C$-D BUF1
FILE     C$-S BUF1
FILE     C$-T BUF1
FILE     TLOG BUF2
FILE     C$N1 BUF3
FILE     E$BR BUF4
```

In this example, the facility will maintain four cache pools of varying block sizes above the 16 MB line. The Directory files will use the first pool. The TLF will use the second pool. A scratch pad file is assigned to the third pool, and a file from the Burrys example database is assigned to the fourth pool.

You must also modify the JCL to define the input file for the facility. The input file is defined with DDNAME PDMXIN.

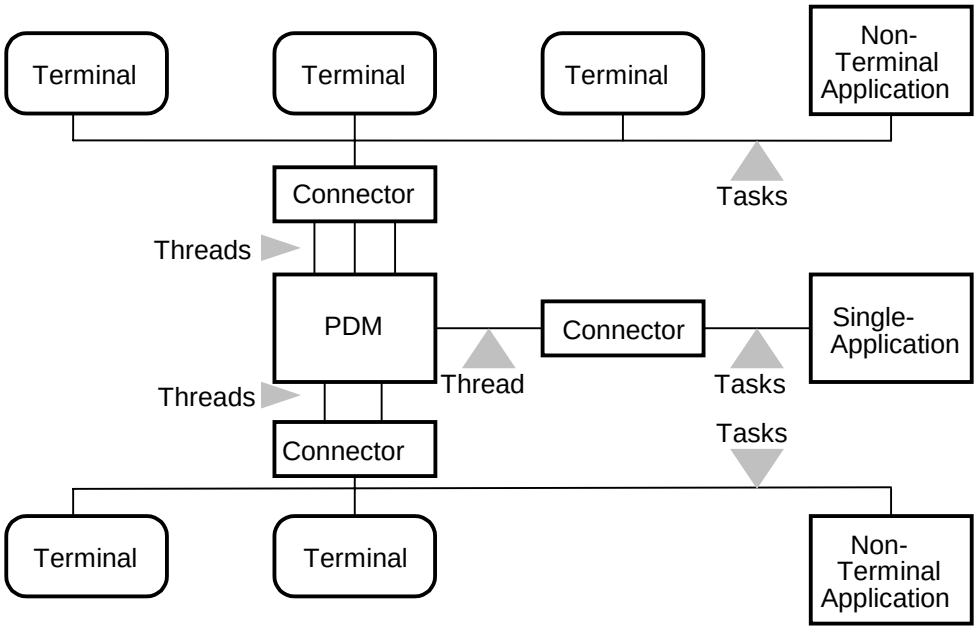
Defining tasks, threads, and interfaces

Every task communicates with a PDM through a connector, called an interface. The interface handles communication between its tasks and the PDM.

Work is processed through a line of access called a thread. A thread can exist between an interface and the PDM, within the interface or within the PDM. The number of requests in progress is limited by the number of threads.

When coding the REALM parameter in the CSIPARM file, you name Environment Description values for maximum connected interfaces, maximum connected threads, and maximum tasks signed on. When the PDM initializes, interface, thread, and task, control blocks are built based on these values.

The following figure illustrates the relationships between tasks, threads, and interfaces.



Enough control blocks to handle all concurrently connected interfaces (CICS and single-task) must be available. Use the following guidelines to calculate the amount you need to specify:

PDM interfaces	PDM threads	PDM tasks
Number of single-task central applications + number of TP Monitors	Concurrent, single-task central applications + for each TP Monitor use:	Concurrent, single-task central applications + for each TP Monitor use:
	For CICS - OPER CONNECT THREADS=parameter or CSTXOTBL THREADS default	For CICS-OPER CONTROL TASKS = parameter +1 or CSTXOTBL TASKS default + 1

The number of threads and tasks in a connector is limited by both the number of PDM threads/tasks and the number of interfaces. The PDM will not allow an interface to connect with a number of tasks/threads that does not leave at least one task and one thread for remaining interfaces. For example, if the PDM has 20 threads, 5 interfaces, and 60 tasks, the maximum number of threads that any one interface can connect with is 16, leaving one free for each of the remaining interfaces.

NOTE

Under the PDM Tasks column, the CICS Connector will request a number of TASKS + 1. This means the maximum number of tasks you can specify in the OPER command or in the CSTXOTBL module is 39.

For single-task execution:

- The single-task version of the CSTEDBMI program loads the single-task version of the interface (CSTESTST) into memory.
- CSTEDBMI is released in the main SUPRA Server linklib in OS/390, and in the core image library in VSE.

For multitask execution:

- The multitask version of the CSTEDBMI program loads the multitask version of the interface (CSTESTMT) into memory.
- The multitask CSTEDBMI is released in the PDM multitask linklib for OS/390, and the core image library for VSE.

NOTE

Using this version in other environments may cause operational problems.

Use the following table to determine which CSTN999x linkdecks to use to create the single-task and multitask CSTEDBMI programs for your environment:

Operating system	ST CSTEDBMI linkdeck	MT CSTEDBMI linkdeck
OS/390	CSTN999Q	CSTN999R
VSE	CSTN999J	CSTN999K

NOTE

CICS does not use the CSTEDBMI module to interface with the PDM.

The following table shows which linkdecks are used to create the PDM interfaces:

Operating system	ST interface CSTESTST	MT interface CSTESTMT	CICS interface CSTXCSMT	MTMT HDMP interface CSTEDCMT
OS/390	CSTN009Q	CSTN999R	CSTN998S	CSTN998T
VSE	CSTN998J	CSTN998K	CSTN998L	CSTN998M

Whether the single-task or multitask version of the CSTEDBMI is used depends on your Load library sequence.

- For single-task: TISXA Main Load Library
- For multitask: TISXA Multitask Interface Library
TISXA Main Load Library

You must use a CSIPARM file to describe the execution environment to the PDM. The sample CSIPARM members are in the sample library on your installation tape.

4

Defining the SUPRA Server database

Definition process

The SUPRA Server database is defined when you organize your company's data into a relational model and then define this model to SUPRA. Defining the SUPRA Server database involves the following steps:

1. Design the relational database model
2. Define the relational entities on the SUPRA Directory
3. Authorize user access to the database

The relational model you design translates to SUPRA Server terminology as the Conceptual Schema, the Internal Schema, and the External Schema. Cincom recommends that you create and add these definitions in the following order:

1. Create and add the Conceptual Schema
2. Create and add the Internal Schema
3. Create and add the External Schema

Each of these steps is discussed in the following sections.

Creating and adding the Conceptual Schema

In the Conceptual Schema you store a definition of your company's data model. This is where you define your company's unique, business integrity rules.

Properly defining a Conceptual Schema is important for the following reasons:

- It represents the "truth" about the data in your organization
- It provides for logical and physical data independence
- It insulates the users from changes in data storage

Creating the Conceptual Schema

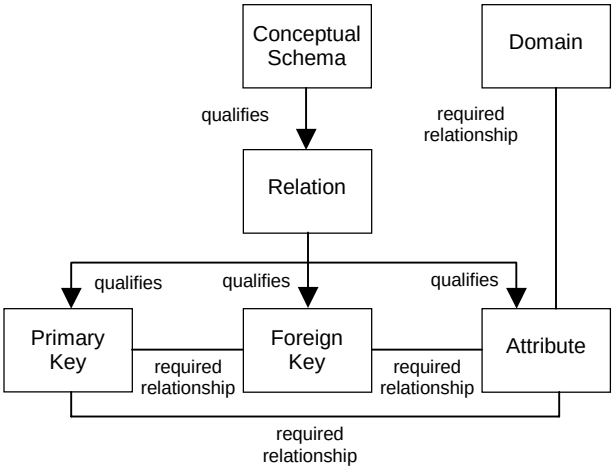
The following categories on the Directory relate to your conceptual model:

- **Conceptual Schema.** A collection of relations that define your database in general relational terms.
- **Relation.** A collection of attributes defining a row. Created from data views.
- **Attribute.** Defines the characteristics of associated domains and relations.
- **Domain.** Defines all possible values of an attribute. It specifies the function of the entity and provides validation options.
- **Foreign Key.** Identifies the attribute or combination of attributes in one relation that is defined across the same domains as the attribute(s) that compose the primary key in the same or another relation.

Certain relationships are required for a Conceptual Schema to be considered consistent. You must add entities according to their position in the hierarchy, but add those at the highest level first. Both Conceptual Schema and Domain are at the highest level of the hierarchy. The following figure shows the hierarchy of the categories in a Conceptual Schema.

NOTE

In this diagram, Primary Key is not a category you may choose. Primary Key data is determined by descriptive data in the Attribute category.



Adding the Conceptual Schema

There are two ways to add a Conceptual Schema:

- Use Online Directory Maintenance to enter all entities.
- Write Batch Directory Maintenance transactions and submit a batch job.

To add the required entities for a Conceptual Schema, follow these steps:

1. Add Domains
2. Add Conceptual Schema
3. Add Relations
4. Add Attributes for each Relation
   a. Enter Y for the Primary Key Attribute(s)
   b. Assign the Domain for each Attribute to create the required relationship between Attribute and Domain
5. Add all Foreign Keys for each Relation (the Target Relation must already exist.)
6. Relate Foreign Keys to Source Attribute(s)
7. Consistency Check the Conceptual Schema

The procedures for adding the entities in these steps are in the *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260, or the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261.

When adding certain entities on the Directory, some descriptive data about that entity may be required. This descriptive information is known as Fixed Attribute Data. Default values for Fixed Attribute Data are defined for each category and may be changed. When adding an entity that requires additional data, you will see an additional screen(s) for input.

Creating and adding the Internal Schema

After you have logically defined your database in the conceptual schema, you “tune” those fields to create the Internal Schema, (you group fields together in data files). These files are defined on the Directory as primary and related files. These files provide the PDM with the most efficient way to access the information defined in the Conceptual Schema.

Creating the Internal Schema

The Internal Schema you create on the Directory describes the following PDM files:

- Data files (primary and related files containing your data)
- Index files consisting of Secondary Key information used to retrieve data from data files
- Directory database files containing definitions of your system, your data, and your users
- System Logs and Task Logs for SLR and TLR
- Statistics Files used to provide PDM statistics

This information is stored in the following entities:

- **Schema Entity.** Definitions of an entire database, including both database files and execution information
- **Environment Description Entity.** Execution-time information describing the operating environment of the PDM
- **File Entity.** Database definitions of the physical files you use to maintain and control your data
- **Internal Record Entity.** Database definitions of the record types (or layouts) within a physical file
- **Physical Field Entity.** Database definitions of the physical data elements which make up an internal record
- **Secondary Key Entity.** Collection of key codes used as a key
- **Key Code Entity.** Physical field(s) constituting a Secondary Key for a single, internal record
- **Buffer Pool Entity.** I/O buffer information for PDM files
- **Log Group Entity.** Characteristics of the System Logs

To define your Internal Schema, you need to decide how you want the PDM to access the relations from the Conceptual Schema and then group these relations accordingly. These groups become the database files. The following table shows you from what entities in the Conceptual Schema you derive the entities in the Internal Schema.

Information from	Is used to build
Relation, Foreign Key, Attribute	Files, Secondary Keys
Relation, Attribute	Internal Records
Domain, Attribute	Physical Field
Primary Key, Foreign Key	Key Code

Adding the Internal Schema

Once you have made the decisions concerning the physical implementation of your database, you can begin to define the Internal Schema's entities to the Directory. As with the Conceptual Schema, you must add the Internal Schema's entities according to the Directory hierarchy. The following steps add the entities in the necessary order:

1. Add Schema

2. Add File
   Add Internal Record
   Add Physical Field
   Add Secondary Key and Key Code
   Relate Key Code to Physical Field
   Relate Secondary Key to Index File

3. Add Environment Description
   Add Buffer Pool
   Relate File to Environment Description
   Add Log Group

For more information on adding files, Environment Descriptions, buffer pools, and related entities, see "Defining the operational environment" on page 115.

Defining database files

The File category of the Internal Schema describes the characteristics of your database files. A SUPRA Server database consists of primary files, related files, and secondary keys. Secondary keys provide alternate paths to the database files and are contained in index files.

- Primary files:
  - Can exist independently of other files
  - Contain single-occurrence data records of the same type and format
  - Are entry points to associated information contained in related files
  - Can be a complete database or can be combined with other primary and/or related files
- Related files:
  - Cannot exist independently of other files
  - Must be connected by at least one linkpath to a primary file
  - Can contain records that are not uniform in content or format
- Secondary keys:
  - Provide alternate paths to primary-file or related-file records. Secondary key information is contained in index files. An index file can contain more than one Secondary Key.
  - Index files are not database files, but the secondary-key information they contain points to primary and related database files.

Primary files

Each record in a primary file is called a primary record. The primary record contains:

- **Root.** Used internally by SUPRA
- **Control key.** An assigned value unique to that record (records are stored and accessed through their control key)
- **Linkpath(s).** Connects any associated related record(s) to the primary record
- **Data item(s).** Entries of data associated with this record's control key

One primary record can be connected with any number of related record groups and can be connected to a single record more than once. Using the linkpath, plus additional data, the PDM can locate associated related records.

The following considerations apply to primary files:

- Primary records can be connected only to related records, not to other primary records.
- Because a primary file can exist independently, linkpaths are not required.
- Linkpaths may be placed anywhere in the record following the control key.
- No duplicate records are permitted for the same control key.
- Primary files cannot contain coded records.

Related files

The records on a related file are called related records. Related records are stored as chains of records connected with the related record's linkpath. Related records are located using the linkpath in the associated primary record. Additional associated records are located using the linkpath in the related record.

Related records contain:

- Linkpath (can be multiple)
- Refer back containing CTRL field values for the primary record to which it is chained to
- Data item(s)
- Record codes

Coded and noncoded related record formats

You can create related records with either a noncoded or coded record format. Noncoded related records are the same length and format. They contain one or more control keys, data, and linkpaths.

When a file has more than one record format, code the first two bytes as the record code. The PDM examines this code to determine the format of the data following it. You can use coded records to:

- Perform a code-directed read which returns only those records in a chain with a particular code(s)
- Build hierarchical structures within a file (header records, comment records)
- Create different access paths

To accommodate a variety of format types, coded related records contain two portions: base data and redefined data. A data item (except linkpaths) in a coded record is either common to all record codes (base data) or valid for only one record code (redefined data).

Secondary keys

Secondary keys are contained in index files. An index file contains:

- Secondary key name(s)
- Secondary key value(s)
- List of pointers to occurrences of the key in the primary or related database file

The value of a Secondary Key for a file is contained in the index file in ascending sequence, which is dependent on data type. This reduces and sometimes eliminates the need for sorting data. Additionally, the Secondary Key can be made up of noncontiguous pieces of data in a primary-file or related-file record. By not requiring a contiguous key value, more keys can be maintained without having redundant data in the record.

While a Secondary Key is populated, you cannot perform the following functions:

- Remove the index file to a Secondary Key relationship
- Delete the index file
- Delete the database file
- Change, delete, or rename the Internal Record
- Add or delete the key code
- Relate or remove the key code to a Physical Field relationship
- Change or delete the Physical Field
- Change or delete the Secondary Key

Use the following formula to calculate the maximum number of Secondary Keys in an index file:

```
(blocksize/128) - 1
```

Maintenance of all secondary keys is automatic and is transparent to PDM applications.

However, in order to access data with the Secondary Key, PDM applications must use the READX DML command. For read-only processing, having all data frequently accessed as part of the Secondary Key can improve I/O performance. For information on DML commands, refer to the *SUPRA Server PDM DML Programming Guide (OS/390 & VSE)*, P26-4340.

For more information on secondary keys, refer to the *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260, or the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261.

Creating and adding the External Schema

The External Schema contains derived views of the database for a particular application. These views are derived from the Conceptual Schema's entities. The following table shows you from what entities in the Conceptual Schema you derive the entities in the External Schema and create derived views:

Information from	Is used to build
Attribute	External Field
Relation	Logical View
Attribute, Primary Key, Foreign Key	Access Set

To add an External Schema, follow these steps:

1. Add External Field
2. Add Logical View
3. Add Access Set

For more information on the External Schema, refer to the *SUPRA Server PDM RDM Administration Guide (OS/390 & VSE)*, P26-8220. For more information on adding the above entities, refer to the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261.

5

Defining the operational environment

Definition process

This chapter provides information on how to define the operational environment to run and control your PDM and Directory. This includes:

- Defining the Internal Schema, including:
  - Adding Environment Descriptions
  - Adding Task Log, System Log, and Statistics Files
- Defining the Directory to the PDM, including:
  - Creating Bootstrap Schema
  - Creating Bootstrap Environment Description
  - Creating Bootstrap Validation Module
- Establishing Directory Security, including:
  - Defining Security Groups
  - Defining Maintenance Restrictions

Defining the Internal Schema

Use Online Directory Maintenance to define the entities used by the Internal Schema to your Directory. You need to add the Internal Schema's entities according to the Directory hierarchy. See "Introduction to the SUPRA Server Directory and Physical Data Manager (PDM)" on page 21 for an overview of the Directory hierarchy. The instructions for adding the Internal Schema's entities are organized to reflect this order.

Add the Internal Schema's entities in the following order:

- Database files (primary, related, and index files)
- Task Log, System Log, and Statistics Files
- Environment Descriptions
- Buffer Pools
- Log Groups

See "Using SUPRA Server Directory schemas" on page 489 for a list of the Directory schemas provided with SUPRA. For more information on the options available for each step and information on adding file types other than those presented here, refer to the *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260.

You can also add these entities using the Batch Directory Maintenance facility. For more information, refer to the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261.

The following are general considerations for Internal Schema definition:

- Throughout the example screens that follow:
  - Press ENTER for the next screen.
  - Press PA2 to return to the Category or Commands screens.
  - All examples assume Active Schema Maintenance has been enabled. See "Using Active Schema Maintenance" on page 39.
- Screens in Release 2.7 will differ slightly from the screens in Release 2.1.6.

Adding primary or related files

If you want to add a file to several Schemas, you should first define the file in a single schema, then copy it. This allows you to avoid duplicate effort and possible errors. The schema COPY command copies all entities constituting the file (all Internal Records, Physical Fields, Secondary Keys, and Key Codes) into the target schema. You may set the Copy External Fields option to Y, which copies all External Fields related to the Physical Fields in the source file along with their relationships.

To add a primary or related file, follow these steps: (The remainder of this section describes each of these steps.)

- File
- Internal Record
- Physical Fields to the File
- Secondary Key to the File

After you have defined the file, perform the following steps for Primary and Related files. For more information on these steps, see the sections indicated.

1. Relate the File to an Environment Description (“Relate the file to an Environment Description” on page 144).
2. Add a Buffer Pool, if necessary (“Adding Buffer Pools” on page 147).
3. Perform a consistency check on the Schema (“Consistency check the Schema” on page 151).
4. Allocate the File (“Allocate the file” on page 153).
5. Format and open the File (“Format and open the file” on page 154).

Adding the file

To add a primary or related database file, follow these steps:

1. Enter FI (File) on the Directory Category screen:

```
                        TIS/XA DIRECTORY MAINTENANCE

ENTER CATEGORY:  FI

AS    ACCESS SET                      LV    LOGICAL VIEW
AT    ATTRIBUTE                       MR    MAINTENANCE RESTRICTION
BP    BUFFER POOL                     PR    PROCEDURE
CS    CONCEPTUAL SCHEMA               RE    RELATION
DC    DIRECTORY COMPONENT DSC         RW    RESERVED WORD
DM    DOMAIN                          SC    SCHEMA
ED    ENVIRONMENT DESCRIPTION         SG    SECURITY GROUP
EM    EDIT MASK                       SK    SECONDARY KEY
FI    FILE                            TA    TABLES
FK    FOREIGN KEY                     US    USER
IR    INTERNAL RECORD                 XF    EXTERNAL FIELD
KC    KEY CODE
LG    LOG GROUP

ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

2. At the Commands screen, enter AD (Add) for selection code and the names of the schema and the file:

```
 FILE:    COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                             RD    REL DSPLY
CG    CHANGE                          RL    RELATE
CK    CHECK                           RM    REMOVE
CO    COPY                            RN    RENAME
DE    DELETE                          SD    STRU DSPLY
DI    DISPLAY                         SE    SHORT EDIT
LE    LONG EDIT                       ST    SHORT TEXT
LT    LONG TEXT                       UT    UTILITIES
RC    REL CHANGE

ENTER SELECTION CODE:  AD             SUBCATEGORY CODE:

ENTER NAMING DATA:
SCHEMA:                       BURRYSCH
FILE:                         BRCH

ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

3. Enter the appropriate attribute data to define the file on the following three screens:

```
  FILE:    ADD          TIS/XA DIRECTORY MAINTENANCE                    1 of 3
SCHEMA:  BURRYSCH                      FILE:  BRCH

LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

DDNAME:                                BRANCH
DATA SET NAME:                         USER.DATASET.NAME

FILE TYPE:                             PRIMARY
FILE ACCESS METHOD:                    BDAM
FILE DEVICE TYPE:                      3350
FILE DEVICE ASSIGNMENT:
LOGICAL RECORD LENGTH:                 21
TOTAL LOGICAL RECORDS:                 1
TOTAL TRACKS:                          0
RECORDS PER BLOCK/CI:                  0
BLOCKS PER TRACK:                      3
VSAM CONTROL INTERVAL:                 0
TOTAL VSAM CNTRL INTVL:                0

COMMAND:
```

```
  FILE:    ADD          TIS/XA DIRECTORY MAINTENANCE                    2 of 3
SCHEMA:  BURRYSCH                       FILE:  BRCH

LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

SITE EXIT NAME:
SITE TABLE NAME:
CODED RECORD INDICATOR:                N
VSE INDICATOR:                         N
OLD FILE INDICATOR:                    N
DL/I ROOT INDICATOR:
VAR. LENGTH INDICATOR:
DL/I PARENT SEGMENT:





COMMAND:
```

```
  FILE:    ADD          TIS/XA DIRECTORY MAINTENANCE                    3 of 3
SCHEMA:  BURRYSCH                      FILE:  BRCH

LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

** BASE IR DATA **:
GET VALID:                            Y
BEFORE GET EXIT:
AFTER GET EXIT:
REPLACE VALID:                        N
BEFORE REPLACE VALID:
AFTER REPLACE VALID:
INSERT VALID:                         N
BEFORE INSERT VALID:
AFTER INSERT VALID:
DELETE VALID:                         N
BEFORE DELETE VALID:
AFTER DELETE VALID:

COMMAND:
```

NOTE

Directory Maintenance automatically adds a BASE. Internal Record for each file. The BASE. Internal Record is the only Internal Record supported for primary files. If you add a coded related file, you may add other Internal Records.

Add Internal record to the file

If you are adding a Primary file or a noncoded related file, the Directory automatically adds the BASE. Internal Record for you. You can proceed to the next section.

If you are adding a coded Related file, proceed to the next section to define the Physical Fields associated with the base (BASE.) portion of the record. However, after adding the Physical Field that will be redefined (coded portion of the record), you must return to this section and add the first user-defined record code for this file.

Enter IR on the “Enter Selection Code” line of any Directory Maintenance screen to bring up the screens for adding an Internal Record. Adding Internal Records is a two-step process:

1. Add an Internal Record
2. Add the Physical Fields for that Internal Record

Repeat these two steps until you have added all of the Internal Records (and corresponding Physical Fields) you need.

Add Physical Fields to the file

The Physical Field category contains information about the physical characteristics of a database element, including storage and format information.

Before adding a Physical Field to a file you must define the Physical Fields in each record of the file.

- If RDM accesses the file, you must add an External Field for each Physical Field and then relate it to the Physical Field. The overlay Physical Field is the last nonlink Physical Field in the BASE. Internal Record.
- You must define several Physical Fields for every PDM primary file, and you may need to define some for a PDM related File. Descriptions and considerations for these Physical Fields follow.

*pppp*ROOT

Description *Conditional*. Required for ESDS and BDAM files. Optional for KSDS files. Contains pointers for hashing synonym chains.

Format *pppp* is the name of the existing primary file

Considerations

- You must define this Physical Field first.
- You may not subdefine this Physical Field.
- The Physical Field Length must be 8, the Parent must be TOP., and the Position must be BEG.

*pppp*CTRL

Description *Required*. Defines the control key and its length.

Format *pppp* is the name of the existing primary file

Considerations

- You must position this Physical Field before all link Physical Fields in a primary file and after the ROOT Physical Field.
- You may subdefine this Physical Field.
- The Physical Field Length must reflect the length of the control key and may not exceed 256 bytes for BDAM and ESDS files or 255 bytes for KSDS files. The Parent field must be TOP.

*pppp*LK*xx*

Description *Conditional*. Required to connect the primary file to a related file.

Format *pppp* is the name of the existing primary file, and *xx* are any unique alphanumeric characters

Considerations

- You must also define this Physical Field in the associated related file.
- This Physical Field must be at the same level as the ROOT field. You may not subdefine this Physical Field.
- The Physical Field Length must be 8.
- In the related file, the Physical Field name must be the same as the link Physical Field defined in the associated primary file. TOP. must be the value of the Parent field. You must also specify a value for the Key Referback field. You must define the Key Referback Physical Field in the same Internal Record as the link Physical Field. The length of the Key Referback Physical Field must be the same as the *pppp*CTRL Physical Field in the associated primary file.

ppppCODE

Description *Conditional.* Required if you use coded records in a related file.

Format *pppp* is the name of the existing related file

Considerations

- This Physical Field must be the first Physical Field in the BASE. Internal Record.
- You may not subdefine this Physical Field.
- The Physical Field Length must be 2.

To add a Physical Field to your file, follow these steps:

1. Enter PF (Physical Field) on the Category screen (see "Adding Environment Descriptions" on page 140) or at the Enter Selection Code prompt of the Commands screen.
2. At the Commands screen enter AD (Add) for the selection code. Directory Maintenance automatically displays the naming data from the previous transaction.

```
 PHYFLD:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                            RD    REL DSPLY
CG    CHANGE                         RL    RELATE
DE    DELETE                         RM    REMOVE
DI    DISPLAY                        RN    RENAME
LE    LONG EDIT                      SD    STRU DSPLY
LT    LONG TEXT                      SE    SHORT EDIT
RC    REL CHANGE                     ST    SHORT TEXT

ENTER SELECTION CODE:  AD            SUBCATEGORY CODE:

ENTER NAMING DATA:
SCHEMA:                      BURRYSCH
FILE:                        BRCH
INTERNAL RECORD:             BASE.
PHYSICAL FIELD:              BRCHROOT

ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

3. Enter the appropriate attribute data for defining the Physical Field on the following two screens:

```
  PHYFLD:  ADD          TIS/XA DIRECTORY MAINTENANCE                  1 of 2
SCHEMA:  BURRYSCH                        FILE:  BRCH
INTREC:  BASE.                         PHYFLD:  BRCHROOT
LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

PARENT:                                TOP.
POSITION:                              BEG.
FUNCTION:                              STRING
UNIT:
DATA FORMAT:                           C
PHYSICAL FIELD LENGTH:                 8
NUMBER DECIMAL PLACES:                 0
SIGNED OPTION:                         N
KEY REFERBACK:
LINKPATH TYPE:                         P
SEQUENCE FIELD:
SEQUENCE FIELD TYPE:                   A
NULLS ALLOWED OPTION:                  N
NULL VALUE:
DEFAULT VALUE:

COMMAND:
```

```
  PHYFLD:  ADD          TIS/XA DIRECTORY MAINTENANCE                  2 of 2
SCHEMA   BURRYSCH                        FILE:  BRCH
INTREC:  BASE.                         PHYFLD:  BRCHROOT
LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

RETRIEVAL VALIDATION:                  N
VALIDATION OPTION:
VALIDATION MINIMUM:
VALIDATION MAXIMUM:
VALIDATION TABLE:
VALIDATION EXIT:
DBD FIELD INDICATOR:





COMMAND:
```

4. Repeat Steps 1 through 3 for each Physical Field in the file.

Add Secondary Key to the file

To add a Secondary Key entity to your File, follow these steps:

1. Enter SK (Secondary Key) on the Category screen (see “Adding Environment Descriptions” on page 140) or at the Enter Selection Code prompt on the Commands screen.

2. At the Commands screen, enter AD (Add) for selection code and the names of the File and the Secondary Key.

```
 SECKEY:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                            LT    LONG TEXT
CG    CHANGE                         RL    RELATE
CO    COPY                           RM    REMOVE
CK    CHECK                          RN    RENAME
DE    DELETE                         SD    STRU DSPLY
DI    DISPLAY                        SE    SHORT EDIT
LE    LONG EDIT                      ST    SHORT TEXT

ENTER SELECTION CODE:  AD            SUBCATEGORY CODE:

ENTER NAMING DATA:
SCHEMA:                      BURRYSCH
FILE:                        BRCH
SECONDARY KEY                BRCHSK01



ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

3. Enter the appropriate attribute data for the Secondary Key on the next screen.

```
 SECKEY:  ADD          TIS/XA DIRECTORY MAINTENANCE                  1 of 1
 SCHEMA:  BURRYSCH                     FILE:  BRCH
 SECKEY:  BRCHSK01
LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

DAMAGED SK ACTION:                   0
PDM MAINTENANCE DENSITY:             50
LOAD DENSITY:                        75
SECONDARY KEY LENGTH:                0
ELIGIBLE TO RDM:                     Y
SK MAINTENANCE ALLOWED:              Y
POPULATED STATUS                     PURGED
CONSISTENT:                          N
GENERATION INDICATOR:                N
UNIQUENESS OPTION:                   N
FILE RECORD POINTER:                 DIRECT
RECORD POINTER ORDERING:             SORTED
DATA TYPE SORTING:                   N
PRIMARY SECONDARY KEY:               N

COMMAND:
```

4. After you add the Secondary Key, you must add a Key Code to the Secondary Key. The Key Code name must be the same as the Internal Record name that qualifies the Physical Fields used to define the Secondary Key. You do not need to enter any attribute data when adding a Key Code. Directory Maintenance displays an attribute screen showing the Key Code length and whether the Key Code is the result of a Directory Maintenance GENERATE command. To add a Key Code , follow these steps:

   a. Enter KC (Key Code) on the Category screen (see “Adding Environment Descriptions” on page 140) or at the Enter Selection Code prompt on the Commands screen.

b. Enter AD (Add) for selection code and the name of the Key Code. Directory Maintenance automatically displays naming data from the previous transaction.

```
  KEY CD:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                           RD    REL DSPLY
DE    DELETE                        RL    RELATE
DI    DISPLAY                       RM    REMOVE
LE    LONG EDIT                     SD    STRU DSPLY
LT    LONG TEXT                     SE    SHORT EDIT
RC    REL CHANGE                    ST    SHORT TEXT

ENTER SELECTION CODE:  AD           SUBCATEGORY CODE:

ENTER NAMING DATA:
SCHEMA:                      BURRYSCH
FILE:                        BRCH
SECONDARY KEY:               BRCHSK01
KEY CODE:                    BASE.


ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

Directory Maintenance displays the attribute data on this screen:

```
  KEY CD:  ADD          TIS/XA DIRECTORY MAINTENANCE                  1 of 1
  SCHEMA:  BURRYSCH                      FILE:  BRCH
  SECKEY:  BRCHSK01                    KEY CD:  BASE.
LAST UPDATE   19.34.17   1125/19xx   V:  0001   USER:  CSI-DBA

KEY CODE LENGTH:                     0
GENERATION INDICATOR:                N



COMMAND:
```

c. You then relate the Key Code to the Physical Fields by following these steps:

   i. Enter KC (Key Code) on the Category screen (see “Adding Environment Descriptions” on page 140) or at the Enter Selection Code prompt of the Commands screen.

   ii. Enter RL (Relate) for selection code and PF (Physical Field) for the subcategory code. Directory Maintenance automatically displays the naming data from the previous transaction.

   iii. Enter the appropriate attribute data on the next two screens. List the names of the related Physical Fields. The Physical Fields must belong to the data file and the Internal Record with the same name as the Key Code.

```
  KEY CD:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                             RD    REL DSPLY
DE    DELETE                          RL    RELATE
DI    DISPLAY                         RM    REMOVE
LE    LONG EDIT                       SD    STRU DSPLY
LT    LONG TEXT                       SE    SHORT EDIT
RC    REL CHANGE                      ST    SHORT TEXT

ENTER SELECTION CODE:  RL             SUBCATEGORY CODE:  PF

ENTER NAMING DATA:
SCHEMA:                       BURRYSCH
FILE:                         BRCH
SECONDARY KEY:                BRCHSK01
KEY CODE:                     BASE.

ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

```
 KEY CD:  RELATE       TIS/XA DIRECTORY MAINTENANCE                  1 of 1
 SCHEMA:  BURRYSCH                       FILE:  BRCH
 SECKEY:  BRCHSK01                     KEY CD:  BASE.
LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

PHYSICAL FIELD LENGTH:                BRCHNAME



COMMAND:
```

iv. This screen displays, containing the name of the last Physical Field you added to this file. If the name is correct, press ENTER. If the name is incorrect, supply the name of the Physical Field you want to be included in the Key Code. Note that you may specify multiple Physical Field names. Press RETURN to position the cursor on the next line to enter another Physical Field name. The example above, however, specifies only one Physical Field for this Key Code.

v. Directory Maintenance returns an attribute screen for each Physical Field. In the Position field, enter the position of the field in the Secondary Key. To retrieve the Physical Fields in ascending sequence, enter A in the Sequence Field Type field.

```
 KEY CD:  RELATE       TIS/XA DIRECTORY MAINTENANCE                  1 of 1
 SCHEMA:  BURRYSCH                      FILE:  BRCH
 SECKEY:  BRCHSK01                    KEY CD:  BASE.
LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USEER:  CSI-DBA

PHYSICAL FIELD LENGTH:               BRCHNAME
POSITION:                            END.
SEQUENCE FIELD TYPE:                 A



COMMAND:
```

vi. If you have multiple Key Codes, repeat the process for each.

d. You then relate the Secondary Key to an Index File that already exists on the Directory. See “Adding index files” on page 132, for information on adding index files. Perform the following steps to relate the Secondary Key to the index file:

   i. Enter SK on the Directory Category screen (see “Adding Environment Descriptions” on page 140) or at the Enter Selection Code prompt of the Commands screen.

   ii. At the Commands screen, enter RL (Relate) for selection code and FI for the subcategory code.

```
  SECKEY:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                             LT    LONG TEXT
CG    CHANGE                          RL    RELATE
CO    COPY                            RM    REMOVE
CK    CHECK                           RN    RENAME
DE    DELETE                          SD    STRU DSPLY
DI    DISPLAY                         SE    SHORT EDIT
LE    LONG EDIT                       ST    SHORT TEXT

ENTER SELECTION CODE:  RL             SUBCATEGORY CODE:  FI

ENTER NAMING DATA:
SCHEMA:                       BURRYSCH
FILE:                         BRCH
SECONDARY KEY                 BRCHSK01





ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

iii. Enter the name of the appropriate index file at the next screen:

```
 SECKEY:  RELATE       TIS/XA DIRECTORY MAINTENANCE                  1 of 1
 SCHEMA:  BURRYSCH                     FILE:  BRCH
 SECKEY:  BRCHSK01
LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

FILE:                                IXOX



COMMAND:
```

Adding index files

To add an index file, follow these steps:

1. Add the File.
2. Define Physical Fields to the File.

The following sections explain these steps in detail.

After you have defined the file, perform the following steps for an Index file:

1. Relate the File to an Environment Description (see “Relate the file to an Environment Description” on page 144).
2. Add a Buffer Pool, if necessary (see “Adding Buffer Pools” on page 147).
3. Perform a consistency check on the Schema (see “Consistency check the Schema” on page 151).
4. Allocate the File (see “Allocate the file” on page 153).
5. Format and open the File (see “Format and open the file” on page 154).
6. Activate the Secondary Key using the File Utilities (see “Activate Secondary Key using the file utilities” on page 157).

Adding an index file

To add an index file, follow these steps:

1. Enter FI (File) on the Directory Category screen (see “Adding Environment Descriptions” on page 140) or at the Enter Selection Code prompt on the Commands screen.

2. At the Commands screen, enter AD (Add) for selection code and the names of the schema and the file.

```
  FILE:    COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                            RD    REL DSPLY
CG    CHANGE                         RL    RELATE
CK    CHECK                          RM    REMOVE
CO    COPY                           RN    RENAME
DE    DELETE                         SD    STRU DSPLY
DI    DISPLAY                        SE    SHORT EDIT
LE    LONG EDIT                      ST    SHORT TEXT
LT    LONG TEXT                      UT    UTILITIES
RC    REL CHANGE

ENTER SELECTION CODE:  AD            SUBCATEGORY CODE:

ENTER NAMING DATA:
SCHEMA:                      BURRYSCH
FILE:                        INDX


ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

3. Enter the appropriate attribute data defining the file on the following three screens exactly as shown:

```
 FILE:    ADD          TIS/XA DIRECTORY MAINTENANCE                    1 of 3
SCHEMA:  BURRYSCH                      FILE:  INDX

LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

DDNAME:                                INDEX
DATA SET NAME:                         USER.DATASET.NAME

FILE TYPE:                             INDEX
FILE ACCESS METHOD:                    BDAM
FILE DEVICE TYPE:                      3350
FILE DEVICE ASSIGNMENT:
LOGICAL RECORD LENGTH:                 4096
TOTAL LOGICAL RECORDS:                 1000
TOTAL TRACKS:                          0
RECORDS PER BLOCK/CI:                  0
BLOCKS PER TRACK:                      3
VSAM CONTROL INTERVAL:                 0
TOTAL VSAM CNTRL INTVL:                0

COMMAND:
```

```
  FILE:    ADD          TIS/XA DIRECTORY MAINTENANCE                    2 of 3
                                FILE:  INDX

LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

SITE EXIT NAME:
SITE TABLE NAME:
CODED RECORD INDICATOR:                N
VSE INDICATOR:                         N
OLD FILE INDICATOR:                    N
DL/I ROOT INDICATOR:
VAR. LENGTH INDICATOR:
DL/I PARENT SEGMENT:









COMMAND:
```

```
  FILE:    ADD          TIS/XA DIRECTORY MAINTENANCE                    3 of 3
SCHEMA:  BURRYSCH                      FILE:  BRCH

LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

** BASE IR DATA **:
GET VALID:                            Y
BEFORE GET EXIT:
AFTER GET EXIT:
REPLACE VALID:                        N
BEFORE REPLACE VALID:
AFTER REPLACE VALID:
INSERT VALID:                         N
BEFORE INSERT VALID:
AFTER INSERT VALID:
DELETE VALID:                         N
BEFORE DELETE VALID:
AFTER DELETE VALID:



COMMAND:
```

NOTE

Directory Maintenance automatically adds a BASE. Internal Record for each file. If you add a coded related file, you may add other Internal Records.

Add Physical Fields to the file

The Physical Field category contains information about the physical characteristics of a database element, including storage and format information. Index files require only one Physical Field. Perform the following steps to add a Physical Field to your file:

1. Enter PF (Physical Field) on the Directory Category screen (see "Adding Environment Descriptions" on page 140) or at the Enter Selection Code prompt of the Commands screen.

2. At the Commands screen, enter AD (Add) for the selection code. Directory Maintenance automatically displays the naming data.

```
PHYFLD:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                           RD    REL DSPLY
CG    CHANGE                        RL    RELATE
DE    DELETE                        RM    REMOVE
DI    DISPLAY                       RN    RENAME
LE    LONG EDIT                     SD    STRU DSPLY
LT    LONG TEXT                     SE    SHORT EDIT
RC    REL CHANGE                    ST    SHORT TEXT

ENTER SELECTION CODE:  AD           SUBCATEGORY CODE:

ENTER NAMING DATA:
SCHEMA:                      BURRYSCH
FILE:                        INDX
INTERNAL RECORD:             BASE.
PHYSICAL FIELD:              INDXDATA




ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

3. Enter the appropriate attribute data to define the Physical Field on the next two screens. Complete these screens using the following values:

```
  PHYFLD:  ADD          TIS/XA DIRECTORY MAINTENANCE                  1 of 2
SCHEMA:  BURRYSCH                       FILE:  INDX
INTREC:  BASE.                        PHYFLD:  INDXDATA
LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

PARENT:                               TOP.
POITION:                              END.
FUNCTION:                             STRING
UNIT:
DATA FORMAT:                          C
PHYSICAL FIELD LENGTH:                4096
NUMBER DECIMAL PLACES:                0
SIGNED OPTION:                        N
KEY REFERBACK:
LINKPATH TYPE:
SEQUENCE FIELD:
SEQUENCE FIELD TYPE:
NULLS ALLOWED OPTION:
NULL VALUE:
DEFAULT VALUE:

COMMAND:
```

```
PHYFLD:  ADD          TIS/XA DIRECTORY MAINTENANCE                  2 of 2
SCHEMA:  BURRYSCH                       FILE:  BRCH
INTREC:  BASE.                        PHYFLD:  BRCHROOT
LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

RETRIEVAL VALIDATION:                 N
VALIDATION OPTION:
VALIDATION MINIMUM:
VALIDATION MAXIMUM:
VALIDATION TABLE:
VALIDATION EXIT:
DBD FIELD INDICATOR:









COMMAND:
```

NOTE

Only attributes need to be defined for this Physical Field. All other attributes should be blank.

Adding Task Log, System Log, or Statistics files

The steps for adding a Task Log file, System Log file, or Statistics file are listed below. The screens are the same as for adding primary or related files (see “Adding primary or related files” on page 117). Considerations are listed where applicable.

1. Add file entity. The following considerations apply to the attribute data screens:

   a. In the File Type field, enter TASKLOG, SYSTLOG, or STATLOG.

   b. In the Logical Record Length field, enter the value for the record length of the file.

   c. In the Records Per Block/Control Interval field, enter 1.

   d. In the Old File Indicator field, enter N.

2. Add one physical field for the file. The following considerations apply to the attribute data screens:

   a. In the Parent field, enter TOP.

   b. In the Position field, enter BEG. or END.

   c. In the Function field, enter STRING.

   d. In the Data Format field, enter C.

   e. In the Field Length, enter the value of the Logical Record Length field for the associated file.

   f. In the Number Decimal Places field, enter 0.

   g. In the Signed Option field, enter N.

   h. In the Required Field, enter N.

3. Add Buffer Pool. See “Adding Buffer Pools” on page 147 for the steps for adding Buffer Pools.

4. Relate file to an Environment Description.

5. Check the schema for consistency.

6. Allocate the file. The file can be allocated in either blocks or tracks (for CKD devices). For the VSE operating system, you may also allocate the file in blocks for FBA devices. To obtain the values needed for the variables, display the file after checking for consistency. If you use Batch Directory Maintenance, use the output of the run.

   To allocate an ESDS VSAM file, use the calculated blocks (for CKD devices) as the RECORDS parameter of IDCAMS.

7. Format the file. Use the DBA utilities Format function.

Adding Environment Descriptions

The Environment Description category describes logging options and other operational information used by the PDM. You will need one Environment Description for the DBA Utilities, and at least one for production use. For more information on logging and recovery for your database, refer to the *SUPRA Server PDM Logging and Recovery Guide (OS/390 & VSE)*, P26-2223.

To add an Environment Description, follow these steps:

1. Enter ED (Environment Description) at the Category screen.

```
                        TIS/XA DIRECTORY MAINTENANCE

ENTER CATEGORY:  ED

AS    ACCESS SET                      LV    LOGICAL VIEW
AT    ATTRIBUTE                       MR    MAINTENANCE RESTRICTION
BP    BUFFER POOL                     PF    PHYSICAL FIELD
CS    CONCEPTUAL SCHEMA               PR    PROCEDURES
DC    DIRECTORY COMPONENT DSC         RE    RELATION
DM    DOMAIN                          RW    RESERVED WORD
ED    ENVIRONMENT DESCRIPTION         SC    SCHEMA
EM    EDIT MASK                       SG    SECURITY GROUP
FI    FILE                            SK    SECONDARY KEY
FK    FOREIGN KEY                     TA    TABLES
IR    INTERNAL RECORD                 US    USER
KC    KEY CODE                        XF    EXTERNAL FIELD
LG    LOG GROUP


ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

2. Enter AD (Add) for selection code and the names of the Schema and Environment Description at the Commands screen.

```
  ENVDES:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                           RD    REL DSPLY
CG    CHANGE                        RL    RELATE
CO    COPY                          RM    REMOVE
DE    DELETE                        RN    RENAME
DI    DISPLAY                       SD    STRU DSPLY
LE    LONG EDIT                     SE    SHORT EDIT
LT    LONG TEXT                     ST    SHORT TEXT
RC    REL CHANGE

ENTER SELECTION CODE:  AD           SUBCATEGORY CODE:

ENTER NAMING DATA:
SCHEMA:                       BURRYSCH
ENVIRONMENT DESCRIPTION:      ENVDESC1


ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

3. Enter appropriate values on both screens of attribute data. For your production Environment Description, use the following values:

   a. If you use task logging, set the Task Log Option to Y.

   b. If you use system logging, set the Log Options to some valid value.

   c. If you use the Recover and Restore utilities, set the End Log Option to a value other than C.

   d. For your DBA Utilities Environment Description, use the following values:

      - Set the Access Mode to UPDATE or RDONLY.

      - Set the OPENX Option to P.

```
 ENVDES:  ADD          TIS/XA DIRECTORY MAINTENANCE                    1 of 3
SCHEMA   BURRYSCH                      ENVDES:  ENVDESC1

LAST UPDATE   19.34.17   11/2519xx   V:  0000   USER:  CSI-DBA

MAX. CONNECTED INTERF.:              1
MAX. CONNECTED THREADS:              1
MAX. SIGNED-ON TASK:                 1
PDM STORAGE ALLOCATION               200K
MAXIMUM HELD RECORDS                 1000
LOG OPTION:                          NNNN
TASK LOG OPTION:                     Y
END LOG OPTION:                      F
OPENX OPTION:                        P
PDM CONSOLE OPTION:                  Y
STATISTICS INDICATOR:                N
ACCESS MODE:                         UPDATE
DIRECTORY ACCESS METHOD:             READ



COMMAND:
```

For more information on recommended options in the Environment Description, refer to the *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260, or the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261.

4. At the next screen add the appropriate data values as follows:

```
 ENVDES:  ADD          TIS/XA DIRECTORY MAINTENANCE                    2 of 3
SCHEMA:  BURRYSCH                       ENVDES:  ENVDESC1

LAST UPDATE   19.34.17   11/25/19xx   V:  0000   USER:  CSI-DBA

BATCH DELAY TIME:                    60
TP MONITOR DELAY TIME:               5
FUNC. INPUT EXIT NAME:
OPEN EXIT NAME:
CLOSE EXIT NAME:
READ EXIT NAME:
WRITE EXIT NAME:
CHECK EXIT NAME:
SHUTDOWN PASSWORD:                   **NONE**
LOGICAL WRITE EXIT NAME:
NEW VOLUME EXIT NAME:
LOCAL SITE NAME:
BOUND NAME                           ENVDESC1
MAX MONITORED RECORDS                100
GENERATION INDICATOR:                N


COMMAND:
```

If you are using the optional Buffer Cache facility, enter CSTTEXIT in all exit name fields on this screen. If, however, you have your own exits coded, you must link-edit those exits with CSTTEXIT (see “Using PDM exits” on page 249).

5. At the final screen, add the appropriate values as follows:

```
  ENVDES:  ADD          TIS/XA DIRECTORY MAINTENANCE                    3 of 3
SCHEMA:  BURRYSCH                       ENVDES: ENVDESC1

LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

INTERFACE SCAN INTERVAL:   300
CLEANUP THRESHOLD:         90









COMMAND:
```

Relate the file to an Environment Description

The Environment Description category contains execution-time information describing the operating environment to the PDM. Because you may use a file in several execution environments, you can relate a file to multiple Environment Descriptions.

The PDM uses this relationship to determine which files to access when using an Environment Description. You need to relate all associated PDM files to an Environment Description. The following considerations apply:

- Only PDM files (primary, related, index, TASKLOG, STATLOG, and SYSTLOG files) can be related to an Environment Description.
- If a PDM-related file is related to an Environment Description, every PDM primary file to which the related file is linked must be related to the Environment Description, and all index files related to primary or related file Secondary Keys must be related to the Environment Description.
- If the Environment Description specifies task logging, you must relate one file of type TASKLOG to the Environment Description.
- If the Environment Description specifies statistics logging, you must relate one file of type STATLOG to the Environment Description.
- If you specify any of the four system logging options in the Environment Description, then you must relate each file identified in any of your Log Groups to the Environment Description.
- Files in the same Log Group should share the same Buffer Pool.

To relate the Environment Description to one or more files, follow these steps:

1. Enter ED on the Directory Category screen (see "Adding Environment Descriptions" on page 140) or at the Enter Selection Code prompt of the Commands screen.

2. Enter FI for selection code and ED for the subcategory code:

```
ENVDES:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                           RD    REL DSPLY
CG    CHANGE                        RL    RELATE
CO    COPY                          RM    REMOVE
DE    DELETE                        RN    RENAME
DI    DISPLAY                       SD    STRU DSPLY
LE    LONG EDIT                     SE    SHORT EDIT
LT    LONG TEXT                     ST    SHORT TEXT
RC    REL CHANGE

ENTER SELECTION CODE:  FI           SUBCATEGORY CODE:  ED

ENTER NAMING DATA:
SCHEMA:                      BURRYSCH
ENVIRONMENT DESCRIPTION:     ENVDESC1



ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

3. Enter the name of each file (specified in the schema) to which you want to relate the Environment Description (up to 15 per screen).

```
  ENVDES:  RELATE       TIS/XA DIRECTORY MAINTENANCE
SCHEMA:  BURRYSCH                      ENVDESC: ENVDESC1

LAST UPDATE   19.34.17   11/25/19xx   V:  0000   USER:  CSI-DBA

FILE:                       E$BR









COMMAND:
```

4. Enter appropriate values on the relationship attribute data screen for each file to be related. For your DBA Utilities Environment Description, the Open Mode must be NONE.

```
 ENVDES:  RELATE       TIS/XA DIRECTORY MAINTENANCE                    1 of 1
SCHEMA:  BURRYSCH                      ENVDESC: ENVDESC1

LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

FILE:                      E$BR
OPEN MODE:                 NONE
BUFFER POOL:               PRI1
TASK LOG OPTION:           Y
FUNCTION LOG OPTION:       Y
AFTER IMAGE LOG OPTION:    Y
BEFORE IMAGE LOG OPTION:   Y









COMMAND:
```

NOTE

Specific logging options are available for each file related to the Environment Description. For more information, see “Using SUPRA Server logging” on page 44. Refer also to the *SUPRA Server PDM Logging and Recovery Guide (OS/390 & VSE)*, P26-2223.

Adding Buffer Pools

The Buffer Pool category describes the file I/O buffers used by the PDM. Each file must have a Buffer Pool defined for its use. Task Log, System Log, and Statistics Log files may share a Buffer Pool. Primary, related, and index files may share a Buffer Pool. However, a database or an index file cannot share the Buffer Pool used for the Task Log, System Log, or Statistics Log File. For more information on sharing Buffer Pools, refer to the *SUPRA Server PDM Tuning Guide (OS/390 & VSE),* P26-0225.

To add a Buffer Pool, follow these steps:

1. Enter BP (Buffer Pool) on the Directory Category screen or at the Enter Selection Code prompt of the Commands screen.

2. At the Commands screen enter AD (Add) for selection code and the names of the Schema, Environment Description, and Buffer Pool.

```
  BUFFER:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                            DI    DISPLAY
CG    CHANGE                         RN    RENAME
DE    DELETE                         SD    STRU DSPLY

ENTER SELECTION CODE:  AD             SUBCATEGORY CODE:

ENTER NAMING DATA:
SCHEMA:                       BURRYSCH
ENVIRONMENT DESCRIPTION:      ENVDESC1
BUFFER POOL:                  PRI1





ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

3. At the next screen enter a value for the Direct Buffer Count field on the attribute data screen. The minimum number of direct buffers should equal the number of Task Log Files, Statistics Log Files, and Log Groups sharing the Buffer Pool.

```
  BUFFER:  ADD          TIS/XA DIRECTORY MAINTENANCE                  1 of 1
SCHEMA   BURRYSCH                     ENVDES:  ENVDESC1
BUFFER:  PRI1
LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USU:  CSI-DBA

BUFFER SIZE:                          0
DIRECT BUFFER COUNT:                  2
SERIAL BUFFER COUNT:                  0
SERIAL THREAD COUNT:                  0
MIN% DIR BUFFERS AVAIL:               50
GENERATION INDICATOR:                 N









COMMAND:
```

Serial processing is not available in this release. However, if you supply values for the Serial Buffer Count and Serial Thread Count fields, the PDM combines these values with the value of the Direct Buffer Count field using the following formula:

```
buffer allocation = d + (b * t)
```

where:

d= Direct Buffer Count

b= Serial Buffer Count

t= Serial Thread Count

Adding Log Groups

The Log Group category connects one or more system log data sets to form a logical system log. If your log files do not already exist, you need to add them.

While Log Groups are always optional, Cincom recommends that you add them if you are using system logging. (System logging is not required for the Environment Description used for DBA Utilities processing.)

To add a Log Group, follow these steps:

1. Enter LG (Log Group) on the Directory Category screen (see "Adding Environment Descriptions" on page 140) or at the Enter Selection Code prompt of the Commands screen.

2. At the Commands screen, enter AD (Add) for selection code and the names of the Schema, Environment Description, and Log Group.

```
 LOG GR:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                            DI    DISPLAY
CG    CHANGE                         RN    RENAME
DE    DELETE

ENTER SELECTION CODE:  AD             SUBCATEGORY CODE:

ENTER NAMING DATA:
SCHEMA:                       BURRYSCH
LOG GROUP:                    LOGGRP01









ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

3. At the attribute data screen, enter appropriate values:

```
  LOG GR:  ADD          TIS/XA DIRECTORY MAINTENANCE                  1 of 1
SCHEMA   BURRYSCH                     ENVDES:  ENVDESC1
LOG GR:  LOGGRP01
LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

LOG WRAP OPTION:                     N
LOGGING SYNCH OPTION:                Y
LOG GROUP FILE1 NAME:                CDFI
LG FILE1 VOL MX BLOCK#:              0
LOG GROUP FILE2 NAME:
LG FILE2 VOL MX BLOCK#:              0
LOG GROUP FILE3 NAME:
LG FILE3 VOL MX BLOCK#:              0
LOG GROUP FILE4 NAME:
LG FILE4 VOL MX BLOCK#:              0





COMMAND:
```

Consistency check the Schema

To verify that the Schema is consistent and check all entities qualified by the Schema, follow these steps:

1. Enter SC (Schema) on the Directory Category screen (see “Adding Environment Descriptions” on page 140) or at the Enter Selection Code prompt of the Commands screen.

2. Enter CK (Check) for selection code and the name of the Schema.

```
  SCHEMA:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                             RL    RELATE
CG    CHANGE                          RM    REMOVE
CK    CHECK                           RN    RENAME
CO    COPY                            SD    STRU DSPLY
DE    DELETE                          SE    SHORT EDIT
DI    DISPLAY                         SF    SPEC FUNC
LT    LONG TEXT                       ST    TEXT

ENTER SELECTION CODE:  CK             SUBCATEGORY CODE:

ENTER NAMING DATA:
SCHEMA:                       BURRYSCH





ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

3. Press ENTER for the next screen. Set the check options. Check all inconsistent physical entities within the Schema.

```
 SCHEMA:  CHECK        TIS/XA DIRECTORY MAINTENANCE                  1 of 1
SCHEMA:  BURRYSCH

LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

** CHECK OPTION **:
ALL PHYSICAL ENTITIES:               N
INCON PHYSICAL ENTITIES:             Y
ALL LOGICAL ENTITIES:                N
INCON LOGICAL ENTITIES:              Y









COMMAND:
```

4. Press ENTER. The screen is repeated with the prompt YES/NO at the bottom. Enter Y and press ENTER.

```
  SCHEMA:  CHECK        TIS/XA DIRECTORY MAINTENANCE                  1 of 1
SCHEMA:  BURRYSCH

LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

** CHECK OPTION **:
ALL PHYSICAL ENTITIES:               N
INCON PHYSICAL ENTITIES:             Y
ALL LOGICAL ENTITIES:                N
INCON LOGICAL ENTITIES:              Y





YES/NO:  Y
```

Allocate the file

You can allocate the file in either blocks or tracks (for CKD devices). For the VSE operating system, you may also allocate the file in blocks for FBA devices. To obtain the values needed for the variables, display the file after consistency checking the file. If you use Batch Directory Maintenance, use the output of the run.

To allocate a VSAM ESDS, use the calculated blocks (for CKD devices) as the RECORDS parameter of IDCAMS.

Format and open the file

Use either the Directory Maintenance File utilities or the DBA utilities Format function. If you use the Directory Maintenance File utilities you must use Active Schema Maintenance (see "Introduction to the SUPRA Server Directory and Physical Data Manager (PDM)" on page 21). For more information on the Format function and the DBA utilities, refer to the *SUPRA Server PDM DBA Utilities User's Guide (OS/390 & VSE)*, P26-6260.

Perform the following steps to format your file:

1. Enter FI on the Directory Category screen (see "Adding Environment Descriptions" on page 140) or at the Enter Selection Code prompt of the Commands screen.

2. Enter UT (Utility) for selection code and the names of the Schema and File.

```
 FILE:  COMMANDS       TIS/XA DIRECTORY MAINTENANCE

AD    ADD                              RD    REL DSPLY
CG    CHANGE                           RL    RELATE
CK    CHECK                            RM    REMOVE
CO    COPY                             RN    RENAME
DE    DELETE                           SD    STRU DSPLY
DI    DISPLAY                          SE    SHORT EDIT
LE    LONG EDIT                        ST    SHORT TEXT
LT    LONG TEXT                        UT    UTILITIES
RC    REL CHANGE

ENTER SELECTION CODE:  UT              SUBCATEGORY CODE:

ENTER NAMING DATA:
SCHEMA:                        BURRYSCH
FILE:                          BRCH


ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

NOTE

If you want to format a previously existing file, you must CL (Close) the file with a close mode of COMP before you issue the FT command shown in the next step.

3. Enter FT to format the file.

```
 UT:  SUBCOMMANDS       TIS/XA DIRECTORY MAINTENANCE

CL    CLOSE                           OP    OPEN
DP    DEPOPULATE                      PP    POPULATE
FT    FORMAT                          RO    REORGANIZE

ENTER SELECTION CODE:  FT

ENTER NAMING DATA:
SCHEMA:                       BURRYSCH
FILE:                         BRCH


ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

4. After the format completes, enter OP on the Utility Subcommands screen to open the file.

```
 UT:  SUBCOMMANDS       TIS/XA DIRECTORY MAINTENANCE

CL    CLOSE                           OP    OPEN
DP    DEPOPULATE                      PP    POPULATE
FT    FORMAT                          RO    REORGANIZE

ENTER SELECTION CODE:  OP

ENTER NAMING DATA:
SCHEMA:                       BURRYSCH
FILE:                         BRCH


ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

5. Enter the applicable open mode (shared update, exclusive update, intent to update, or read only) at the next screen.

```
  FILE:  UTILITIES      TIS/XA DIRECTORY MAINTENANCE                  1 of 1
SCHEMA:  BURRYSCH                     FILE:  BRCH

LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

OPEN MODE:                            SUPD



COMMAND:
```

Activate Secondary Key using the file utilities

Perform the following steps to activate a new index file. You can perform these functions using either Online or Batch Directory Maintenance. The following examples show Online Directory Maintenance. You must use Active Schema Maintenance to use the file utilities (see "Introduction to the SUPRA Server Directory and Physical Data Manager (PDM)" on page 21).

1. To Purge Original Data File, follow these steps:

   a. Enter FI on the Directory Category screens (see "Adding Environment Descriptions" on page 140) or at the Enter Selection Code prompt of the Commands screen.

   b. Enter UT (Utilities) for selection code and CL (Close) for the subcategory code.

   c. Enter PURG to purge the original file description from the PDM.

```
  FILE:  UTILITIES      TIS/XA DIRECTORY MAINTENANCE
SCHEMA:  BURRYSCH                          FILE:  BRCH

LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USEER:  CSI-DBA

CLOSE MODE:                            PURG



COMMAND:
```

2. Populate the Secondary Key.

   You need to populate the Secondary Key so the PDM can maintain the index information and applications can use the index. There are two ways to populate the Secondary Key:

   a. Directory Maintenance file utilities (online or batch)

   b. Sorted Populate function on DBA utilities (batch)

   For large PDM files, Cincom recommends using the Sorted Populate function. For information on the Sorted Populate function, refer to the *SUPRA Server PDM DBA Utilities User's Guide (OS/390 & VSE)*, P26-6260.

NOTE

If you want to move a Secondary Key from one index file to another, you must depopulate/purge the old Secondary Key before you close/purge the data file.

To populate the Secondary Key using the Directory Maintenance file utilities, follow these steps:

a. Enter FI on the Directory category screen or at the Enter Selection Code prompt of the Commands screen.

b. Enter UT (Utilities) for selection code, PP (Populate) for the subcategory code, and the name of the data file.

c. Enter the Secondary Key name(s) used for this index file.

```
    UT:  POPULATE       TIS/XA DIRECTORY MAINTENANCE
SCHEMA:  BURRYSCH                        FILE:  BRCH

LAST UPDATE   19.34.17   11/25/19xx   V:  0001   USER:  CSI-DBA

SECONDARY KEY:                       BRCHSK01









COMMAND:
```

3. Open the data file.

   Perform the following steps to open the data file:

   a. Enter FI on the Directory Category screen or at the Enter Selection Code prompt of the Commands screen.

   b. Enter UT (Utilities) for selection code, OP (Open) for the subcategory code, and the name of the data file.

   c. Enter the applicable open mode (shared update, exclusive update, intent to update, or read only). Directory Maintenance automatically opens the index file in a compatible mode.

Defining the Directory to the PDM

While you can use the tailored bootstrap environment on your installation tape to bring up your Directory files, you may choose to modify the boot environment to suit your needs. If you want to modify the bootstrap environment, you need to do so before you begin defining your Directory database files.

The SUPRA Server bootstrap environment contains the information the PDM requires to open and read your Directory files during initialization. In other words, a bootstrap environment contains the physical descriptions of the information in your Directory files (how they are accessed, how they are allocated, file size, etc.).

The bootstrap environment consists of the bootstrap schema, the bootstrap Environment Description, and the validation module. These are modified using the following utility programs:

- **Modify Schema.** Modifies the bootstrap schema module (CSTASCHM) which contains the definition of the Directory files and the TLF. Run this program if you want to change any of the following: device types or access methods, ddnames, block sizes of the files, or physical sizes of the files.

- **Create Environment Description.** Creates the bootstrap-environment-description modules which contain the execution parameters (the same information that is in a Directory Environment Description). Run this program if you want to add an Environment Description or replace an existing one.

- **Create Validation Module.** Creates the validation module that validates the consistency of a schema and Environment Description combination. You must run this program if you have run Modify Schema and/or Create Environment Description.

See “Using the SUPRA Server bootstrap schema” on page 487 for a list of the bootstrap schemas provided with SUPRA. It is possible to have many sets of bootstrap environments with the same Directory files, and you can define many bootstrap Environment Descriptions to use the same bootstrap schema. However, only one set can be accessed during an execution of the PDM. The DIRECTORY parameter in the CSIPARM file is used to pass the names of the bootstrap schema and Environment Description to the PDM. SUPRA Server ensures the integrity of physical data by requiring that the bootstrap environment be consistent (the three bootstrap modules are consistent with each other). This task is handled by the validation module.

When you modify the bootstrap schema or the bootstrap Environment Descriptions, the validation module monitors the consistency of the bootstrap environment by placing a time and date stamp in the resulting bootstrap schema and Environment Description. The validation module saves a list of the time and date stamps, and the PDM verifies the time and date stamps in the bootstrap schema and Environment Description against the list in the validation module. If the time and date stamps do not match, the PDM abends.

NOTE

To avoid an abend, you need to run the bootstrap utility programs in the proper order.

If you are changing the bootstrap schema, do the following:

1. Run the Modify Schema program.
2. Run the Create Environment Description program for any bootstrap Environment Description you are changing or adding.
3. Run the Create Validation Module program.

If you are not changing the bootstrap schema, do the following:

1. Run the Create Environment Description program for each bootstrap Environment Description you are changing or adding.
2. Run the Create Validation Module program.

NOTE

You need to run the Create Validation Module program even if you do not modify its input. If you do not run the Create Validation Module program when required, or if you do not execute the steps in order, the PDM will abend.

NOTE

Warning: If your PDM abends, do not execute the Modify Schema, Create Environment Description, or Create Validation Module programs. This changes your environment, and you will be unable to warm start your PDM.

Modifying the bootstrap schema

You modify the bootstrap schema by running the Modify Schema program when you want to change any of the following:

- Physical description of files
- Block sizes of files
- Physical sizes of the files
- Device types
- Access methods
- ddnames

Enter the input to Modify Schema in the following order:

```
SCHEMA statement
File-definition group(s)
END-SCHEMA: statement
```

Within a file definition group, the first statement must be a FILE statement. The physical description statements (ACCESS-METHOD, OLDFILE, DDNAME, UNIT, TOTAL-LOGICAL-RECORDS, BLOCKSIZE, CONTROL-INTERVAL-SIZE, and LOGICAL-BLOCKS-PER-TRACK) may be in any order, but they must be terminated by the END-FILE: statement. All statements have to begin in column 1 of the 80-byte record. You can include comment statements by coding a blank or asterisk (*) in column 1 of the 80-byte record.

The following are various operating system considerations for running the Modify Schema utility program:

- **OS/390** To run Modify Schema in OS/390, use the input member MODSC*xxx* (where *xxx* is 375 for 3375s, 380 for 3380s, or 390 for 3390s), the procedure TISBUSCH, and the sample JCL member TXJBMODS. When running Modify Schema, you should be aware of the location of the CSTASCHM module you are modifying and which library the modified schema will be linked to.

- **VSE** To run Modify Schema in VSE, use the procedure TISBUS*xx* (where *xx* is 30, 40, 50, 75, 80, 90 or VS, depending on the device type or access method your installation uses for the Directory files) and the sample JCL member TXJBMODS. When running Modify Schema, you should be aware of the location of the CSTASCHM module you are modifying and which library the modified schema will be linked to.

Input specifications for the Modify Schema program

This section describes some input specification and considerations for coding the Modify Schema program. This includes input specifications for the Directory files, the Task Log file, the System Log file and the Statistics file.

The following table shows the Modify Schema statements you use for a BDAM or ESDS Directory file when you code an OLDFILE statement. Using the OLDFILE statement indicates you are not changing the access method. See the table labeled “Input specifications for the Task Log File (changing access method)” later in this section for the Modify Schema statements you use to change the access method.

Input specifications for the directory files (using the OLDFILE statement)

Statement	BDAM	ESDS
OLDFILE= (*) OLDFILE= (DDNAME/DLBLNAME)	SUPPLIED	SUPPLIED
CONTROL-INTERVAL-SIZE	INVALID	OPTIONAL
DDNAME/DTFNAME	INVALID	INVALID
LOGICAL-BLOCKS-PER-TRACK	INVALID	INVALID
TOTAL-LOGICAL-RECORDS	OPTIONAL	OPTIONAL
UNIT/DEVICE	INVALID	INVALID

NOTE

OPTIONAL means that if you do not supply a value, Modify Schema uses the value from the file in the Schema being modified. For more information, refer to “OLDFILE statement” on page 181.

The following table lists the Modify Schema statements you use to change the access method without using the OLDFILE statement:

Input specifications for the directory files (not using the OLDFILE statement)

Statement	BDAM to ESDS	BDAM to KSDS	ESDS to BDAM	KSDS to BDAM	ESDS to KSDS	KSDS to ESDS
ACCESS-METHOD	ESDS	KSDS	BDAM	BDAM	KSDS	ESDS
CONTROL-INTERVAL SIZE	REQUIRED	INVALID	INVALID	INVALID	INVALID	REQUIRED
DDNAME / DTFNAME	OPTIONAL	OPTIONAL	OPTIONAL	OPTIONAL	OPTIONAL	OPTIONAL
LOGICAL-BLOCKS-PER-TRACK	INVALID	INVALID	OPTIONAL DEFAULT=2	OPTIONAL DEFAULT=2	INVALID	INVALID
TOTAL-LOGICAL-RECORDS	OPTIONAL	INVALID	OPTIONAL	REQUIRED	INVALID	REQUIRED
UNIT/DEVICE	INVALID	INVALID*	INVALID*	INVALID*	INVALID*	INVALID*

* Primary files only

NOTE

Using the ACCESS-METHOD statement and not using the OLDFILE statement indicates you are changing the access method for the file. OPTIONAL means that if you do not supply a value, Modify Schema uses the value from the file in the Schema being modified (except where noted in the table).

The following table lists the Modify Schema statements you code when you are not changing the access method of the Task Log File:

Input specifications for the Task Log File (not changing access method)

Statement	BDAM	ESDS
OLDFILE	INVALID	INVALID
ACCESS-METHOD	OPTIONAL	OPTIONAL
BLOCKSIZE	OPTIONAL*	OPTIONAL
CONTROL-INTERVAL SIZE	INVALID	OPTIONAL
DDNAME/DTFNAME	OPTIONAL	OPTIONAL
LOGICAL-BLOCKS-PER-TRACK	OPTIONAL	INVALID
TOTAL-LOGICAL-RECORDS	OPTIONAL	OPTIONAL
UNIT/DEVICE	OPTIONAL*	INVALID

* You should supply a UNIT statement and not supply a BLOCKSIZE statement so you will not have to be aware of device capacities. A UNIT statement is required if you do not supply a BLOCKSIZE statement. The minimum blocksize is 512.

NOTE

OPTIONAL indicates that the utility uses the current value, if present.

The following table lists the Modify Schema statements you use to change the access method of the Task Log File:

Input specifications for the Task Log File (changing access method)

Statement	from BDAM to ESDS	from ESDS to BDAM
OLDFILE	INVALID	INVALID
ACCESS-METHOD	ESDS	BDAM
BLOCKSIZE	OPTIONAL-DEFAULTS TO CURRENT	OPTIONAL-DEFAULTS TO CURRENT
CONTROL-INTERVAL SIZE	REQUIRED	INVALID
DDNAME/DTFNAME	OPTIONAL-DEFAULTS TO CURRENT	OPTIONAL-DEFAULTS TO CURRENT
LOGICAL-BLOCKS-PER-TRACK	INVALID	OPTIONAL DEFAULT=2
TOTAL-LOGICAL-RECORDS	OPTIONAL-DEFAULTS TO CURRENT	OPTIONAL-DEFAULTS TO CURRENT
UNIT/DEVICE	INVALID	OPTIONAL IF BLOCKSIZE IS SUPPLIED. REQUIRED IF NOT SUPPLIED.

NOTE

The Task Log File cannot be a KSDS or BSAM File. Supply a UNIT statement but not a BLOCKSIZE statement so you do not have to know device capacities.

The following table lists the Modify Schema statements you code when you are not changing the access method of the System Log File:

Input for the System Log File (not changing access method)

Statement	BDAM	ESDS	BSAM
OLDFILE	INVALID	INVALID	INVALID
ACCESS-METHOD	OPTIONAL	OPTIONAL	OPTIONAL
BLOCKSIZE	OPTIONAL*	OPTIONAL	OPTIONAL
CONTROL-INTERVAL SIZE	INVALID	OPTIONAL	INVALID
DDNAME/DTFNAME	OPTIONAL	OPTIONAL	OPTIONAL
DEVADDR=SYS*xxx*	INVALID	INVALID	INVALID
LOGICAL-BLOCK-PER-TRACK	OPTIONAL	INVALID	INVALID
TOTAL-LOGICAL-RECORDS	OPTIONAL	OPTIONAL	INVALID
UNIT/DEVICE	OPTIONAL*	INVALID	INVALID

* You should supply a UNIT statement and not supply a BLOCKSIZE statement so you will not have to know device capacities. A UNIT statement is required if a BLOCKSIZE statement is not supplied.

NOTE

OPTIONAL indicates that the utility uses the current value, if present. The System Log File cannot be a KSDS File. DEVADDR=SYS*xxx* is valid only when you specify TAPE on the UNIT or DEVICE statement (VSE only).

The following table lists the Modify Schema statements you code when you are changing the access method of the System Log File:

Input specifications for the System Log File (changing access method)

Statement	BDAM to ESDS	BDAM to BSAM	BSAM to BDAM	BSAM to ESDS	ESDS to BDAM	ESDS to BSAM
OLDFILE	INVALID	INVALID	INVALID	INVALID	INVALID	INVALID
ACCESS-METHOD	ESDS	BSAM	BSAM	ESDS	BSAM	BSAM
BLOCKSIZE	OPTIONAL	OPTIONAL	OPTIONAL**	OPTIONAL	OPTIONAL**	OPTIONAL
CONTROL-INTERVAL SIZE	REQUIRED	INVALID	INVALID	REQUIRED	INVALID	REQUIRED
DDNAME / DTFNAME	OPTIONAL; DEFAULTS TO CURRENT	OPTIONAL; DEFAULTS TO CURRENT	OPTIONAL; DEFAULTS TO CURRENT	OPTIONAL; DEFAULTS TO CURRENT	OPTIONAL; DEFAULTS TO CURRENT	OPTIONAL; DEFAULTS TO CURRENT
LOGICAL-BLOCKS-PER-TRACK	INVALID	INVALID	OPTIONAL; DEFAULT=2	INVALID	OPTIONAL; DEFAULT=2	INVALID
TOTAL-LOGICAL-RECORDS	OPTIONAL; DEFAULTS TO CURRENT	INVALID	RECOM MENDED=2	RECOM MENDED=2	OPTIONAL; DEFAULTS TO CURRENT	INVALID
UNIT/DEVICE	INVALID	INVALID	OPTIONAL**	INVALID	OPTIONAL**	INVALID

* If you omit the TOTAL-LOGICAL-RECORDS statement when you change a BSAM System Log File to BDAM or ESDS, the allocation requirement for the new System Log File is 2,147,483,647 records.

** You should supply a UNIT statement and not a BLOCKSIZE statement so you do not have to know device capacities. A UNIT statement is required if a BLOCKSIZE statement is not supplied.

NOTE

OPTIONAL indicates that the utility will use the current value, if present. The System Log File cannot be a KSDS file. DEVADDR=SYS*xxx* is valid only when you specify TAPE on the UNIT or DEVICE statement (VSE only).

The following table lists the Modify Schema statements you code for the Statistics File. Because the Statistics File must be a BSAM file, you cannot change the access method.

Statement	Coding requirement
OLDFILE	INVALID
ACCESS-METHOD	BSAM
BLOCKSIZE*	OPTIONAL
CONTROL-INTERVAL SIZE	INVALID
DDNAME/DTFNAME	OPTIONAL-DEFAULTS TO CURRENT
DEVADDR=SYS*xxx*	INVALID
LOGICAL-BLOCK-PER-TRACK	INVALID
TOTAL-LOGICAL-RECORDS	INVALID
UNIT/DEVICE	INVALID

* If you omit the BLOCKSIZE statement, the system uses the current value in the schema being modified.

NOTE

The OLDFILE statement is invalid for the Statistics File. The Statistics File cannot be a BDAM, ESDS, or KSDS File. The Statistics File cannot be TAPE under VSE.

The Statistics File described in the schema on your installation tape is coded as follows:

FILE=STAT	
BLOCKSIZE=1024	(*or block size you supplied to the Technical Service Center (TSC) before installation*)
ACCESS-METHOD=BSAM	
DDNAME=C$TSTAT	(*or ddname you supplied to the TSC before installation*)
END-FILE:	

BSAM is the only valid access method for the Statistics File. Refer to the *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260, or the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261, for considerations on the size of the statistics record.

The following table lists all of the Modify Schema statements, with brief descriptions and section numbers where each is discussed in the text:

Statement	Description	Section
ACCESS-METHOD	Names the access method you want to use.	"ACCESS-METHOD statement" on page 173
BLOCKSIZE	Specifies block size of the System Log, Task Log, or Statistics File.	"BLOCKSIZE statement" on page 174
CONTROL-INTERVAL-SIZE	Specifies the VSAM control interval size of an ESDS VSAM file.	"CONTROL-INTERVAL-SIZE statement" on page 175
DDNAME	OS/390 Specifies the ddname which defines the file to the PDM.	"DDNAME statement" on page 176
DEVADDR	VSE only. Specifies the SYSNO of a tape file defined to Modify Schema.	"DEVADDR statement" on page 176
DEVICE	VSE Specifies device type of the file.	"DEVICE statement" on page 177
DTFNAME	Specifies the DLBL which defines the file to the PDM (VSE).	"DTFNAME statement" on page 178
END-FILE:	Indicates the end of the file input for the current file.	"END-FILE: statement" on page 178
END-SCHEMA:	Indicates the end of input to Modify Schema.	"END-SCHEMA: statement" on page 178
FILE	Names the file whose physical characteristics you want to modify.	"FILE statement" on page 179
LOGICAL-BLOCKS-PER-TRACK	Specifies the number of blocks per track on a BDAM file.	"LOGICAL-BLOCKS-PER-TRACK statement" on page 180
OLDFILE	OS/390 VSE Indicates that a particular file exists on your system.	"OLDFILE statement" on page 181
SCHEMA	Names the schema you want to modify.	"SCHEMA statement" on page 182
TOTAL-LOGICAL-RECORDS	Specifies the total logical records for a BDAM file or ESDS VSAM file.	"TOTAL-LOGICAL-RECORDS statement" on page 183
OS/390 UNIT	Specifies the ddname of a DD statement OS/390. The DD statement names the type of unit on which to allocate a file.	"UNIT statement (OS/390)" on page 184
VSE UNIT	Specifies file's device type.	"UNIT statement (VSE)" on page 185

ACCESS-METHOD statement

The ACCESS-METHOD statement identifies the access method to use.

$$\left\{\begin{array}{l}\textbf{ACCESS}\\ \textbf{ACCESS– METHOD}\end{array}\right\} = \textbf{\textit{access – method}}$$

access-method

Description *Optional*. Identifies the access method.

Default See options.

Options Valid access methods depend on the file type. If you specify an invalid access method, the Modify Schema program generates a message and uses the default.

File type	Valid access method options	Default, if error
Primary File	KSDS, ESDS, BDAM	BDAM
Related File	ESDS, BDAM	BDAM
System Log File	BDAM, ESDS, BSAM, WORK, OUTP	BSAM
Task Log File	ESDS, BDAM	BDAM
Statistics File	BSAM, WORK, OUTP	BSAM

General considerations

- If you change the access method of the System Log File from BSAM to BDAM, supply the TOTAL-LOGICAL-RECORDS statement. If you omit the TOTAL-LOGICAL-RECORDS statement, the allocation requirement for a BDAM System Log File may be 2,147,483,647 records.
- WORK and OUTP are identical to BSAM except for DOS/VSE tape System Log Files. For those files, WORK specifies a file whose DTF specifies TYPEFLE=OUTPUT. WORK allows the PDM to overlap I/O and PDM processing. The PDM simulates multivolume support by closing the file, issuing volume-change messages to the operator, and reopening the file. OUTP uses the system’s real multivolume support and does not overlap I/O.

BLOCKSIZE statement

The BLOCKSIZE statement specifies the block size of the System Log, the Task Log, or the Statistics File.

$$\left\{\begin{matrix}\textbf{BKSZ}\\ \textbf{BLOCKSIZE}\end{matrix}\right\} = \textbf{\textit{blocksize}}$$

blocksize

Description *Optional*. Specifies the block size of the System Log File, the Task Log File, or the Statistics File.

Default If you omit this statement, the existing block size in the bootstrap Directory schema is used.

Options 1–2,147,483,647

General considerations

- The BLOCKSIZE statement is not valid for primary, related, or index files.
- Modify Schema uses this statement to set the block size. If you use this statement, DO NOT use a value that exceeds the capacity of the device. Unless you have a reason for changing the block size in the bootstrap Directory Schema (supplied on the installation tape), you should either omit this statement or allow the block size to be set as follows:
  - For BDAM files either allow the block size to default, or code the UNIT statement and the BLOCKSIZE statement using the block size value you specify.
  - For ESDS files, use the CONTROL-INTERVAL-SIZE statement.
- If you change the unit and you do not specify the block size, the program will use the block size already contained in the Schema being modified. If you change the block size, check the appropriate IBM Reference Summary to make sure the new block size is optimum for the device being used.
- You need to specify a statistics log block size as large as the largest statistics record; otherwise, the DBA Execution Statistics utility will not process the file.
- When you omit this statement and use the UNIT statement, use the block size calculated by Modify Schema to allocate this file.
- 512K is the minimum block size for the Task Log File.

CONTROL-INTERVAL-SIZE statement

The CONTROL-INTERVAL-SIZE statement specifies the VSAM control interval size of an ESDS.

$$\left\{\begin{array}{l}\textbf{CVCI}\\ \textbf{CONTROL}-\textbf{INTERVAL}-\textbf{SIZE}\end{array}\right\}=\textbf{\textit{control}}-\textbf{int}\textbf{\textit{erval}}-\textbf{\textit{size}}$$

control-interval-size

Description *Optional*. Specifies the VSAM control-interval-size in bytes of an ESDS.

Default If you omit this statement, the existing control interval size (if present) in the Directory bootstrap schema is used. If the control interval size is not present, an error message is displayed.

Options

512–4096	Must be an integral multiple of 512
4096–32,768	Must be an integral multiple of 2048

General consideration

The CONTROL-INTERVAL-SIZE statement is invalid for BDAM, KSDS, and BSAM access methods.

DDNAME statement

The DDNAME statement specifies the ddname that defines the file to the PDM. The DDNAME statement is valid for OS/390.

DDNAME=*ddname*.

***ddname*.**

Description *Optional*. Specifies the ddname that defines the file to the PDM.

Default If you omit this statement, the existing ddname for the file is used.

Format 1–8 characters. First character must be alphabetic, $, #, or @. Remaining characters must be alphanumeric, $, #, or @.

General consideration

If you code a DDNAME statement, you must omit the OLDFILE statement.

DEVADDR statement

The DEVADDR statement specifies the SYSNO associated with a tape file defined to Modify Schema. The DEVADDR statement is valid only for VSE.

DEVADDR=SYS*xxx*.

SYS*xxx*.

Description *Conditional*. Required for tape. Optional and not used for disk. Indicates a valid SYSNO for the tape file.

Options 000–253

Consideration You should use the DEVADDR statement only for a tape file. The DEVICE statement determines the device type (TAPE). If you specify DEVADDR for disk, the value you enter is edited and then discarded.

General consideration

To use DEVADDR properly, you should specify DEVICE=TAPE and ACCESS-METHOD=BSAM.

DEVICE statement

The DEVICE statement specifies the device type of the file. The DEVICE statement is valid only for VSE.

$$\textbf{DEVICE} = \left[\frac{\textbf{3330}}{\textbf{xxxx}}\right]$$

$$\left[\frac{\textbf{3330}}{\textbf{xxxx}}\right]$$

Description *Optional*. Indicates the device type associated with this file.

Default 3330

Options TAPE, 2314, 3330, 3340, 3344, 3350, 3375, 3380, 3390, or 9345

Considerations

- This statement affects the calculation of the number of records in the file for BDAM files.
- The device you enter affects the total logical records calculated.
- This statement is invalid for ESDSs, KSDSs, and BSAM disk files.

General considerations

- The TAPE option is valid only for BSAM files.
- VSE You must not use the TAPE option for your Statistics file.
- VSE To specify the device type of a file, you can use the DEVICE statement or the UNIT statement (see “UNIT statement (OS/390)” on page 184).

DTFNAME statement

The DTFNAME statement specifies the DLBL that defines the file to the PDM. The DTFNAME statement is valid only for VSE.

DTFNAME=*dlblname*

dlblname

Description *Optional*. Specifies the DLBL name that defines the file to the PDM.

Default If you omit this statement, the existing DLBL name for this file in the bootstrap schema is used.

Format 1–7 characters. First character must be alphabetic, $, #, or @. Remaining characters must be alphanumeric, $, #, or @.

General consideration

If you code a DTFNAME statement, omit the OLDFILE statement.

END-FILE: statement

The END-FILE: statement indicates the end of the file input for the current file.

END-FILE:

General consideration

This statement is required.

END-SCHEMA: statement

The END-SCHEMA: statement indicates the end of the input to Modify Schema.

END-SCHEMA:

General consideration

This statement is for documentation purposes only. This statement causes no processing.

FILE statement

The FILE statement names the file whose physical characteristics you want to modify.

FILE=*file-name*

file-name

Description *Required*. Defines the name of the file whose physical characteristics you want to modify.

Options The valid options are listed under File Name:

File name	DDNAME/ DLBL name	Description	File type
C$-#	C$TDEF#	Definition Number File	Primary
C$-N	C$TNAME	Name File	Primary
C$-T	C$TTEXT	Text File	Related
C$-S	C$TSTRU	Structure File	Related
C$-D	C$TDATA	Data File	Related
SLOG	C$TSLOG	PDM System Log File	System log
TLOG	C$TTLOG	PDM Task Log File	Task log
STAT	C$TSTAT	PDM Statistics File	Statistics

Consideration These DDNAMEs or DLBL names exist on the installation tape. However, you can change them using Modify Schema.

LOGICAL-BLOCKS-PER-TRACK statement

The LOGICAL-BLOCKS-PER-TRACK statement specifies the number of blocks per track on a BDAM File.

$$\left\{\begin{array}{l}\textbf{BTRK}\\ \textbf{LOGICAL-BLOCKS-PER-TRACK}\end{array}\right\} = \textbf{\textit{blocks-per-track}}$$

blocks-per-track

Description *Conditional*. Specifies the number of blocks per track on a BDAM file. Required if you change the access method to BDAM. Optional if you do not change to BDAM.

Default If you omit this statement and an existing blocks per track value is present in the bootstrap schema, the existing blocks per track value is used. If an existing blocks per track is not present, 2 is used.

Options 1–2,147,483,647

General considerations

- The LOGICAL-BLOCKS-PER-TRACK statement is invalid for BSAM files, ESDS, and KSDS.
- If you code a LOGICAL-BLOCKS-PER-TRACK statement, omit the OLDFILE statement.
- If you run Modify Schema against a BDAM file and you do not change the access method, the logical blocks per track is retrieved from the original file (in the bootstrap schema being modified). You should supply this statement if you want to change the original blocking factor.

OLDFILE statement

The OLDFILE statement indicates that a particular file exists on your system. Modify Schema accesses this File to retrieve the block size for a BDAM file or the control interval size for an ESDS.

$$\textbf{OLDFILE} = \begin{bmatrix} \textbf{(*)} \\ \textbf{(ddname)} \\ \textbf{(dlblname)} \end{bmatrix}$$

$$\begin{bmatrix} \textbf{(*)} \\ \textbf{(ddname)} \\ \textbf{(dlblname)} \end{bmatrix}$$

Description *Optional*. Specifies the ddname or DLBL name of the file from which to retrieve either the block size for a BDAM file (OS/390 only) or control interval size for an ESDS.

Default If you omit this statement and the DDNAME and DTFNAME statements, the existing ddname or DLBL name in the bootstrap Directory Schema is used.

Format 1–8 characters for ddname. 1–7 characters for dlblname. First character must be alphabetic, $, #, or @. Remaining characters must be alphanumeric, $, #, or @.

Considerations

- Obtain the block size or control interval size by accessing the file to which the ddname or DLBL name on the OLDFILE statement points.
- OLDFILE=(*) sets the ddname or DLBL name to the old ddname or DLBL name in the bootstrap Directory Schema. See “FILE statement” on page 205 for FILE statement options.
- Include the ddname or DLBL name referred to on the OLDFILE statement in the JCL when you execute Modify Schema.
- VSE You cannot use the OLDFILE statement for BDAM files.

General considerations

- The OLDFILE statement indicates that you are not changing the access method for the file.
- The OLDFILE statement is invalid for the System Log File, Task Log File, Statistics File, BSAM Files, and KSDSs.
- If you code an OLDFILE statement, omit all of the following statements: DDNAME, DTFNAME, UNIT, and LOGICAL-BLOCKS-PER-TRACK.

SCHEMA statement

The SCHEMA statement names the Schema you want to modify.

SCHEMA = [**CSTASCHM** / ***schema – id***]

[**CSTASCHM** / ***schema – id***]

Description *Optional*. Defines the name of the schema you want to modify.

Default CSTASCHM

Format 1–8 characters. First character must be alphabetic. Remaining characters must be alphanumeric, $, #, or @.

General considerations

- The Modify Schema input statements used to create the bootstrap schema supplied in the installation tape (CSTASCHM) are supplied in the MACLIB member. The MACLIB member MODSC*xxx* was used for your installation where *xxx* equals 375 for 3375s, 380 for 3380s, or 390 for 3390s.
- If the STEPLIB (OS/390), or core-image library (VSE) does not contain the Schema during an execution of Modify Schema, the Modify Schema program generates a fatal error message.

TOTAL-LOGICAL-RECORDS statement

The TOTAL-LOGICAL-RECORDS statement specifies the total logical records for a BDAM file or an ESDS.

$$\left\{\begin{array}{l}\textbf{RFIL}\\ \textbf{TOTAL}-\textbf{LOGICAL}-\textbf{RECORDS}\end{array}\right\} = \textit{total} - \textit{logical} - \textit{records}$$

total-logical-records

Description *Optional*. Specifies the total logical records.

Options 1–2,147,483,647

General considerations

- If you do not supply this statement for a BSAM file, the total logical records are set to the highest positive number contained in a full word.
- The TOTAL-LOGICAL-RECORDS statement is invalid for a KSDS file.
- Supply this statement when you change a BSAM System Log File to BDAM. If you omit this statement, the allocation requirement for a BDAM System Log File is 2,147,483,647 records.
- If the file you modify already has a value for total logical records, you do *not* need to supply this statement. For example, if you are converting a BDAM file to an ESDS and the total logical records previously specified for the BDAM file is correct, omit this statement.

UNIT statement (OS/390)

The UNIT statement specifies the ddname of a DD statement. The DD statement names the type of unit on which to allocate a BDAM file. The UNIT statement has different meanings for OS/390 and VSE. For VSE, see “UNIT statement (VSE)” on page 185.

UNIT=*ddname*

ddname

Description *Optional*. Indicates the ddname of the DD statement that defines the desired type of unit.

Format 1–8 characters. First character must be alphabetic, $, #, or @. Remaining characters must be alphanumeric, $, #, or @.

Considerations

- The DD statement in your JCL stream must contain UNIT and SPACE parameters. For example:

```
//#3350 DD UNIT=3350,SPACE=(TRK,1)
```

- In an OS/390 environment, you may need to code VOL=SER=volser on the DD statement, where *volser* represents a volume label, to allow the utility to locate a device type.

General considerations

- The UNIT statement is invalid for ESDSs, KSDSs, and BSAM disk files.
- If you code a UNIT statement, you must omit the OLDFILE statement.
- Use a UNIT statement when modifying a Task Log File, System Log File or Statistics File. Allow the block size to default to the operating system page size. See “BLOCKSIZE statement” on page 174 for considerations for the BLOCKSIZE statement. Use the block size calculated by Modify Schema to allocate these files.

UNIT statement (VSE)

The UNIT statement specifies the device type of the file. The UNIT statement has different meanings for OS/390 and VSE. For OS/390, see "UNIT statement (OS/390)" on page 184.

UNIT = [**3330** / ***xxxx***]

[**3330** / ***xxxx***]

Description *Optional*. Indicates the device type associated with this file.

Default 3330

Options TAPE, 2314, 3330, 3340, 3344, 3350, 3375, 3380, 3390, or 9345

Consideration This statement affects the calculation of the number of records in the file for BDAM files. The device you enter affects the total logical records calculated.

General considerations

- The TAPE option is valid only for BSAM files.
- VSE You must not use the TAPE option for your Statistics File.
- This statement is equivalent to the DEVICE statement.
- If you code a UNIT statement, you must omit the OLDFILE statement.
- This statement is invalid for ESDSs, KSDSs, and BSAM disk files.

Modify Schema example

The following is an example of a Modify Schema program-execution listing:

```
CINCOM SYSTEMS, INC. SUPRA MODIFY SCHEMA   16:27:44  TUESDAY SEPTEMBER 01, 1997  PAGE 1
                                0001000
* MODIFY SCHEMA STATEMENTS FOR SUPRA 2.6 BOOT SCHEMA  0002000
* FOR 3380S                     0002100
*                               0003000
SCHEMA=CSTASCHM                 0004000
FILE=C$-N               NAME FILE          0005000
ACCESS-METHOD=BDAM              0006000
UNIT=#UNIT1                     0007000
DDNAME=C$TNAME                  0008000
LOGICAL-BLOCKS-PER-TRACK=10         0009000
TOTAL-LOGICAL-RECORDS=18360         0011000
END-FILE:                       0013000

                        GIVEN   CALCULATED
  ACCESS-METHOD         BDAM    BDAM
  TOTAL-LOGICAL-RECORDS 18360   18360
  LOGICAL-RECORD-LENGTH 0   78  *
  LOGICAL-BLOCKS-PER-TRACK  10  10
  BLOCK-SIZE            0   4212    *
  LOGICAL-RECORDS-PER-BLOCK 0   54  *
  DDNAME                            C$TNAME

*                                   0014000
FILE=C$-#               DEF# FILE              0015000
ACCESS-METHOD=BDAM                  0016000
UNIT=#UNIT1                         0017000
DDNAME=C$TDEF#                      0018000
LOGICAL-BLOCKS-PER-TRACK=10             0019000
TOTAL-LOGICAL-RECORDS=18260             0021000
END-FILE:                           0023000

                        GIVEN   CALCULATED
  ACCESS-METHOD         BDAM    BDAM
  TOTAL-LOGICAL-RECORDS 18260   18260
  LOGICAL-RECORD-LENGTH 0   374 *
```

```
  LOGICAL-BLOCKS-PER-TRACK  10  10
  BLOCK-SIZE            0   4114    *
  LOGICAL-RECORDS-PER-BLOCK 0   11  *
  DDNAME                        C$TDEF#

*                                   0024000
FILE=C$-D               DATA FILE               0025000
ACCESS-METHOD=BDAM                  0026000
UNIT=#UNIT1                         0027000
DDNAME=C$TDATA                      0028000
LOGICAL-BLOCKS-PER-TRACK=10             0029000
TOTAL-LOGICAL-RECORDS=19823             0031000
END-FILE:                           0033000

                        GIVEN   CALCULATED
  ACCESS-METHOD         BDAM    BDAM
  TOTAL-LOGICAL-RECORDS 19823   20210   *
  LOGICAL-RECORD-LENGTH 0   98  *
  LOGICAL-BLOCKS-PERTRACK   10  10
  BLOCK-SIZE            0   4214    *
  LOGICAL-RECORDS-PER-BLOCK 0   43  *
  DDNAME                        C$TDATA
 *                              0034000
FILE=C$-T               TEXT FILE               0035000
ACCESS-METHOD=BDAM                  0036000
UNIT=#UNIT1                         0037000
DDNAME=C$TTEXT                      0038000
LOGICAL-BLOCKS-PER-TRACK=10             0039000
TOTAL-LOGICAL-RECORDS=14960             0041000
END-FILE:                           0043000
                        GIVEN   CALCULATED
  ACCESS-METHOD         BDAM    BDAM
  TOTAL-LOGICAL-RECORDS 14960   14960
  LOGICAL-RECORD-LENGTH 0   96  *
  LOGICAL-BLOCKS-PER-TRACK  10  10
  BLOCK-SIZE            0   4224    *
  LOGICAL-RECORDS-PER-BLOCK 0   44  *
  DDNAME                    C$TTEXT

*                                   0044000
FILE=C$-S               STRU FILE               0045000
```

```
ACCESS-METHOD=BDAM                  0046000
UNIT=#UNIT1                         0047000
DDNAME=C$TSTRU                      0048000
LOGICAL-BLOCKS-PER-TRACK=10             0049000
TOTAL-LOGICAL-RECORDS=27880             0051000
END-FILE:                           0053000

                       GIVEN   CALCULATED
  ACCESS-METHOD        BDAM    BDAM
  TOTAL-LOGICAL-RECORDS 27880   27880
  LOGICAL-RECORD-LENGTH 0   52  *
  LOGICAL-BLOCKS-PER-TRACK  10  10
  BLOCK-SIZE           0   4264    *
  LOGICAL-RECORDS-PER-BLOCK 0   82  *
  DDNAME               0   C$TSTRU

*
*   TASK LOG FILE
*                                   0064200
FILE=TLOG              TLOG FILE              0064300
ACCESS-METHOD=BDAM                  0064400
DDNAME=C$TTLOG                      0064500
BLOCKSIZE=8192                      0064600
LOGICAL-BLOCKS-PER-TRACK=5              0064700
TOTAL-LOGICAL-RECORDS=900               0064800
END-FILE:                           0064900
      CSTG091I - NOTE    LOGICAL RECORD LENGTH ASSUMED TO BE BLOCKSIZE FOR BDAM SLOG OR
  TLOG FILE.

                       GIVEN   CALCULATED
  ACCESS-METHOD        BDAM    BDAM
  TOTAL-LOGICAL-RECORDS 900 900
  LOGICAL-BLOCKS-PER-TRACK  5   5
  BLOCK-SIZE           0   8192    *
  DDNAME                   C$TTLOG

*                                   0064910
END-SCHEMA:                         0065000

  0 WARNINGS
```

Modifying the bootstrap Environment Description

To modify the bootstrap Environment Description, run the Create Environment Description program:

- Add new execution parameters
- Replace existing execution parameters

The input statements for Create Environment Description are presented in alphabetical sequence for easy reference. However, you can code the statements in any order. All statements must begin in column 1 of the 80-byte record. You can include comment statements by coding a blank or asterisk (*) in column 1 of the 80-byte record. The next table in this section lists the Create Environment Description statements with brief descriptions and section references. These statements are optional. Include only those statements appropriate to the environment you are creating:

OS/390

To run Create Environment Description in OS/390, use the procedure TISBUEVD and the sample JCL member TXJBENVD. The bootstrap Environment Descriptions supplied on the installation tape are named CSTANONE, CSTAOLDM, CSTAREAD, CSTASUPD, and CSTATLOG. These Environment Descriptions are provided, already link edited, in the SUPRA Server LINKLIB. The source input to Create Environment Description is supplied on the SUPRA Server MACLIB.

VSE

To run Create Environment Description in VSE, use the procedure TISBUEVx,

where:
x is 1 (CSTANONE)
2 (CSTATLOG)
3 (CSTAREAD)
4 (CSTASUPD)
5 (CSTAOLDM)

and the sample JCL member TXJBENVD. The bootstrap Environment Descriptions supplied on the installation tape are named CSTANONE, CSTAREAD, CSTASUPD, and CSTATLOG. The bootstrap Environment Descriptions are link-edited into the corelib.

The various operating system considerations for running the Create Environment Description utility program are listed below:

Statement	Description	Section
ACCESS-MODE	Indicates if database updates and recovery functions are permitted.	"ACCESS-MODE statement" on page 195
AFTER	Indicates if after images are to be logged to the System Log File.	"AFTER statement" on page 196
BEFORE	Indicates if before images are to be logged to the System Log File.	"BEFORE statement" on page 197
BUFFER	Defines the following information about buffering: ♦ A symbolic name for the buffer pool ♦ The number of buffers in the pool used for direct processing	"BUFFER statement" on page 198
CONNECTED-INTERFACES	Indicates the maximum number of connected interfaces permitted during execution with this Environment Description.	"CONNECTED-INTERFACES statement" on page 199
CONNECTED-THREADS	Indicates the maximum number of connected threads permitted during execution with this Environment Description.	"CONNECTED-THREADS statement" on page 200
CONSOLE	Indicates if optional console messages are to be generated during execution with this Environment Description.	"CONSOLE statement" on page 201
END-LOG-OPTIONS	Indicates how to handle the System Log File during an ENDLG command.	"END-LOG-OPTIONS statement" on page 202
ENVDESC	Defines the name of the bootstrap Directory Environment Description to be created and used by the PDM.	"ENVDESC statement" on page 203

Statement	Description	Section
EXITS	Enables and names PDM exits, as follows: ♦ CHECK. Invoke an exit before checking the result of I/O on the System Log File. ♦ CLOSE. Invoke an exit before closing a primary, related, or System Log File. ♦ COMMAND-INIT. Invoke an exit before accessing any file during processing of each command. ♦ OPEN. Invoke an exit before opening a primary, related, or System Log File. ♦ READ. Invoke an exit before reading a physical record of a primary or related file. ♦ WRITE. Invoke an exit before writing a physical record of a primary, related, or System Log File.	"EXITS statement" on page 204
FILE	Defines the following information about each file in the bootstrap Directory Schema: ♦ Whether to permit access to the file ♦ Which buffer pool to use for the file ♦ How to open and whether to open the file during system initialization	"FILE statement" on page 205
FUNCTION	Indicates whether to log commands and whether to log all functions or only update functions in the System Log File.	"FUNCTION statement" on page 207
HELD-RECORDS	Indicates the maximum number of records to monitor, lock, or hold.	"HELD-RECORDS statement" on page 208

Statement	Description	Section
MEMORY	Specifies the amount of memory to allocate as work space by and for the PDM.	"MEMORY statement" on page 209
OPENX-OPTION	Indicates whether to process, check, or ignore OPENX and CLOSX commands.	"OPENX-OPTION statement" on page 211
PASSWORDS	Indicates whether the PDM requires a password for termination, and if so, identifies the password.	"PASSWORDS statement" on page 212
SCHEMA	Defines the bootstrap Directory Schema associated with this Environment Description.	"SCHEMA statement" on page 213
SIGN-ON	Indicates whether to log SINON and SINOF commands to the System Log File.	"SIGN-ON statement" on page 214
SIGNED-ON-TASKS	Indicates the maximum number of tasks that can sign on at one time.	"SIGNED-ON-TASKS statement" on page 215
STATISTICS	Indicates whether to accumulate statistics.	"STATISTICS statement" on page 217
TASK-LOGGING	Indicates whether to use task logging.	"TASK-LOGGING statement" on page 218

Before you code any Create Environment Description statements, consider the following points, which describe how to derive the various Environment Description options when you use both the REALM and DIRECTORY parameters of the CSIPARM File. For more information on the CSIPARM File, see “Initializing the PDM” on page 63.

- The REALM parameter defines the User Schema and Environment Description.
- The DIRECTORY parameter defines the bootstrap Directory Schema and Environment Description.
- When creating any bootstrap Directory Environment Description, always code FILE statements for Directory files.
- When you create a bootstrap Directory Environment Description for running the DBA utilities, you must specify the following:
  - File open mode of NONE
  - Access mode of UPDATE
  - OPENX option of PROCESS
- Once you initialize the system, the PDM obtains the following options from the Environment Description named in the REALM parameter. The options defined in the user Environment Description override the options you defined in the bootstrap Directory Environment Description. Two exceptions (memory and task logging) are discussed in the remainder of these considerations.

Maximum connected threads	Function logging
Maximum connected interfaces	End log option
Statistics option	Access mode
Console option	Command initialization
Shutdown password	Exit name
Maximum signed on tasks	Open exit name
Maximum held records	Read exit name
SINON logging	Write exit name
Before logging	Check exit name
After logging	Close exit name
Directory open mode	OPENX option

- The PDM obtains the task logging option from the Environment Description in the DIRECTORY parameter of the CSIPARM file. If the task logging option in the DIRECTORY and REALM parameters are not the same, the PDM abends. For further information, see the following third point.

- The memory allocations in both the DIRECTORY and REALM Environment Description parameters create two separate memory pools. See “MEMORY statement” on page 209 for more information on the MEMORY statement.

- If you are using task logging and a forced termination or system failure occurs, the next time you execute TIS/XA, you must use the same Task Log File and PDM name used when the forced termination or failure occurred. The system then initializes the files, buffers, and so on, according to the information contained in the task log system file table. Refer to the *SUPRA Server PDM Logging and Recovery Guide (OS/390 & VSE)*, P26-2223, for information on recovery.
- If you define a Directory file or Task Log File in both the DIRECTORY and REALM parameters on the CSIPARM file, the following statements are true:

  - The physical file description and buffering for these files in the REALM schema is ignored.

  - The open modes for the Directory files are obtained from the Directory open mode in the User Environment Description defined in the REALM parameter.

  - The buffer names for the Directory files are obtained from the Environment Description defined in the REALM parameter.

- If you defined the System Log File or the Statistics File in both the DIRECTORY and REALM parameters on the CSIPARM file, the physical description and buffering in the Directory schema is ignored.

- To define a Directory-driven environment, you must name both the REALM parameter (User Schema and Environment Description) and the DIRECTORY parameter (bootstrap schema and Environment Description) in the CSIPARM file.

ACCESS-MODE statement

The ACCESS-MODE statement indicates whether to permit database updates.

$$\left\{\begin{matrix}\textbf{ACCESS}\\ \textbf{ACCESS-MODE}\end{matrix}\right\}=\left[\begin{matrix}\textbf{RDONLY}\\ \underline{\textbf{UPDATE}}\end{matrix}\right]\textbf{ACCESS=RDONLY}$$

$$\left[\begin{matrix}\textbf{RDONLY}\\ \underline{\textbf{UPDATE}}\end{matrix}\right]$$

Description *Optional*. Indicates whether to permit database updates.

Default UPDATE

Options

RDONLY Allows only DML READ functions.

UPDATE Allows UPDATE and READ functions.

Considerations

- When you create an Environment Description for running the DBA utilities, code ACCESS-MODE=UPDATE.
- When running Directory-driven (DIRECTORY and REALM parameters in CSIPARM file), the access mode in the Environment Description in the DIRECTORY parameter is ignored.

AFTER statement

The AFTER statement indicates whether to log after images to the System Log file.

$$\begin{Bmatrix} \textbf{ALOG} \\ \textbf{AFTER} \end{Bmatrix} = \begin{bmatrix} \textbf{NO} \\ \underline{\textbf{YES}} \end{bmatrix}$$

$$\begin{bmatrix} \textbf{NO} \\ \underline{\textbf{YES}} \end{bmatrix}$$

Description *Optional*. Indicates whether to log after images to the System Log File.

Default YES

Options NO Do not log after images to the System Log File.

YES Log after images to the System Log File.

Considerations

- If you code AFTER=YES, the following considerations apply:
  - Before adding a record to a KSDS, a before image record is added to the System Log File.
  - After adding a record to a KSDS, an after image record is added to the System Log File.
  - Before deleting a record from a KSDS, a before image record is added to the System Log File.
  - After deleting a record from a KSDS, an after image record is added to the System Log File.
- When running Directory-driven (DIRECTORY and REALM parameters in CSIPARM file), the AFTER LOGGING option in the Environment Description in the DIRECTORY parameter is ignored.

BEFORE statement

The BEFORE statement indicates whether to log before images to the System Log File.

$$\begin{Bmatrix}\textbf{BLOG}\\ \textbf{BEFORE}\end{Bmatrix} = \begin{bmatrix}\textbf{NO}\\ \underline{\textbf{YES}}\end{bmatrix}$$

$$\begin{bmatrix}\textbf{NO}\\ \underline{\textbf{YES}}\end{bmatrix}$$

Description *Optional*. Indicates whether to log before images to the System Log File.

Default YES

Options NO Do not log before images to the System Log File.

YES Log before images to the System Log File.

Considerations

- If you code BEFORE=YES, certain actions are taken before updating a database record:
  - If task logging is active and this is the first time the task has updated this particular record because it last committed or signed on, a before image record is added to the System Log File.
  - If task logging is not active, a before image record is added to the System Log File.
- If you code BEFORE=YES, the following considerations apply:
  - Before adding a record to a KSDS, a before image record is added to the System Log File.
  - After adding a record to a KSDS, an after image record is added to the System Log File.
  - Before deleting a record from a KSDS, a before image record is added to the System Log File.
  - After deleting a record from a KSDS, an after image record is added to the System Log File.
- When running Directory-driven (DIRECTORY and REALM parameters in CSIPARM file), the BEFORE LOGING option in the Environment Description in the DIRECTORY parameter is ignored.

BUFFER statement

The BUFFER statement defines information about buffering.

BUFFER= *buffer-name*[=(*buffer-allocation*)]

buffer-name

Description *Required*. Defines a symbolic name for the buffer pool.

Format 4 alphanumeric characters

Consideration Task log, system log, and statistics log files may share a Buffer Pool. However, the Directory files cannot share the Buffer Pool used for the task log, system log, or statistics log file.

(*buffer-allocation*)

Description *Optional*. Defines the number of buffers to be allocated.

Format

(d)
(d,t,b)
(t,b)

where:

d = Number of direct buffers
1–10 numeric characters; default = 1 direct buffer

t = Number of serial threads
1–10 numeric characters; default = 0 sequential threads

b = Number of serial buffers
1–10 numeric characters; default = 0 sequential buffers

Considerations

- For KSDS, the number of direct buffers (d) sets the *STRNO* (*string number*) parameter when the file is opened. Do not use *t* and *b* for KSDS. For more information on the string number when used with the define cluster operation, refer to the appropriate IBM Access Method Services manual.
- Serial processing (also known as read ahead processing) is not available in this release. However, if you supply values for sequential threads and sequential buffers, the values will be combined with the value for direct buffers using the following formula:

```
direct buffers = d + (t * b)
```

- Buffer pools used in the DIRECTORY (bootstrap) Environment Description and the REALM statement Environment Description must not have the same name.

CONNECTED-INTERFACES statement

The CONNECTED-INTERFACES statement indicates the maximum number of connected interfaces permitted during execution with this Environment Description.

$$\begin{Bmatrix} \textbf{MXIF} \\ \textbf{CONNECTED– INTERFACE} \end{Bmatrix} = \left[\frac{\textbf{5}}{\textbf{\textit{maximum – interfaces}}} \right]$$

$$\left[\frac{\textbf{5}}{\textbf{\textit{maximum – interfaces}}} \right]$$

Description *Optional*. Indicates the maximum number of connected interfaces.

Default 5

Options 1–2,147,483,647

General considerations

- When executing an application program in the attached operating mode, maximum connected interfaces must be two or greater (one for the application program and one for the Attacher). If only one interface is allowed and the application does not sign off, the Attacher gets a TFUL status when it tries to terminate the PDM using an ENDTO.
- Make this parameter large enough to handle all possible applications' interfaces and CICS interfaces.
- When running Directory-driven (DIRECTORY and REALM parameters in the CSIPARM file), the connected interfaces value in the Environment Description in the DIRECTORY parameter is ignored.

CONNECTED-THREADS statement

The CONNECTED-THREADS statement indicates the maximum number of connected threads permitted during execution with this Environment Description.

$$\begin{Bmatrix} \textbf{\textit{MXTH}} \\ \textbf{\textit{CONNECTED – THREADS}} \end{Bmatrix} = \left[\frac{\textbf{10}}{\textbf{\textit{maximum – connected – threads}}} \right]$$

$$\left[\frac{\textbf{10}}{\textbf{\textit{maximum – connected – threads}}} \right]$$

Description *Optional*. Indicates the maximum number of connected threads.

Default 10

Options 2-2,147,483,647

Considerations

- If the maximum connected threads is less than the maximum number of connected interfaces, the Create Environment Description program displays a fatal error message.
- The PDM allocates memory for the number of threads you specify. If you specify more threads than signed-on tasks, the PDM never uses the memory allocated for the extra threads. Furthermore, this memory is unavailable for other uses for the duration of the run.
- Make this parameter large enough to allow for threads associated with tasks communicating through CICS.
- When running Directory-driven (DIRECTORY and REALM parameters in the CSIPARM file), the connected threads value in the Environment Description in the DIRECTORY parameter is ignored.

CONSOLE statement

The CONSOLE statement indicates whether to generate optional console messages during execution with this Environment Description.

$$\left\{\begin{matrix}\textbf{CONS}\\ \textbf{CONSOLE}\end{matrix}\right\} = \left[\begin{matrix}\textbf{NO}\\ \underline{\textbf{YES}}\end{matrix}\right]$$

$$\left[\begin{matrix}\textbf{NO}\\ \underline{\textbf{YES}}\end{matrix}\right]$$

Description *Optional*. Indicates whether to generate optional console messages.

Default YES

Options NO The PDM does *not* generate console messages.

YES The PDM generates console messages.

General considerations

- The PDM generates some console messages (all abend messages) regardless of how you set this option. If CONSOLE=YES, the PDM produces additional console messages.
- For explanations of PDM messages, refer to the *SUPRA Server PDM Messages and Codes Reference Manual (RDM/PDM Support for OS/390 & VSE)*, P26-0126.
- If you start the central PDM with the TIS/XA Software Selection Facility, it issues a VM READ when you terminate the PDM. If the primary user responds to this, the Software Selection Facility issues full screen 3270 writes, which the Single Console Image Facility does not send to the primary user. To respond to the VM READ, the secondary user should log on again. For more information on the Single Console Image Facility, refer to the *IBM VM/SP System Programmer's Guide*, SC19-6224-5.
- When running Directory-driven (DIRECTORY and REALM parameters in the CSIPARM file), the console option in the Environment Description in the DIRECTORY parameter is ignored.

END-LOG-OPTIONS statement

The END-LOG-OPTIONS statement indicates how to handle the System Log File during an ENDLG (End Log) command.

$$\left\{\begin{matrix}\textbf{ELOG}\\ \textbf{END}-\textbf{LOG}-\textbf{OPTIONS}\end{matrix}\right\} = \left[\begin{matrix}\textbf{FEOV}\\ \hline \textbf{CLOSE}\end{matrix}\right]$$

$$\left[\begin{matrix}\textbf{FEOV}\\ \hline \textbf{CLOSE}\end{matrix}\right]$$

Description *Optional*. Indicates how to handle the System Log File during an ENDLG (End Log) command.

Default FEOV

Options

FEOV	ENDLG issues an FEOV (Force End of Volume) macro, whose effects depend on the operating system and the device type of the System Log File.
CLOSE	The PDM closes and then opens the same System Log File and reinitializes the file. This option might create a System Log File that cannot be used for Recovery and Restore.

Considerations

- For a detailed explanation of the FEOV macro, refer to the IBM manual *OS/VS2 OS/390 Data Management Macro Instructions,* GC26-3873, or *DOS/VS Supervisor and I/O Macros*, GC33-5373.
- When running Directory-driven (DIRECTORY and REALM parameters in the CSIPARM file), the END-LOG option in the Environment Description in the DIRECTORY parameter is ignored.

ENVDESC statement

The ENVDESC statement defines the name of the bootstrap Environment Description the PDM creates and uses.

$$\textbf{ENVDESC} = \left[\frac{\textbf{CSTAENVD}}{\textbf{\textit{env – desc – id}}}\right]$$

$$\left[\frac{\textbf{CSTAENVD}}{\textbf{\textit{env – desc – id}}}\right]$$

Description *Optional*. Defines the name of the bootstrap Environment Description the PDM creates.

Default CSTAENVD

Format 1–8 characters. First character must be alphabetic. Remaining characters must be alphanumeric, $, #, or @.

General considerations

- Five bootstrap Environment Descriptions are on the installation tape.
  - **OS/390** In OS/390, the SUPRA Server MACLIB contains the Create Environment Description statements used to generate these Environment Descriptions.
  - **VSE** They are contained in the VSE procedures TISBUEVx.
- The five Environment Descriptions and their generalized purposes are:
  - CSTANONE—Directory files open modes = NONE.
  - CSTAREAD—Directory files open modes = READ.
  - CSTASUPD—Directory files open modes = SUPD.
  - CSTATLOG—Directory files open modes = SUPD. Also enables task logging.
  - CSTAOLDM—Intended for certain situations in which you may want to perform Online Directory Maintenance in bootstrap mode.
- The input statements for these modules contain further documentation.

EXITS statement

The EXITS statement names and enables PDM exits. For more information on PDM exits, see “Using PDM exits” on page 249.

EXITS= (*exit-type1=program1*,[*exit-type2=program2*...])

(*exit-type1=program1*,[*exit-type2=program2*...])

Description *Optional*. Defines the PDM exits and the name of the program containing the exit.

Format 1–8 characters. First character must be alphabetic. Remaining characters must be alphanumeric, $, #, or @.

Options The following exit types are possible:

CHECK	Invoke an exit before checking the result of I/O on the System Log File.
CLOSE	Invoke an exit before closing a primary, related, or System Log File.
COMMAND-INIT	Invoke an exit before accessing any file during processing of each command.
OPEN	Invoke an exit before opening a primary, related, or System Log File.
READ	Invoke an exit before reading a physical record of a primary or related file.
WRITE	Invoke an exit before writing a physical record to a primary, related, or System Log File.

General considerations

- If you code more than one exit, code a comma after each one except the last.
- You may code more than one EXITS statement, each with one or more exits.
- When running Directory-driven (DIRECTORY and REALM parameters in the CSIPARM file), the exits defined in the bootstrap Environment Description (DIRECTORY in the CSIPARM file) are ignored.
- If you are using the optional Buffer Cache facility, specific entry-point names are required. See “Using PDM exits” on page 249 for more information.

FILE statement

The FILE statement defines information about each file in the bootstrap schema. To access a file in a schema using the bootstrap Environment Description, you must have a FILE statement.

$$\mathbf{FILE} = \left\{ \begin{array}{l} \textit{file-name}, \textit{buffer-name} \\ \textit{file-name}, \textit{buffer-name}[, \textit{file-open-mode}] \end{array} \right\}$$

file-name

Description *Required*. Defines information about each file in the bootstrap schema.

Options

C$-#	Directory Definition Number file
C$-N	Directory Name file
C$-T	Directory Text file
C$-S	Directory Structure file
C$-D	Directory Data file
SLOG	PDM System Log file
TLOG	PDM Task Log file
STAT	PDM Statistics file

buffer-name

Description *Required*. Defines the buffer pool used for this file.

Format 4 characters. First character must be alphabetic. Remaining characters must be alphanumeric, $, #, or @.

Considerations

- Task Log, System Log, and Statistics Log Files may share a Buffer Pool. However, the Directory files cannot share the Buffer Pool used for the Task Log, System Log, or Statistics Log File.
- For any buffer name in a FILE statement, there must be a corresponding BUFFER statement for that buffername (see “BUFFER statement” on page 198).

file-open-mode

Description *Optional*. Defines how the file is to be opened during system initialization.

Options

SUPD	File opened for shared update
READ	File opened for read only
NONE	File not opened

Considerations

- If you do not code this parameter, either the definition on the Directory for user files or in the Environment Description for Directory files determines the default.
- When you create an Environment Description for running the DBA utilities, code a file open mode of NONE. The utilities open the files; therefore, you should not automatically open them at system initialization.
- The open mode for an index file must be NONE. The index file is opened to the highest open mode of any related data file. (SUPD is the highest mode, and NONE is the lowest.)

General considerations

- If you omit a FILE statement for a file in the bootstrap schema, the PDM cannot access that file using this bootstrap Environment Description.
- If you do not define the file in the bootstrap schema, the PDM generates a fatal error message.

FUNCTION statement

The FUNCTION statement indicates whether to log no commands, all commands, or only update commands to the System Log File.

$$\left\{\begin{matrix}\textbf{FLOG}\\ \textbf{FUNCTION}\end{matrix}\right\} = \left[\begin{matrix}\textbf{ALL}\\ \textbf{NONE}\\ \underline{\textbf{UPDATE}}\end{matrix}\right]$$

$$\left[\begin{matrix}\textbf{ALL}\\ \textbf{NONE}\\ \underline{\textbf{UPDATE}}\end{matrix}\right]$$

Description *Optional*. Indicates which commands or functions to log.

Default UPDATE

Options

ALL	Log all functions
NONE	Log no functions
UPDATE	Log only update functions (ADD-M)

General considerations

- If you use any logging options and a CLOSX, COMIT, ENDLG, FREEX, MARKL, OPENX, QMARK, QUIET, or RESET command executes successfully, a Command Log record is added to the System Log File.
- If you choose FUNCTION=ALL and a FINDX, RDNXT, READD, READM, READR, READV, or RQLOC command executes successfully, the PDM adds a command log record to the System Log File.
- If you choose FUNCTION=UPDATE or FUNCTION=ALL and an ADD-M, ADDVA, ADDVB, ADDVC, ADDVR, DEL-M, DELVD, WRITD, WRITM, or WRITV command executes successfully, the PDM adds a Command Log record to the System Log File.
- When running Directory-driven (DIRECTORY and REALM parameters in the CSIPARM file), the function logging option in the Environment Description in the DIRECTORY parameter is ignored.

HELD-RECORDS statement

The HELD-RECORDS statement indicates the maximum number of records to monitor, lock, or hold.

$$\left\{\begin{matrix}\textbf{HREC} \\ \textbf{HELD}-\textbf{RECORDS}\end{matrix}\right\} = \textbf{\textit{maximum}} - \textbf{\textit{held}} - \textbf{\textit{records}}$$

maximum-held-records

Description *Required*. Indicates the maximum number of records to monitor, lock, or hold.

Options 1–2,147,483,647

Consideration If you specify 0, the PDM generates a fatal error message.

General considerations

- In a single-task environment where task logging is not active, record holding is not active. However, structural maintenance of update commands requires record holding entries.
- For a non-Directory-driven, nontask-logging environment, set this statement to at least 15 times the number of tasks.
- For a non-Directory-driven, task logging environment, set this statement based on the maximum transaction needs.
- When running Directory-driven (DIRECTORY and REALM parameters in the CSIPARM file), the number of held records in the Environment Description in the DIRECTORY parameter is ignored.

MEMORY statement

The MEMORY statement specifies the amount of memory for the PDM to allocate as work space.

$$\textbf{MEMORY} = \left\{\begin{matrix} \textbf{\textit{numberK}} \\ \textbf{\textit{numberM}} \\ \textbf{\textit{number.digitM}} \end{matrix}\right\}$$

$$\left\{\begin{matrix} \textbf{\textit{numberK}} \\ \textbf{\textit{numberM}} \\ \textbf{\textit{number.digitM}} \end{matrix}\right\}$$

Description *Required*. Specifies the amount of memory the PDM allocates as work space.

Format 1–10 numeric characters for number

1–9 numeric characters for digit

Considerations

- If you code K, memory is set to 1024 times the number.
- If you code M, memory is set to 1,048,576 times the number.
- If you code M, one digit may follow the decimal point.

General considerations

- This memory is aligned on a page boundary.
- In the user Environment Description, this parameter is the PDM Storage Allocation. Memory from the bootstrap Environment Description is for one pool of storage (Directory memory pool); PDM storage allocation value is used for a second pool of storage. The storage pools are where the PDM defines its internal control blocks. Memory from the bootstrap Environment Description is added to PDM Storage Allocation from the user Environment Description.
- Refer to the *SUPRA Server PDM Tuning Guide (OS/390 & VSE)*, P26-0225, to calculate the memory requirements for the Environment Description (for the Directory memory pool) based on entities you define in the bootstrap schema and Environment Description. Calculate the memory requirements for the memory pool as though all entities for the Directory are allocated from one pool.

NOTE

Warning: Do *not* try to calculate too closely. These calculations have been figured loosely to allow for variable environments, and so on. Cincom suggests you add 2–3% of the calculated amount for total memory. This additional memory will allow for dynamic DML memory allocation, and will minimize memory fragmentation problems.

OPENX-OPTION statement

The OPENX-OPTION statement indicates whether to process, check, or ignore OPENX and CLOSX commands.

$$\textbf{OPENX} - \textbf{OPTION} = \begin{bmatrix} \textbf{IGNORE} \\ \textbf{CHECK} \\ \underline{\textbf{PROCESS}} \end{bmatrix}$$

$$\begin{bmatrix} \textbf{IGNORE} \\ \textbf{CHECK} \\ \underline{\textbf{PROCESS}} \end{bmatrix}$$

Description *Optional*. Indicates whether to process, check, or ignore OPENX and CLOSX commands.

Default PROCESS

Options

IGNORE	Ignore OPENX and CLOSX commands.
CHECK	Check OPENX and CLOSX commands against the current file mode list. (Does not open or close files.)
PROCESS	Process OPENX and CLOSX commands.

General considerations

- When you create an Environment Description for running the utilities, code OPENX-OPTION=PROCESS.
- The combination of CLOSX and OPENX is not allowed in the same logical unit of work (LUW). For example, if an application issues an OPENX or CLOSX, it must issue a COMIT before issuing another OPENX or CLOSX on the same file.
- For information on using the OPENX and CLOSX commands and their processing, refer to the *SUPRA Server PDM DML Programming Guide (OS/390 & VSE)*, P26-4340.
- When running Directory-driven (DIRECTORY and REALM parameters in the CSIPARM file), the OPENX option in the Environment Description in the DIRECTORY parameter is ignored.

PASSWORDS statement

The PASSWORDS statement indicates whether the PDM requires a password for termination and, if so, identifies the password.

PASSWORDS=(SHUTDOWN(shutdown-password))

shutdown-password

Description *Optional*. Specifies the termination password, if required.

Format 1–8 alphanumeric characters, $, #, or @

General considerations

- If you omit this statement, the PDM issues a message indicating that it does not require a password for termination.
- As a security feature, the output listing does not display the PASSWORDS statement.
- When running Directory-driven, the Environment Description in the DIRECTORY parameter is ignored. The shutdown password (if any) is taken from the User Environment Description named in the REALM statement in the CSIPARM file.

SCHEMA statement

The SCHEMA statement defines the bootstrap schema associated with the bootstrap Environment Description.

$$\textbf{SCHEMA} = \left[\frac{\textbf{CSTASCHM}}{\textbf{\textit{schema – id}}}\right]$$

$$\left[\frac{\textbf{CSTASCHM}}{\textbf{\textit{schema – id}}}\right]$$

Description *Optional*. Defines the name of the bootstrap schema associated with the bootstrap Environment Description.

Default CSTASCHM

Format 1–8 characters. First character must be alphabetic. Remaining characters must be alphanumeric, $, #, or @.

General consideration

If the STEPLIB (OS/390), or core-image library (VSE), does not contain the schema during execution of Create Environment Description, the Create Environment Description program generates a fatal error.

SIGN-ON statement

The SIGN-ON statement indicates whether to log SINON and SINOF commands to the System Log File.

$$\begin{Bmatrix} \textbf{SLOG} \\ \textbf{SIGN-ON} \end{Bmatrix} = \begin{bmatrix} \textbf{NO} \\ \textbf{ALL} \\ \underline{\textbf{UPDATE}} \end{bmatrix}$$

$$\begin{bmatrix} \textbf{NO} \\ \textbf{ALL} \\ \underline{\textbf{UPDATE}} \end{bmatrix}$$

Description *Optional*. Indicates whether to log SINON and SINOF commands.

Default UPDATE

Options

NO	Do not log SINON and SINOF commands.
ALL	Log all SINON and SINOF commands.
UPDATE	Log all SINON and SINOF commands if ACCESS-MODE=UPDATE.

Considerations

- If SIGN-ON=UPDATE, SUPRA Server logs SINON commands at sign-on, instead of when the first UPDATE command is processed.
- If you specify TASK-LOGGING=YES and AFTER=YES and/or BEFORE=YES, SIGN-ON is set to ALL.
- When running Directory-driven (DIRECTORY and REALM parameters in the CSIPARM file), the sign-on logging option in the Environment Description in the DIRECTORY parameter is ignored.

SIGNED-ON-TASKS statement

The SIGNED-ON-TASKS statement indicates the maximum number of tasks that can sign on at one time.

$$\left\{\begin{array}{l}\textbf{SGTK}\\ \textbf{SIGNED– ON– TASKS}\end{array}\right\} = \textbf{\textit{signed – on – tasks}}$$

signed-on-tasks

Description *Optional*. Indicates the maximum number of tasks that can sign on at one time.

Options 1–2,147,483,647

General considerations

- If you omit this statement, the PDM issues a warning message. The maximum signed-on-tasks is set to the maximum connected-threads (10 or the value specified in your input stream).
- Make this parameter large enough to allow for tasks communicating through CICS, as well as any batch tasks.
- In cases where this Environment Description is used by a PDM servicing a CICS system(s), you must reserve one task per CICS system for use by the CICS Connector, so that the Connector can process a Disconnect function even if all other tasks are in use.
- In this situation, you will need an Environment Description that specifies at least:
  - Connected Interfaces=10
  - Connected-Threads=10
  - Signed-on-Tasks=15

  During processing of the CICS Connect request, one of the interfaces is reserved for this CICS, two of the connected-threads are reserved for this CICS, five of the signed-on-tasks are reserved for your PDM requests, and one additional signed-on-task is reserved for the CICS Disconnect task. Therefore, when the Connect command completes, there are nine interfaces still available for other uses (the batch tasks in the original assumption above), nine threads left (one for each interface), and nine tasks (one for each interface).

- In this situation, if you set the signed-on-tasks value to anything less than 15, you will receive an IPAR status, and the Connect request will fail. Alternatively, you could decrease the number of connected-threads to avoid the IPAR. In any event, this example demonstrates the interaction of these three parameters in setting up an Environment Description that will be usable in a CICS environment.

NOTE

Even though the formula in the following bullet is correct, if you do not consider the requirements of a CICS system, you may receive the IPAR error.

If you are using a Directory-driven environment (DIRECTORY and REALM parameters in the CSIPARM file), this consideration applies to the Active Environment Description you specify in the REALM parameter.

Use the following formula when you set values for interfaces, threads, and tasks:

```
MXIF  ≤  MXTH  ≤  SGTK
```

where:

MXIF = Maximum Connected Interfaces

MXTH = Maximum Connected Threads

SGTK = Maximum Signed-on Tasks

- When running Directory-driven (DIRECTORY and REALM parameters in the CSIPARM file), the signed-on tasks value in the Environment Description in the DIRECTORY parameter is ignored.

STATISTICS statement

The STATISTICS statement indicates whether to accumulate statistics.

$$\left\{\begin{matrix}\textbf{STOP}\\ \textbf{STATISTICS}\end{matrix}\right\} = \left[\begin{matrix}\textbf{NO}\\ \hline \textbf{YES}\end{matrix}\right]$$

$$\left[\begin{matrix}\textbf{NO}\\ \hline \textbf{YES}\end{matrix}\right]$$

Description *Optional*. Indicates whether to accumulate statistics.

Default NO

General considerations

- If you code STATISTICS=YES, code a FILE statement describing the Statistics File. Otherwise, the PDM issues a warning message.
- The options from the User Environment Description override the options from the bootstrap Environment Description.
- For information on the Execution Statistics utility, refer to the *SUPRA Server PDM DBA Utilities User's Guide (OS/390 & VSE)*, P26-6260.
- When running Directory-driven (DIRECTORY and REALM parameters in the CSIPARM file), the Statistics option in the Environment Description in the DIRECTORY parameter is ignored.
- Statistics are written to the Statistics Log File when an RSTAT DML command is issued and during PDM initialization and termination. For additional information, refer to the *SUPRA Server PDM DML Programming Guide (OS/390 & VSE)*, P26-4340.
- If you do not accumulate statistics, the data returned by the SHOWX DML command for some data items may not be valid. For additional information, refer to the *SUPRA Server PDM DML Programming Supplement (OS/390 & VSE)*, P26-4340.

TASK-LOGGING statement

The TASK-LOGGING statement indicates whether to use task logging.

$$\left\{\begin{matrix}\textbf{TLOG}\\ \textbf{TASK}-\textbf{LOGGING}\end{matrix}\right\}=\left[\begin{matrix}\textbf{NO}\\ \underline{\textbf{YES}}\end{matrix}\right]$$

$$\left[\begin{matrix}\textbf{NO}\\ \underline{\textbf{YES}}\end{matrix}\right]$$

Description *Optional*. Indicates whether to use task logging.

Default YES

Considerations

- If you code TASK-LOGGING=YES, AFTER=YES, and SIGN-ON=NO or UPDATE, the PDM issues a warning message and sets SIGN-ON to ALL.
- If you code TASK-LOGGING=YES, BEFORE=YES, and SIGN-ON=NO or UPDATE, the PDM issues a warning message and sets SIGN-ON to ALL.
- If you code TASK-LOGGING=YES, the PDM writes the following types of records to the Task Log File:
  - Record before images
  - File before images (For example, if a task opens a file, the file before image shows that this file is to be closed. Thus, if the task issues a RESET, the file is restored to its original closed state.)
  - Commit records for each task

Create Environment Description example

```
CINCOM SYSTEMS, INC. SUPRA ENVIRONMENT DESCRIPTION GENERATOR16:33:31  TUESDAY SEPTEMBER 01, 1997  PAGE1

**STARTMODDB*********************************************************** 0000010
*                                                                      * 0000020
* PROCEDURE NAME        :: CSTANONE - BOOTSTRAP ENVDESC                * 0000030
*                                                                      * 0000040
* CODING CONVENTIONS    :: DESCRIPTOR GENERATOR INPUT STATEMENTS       * 0000050
*                                                                      * 0000060
* CALL STATEMENT        :: NONE; THIS IS AN INPUT DECK                 * 0000070
*                                                                      * 0000080
* PARAMETERS            :: NONE                                        * 0000090
*                                                                      * 0000100
* REGISTERS AT ENTRY    :: NOT APPLICABLE                              * 0000110
*                                                                      * 0000120
* REGISTERS AT EXIT     :: NOT APPLICABLE                              * 0000130
*                                                                      * 0000140
* REGISTER CONVENTIONS  :: NOT APPLICABLE                              * 0000150
*                                                                      * 0000160
* RETURN CODES          :: SEE PDM UTILITIES USER'S GUIDE              * 0000170
*                                                                      * 0000180
* FUNCTIONAL NARRATIVE  :: NONE                                        * 0000190
*                                                                      * 0000200
* OPERATION             :: GENERATES CSTANONE BOOTSTRAP ENVDESC.      * 0000210
*                                                                      * 0000220
* NOTES                 ::                                             * 0000230
*                                                                      * 0000240
* 1. THIS ENVIRONMENT DESCRIPTION IS PRIMARILY INTENDED FOR USE IN     * 0000250
*    EXECUTING THE SUPRA UTILITIES AGAINST THE DIRECTORY IN A BOOT-    * 0000260
*    STRAP-ONLY ENVIRONMENT, I.E., DO NOT SPECIFY A REALM= IN THE      * 0000270
*    CSIPARM FILE. THIS ENVDESC IS ALSO USED TO FORMAT THE TASK       * 0000280
*    LOG FILE. EXAMPLES OF HOW THIS ENVDESC CAN BE USED ARE:          * 0000290
*    UNLOADING/LOADING THE C$ND NORMAL FILE FOR MIGRATION PURPOSES,    * 0000300
*    FORMATTING FILES, RECOVER/RESTORE FUNCTIONS, AND UNLOCKING        * 0000310
*    USER FILES. PLEASE SEE THE JCL SAMPLES SUPPLIED WITH THE         * 0000320
*    SUPRA PRODUCT. THIS ENVIRONMENT DESCRIPTION CAN ALSO BE USED     * 0000330
*    IN A DIRECTORY-DRIVEN ENVIRONMENT TO EXECUTE BATCH DBAID.        * 0000340
*                                                                      * 0000350
* 2. THIS ENVIRONMENT DESCRIPTION SPECIFIES AN OPEN MODE FOR THE       * 0000360
*    DIRECTORY FILES OF 'NONE'.                                       * 0000370
*                                                                      * 0000380
* 3. THE FOLLOWING LISTS THE DEFAULTS FOR THE PARAMETERS TO THE        * 0000390
*    ENVIRONMENT DESCRIPTOR GENERATOR PROGRAM:                         * 0000400
*                                                                      * 0000410
```

```
*    ENVDESC=                NO DEFAULT - REQUIRED PARAMETER          * 0000420
*    SCHEMA=                 NO DEFAULT - REQUIRED PARAMETER          * 0000430
*    ACCESS-MODE=            DEFAULT = UPDATE                         * 0000440
*    CONSOLE=YES             DEFAULT = YES                            * 0000450
*    TASK-LOGGING=           DEFAULT = YES                            * 0000460
*    AFTER=NO                DEFAULT = YES                            * 0000470
*    BEFORE=NO               DEFAULT = YES                            * 0000480
*    FUNCTION=NONE           DEFAULT = UPDATE                         * 0000490
*    SIGN-ON=NO              DEFAULT = UPDATE                         * 0000500
*    FILE=                   NO DEFAULT - REQUIRED PARAMETER          * 0000510
*    BUFFER=                 NO DEFAULT - REQUIRED PARAMETER          * 0000520
*    HELD-RECORDS=           NO DEFAULT - REQUIRED PARAMETER          * 0000530
*    MEMORY=                 NO DEFAULT - REQUIRED PARAMETER          * 0000540
*    OPENX-OPTION=           DEFAULT = PROCESS                        * 0000550
*    CONNECTED-THREADS=      DEFAULT = 10                             * 0000560
*    CONNECTED-INTERFACES=   DEFAULT = 5                              * 0000570
*    SIGNED-ON-TASKS=        DEFAULT = CONNECTED THREADS              * 0000580
*    STATISTICS=             DEFAULT = NO                             * 0000590
*    PASSWORDS=              NO DEFAULT - OPTIONAL PARAMETER          * 0000600
*    END-LOG-OPTIONS=        DEFAULT = FEOV                           * 0000610
*    EXITS=                  NO DEFAULT - OPTIONAL PARAMETER          * 0000620
*                                                                     * 0000630
**ENDMODDB************************************************************ 0000640
*                                                                      0000650
ENVDESC=CSTANONE        THIS ENVDESC NAME                              0000660
SCHEMA=CSTASCHM         TO WORK WITH THIS SCHEMA                       0000670
********************************************************************** 0000680
ACCESS-MODE=UPDATE      DIRECTORY ACCESS MODE                          0000690
```

```
********************************************************************** 0000700
CONSOLE=YES             CONSOLE OPTION IS ENABLED                      0000710
********************************************************************** 0000720
TASK-LOGGING=NO         INDICATE TASK-LOGGING IS NOT IN EFFECT         0000730
********************************************************************** 0000740
*                       SYSTEM LOGGING OPTIONS:                        0000750
AFTER=NO                  AFTER-IMAGE LOGGING = NO                     0000760
BEFORE=NO                 BEFORE-IMAGE LOGGING = NO                    0000770
FUNCTION=NONE             FUNCTION-LOGGING = NONE                      0000780
SIGN-ON=NO                SIGN-ON-LOGGING = NO                         0000790
********************************************************************** 0000800
*                       SELECT THE DIRECTORY FILES, SPECIFY BUFFER     0000810
*                        POOL AND OPEN MODE                            0000820
FILE=C$-#,C$BP,NONE     ...                                            0000830
FILE=C$-N,C$BP,NONE     ...                                            0000840
FILE=C$-T,C$BP,NONE     ...                                            0000850
FILE=C$-S,C$BP,NONE     ...                                            0000860
FILE=C$-D,C$BP,NONE     ...                                            0000870
*                                                                      0000880
FILE=TLOG,C$BL,SUPD     SELECT A TASK LOG FILE, SPECIFY BUFFER         0000890
*                        POOL AND OPEN MODE                            0000900
********************************************************************** 0000910
BUFFER=C$BP=(8,0,0)     DEFINE THE BUFFERS                             0000920
BUFFER=C$BL=(2,0,0)     ...                                            0000930
********************************************************************** 0000940
HELD-RECORDS=100        # OF HELD RECORD ENTRIES                       0000950
********************************************************************** 0000960
MEMORY=400K             MEMORY                                         0000970
OPENX-OPTION=PROCESS    INDICATES TO PROCESS OPENX/CLOSX COMMANDS      0000980
********************************************************************** 0000990
*          SPECIFY THREADS/INTERFACES, ETC. FOR ENVIRONMENT.           0001000
CONNECTED-THREADS=1                                                    0001010
CONNECTED-INTERFACES=1                                                 0001020
SIGNED-ON-TASKS=2                                                      0001030
```

```
CSTG052I - NOTE       STATISTICS OPTION NOT RECEIVED AS INPUT. DEFAULT IS NO.
CSTG042I - NOTE       SHUTDOWN PASSWORD NOT RECEIVED AS INPUT. NO DEFAULT SUPPLIED.
CSTG033I - NOTE       END LOG OPTIONS NOT RECEIVED AS INPUT. DEFAULT IS FEOV
CSTG028I - NOTE       COMMAND INITIALIZATION EXIT NOT RECEIVED AS INPUT. NO DEFAULT
  SUPPLIED.
CSTG026I - NOTE       OPEN EXIT NOT RECEIVED AS INPUT. NO DEFAULT SUPPLIED.
CSTG023I - NOTE       READ EXIT NOT RECEIVED AS INPUT. NO DEFAULT SUPPLIED.
CSTG005I - NOTE       WRITE EXIT NOT RECEIVED AS INPUT. NO DEFAULT SUPPLIED.
CSTG004I - NOTE       CHECK EXIT NOT RECEIVED AS INPUT. NO DEFAULT SUPPLIED.
CSTG003I - NOTE       CLOSE EXIT NOT RECEIVED AS INPUT. NO DEFAULT SUPPLIED.
CSTG010I - NOTE       ENVDESC BUFFER FOUND FOR FILE - C$-#.
CSTG010I - NOTE       ENVDESC BUFFER FOUND FOR FILE - C$-N.
CSTG010I - NOTE       ENVDESC BUFFER FOUND FOR FILE - C$-T.
CSTG010I - NOTE       ENVDESC BUFFER FOUND FOR FILE - C$-S.
CSTG010I - NOTE       ENVDESC BUFFER FOUND FOR FILE - C$-D.
CSTG010I - NOTE       ENVDESC BUFFER FOUND FOR FILE - TLOG.
CSTG008I - NOTE       SCHEMA FILE FOUND FOR FILE - C$-#.
CSTG008I - NOTE       SCHEMA FILE FOUND FOR FILE - C$-N.
CSTG008I - NOTE       SCHEMA FILE FOUND FOR FILE - C$-T.
CSTG008I - NOTE       SCHEMA FILE FOUND FOR FILE - C$-S.
CSTG008I - NOTE       SCHEMA FILE FOUND FOR FILE - C$-D.
CSTG008I - NOTE       SCHEMA FILE FOUND FOR FILE - TLOG.

0 WARNINGS
```

Creating the bootstrap Validation Module

The consistency of the data stored in the bootstrap environment is ensured by the validation module. Running the Create Validation Module program verifies that any bootstrap Environment Descriptions that point to a particular bootstrap Directory schema are valid.

Each time you modify the bootstrap environment by running the Modify Schema or the Create Environment Description programs, you must run the Create Validation Module program.

The input to Create Validation Module defines the Schema and environment Description pair(s) you want to verify. The program performs the following tasks:

- Produces a list of valid schema(s) and Environment Description(s) in a form suitable for use by the PDM. This list is a validation module.
- Revalidates the schema and its associated Environment Description (the Create Validation Module program verifies that the file exists in the schema).

If an error occurs during the execution of Create Validation Module, correct the errors and regenerate only the flagged Environment Description(s).

This chapter presents the statements in the order in which you must code them. Code a SCHEMA statement followed by one or more ENVDESC statements. You can also have one or more groups of SCHEMA and ENVDESC statements. When you run Create Validation Module, you can include comment statements by coding a blank (b) or asterisk (*) in column 1 of the 80-byte record.

Operating system considerations for running Create Validation Module are listed below:

OS/390

To run Create Validation Module in OS/390, use the input member CSTAVMOD (on MACLIB), the procedure TISBUVAL, and the sample JCL member TXJBVMOD.

VSE

To run Create Validation Module in VSE, use the procedure TISBUVAL and the sample JCL member TXJBVMOD.

SCHEMA statement

The SCHEMA statement names the bootstrap schema associated with the following Environment Descriptions.

SCHEMA=*schema-name*

schema-name

Description *Required*. Names the bootstrap schema associated with the following Environment Descriptions.

Format 1–8 alphanumeric characters

Consideration The Create Validation Module program generates a fatal error message if the schema is not present in the steplib (OS/390), or core-image library (VSE).

General consideration

You must code the SCHEMA statement before the corresponding ENVDESC statement(s).

ENVDESC statement

The ENVDESC statement names the bootstrap Environment Descriptions you want to validate.

ENVDESC=*env-desc-name-list*

env-desc-name-list

Description *Required*. Names the bootstrap Environment Descriptions you want to validate.

Format One or more names, each 1–8 alphanumeric characters, each followed by a comma except the last.

Considerations

- For each Environment Description, the Create Validation Module program generates a fatal error message if:
  - A file is missing from the schema.
  - The Environment Description does not include the buffer for a file.
  - You enable task logging, but the schema contains no Task Log File definition.
- For each environment Description, the Create Validation Module program generates a warning message if:
  - You enable statistics gathering, but the schema contains no Statistics File definition.
  - You enable system logging, but the schema contains no System Log File definition.
- The Create Validation Module program generates a fatal error message if the schema is not present in the steplib (OS/390), or core-image library (VSE).

General consideration

You must code the ENVDESC statement(s) after the corresponding SCHEMA statement.

Create Validation Module example

```
CINCOM SYSTEMS, INC. SUPRA VALMOD GENERATOR    16:41:34    TUESDAY, SEPTEMBER 01, 1997  PAGE 1

**STARTMODDB************************************************************ 0000010
*                                                                      * 0000020
* PROCEDURE NAME        :: CSTAVMOD - VALIDATION MODULE                * 0000030
*                                                                      * 0000040
* CODING CONVENTIONS    :: DESCRIPTOR GENERATOR INPUT STATEMENTS       * 0000050
*                                                                      * 0000060
* CALL STATEMENT        :: NONE; THIS IS AN INPUT DECK                 * 0000070
*                                                                      * 0000080
* PARAMETERS            :: NONE                                        * 0000090
*                                                                      * 0000100
* REGISTERS AT ENTRY    :: NOT APPLICABLE                              * 0000110
*                                                                      * 0000120
* REGISTERS AT EXIT     :: NOT APPLICABLE                              * 0000130
*                                                                      * 0000140
* REGISTER CONVENTIONS  :: NOT APPLICABLE                              * 0000150
*                                                                      * 0000160
* RETURN CODES          :: SEE PDM UTILITIES USER'S GUIDE              * 0000170
*                                                                      * 0000180
* FUNCTIONAL NARRATIVE  :: NONE                                        * 0000190
*                                                                      * 0000200
* OPERATION             :: GENERATES CSTAVMOD (VALMOD)                 * 0000210
*                                                                      * 0000220
* NOTES                 ::                                             * 0000230
*                                                                      * 0000240
* 1. THE VALMOD SHOULD CONTAIN THE BOOTSTRAP SCHEMA AND ANY            * 0000250
*    ASSOCIATED ENVIRONMENT DESCRIPTIONS.                             * 0000260
*                                                                      * 0000270
* 2. ONLY TWO KEYWORD PARAMETERS ARE USED AS INPUT TO THE VALMOD       * 0000280
*    GENERATOR PROGRAM, AND THEY ARE BOTH REQUIRED PARAMETERS.        * 0000290
*                                                                      * 0000300
*    ENVDESC=                    NO DEFAULT - REQUIRED PARAMETER       * 0000310
*    SCHEMA=                     NO DEFAULT - REQUIRED PARAMETER       * 0000320
*                                                                      * 0000330
**ENDMODDB************************************************************ 0000340
*    THIS IS A LOGICAL GROUP. NAME THE BOOTSTRAP SCHEMA FOLLOWED BY    0000350
*    ITS ASSOCIATED BOOTSTRAP ENVIRONMENT DESCRIPTIONS.                0000360
SCHEMA=CSTASCHM                                                        0000370
ENVDESC=CSTANONE                                                       0000380
```

```
CSTG008I - NOTE    VALIDATION FOR SCHEMA =CSTASCHM.
CSTG009I - NOTE    ENVDESC = CSTANONE.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-#.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-N.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-T.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-S.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-D.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - TLOG.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-#.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-N.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-T.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-S.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-D.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - TLOG.
ENVDESC=CSTAOLDM                                                    0000390


CSTG008I - NOTE    VALIDATION FOR SCHEMA =CSTASCHM.
CSTG009I - NOTE    ENVDESC =CSTAOLDM.
CSTG017I - NOTE    TASK LOG FILE FOUND IN SCHEMA FOR TASK-LOGGING=YES
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-#.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-N.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-T.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-S.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-D.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - TLOG.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-#.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-N.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-T.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-S.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-D.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - TLOG.

ENVDESC=CSTAREAD                                                    0000400
CSTG008I - NOTE    VALIDATION FOR SCHEMA = CSTASCHM.
CSTG009I - NOTE    ENVDESC = CSTAREAD.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-#.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-N.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-T.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-S.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-D.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-#.
```

```
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-N.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-T.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-S.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-D.
ENVDESC=CSTASUPD                                                    0000410

CSTG008I - NOTE    VALIDATION FOR SCHEMA = CSTASCHM.
CSTG009I - NOTE    ENVDESC = CSTASUPD.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-#.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-N.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-T.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-S.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-D.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-#.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-N.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-T.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-S.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-D.
ENVDESC=CSTATLOG                                                    0000420

CSTG008I - NOTE    VALIDATION FOR SCHEMA = CSTASCHM.
CSTG009I - NOTE    ENVDESC = CSTATLOG.
CSTG017I - NOTE    TASK LOG FILE FOUND IN SCHEMA FOR TASK-LOGGING=YES
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-#.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-N.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-T.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-S.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - C$-D.
CSTG011I - NOTE    ENVDESC BUFFER FOUND FOR FILE - TLOG.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-#.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-N.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-T.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-S.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - C$-D.
CSTG013I - NOTE    SCHEMA FILE FOUND FOR FILE - TLOG.

0 WARNINGS
```

Establishing Directory security

The DBA can control access to the Directory on a user-by-user basis by using the Directory security options. Virtually all data in the Directory may be protected at any level, including the instances of an entity.

Directory security is implemented through two options on the Directory Maintenance menu: Security Group (SG) and Maintenance Restriction (MR). A Security Group is created by defining the appropriate Maintenance Restrictions. A Maintenance Restriction is a rule or set of rules that permit or deny access through Directory Maintenance to the Directory.

Once you have defined the appropriate MR to a SG you need to relate the Security Group to the appropriate users (see the following screen illustration). If you do not relate users to a Security Group, they will have unlimited access to your Directory. The defined SG applies to both Online and Batch Directory Maintenance.

```
SECGRP: RELATE            TIS/XA DIRECTORY MAINTENANCE               1 OF 1
SECGRP:CSISG002

LAST UPDATE hh.mm.ss mm/dd/yyyy V: nnnn USER: dba-id

USER:                     USERA
USERB
USERC
USERD
```

To define Security Groups and Maintenance Restrictions follow the steps:

1. Decide what type of access a new or existing user will have to your Directory and define a Security Group, or copy an existing one to implement the desired access restrictions. For information on copying the default Security Group, CSISG002, see “Defining Security Groups” on page 231.

2. Define the desired Maintenance Restrictions for the Security Group you have created or are altering. See “Defining Maintenance Restrictions” on page 232 for more information on defining Maintenance Restrictions.

3. Test your Maintenance Restrictions on a dummy user to ensure that they function correctly. In order to implement any changes you make during testing, sign off and then sign back on to Directory Maintenance. When testing is complete, delete the dummy user.

4. Relate the Security Group to the appropriate user(s). If you do not relate a Security Group to a user, the user will have full access to the Directory. See “Relating Security Groups to users” on page 235 for more information on relating Security Groups to users.

Defining Security Groups

You need to define a Security Group for every user on your Directory and relate that Security Group to the appropriate user. If you do not relate a user to a Security Group, the user will have full access to your Directory. When a new user is added to Directory Maintenance, you can either add a new Security Group detailing the Maintenance Restrictions that apply to that user, or you can copy an existing Security Group and change it as necessary. If an existing Security Group already applies to a new user, you can simply relate the new user to the existing Security Group (see "Relating Security Groups to users" on page 235).

If you create a new Security Group, you need to relate it to the appropriate user. Any new Security Group's definitions as well as changes to existing Security Groups or Maintenance Restrictions will be implemented the next time the referenced user signs on to Directory Maintenance.

After installation or migration from a previous release of SUPRA, only the master DBAID defined in the Directory Component Description has access to the Security Group, Maintenance Restriction, and user categories. Every other user is related to the following default Security Group:

Security Group name:	CSISG001
Maintenance Restriction name:	DENYSG (deny access to SG category)
Maintenance Restriction name:	DENYMR (deny access to MR category)
Maintenance Restriction name:	DENYUS (deny access to US category)

NOTE

Only the master DBA can change this default Security Group and its Maintenance Restrictions. Cincom recommends that the master DBA retain sole authority over the ADD command in the User, Security Group, and Maintenance Restriction categories.

Defining Maintenance Restrictions

After you have defined a Security Group, you need to define the Maintenance Restrictions to it. If you copied the Security Group, all Maintenance Restrictions defined for that Security Group will be copied.

A Maintenance Restriction is composed of one or more permit/deny entries that apply to one or more of the following:

- **Categories.** A 2-character selection code from the Directory Maintenance main menu to gain access to a specific entity in the category. For example, category SC (Schema) will display information about the schema entity BURRYSCH. (Burrys is the educational database provided by Cincom.)

- **Commands.** A 2-character selection code on the Directory Category screen describing the function you want to perform on some entity in that category. For example, AD (Add) will add a schema.

- **Entities.** A specific entry of metadata in the Directory used for an organization's processing. For example, BURRYSCH is a schema for Burrys.

Deny entries, restrict access, and permit entries modify the deny entries. For example, you have decided that you want a particular user to have access to the schema BURRYSCH but not to any other schema on the Directory, you would first have to deny the user access to the Schema category as a whole and then permit access to the schema BURRYSCH only. You can add additional deny entries to modify the effects of permit entries.

Because the effects of Maintenance Restrictions are cumulative, keep them simple. For example, one Maintenance Restriction may deny several entities, but another Maintenance Restriction assigned to the user may modify the original security rules. Review all of the Maintenance Restrictions you define to ensure that there are no redundancies.

General considerations. The following are general considerations for defining Maintenance Restrictions:

- Permit and deny entries override each other on an equal level in the Directory hierarchy. Permit and deny entries at a lower level on the Directory hierarchy override entries at a higher level. For more information on the Directory hierarchy, see "Introduction to the SUPRA Server Directory and Physical Data Manager (PDM)" on page 21.

  For example, if you deny access to a schema entity and all that the Schema entity qualifies, you can then permit access to a particular file entity within that schema, because the file is at a lower level than the Schema. You may then deny access to a Secondary Key entity, because the lower level Secondary Key deny overrides the higher-level file permit.

- More specific permit or deny entries override less specific ones. For example, a permit entry for the COPY command for the file category overrides a deny entry for the File category. In other words, the user may access the file only for the COPY command (all other access is still denied).

Defining deny entries

There are three ways of defining deny entries to restrict user access to your Directory:

- **Deny a Category.** Denying access to a category in Directory Maintenance prevents a user from adding, changing, deleting, or viewing an entity from that category. However, denying access to a category does not restrict the user's access to any category qualified by the denied category or prevent the denied category from being used as a subcategory on another category command.

  For example, if you deny the Schema category from a user, the user cannot select the Schema category to add, change, delete, or view schema data. However, denying the Schema category does not restrict the user's access to the File category. Thus, the user can relate a conceptual schema to a schema from the Conceptual Schema category.

- **Deny a Command.** As when you deny a category, denying the use of a particular command does not prevent a user from using the command as a subcommand for another command where appropriate. For example, denying the DELETE command does not prevent the user from performing a VARIABLE EDIT DELETE.

- **Deny an Entity.** When you deny an entity, you are denying a user the ability to perform any command on that particular entity and any entities qualified by it. Denying an entity does not prevent a user from accessing other entities within the category.

  For example, denying the E$PL file in the schema BURRYSCH prevents the user from performing any functions for this file. However, unless you specifically deny it, the user will still be able to perform functions on the Internal Record, Secondary Key, Key Code, and Physical Field entities for the file. Denying the E$PL file in the schema BURRYSCH does not prevent the user from accessing any other file in the schema or relating an Environment Description to the E$PL file from the Environment Description category.

  Filling in the Category Code, Naming Data Category, and Entity Name fields denies access to the named entity only. Filling in the Naming Data Category and the Entity Name fields denies access to that entity and all that it qualifies.

Defining permit entries

Access to a particular category, command, or entity is permitted unless you specifically deny it. Use an explicit permit entry only to modify an explicit deny entry.

Relating Security Groups to users

After you have defined a Security Group and its Maintenance Restrictions, you need to relate the Security Group to the appropriate user(s) by one of the following methods. If you do not relate a Security Group to a user, the user will have full access to the Directory.

- To relate one or more users to one Security Group, select the RELATE command from the Security Group category.
- Add a new user and put the Security Group name in the Main Security Group field. You may assign only one Security Group using this method. For Online Directory Maintenance, the Main Security Group field appears only for the ADD command in the user category.
- To relate one or more Security Groups to one user, select the RELATE command from the user category. You may relate many Security Groups to a user.

Example Security Group

A new user named SMITH has joined your company. Because SMITH's job definition does not require full access to the Directory, you have determined what type of access is appropriate. To implement your security decision you need to add a Security Group, define the appropriate Maintenance Restrictions to the Security Group and relate that Security Group to SMITH's ID. If no Security Group is related, SMITH will have implicit access to the entire Directory.

To add the Security Group, CSISG002, and the appropriate Maintenance Restrictions for SMITH, follow these steps:

1. Add the Security Group CSISG002 for SMITH.
2. Deny access to the Security Group (SG) screen.
3. Deny authority to use the AD (Add) command in any category.
4. Deny authority to use the CG (Change) command in the Access Set category.
5. Deny access to files in the schema BURRYSCH.
6. Permit access to file E$PL.
   a. Deny authority to use the CG (Change) command.
   b. Permit authority to change a Physical Field in Internal Record HD for the file E$PL.
   c. Deny authority to use the FORMAT Utility on the file E$PL.
7. Relate the Security Group CSISG002 to SMITH. You can copy the CSISG001 default Security Group which will automatically take care of the first three restrictions above. You can then create a new Security Group for the remaining restrictions. For more information on the default Security Group, see "Defining Security Groups" on page 231.

The following screens illustrate how to define the Security Group and Maintenance Restrictions outlined in the previous example and how to relate the Security Group to SMITH:

1. Add the Security Group CSISG002 to restrict user SMITH's access to your Directory. Select the SG (Security Group) category from the TIS/XA Directory Maintenance menu and press ENTER. This takes you to the Security Group Commands screen. To add the Security Group select the AD (Add) command and type in the name of the Security Group:

```
  SECGRP:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                             RL    RELATE
CO    COPY                            RM    REMOVE
DE    DELETE                          RN    RENAME
DI    DISPLAY                         SD    STRU DSPLY
LE    LONG EDIT                       SE    SHORT EDIT
LT    LONG TEXT                       ST    SHORT TEXT

ENTER SELECTION CODE:  AD             SUBCATEGORY CODE:

ENTER NAMING DATA:
SECURITY GROUP:              CSISG002









ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

Press ENTER. The following screen is displayed.

```
  SECGRP:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE                1 of 1
SECGRP:  CSISG002

LAST UPDATE   19.34.17   11/25/1997   V:  0001   USER:  CSI-DBA

MAINT RESTRICTION COUNT:  0
```

The Maintenance Restriction count advises you of the number of Maintenance Restrictions defined to a Security Group. Because you are adding a new Security Group, the count is 0. Press ENTER. Press PA2 twice to display to the Directory Maintenance main menu.

2. Deny SMITH access to the Security Group category. Select the MR (Maintenance Restriction) category and press ENTER. The Maintenance Restriction commands screen is displayed. Select AD to add the first Maintenance Restriction as follows:

```
  MRESTR:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                            DI    DISPLAY
CG    CHANGE                         RN    RENAME
DE    DELETE

ENTER SELECTION CODE:  AD            SUBCATEGORY CODE:

ENTER NAMING DATA:
SECURITY GROUP:              CSISG002
MAINTENANCE RESTRICTION:     DENYSG









ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

Press ENTER. The next screen is displayed.

```
     MRESTR:  ADD          TIS/XA DIRECTORY MAINTENANCE                1 of 1
SECGRP:  CSISG002                  MRESTR:  DENYSG

LAST UPDATE   19.34.17   11/25/1997   V:  0001   USER:  CSI-DBA

PERMIT/DENY OPTION:          DENY
CATEGORY CODE:               SG
COMMAND CODE:
SUBCOMMAND/SUBCATEGORY:
NAMING DATA CATEGORY:
ENTITY NAME 1:
ENTITY NAME 2:
ENTITY NAME 3:
ENTITY NAME 4:
```

Because you are completely denying SMITH access to the Security Group category, there are no qualifiers to enter. Press ENTER to display to the Maintenance Restriction commands screen.

3. Deny SMITH authority to use the AD (Add) command in any category. From the Maintenance Restriction commands screen, add the next access restriction for SMITH:

```
  MRESTR:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                            DI    DISPLAY
CG    CHANGE                         RN    RENAME
DE    DELETE

ENTER SELECTION CODE:  AD            SUBCATEGORY CODE:

ENTER NAMING DATA:
SECURITY GROUP:              CSISG002
MAINTENANCE RESTRICTION:     DENYAD









ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

Press ENTER. The next screen is displayed.

```
MRESTR:  ADD          TIS/XA DIRECTORY MAINTENANCE                1 of 1
SECGRP:  CSISG002                  MRESTR:  DENYAD

LAST UPDATE   19.34.17   11/25/1997   V:  0001   USER:  CSI-DBA

PERMIT/DENY OPTION:          DENY
CATEGORY CODE:
COMMAND CODE:                AD
SUBCOMMAND/SUBCATEGORY:
NAMING DATA CATEGORY:
ENTITY NAME 1:
ENTITY NAME 2:
ENTITY NAME 3:
ENTITY NAME 4:
```

Because you are denying the AD command in all categories, press ENTER to display to the Maintenance Restriction commands screen.

4. Deny SMITH authority to use the CG (Change) file in the Access Set category. Add the restriction as follows:

```
  MRESTR:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                           DI    DISPLAY
CG    CHANGE                        RN    RENAME
DE    DELETE

ENTER SELECTION CODE:  AD           SUBCATEGORY CODE:

ENTER NAMING DATA:
SECURITY GROUP:              CSISG002
MAINTENANCE RESTRICTION:     DENYCGAS









ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

Press ENTER. The next screen is displayed.

Add the name of the category you are denying (AS) and the command you are denying (CG) as follows:

```
  MRESTR:  ADD          TIS/XA DIRECTORY MAINTENANCE                1 of 1
SECGRP:  CSISG002                  MRESTR:  DENYCGAS

LAST UPDATE   19.34.17   11/25/1997   V:  0001   USER:  CSI-DBA

PERMIT/DENY OPTION:          DENY
CATEGORY CODE:               AS
COMMAND CODE:                CG
SUBCOMMAND/SUBCATEGORY:
NAMING DATA CATEGORY:
ENTITY NAME 1:
ENTITY NAME 2:
ENTITY NAME 3:
ENTITY NAME 4:
```

5. Deny SMITH access to the BURRYSCH. This restriction means that SMITH cannot access any files in the schema BURRYSCH:

```
  MRESTR:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                           DI    DISPLAY
CG    CHANGE                        RN    RENAME
DE    DELETE

ENTER SELECTION CODE:  AD           SUBCATEGORY CODE:

ENTER NAMING DATA:
SECURITY GROUP:              CSISG002
MAINTENANCE RESTRICTION:     DENYSCHM









ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

Press ENTER. The next screen is displayed.

Enter the name of the category you are denying (SC) and the name of the schema you are denying (BURRYSCH).

Press ENTER. The Commands screen is displayed.

```
 MRESTR:  ADD          TIS/XA DIRECTORY MAINTENANCE                1 of 1
SECGRP:  CSISG002                  MRESTR:  DENYSCHM

LAST UPDATE   19.34.17   11/25/1997   V:  0001   USER:  CSI-DBA

PERMIT/DENY OPTION:          DENY
CATEGORY CODE:
COMMAND CODE:
SUBCOMMAND/SUBCATEGORY:
NAMING DATA CATEGORY:        SC
ENTITY NAME 1:               BURRYSCH
ENTITY NAME 2:
ENTITY NAME 3:
ENTITY NAME 4:
```

6. Allow SMITH access to the E$PL file in the BURRYSCH. Because you want to allow SMITH access to the E$PL file in the BURRYSCH, but the previous DENY command prevents access to any files in the schema, you need to set the Maintenance Restriction to permit access to the E$PL file as follows:

```
  MRESTR:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                            DI    DISPLAY
CG    CHANGE                         RN    RENAME
DE    DELETE

ENTER SELECTION CODE:  AD            SUBCATEGORY CAPTION:

ENTER NAMING DATA:
SECURITY GROUP:              CSISG002
MAINTENANCE RESTRICTION:     PERME$PL









ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

Press ENTER. The next screen is displayed. Type in the Naming Data Category (FI), the Entity Name (BURRYSCH), and the second Entity Name (E$PL) as follows.

Press ENTER. The Maintenance Restriction commands screen is displayed.

```
 MRESTR:  ADD          TIS/XA DIRECTORY MAINTENANCE                1 of 1
SECGRP:  CSISG002                  MRESTR:  PERME$PL

LAST UPDATE   19.34.17   11/25/1997   V:  0001   USER:  CSI-DBA

PERMIT/DENY OPTION:          PERMIT
CATEGORY CODE:
COMMAND CODE:
SUBCOMMAND/SUBCATEGORY:
NAMING DATA CATEGORY:        FI
ENTITY NAME 1:               BURRYSCH
ENTITY NAME 2:               E$PL
ENTITY NAME 3:
ENTITY NAME 4:
```

a. Deny SMITH access to the CG (Change) command for the E$PL file. As it stands, SMITH has complete access to the E$PL file. However, because you decided that SMITH will not be allowed to change the E$PL file, you need to deny the CG command as follows:

```
  MRESTR:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                             DI    DISPLAY
CG    CHANGE                          RN    RENAME
DE    DELETE

ENTER SELECTION CODE:  AD             SUBCATEGORY CODE:

ENTER NAMING DATA:
SECURITY GROUP:              CSISG002
MAINTENANCE RESTRICTION:     DCGE$PL









ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

Press ENTER. The next screen is displayed.

Type in the Command Code (CG), the Naming Data Category (FI), the first Entity Name (BURRYSCH), and the second Entity Name (E$PL) as follows:

```
 MRESTR:  ADD          TIS/XA DIRECTORY MAINTENANCE                1 of 1
SECGRP:  CSISG002                  MRESTR:  DCGE$PL

LAST UPDATE   19.34.17   11/25/1997   V:  0001   USER:  CSI-DBA

PERMIT/DENY OPTION:          DENY
CATEGORY CODE:
COMMAND CODE:                CG
SUBCOMMAND/SUBCATEGORY:
NAMING DATA CATEGORY:        FI
ENTITY NAME 1:               BURRYSCH
ENTITY NAME 2:               E$PL
ENTITY NAME 3:
ENTITY NAME 4:
```

b. Permit SMITH the authority to change the Physical Field (PF) category in the Internal Record HD on the E$PL file. Add the restriction as follows:

```
  MRESTR:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                             DI    DISPLAY
CG    CHANGE                          RN    RENAME
DE    DELETE

ENTER SELECTION CODE:  AD             SUBCATEGORY CODE:

ENTER NAMING DATA:
SECURITY GROUP:              CSISG002
MAINTENANCE RESTRICTION:     PCGIE$PL









ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

Press ENTER. The next screen is displayed.

Type in the Category Code (PF), the Command Code (CG), the Naming Data Category (IR), and the three Entity Names (BURRYSCH, E$PL, HD) as follows:

```
 MRESTR:  ADD          TIS/XA DIRECTORY MAINTENANCE                1 of 1
SECGRP:  CSISG002                  MRESTR:  PCGIE$PL

LAST UPDATE   19.34.17   11/25/1997   V:  0001   USER:  CSI-DBA

PERMIT/DENY OPTION:          PERMIT
CATEGORY CODE:               PF
COMMAND CODE:                CG
SUBCOMMAND/SUBCATEGORY:
NAMING DATA CATEGORY:        IR
ENTITY NAME 1:               BURRYSCH
ENTITY NAME 2:               E$PL
ENTITY NAME 3:               HD
ENTITY NAME 4:
```

c. Deny SMITH the authority to use the FORMAT Utility on the E$PL file. Add the restriction as follows:

```
  MRESTR:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                            DI    DISPLAY
CG    CHANGE                         RN    RENAME
DE    DELETE

ENTER SELECTION CODE:  AD            SUBCATEGORY CODE:

ENTER NAMING DATA:
SECURITY GROUP:              CSISG002
MAINTENANCE RESTRICTION:     DFIUTBUR





ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

Press ENTER. The next screen is displayed.

Type in the Category Code (FI), the Command Code (UT), the Subcommand/Subcategory (FT), and the Naming Data Category (FI), and the two Entity Names (BURRYSCH, E$PL) as follows:

```
 MRESTR:  ADD          TIS/XA DIRECTORY MAINTENANCE                1 of 1
SECGRP:  CSISG002                  MRESTR:  DFIUTBUR

LAST UPDATE   19.34.17   11/25/1997   V:  0001   USER:  CSI-DBA

PERMIT/DENY OPTION:          DENY
CATEGORY CODE:               FI
COMMAND CODE:                UT
SUBCOMMAND/SUBCATEGORY:      FT
NAMING DATA CATEGORY:        FI
ENTITY NAME 1:               BURRYSCH
ENTITY NAME 2:               E$PL
ENTITY NAME 3:
ENTITY NAME 4:
```

Press ENTER to display to the commands screen.

You have now completed the creation of the Security Group CSISG002 that you designed for user SMITH. Press PA2 twice to display to the Directory Maintenance main menu.

7. To relate the Security Group you have created to user SMITH. From the Directory Maintenance main menu, select the SG (Security Group) category and press ENTER. From the Security Group commands screen select RL (Relate) and the subcategory US (Users) as follows:

```
  SECGRP:  COMMANDS     TIS/XA DIRECTORY MAINTENANCE

AD    ADD                            RL    RELATE
CO    COPY                           RM    REMOVE
DE    DELETE                         RN    RENAME
DI    DISPLAY                        SD    STRU DSPLY
LE    LONG EDIT                      SE    SHORT EDIT
LT    LONG TEXT                      ST    SHORT TEXT

ENTER SELECTION CODE:  RL            SUBCATEGORY CODE:  US

ENTER NAMING DATA:
SECURITY GROUP:              CSISG002









ACTIVE SC:  BURRYSCH   ACTIVE ED:  BURRYENV   MAINTENANCE PERMITTED:  YES
```

Press ENTER. The next screen is displayed.

Type in the user SMITH:

```
  SECGRP:  RELATE       TIS/XA DIRECTORY MAINTENANCE
SECGRP:  CSISG002

LAST UPDATE   19.34.17   11/25/1997   V:  0001   USER:  CSI-DBA

USER:                        SMITH









COMMAND:
```

Press ENTER. The user SMITH is now related to the Security Group CSISG002. The Maintenance Restrictions will be implemented the next time SMITH signs on to Directory Maintenance. Press PA2 to exit.

6

Using PDM exits

Overview

PDM interface exits and PDM exits extend the functionality of the PDM.

PDM interface exits are linked with each PDM interface module. The entry point with a specified exit name is resolved in the corresponding link-edited interface module. Interface exits can exist within each address space communicating with the PDM (each copy could perform different operations).

PDM exits are loaded by the PDM program. They exist within the PDM address space and are common for all tasks communicating with the PDM. In a non-Directory-driven mode, the DIRECTORY Environment Description controls the names of the loaded exits. In a Directory-driven mode, the REALM Environment Description controls the names of the loaded exits.

NOTE

You cannot use the exits in the PDM interface used by HDMP:PDM (DCMT).

If you are using the optional Buffer Cache facility (SUPRA Server Release 2.4), you must define the required Buffer Cache exit entry point names within your PDM exit programs. You provide the PDM exit program names in the Environment Description. If you are running Directory-driven (the Environment Description is named in the REALM parameter in the CSIPARM file), you supply the exit names when you add or change an Environment Description using Batch or Online Directory Maintenance. Refer to the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261, or the *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260, for more information.

If you are running non-Directory-driven (the Environment Description is named in the DIRECTORY parameter in the CSIPARM file), you supply the exit names when you use the Create Environment Description program and supply the EXITS statement. See "EXITS statement" on page 204 for more information.

To use the optional Buffer Cache facility, all exit program names must be CSTTEXIT. If you wish to use other exits with CSTTEXIT, you must link-edit them with CSTTEXIT and use the following entry point names:

Exit program name	Entry point name
Command exit	PDMXCMND
Open exit	PDMXOPEN
Physical read exit	PDMXREAD
Physical write exit	PDMXWRIT
Close exit	PDMXCLOS
Check exit	PDMXCHEK
Logical write exit	PDMXLWRI
New volume exit	PDMXNVOL

For more specific information on PDM exits, see "Coding PDM exits" on page 260.

Coding PDM interface exits

There are three PDM interface exits:

- **Precommand Exit (CSTEXT01).** Called once per DML command before the DML command is processed. On entry to this exit, register 2 points to FTBLUPM. The precommand exit can modify any or all user parameters in the DML command or in the parameter list. You can use this exit to gather statistics, as an assist for upgrading old programs to SUPRA, or for additional security. A sample precommand exit (CSTK0001) is included on your SUPRA Server MACLIB.

- **DATBASXT.** Called one or more times per DML command; controllable by the exit. On entry to this exit, register 2 points to FTBLUPM. DATBASXT can terminate the DML command without calling the PDM, call the PDM and return to the DATBASXT, or call the PDM and not return to the DATBASXT. A sample DATBASXT exit (CSTK0003) is included on your SUPRA Server MACLIB.

- **Post-Command Exit (CSTEXT02).** Called once per DML command after the DML command is processed. On entry to this exit, register 2 points to FTBLUPM. The post-command exit can modify any or all user parameters but cannot modify the parameter list itself. A sample post-command exit (CSTK0002) is included on your SUPRA Server MACLIB.

To link the interface exits with an interface, follow these five steps:

1. Assemble or compile the exits.
2. Find the linkdeck for the interface. Cincom provides interface linkdecks on your installation tape MACLIB, as follows:

		Interface linkdeck names	
Interface name	**Interface description**	**OS/390**	**VSE**
CSTESTST	Single-task (ST) Interface	CSTN998Q	CSTN998J
CSTESTMT	Multitask (MT) Interface	CSTN998R	CSTN998K
CSTXCSMT	CICS Interface	CSTN998S	CSTN998L
CSTEDCMT	MTMT HDMP Interface	CSTN998T	CSTN998M

3. Modify the linkdeck by adding an INCLUDE statement for each exit.
4. If you are including a post-command exit based on the sample one (CSTK0002), you must also add an INCLUDE statement for the Cincom console-output module. This module is provided on the load 57library on your installation tape. The module name is CSEOCONS, and its entry point name is CSIECONS.
5. With the modified linkdeck as input, run the linkage editor to relink the interface.

Modification of operating system or Cincom control blocks can produce unpredictable results. To support the EXEC CICS command-level requests from the PDM interface exits, the address of the following parm area is passed to each of the PDM interface exits. The discussion of each exit includes information on where to find this address.

```
FTBLUPM  DS   OF                    USER PARM
FTBLEIB  DS   A                     USEABLE EIB ADDRESS
FTBLTUW  DS   A                     ADDRESS OF 256 BYTE TASK
*                                   UNIQUE WORK AREA
*          END OF PARM ADDRESS LIST
```

If FTBLEIB is zero, a useable EIB (exec interface block) address cannot be passed to the exit. If FTBLTUW is zero, a useable task's unique-work-area address cannot be passed to the exit.

NOTE

The PDM exits receive control in AMODE 31. PDM parameters and data areas can reside above the 16 MB line. If your program accesses these, the programs must operate in AMODE 31.

Coding the precommand exit

The precommand exit can modify a DML command before command initiation. Use it for compatibility to change certain nonstandard parameter lists into standard parameter lists. Modifying DML output parameters has no effect. The precommand exit is called by an assembler routine as if the called routine (the exit) were an assembler routine. You can write this routine in any high-level language that can operate under those conditions (PL/1 or COBOL). This routine is called once for every external DML command.

Processing proceeds in the following sequence and is not affected by the exit:

1. The application calls the PDM interface with a DML request.
2. The PDM interface examines and verifies the DML request and prepares to forward the request to the PDM.
3. Just before relaying the DML request to the PDM, the PDM interface calls the exit (if present).
4. After the PDM interface gets control back from the exit, the PDM interface relays the request to the PDM.

Precommand exit input

Register 0	Unpredictable
Register 1	Address of the DML command parameter list
Register 2	FTBLUPM parm address list (CICS Interface only)
Registers 3–12	Unpredictable
Register 13	Address of an 18 full word register save area
Register 14	Return address
Register 15	CSTEXT01 address

Precommand exit output

All register contents must be the same on output as on input.

General considerations. The following are general considerations for coding the precommand exit:

- See the sample precommand exit (source only) distributed with SUPRA Server (CSTK0001) for an example of how to code a precommand exit.
- You need to link a precommand exit into the interface load module, CSTESTST or CSTESTMT, by adding INCLUDE statements to the linkdecks and running the linkage editor. The appropriate CSTEDBMI module loads the entire interface load module at execution time.
- You can change the parameter list by having the exit call the entry point CSTEEP01 with the new parameter list.
- You can prevent a command from being executed by having the exit call the entry point CSTEEP02. No parameter list is required to call this entry point.
- You must code the entry-point name for the precommand exit as CSTEXT01.
- You can perform I/O in this exit. Use caution if you are using the exit to free memory.
- The precommand exit parameter list contains the addresses of the original parameters that point directly to the parameters even if the original DML call was from a PL/1 program.
- The first two words of the register save area must be the same on output as on input.
- This exit is not required to do any processing.
- VSE If you code this exit in COBOL for use under VSE, it must issue a call to the COBOL routine ILBSET0 before calling the COBOL subprogram. See the appropriate IBM COBOL manual for more information on the ILBSET0 routine.
- The PDM exits receive control in AMODE 31. PDM parameters and data areas can reside above the 16 MB line. If your program accesses these, the programs must operate in AMODE-31.

Coding the DATBASXT exit

For command compatibility, code the DATBASXT module to modify a command before command initiation. An example could be a user-generated command, such as DELVC, which would delete an entire related chain. A DELVC would generate a READV and DELVD for each record until the end of the chain was reached. You can enter the DATBASXT routine one or more times for each DML command. You control this entry by the status return codes.

The DATBASXT exit is called according to standard assembler routine conventions. You can write this routine in any high-level language which can operate under those conditions, for example PL/1 or COBOL.

Processing proceeds in the following sequence and is not affected by the exit:

1. The application calls the PDM interface with a DML request.
2. The PDM interface examines and verifies the DML request and prepares to forward the request to the PDM.
3. Just before relaying the DML request to the PDM, the PDM interface calls the precommand exit (if present); then it calls the DATBASXT exit. Depending on the return code, the PDM interface does one of the following:
    a. Relays the DML instruction to the PDM. When control returns from the PDM, the interface passes control back to the application.
    b. Relays the DML instruction to the PDM. When control returns from the PDM, the interface calls DATBASXT again.
    c. Returns control to the application without transferring the DML command to the PDM. DATBASXT passes a return code to the PDM interface to determine subsequent processing. The return codes are passed in Register 15 and have the following values:
        0 Process the command normally; then return to the application.
        4 Process the command normally; then return to the DATBASXT routine.
        8 Do not process the command; return immediately to the application. You must set the status code.

DATBASXT exit input

Register 0	Unpredictable
Register 1	Address of DML command parameter list
Register 2	FTBLUPM parm address list CICS Interface only)
Registers 3–12	Unpredictable
Register 13	Address of an 18 fullword register save area
Register 14	Return address
Register 15	DATBASXT address

DATBASXT exit output

Register 1	Address of DML command parameter list as modified by the DATBASXT code
Register 15	DATBASXT return code

All other register contents must be the same on output as on input.

General considerations. The following are general considerations for coding the DATBASXT exit:

- For an example of how to code a DATBASXT exit, see the sample DATBASXT exit (CSTK0003) distributed in source on the SUPRA Server MACLIB.
- The CSTEDBMI module in the main SUPRA Server Load Library loads CSTESTST (single-task PDM interface). The CSTEDBMI module in the SUPRA INTERFLM load Library Loads CSTESTMT (multitask PDM interface). You need to link a DATBASXT exit into CSTESTST or CSTESTMT by adding the INCLUDE statement to the linkdecks and running the linkage editor. CSTEDBMI loads the entire interface load module. See "Coding PDM interface exits" on page 251 for the linkdeck name(s) you must update.
- If the original parameter list has been modified by a precommand exit, the modified version is passed to DATBASXT. Otherwise, the original parameter list is passed to DATBASXT. In either case, all parameter addresses point directly to their corresponding parameters.
- The DATBASXT return code must equal 0, 4, or 8. If the return code is not 0, 4, or 8, the PDM interface abnormally terminates.
- The entry point for the DATBASXT exit is DATBASXT.
- The command exit parameter list contains the addresses of the original parameters that point directly to the parameters even if the original call was from a PL/1 program.
- This exit is not required to do any processing.
- The first two words of the register save area must be the same on output as on input.
- VSE If you code this exit in COBOL for use under VSE, it must issue a call to the COBOL routine ILBSET0 before calling the COBOL subprogram. See the appropriate IBM COBOL manual for more information on the ILBSET0 routine.

Coding the post-command exit

You can use the post-command exit to modify the results of the DML command after its completion; for example, you could use it to implement status compatibility. You can also use it to insert processing routines so you can make changes in the output parameters. The post-command exit is called according to standard assembler-routine conventions. You can write this routine in any high-level language that can operate under those conditions (PL/1 or COBOL). If you include this exit, it is always called.

Processing proceeds in the following sequence and is not affected by the exit:

1. The application calls the PDM interface with a DML request.
2. The PDM interface examines and verifies the DML request and prepares to forward the request to the PDM.
3. Just before relaying the DML request to the PDM, the PDM interface calls the precommand exit (if present); then it calls the DATBASXT exit (if present).
4. After the exits have been called, and unless the DATBASXT specified another action, the PDM interface relays the DML's request to the PDM.
5. When control returns from the PDM, the PDM interface calls the post-command exit (if present). Then it returns control to the application.

Post-command exit input

Register 0	Unpredictable
Register 1	Address of the DML command parameter list
Register 2	FTBLUPM parm address list (CICS Interface only)
Registers 3–12	Unpredictable
Register 13	Address of an 18 fullword register save area
Register 14	Return address
Register 15	CSTEXT02 address

Post-command exit output

All register contents must be the same on output as on input.

General considerations. The following lists the general considerations for coding the post-command exit:

- If the original parameter list has not been modified by a precommand exit or DATBASXT, it contains the addresses of the original parameters. These addresses point directly to the parameters even if the original call was from a PL/1 program. The command exit parameter list contains the addresses of the extended status field and the parameters as modified by precommand exit.
- The entry point for the post-command exit is CSTEXT02.
- The first two words of the register save area must be the same on output as on input.
- VSE If you are writing this exit using COBOL under VSE, you must issue a call to the COBOL routine ILBSET0 (required by VSE-COBOL) before calling the COBOL subprogram. See the appropriate IBM COBOL manual for more information on the ILBSET0 routine.
- See the sample post-command exit (source only) distributed with SUPRA Server (CSTK0002) for an example of how to code a post-command exit.
- In the case of severe error conditions, some DML commands may not take the post-command exit.
- This exit is not required to do any processing.

Coding PDM exits

The PDM exits allow you to do the following:

- Duplicate your System Log File (SLF)
- Check security
- Create a Private Log File
- Control the amount and type of logging activity
- Gather database statistics on the number of reads, writes, opens and closes an application program performs

To use PDM exits, you must name the exits in the user Environment Description using Directory Maintenance or in the boot Environment Description using the create boot Environment Description. Refer to the *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260, or the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261, for information on the exit statements of the user Environment Description. For information on the bootstrap Environment Description, see "Defining the operational environment" on page 115.

You can use one or many of these exits, but you need to code the exits to handle all types of entry (an OPEN exit must be able to handle a database file open, a Directory file open, and a System Log File open). There are eight PDM exit points for use on database files, Directory files, or the System Log File. In a non-Directory-driven mode, the PDM exits are command initialization, open, read, write, check, and close.

Directory-driven mode uses those six exits plus the logical write and new volume exits:

- **Command Initialization Exit.** Receives control after the PDM has started processing a command, but before it has read or modified any database file. This exit can only suppress execution of the DML command, in which case an error status is returned to the application. You can use this exit to add more security or as an interface to a user-security subsystem.
- **Open Exit.** Called before the PDM physically opens a file. Permits or denies opening of files or special log file processing.
- **Read Exit.** Called before the PDM physically reads a file record. Permits or denies reads to a file.
- **Write Exit.** Called before the PDM physically writes a file record. Permits or denies writes to an SLF. Allows special processing before writes to any file.
- **Check Exit.** Called after the PDM has completed a write to an SLF, but before the PDM issues the IBM CHECK macro for the write. Allows for special SLF processing.
- **Close Exit.** Called before the PDM physically closes a file.
- **Logical Write Exit.** Called before a logical record is written to the buffer for the System Log or Statistics File. It allows or suppresses logical write operations.
- **System Log New Volume Exit.** Called before a System Log switches to a new, logical volume.

The input parameters to each exit vary by the type of exit being processed. See the following figure for:

- Parameter list requirements for each type of exit
- Valid file types for each exit
- Valid return codes, by file type, for each type of exit

Displacement in Parameter List (DEC/HEX)	0 Command Initialization	4 OPEN	8 READ	12 WRITE	16 CLOSE	20 CHECK	24 LOGICAL WRITE	28 NEW VOLUME
0(00)	A(SAVE)	A(SAVE)	A(SAVE)	A(SAVE)	A(SAVE)	A(SAVE)	A(SAVE)	A(SAVE)
4(04)	A(EXIT-TYPE)	A(EXIT-TYPE)	A(EXIT-TYPE)	A(EXIT-TYPE)	A(EXIT-TYPE)	A(EXIT-TYPE)	A(EXIT-TYPE)	A(EXIT-TYPE)
8(08)	A(INTERFACE-NAME)	A(FILE-NAME)	A(FILE-NAME)	A(FILE-NAME)	A(FILE-NAME)	A(FILE-NAME)	A(ACTION)	A(ACTION)
12(0C)	A(TASK-NAME)	A(FILE-TYPE)	A(FILE-TYPE)	A(FILE-TYPE)	A(FILE-TYPE)	A(FILE-TYPE)	A(FILE-NAME)	A(FILE-NAME)
16(10)	A(USER-DATA)	A(ACB/DCB)	A(DECB/RPL)	A(DECB/RPL)	A(ACB/DCB)	A(DECB/RPL)	A(RBN-RANGE-LIST)	A(CURRENT-VOLUME-NUMBER)
20(14)	(USER-DATA-LENGTH)	-	-	-	-	-	A(ACB/DCB)	A(LAST-COMMITTED-VOLUME-NUMBER)
24(18)	A(DML-PARAMETER-LIST)	-	-	-	-	-	A(LOG-RECORD-HEADER)	A(ACB/DCB)
28(1C)	-	-	-	-	-	-	A(LOG-RECORD-HEADER-LENGTH)	A(NEXT-FILE-NAME)
32(20)	-	-	-	-	-	-	A(LOG-SYSTEM-DATA)	A(NEXT-FILE-USED-FLAG)
36(24)	-	-	-	-	-	-	A(LOG-SYSTEM-DATA-LENGTH)	-
40(28)	-	-	-	-	-	-	A(LOG-DATA)	-
44(2C)	-	-	-	-	-	-	A(LOG-DATA-LENGTH)	-
File Types	- - - -	PRIMARY RELATED SYSTEM LOG -	PRIMARY RELATED - -	PRIMARY RELATED SYSTEM LOG -	PRIMARY RELATED SYSTEM LOG -	- - SYSTEM LOG -	- - SYSTEM LOG STATISTICS	- - SYSTEM LOG -
When Taken	BEFORE COMMAND	BEFORE OPEN	BEFORE READ	BEFORE WRITE	BEFORE CLOSE	BEFORE CHECK	BEFORE LOGICAL WRITE	BEFORE LOGICAL VOLUME SWITCH
If Return Code is R15=0	ACTION IS TO PERFORM COMMAND	ACTION IS TO PERFORM OPEN	ACTION IS TO PERFORM READ	ACTION IS TO PERFORM WRITE	ACTION IS TO PERFORM CLOSE	ACTION IS TO PERFORM CHECK	ACTION IS TO PERFORM LOGICAL WRITE	ACTION IS DEFAULT LOG SWITCH PROCEDURE
or R15-4	STATUS RETURNED IS “FUNC”	ACTION IS TO PERFORM OPEN: IF NOT SYSTEM LOG, “OERR” STATUS IS RETURNED	STATUS RETURNED IS “IOER”	ACTION IS TO PERFORM WRITE	ACTION IS TO PERFORM CLOSE: IF NOT SYSTEM LOG, “CERR” STATUS IS RETURNED	ACTION IS TO PERFORM CHECK	“ACTION” PARAMETER IS EXAMINED AND FOLLOWED	“ACTION” PARAMETER IS EXAMINED AND FOLLOWED

The following is a legend for the terms in the preceding figure:

Term	Meaning
A()	A (parameter) indicates the address of the item shown in parentheses.
EXIT TYPE	The fullword binary integer that identifies the exit (0 - COMMAND INITIALIZATION, 4 - OPEN).
SAVE	An 18 fullword save area which may be used by the exit (for saving/restoring input registers).
FILE TYPE	4—PRIMARY FILE 8—RELATED FILE 12—SYSTEM LOG FILE 16—STATISTICS FILE
USER DATA	An 8-character string optionally provided by the user with the SINON command, and supplied to the command initialization exit for every command by that signed-on task.
ACB	Access method control block (for VSAM files).
DCB	Data control block (for non-VSAM files).
RPL	Request parameter list (for VSAM files).
DECB	Data event control block (for non-VSAM files).
FILE-NAME	4-character logical file name.
DML-PARAMETER-LIST	This is a pseudo-DML command parameter list. It includes the command name; other contents vary by command and by mode (central or noncentral).
ACTION	An 8-character area consisting of two 4-character field words. To be valid, each field must contain spaces or a valid instruction for that exit. (If the return code is 4 and the action is invalid, an abend results.) Valid instructions for the logical write exit are SKIP and NVOL. The only valid instruction for the new volume exit is NWAT.
RBN-RANGE-LAST	The relative block number of the last log file block that would be used to write this logical record.
CURRENT-VOLUME-7 NUMBER	The number of the current logical-log volume for the new volume exit; this is the one that has just been completed. (The first volume is number zero.)
LAST-COMMITTED-VOLUME-NUMBER	The number of the earliest logical-log volume that might be necessary for recovery to last commit.
LOG-RECORD-HEADER, LOG-SYSTEM-DATA, LOG-DATA	The three parts of the logical log record. One or more of these may have zero length(s).
NEXT-FILE-NAME	The 4-character logical name of the log file on which the next logical volume will be written.
NEXT-FILE-USED-FLAG	If the volume switch involves switching log files, this 4-character string is either REUS or UNUS. REUS indicates that the next file has been used; log records have been written to it during this run and will be overwritten by the switch. UNUS indicates that the next file is unused.

Coding the command initialization exit

You can use the command initialization exit routine to insert processing before the beginning of a DML command.

The PDM calls the command initialization exit. Based on the return code from the exit in register 15, the PDM takes one of the following actions:

0: Proceed and process the command

or

4: Not process the command and set an error status of FUNC

In either case, the exit returns control to the PDM's interface that displays control to the application.

Command initialization exit input

Register 0	Unpredictable
Register 1	Address of the command initialization exit parameter list (see the illustration under "Coding PDM exits" on page 260)
Registers 2–12	Unpredictable
Register 13	Address of an 18 fullword register save area
Register 14	Return address
Register 15	Command initialization module address

Command initialization exit output

The exit must set Register 15 to 0 or 4, and it must restore all other register's contents to their original content.

General considerations. The following lists the general considerations for coding the command initialization exit:

- The DML command parameter list (see the table later in this section) is the address of the parameter list for the physical DML command being attempted.
- At the time the command initialization exit receives control, some DML command parameter validation has already occurred. If the exit changes any of the DML command parameter addresses or the storage they address, the results are unpredictable.
- The command initialization exit is not called for the ENDTO, RQLOC or SHOWX commands.
- When the application is in the same address space as the PDM, the command initialization exit is given the list of addresses of the application's actual DML parameters. When the application is in a different address space, the command initialization exit is given a list of addresses in the PDM address space to which application parameters have been copied.

NOTE

Not all parameters of all commands are copied in all cases. When a parameter has not been copied, the address supplied is zero.

- The first two words of the register save area must be the same on output as on input.
- If 7the return code is invalid, the PDM terminates abnormally.
- If you are using the optional Buffer Cache Facility (see "Improving database performance" on page 48, "Coding the Buffer Cache Facility input file" on page 95, and "Adding Environment Descriptions" on page 140) and if you have coded your own name for this exit in an Environment Description, you must link edit this exit using the PDMXCMND entry point name.

Parameter addresses and the command initialization exit

When an application program issues a DML command, the PDM calls the command initialization exit. The PDM passes a list of addresses of the command's parameters to the exit.

OS/390

If the application issues the command to a central or attached central PDM from an interface in a different region, some parameters are not available to the exit. If the parameter is not available, the PDM passes an address of zero.

The following table indicates availability of parameters.

NOTE

The parameters are listed in their order in the command's parameter list. For example, in the ADD-M command, the command parameter is the first, the status parameter is the second parameter, the file parameter is third, the control-key parameter is fourth, and so on. For complete information about the DML commands, refer to the *SUPRA Server PDM DML Programming Guide (OS/390 & VSE)*, P26-4340.

Command	DML parameter list
ADD-M	command, status, file, control key, data list, data area*, end*
ADDVA	command, status, file, refer, linkpath, control key, data list, data area*, end*
ADDVB	command, status, file, refer, linkpath, control key, data list, data area*, end*
ADDVC	command, status, file, refer, linkpath, control key, data list, data area*, end*
ADDVR	command, status, file, refer, linkpath, control key, data list, data area*, end*
CLOSX	command, status, realm, end*
COMIT	command, status, commit ID, length, data area, end*
DEL-M	command, status, file, control key, data list*, data area*, end*
DELVD	command, status, file, refer, linkpath, control key, data list, data area*, end*
ENDLG	command, status, options*, end*
ENDTO	command, status, options*, end*

* Not always available

Command	DML parameter list
FINDX	command, status, file, qualifier, argument, data list, data area*, end*
FREEX	command, status, other, end*
MARKL	command, status, length, data area, end*
OPENX	command, status, realm, end*
QMARK	command, status, length, data area, end*
QUIET	command, status, count*, end*
RDNXT	command, status, file, qualifier, data list, data area*, end*
READD	command, status, file, refer, linkpath, control key*, data list, data area*, end*
READM	command, status, file, control key, data list, data area*, end*
READR	command, status, file, refer, linkpath, control key, data list, data area*, end*
READV	command, status, file, refer, linkpath, control key, data list, data area*, end*
READX	command, status, file, options, qualifier, length, data list, data area*, end*
RESET	command, status, commit ID, length, data area, end*
RSTAT	command, status, qualifier, length, data area, end*
SINOF	command, status, end*
SINOF	command, status, task*, end*
SINON	command, status, end*
SINON	command, status, other*, end*
WRITD	command, status, file, refer, data area*, end*
WRITM	command, status, file, control key, data list, data area*, end*
WRITV	command, status, file, refer, linkpath, control key, data list, data area*, end*

* Not always available

Coding the open exit

You can use the open exit routine to insert processing before the opening of a primary file, a related file, or a System Log File.

Before physically opening a file the PDM calls the open exit, if present. Based on the return code from the exit in register 15, the PDM takes one of the following actions:

0: Open the file

4: Do not open the file (if the file is a System Log File) or open the file and set an error status of OERR (if the file is not a System Log File).

Open exit input

Register 0	Unpredictable
Register 1	Address of open exit parameter list (see the illustration under "Coding PDM exits" on page 260)
Registers 2–12	Unpredictable
Register 13	Address of an 18 fullword register save area
Register 14	Return address
Register 15	Open exit module address

Open exit output

The exit must set Register 15 to 0 or 4, and must restore all other register contents to their original content.

General considerations. The following lists the general considerations for coding the open exit:

- You define the name of this routine in the user Environment Description using Directory Maintenance or in the boot Environment Description using Modify Schema.
- The first two words of the register save area must be the same on output as on input.
- If the return code is invalid, the PDM terminates abnormally.
- If you are using the optional Buffer Cache Facility (see "Improving database performance" on page 48, "Coding the Buffer Cache Facility input file" on page 95, and "Adding Environment Descriptions" on page 140) and if you have coded your own name for this exit in an Environment Description, you must link edit this exit using the PDMXOPEN entry point name.

Coding the read exit

You can use the read exit routine to insert processing before reading a primary file or a related file.

Before physically reading a physical record, the PDM calls the read exit, if present. Based on the return code from the exit in register 15, the PDM takes one of the following actions:

0: Perform the read

4: Do not perform the read and set an error status of IOER

Read exit input

Register 0	Unpredictable
Register 1	Address of the read exit parameter list (see the illustration under "Coding PDM exits" on page 260)
Registers 2–12	Unpredictable
Register 13	Address of an 18 fullword register save area
Register 14	Return address
Register 15	Read exit module address

Read exit output

The exit must set Register 15 to 0 or 4, and must restore all other register contents to their original content.

General considerations. The following lists the general considerations for coding the read exit:

- You define the name of this routine in the user Environment Description through Directory Maintenance or in the boot Environment Description using Modify Schema.
- The first two words of the register save area must be the same on output as on input.
- If the return code is invalid, the PDM terminates abnormally.
- If you are using the optional Buffer Cache Facility (see "Improving database performance" on page 48, "Coding the Buffer Cache Facility input file" on page 95, and "Adding Environment Descriptions" on page 140) and if you have coded your own name for this exit in an Environment Description, you must link edit this exit using the PDMXREAD entry point name.

Coding the write exit

You can use the write exit routine to insert processing before writing data to a primary file, a related file, or a System Log File.

Before physically writing a physical record, the PDM calls the write exit, if present. Based on the return code from the exit in register 15, the PDM takes one of the following actions:

0: Perform the write

4: Do not perform the write

Write exit input

Register 0	Unpredictable
Register 1	Address of the write exit parameter list (see the illustration under “Coding PDM exits” on page 260)
Registers 2–12	Unpredictable
Register 13	Address of an 18 fullword register save area
Register 14	Return address
Register 15	Write exit module address

Write exit output

If the input file is a primary or related file, the exit must set Register 15 to 0. If the input file is a System Log File, the exit must set Register 15 to 0 or 4. The exit must restore all other register contents to their original contents.

General considerations. The following lists the general considerations for coding the write exit:

- You define the name of this routine in the user Environment Description through Directory Maintenance or in the boot Environment Description using Modify Schema.
- The first two words of the register save area must be the same on output as on input.
- If the input file is a primary or related file and Register 15 does not contain 0, the PDM abnormally terminates.
- If the input file is a System Log File and Register 15 contains 4, the physical record is not written to the System Log File, and the System Log File cannot be used for recovery.
- If the return code is invalid, the PDM abnormally terminates.
- If you are using the optional Buffer Cache Facility (see "Improving database performance" on page 48, "Coding the Buffer Cache Facility input file" on page 95, and "Adding Environment Descriptions" on page 140) and if you have coded your own name for this exit in an Environment Description, you must link edit this exit using the PDMXWRIT entry point name.

Coding the check exit

You can use the check exit routine to insert processing before checking the result of the I/O on the System Log File.

During processing, the PDM may output system log records. After the physical I/O to the system log is complete, the PDM calls the check exit, if present. Based on the return code from the exit in register 15, the PDM takes one of the following actions:

0: Issue the IBM Assembler CHECK macro to check the I/O

4: Do nothing

Then, the PDM resumes normal processing.

Check exit input

Register 0	Unpredictable
Register 1	Address of check exit parameter list (see the illustration under "Coding PDM exits" on page 260)
Registers 2–12	Unpredictable
Register 13	Address of an 18 fullword register save area
Register 14	Return address
Register 15	Check exit module address

Check exit output

Register 15 must be set to either 0 or 4. All other register contents must be restored to their original contents.

General considerations. The following lists the general considerations for coding the check exit:

- You define the name of this routine in the user Environment Description using Directory Maintenance or in the boot Environment Description using Modify Schema.
- The first two words of the register save area must be the same on output as on input.
- If the return code is invalid, the PDM will terminates abnormally.
- If you are using the optional Buffer Cache Facility (see "Improving database performance" on page 48, "Coding the Buffer Cache Facility input file" on page 95, and "Adding Environment Descriptions" on page 140) and if you have coded your own name for this exit in an Environment Description, you must link edit this exit using the PDMXCHEK entry point name.

Coding the close exit

You can use the close exit routine to insert processing before closing a primary file, a related file, or a System Log File.

The PDM prepares to physically close a file. Before doing the close, it calls the close exit, if present. Based on the return code from the exit in register 15, the PDM takes one of the following actions:

0 Close the file

4: Do not close the file (if the file is a System Log File) or close the file and set an error status of CERR (if the file is not a System Log File).

Close exit input

Register 0	Unpredictable
Register 1	Address of the close exit parameter list (see the illustration under "Coding PDM exits" on page 260)
Registers 2–12	Unpredictable
Register 13	Address of an 18 fullword register save area
Register 14	Return address
Register 15	Close exit module address

Close exit output

The exit must set Register 15 to 0 or 4, and it must restore all other register contents to their original content.

General considerations. The following lists the general considerations for coding the close exit:

- You define the name of this routine in the user Environment Description using Directory Maintenance or in the boot Environment Description using Modify Schema.
- The first two words of the register save area must be the same on output as on input.
- If the return code is invalid, the PDM terminates abnormally.
- If you are using the optional Buffer Cache Facility (see "Improving database performance" on page 48, "Coding the Buffer Cache Facility input file" on page 95, and "Adding Environment Descriptions" on page 140) and if you have coded your own name for this exit in an Environment Description, you must link edit this exit using the PDMXCLOS entry point name.

Coding the logical write exit

You can use the logical write exit to suppress selectively the writing of System Log or Statistics Records.

Before writing a logical record to any System Log Group or Statistics File, the PDM initializes the 8-character action parameter to blanks and calls the exit. Based on the output returned by the exit (the code in register 15 and the contents of the action parameter), the PDM takes one of the following actions:

Code	Action parameter	PDM action
0	(Ignored)	Write this logical record to this Log Group or Statistics File. (This is the default action.)
4	ƀƀƀƀƀƀƀƀ	Write this logical record to this Log Group or Statistics File.
4 4 4	SKIPƀƀƀƀ ƀƀƀƀSKIP SKIPSKIP	Do not write this logical record to this Log Group or Statistics File.
4 4 4	NVOLƀƀƀƀ ƀƀƀƀNVOL NVOLNVOL	End this current logical log volume and start a new one, then write this logical record to this Log Group.
4 4	NVOLSKIP SKIPNVOL	End this current logical log volume and start a new one; do not write this logical record to this Log Group.

Logical write exit input

Register 0	Unpredictable
Register 1	Address of the logical write exit parameter list (see the illustration under "Coding PDM exits" on page 260)
Registers 2–12	Unpredictable
Register 13	Address of an 18 fullword register save area
Register 14	Return address
Register 15	Logical write exit module address

The action parameter contains spaces on entry.

Logical write exit output

The exit must set Register 15 to 0 or 4, and must restore all other register contents to their original content.

If the exit modifies the 8-character action parameter, it must leave it with valid contents: spaces and/or valid, correctly positioned 4-character commands. The only valid commands are SKIP and NVOL. Any command present in the action parameter must start in the first or fifth column of the parameter.

General considerations. The following lists the general considerations for coding the logical write exit:

- If the return code in register 15 is not 0 or 4, the PDM terminates abnormally.
- The exit can put one or both of the following commands in the action parameter (each command must start in the first or fifth character of the parameter):
  - SKIP—Tells the PDM not to write this logical record to this log group or statistics file.
  - NVOL—Tells the PDM to end this current log volume and start a new one (before it writes a logical record). VALID ONLY FOR THE SYSTEM LOG FILES.
- If the return code in register 15 is 0, the action parameter is ignored.
- If the return code in register 15 is 4, and the action parameter is invalid, the PDM terminates abnormally.
- If the return code in register 15 is 4, and the action parameter includes the NVOL command, and the current file is a statistics file, the PDM terminates abnormally.
- The PDM terminates abnormally when *all four* of the following conditions are present:
  - The return code in register 15 is 4.
  - The action parameter includes the NVOL command.
  - The PDM is in the process of switching the current log group from one logical log volume to another.
  - The current logical record is one that the PDM writes at the beginning or end of each log volume (record types EOVL, INIT, FINT, and TASK).
- This exit can be specified only in the user Environment Description, not in the boot Environment Description.
- If you are using the optional Buffer Cache Facility (see “Improving database performance” on page 48, “Coding the Buffer Cache Facility input file” on page 95, and “Adding Environment Descriptions” on page 140) and if you have coded your own name for this exit in an Environment Description, you must link edit this exit using the PDMXLWRI entry point name.

Coding the new volume exit

You can use the new volume exit to avoid waiting for console operator action before starting a new logical log volume.

Before starting a new logical volume for a system log group, the PDM initializes the 8-character action parameter to blanks and calls the exit. Based on the output returned by the exit (the code in register 15 and the contents of the action parameter), the PDM takes one of the following actions:

Code	Action parameter	PDM action
0	(Ignored)	If the next logical log volume is a disk file which has already been used in this run of the PDM, wait for the console operator's reply before writing to that file again. (This is the default action.)
4	ƀƀƀƀƀƀƀƀ	If the next logical log volume is a disk file which has already been used in this run of the PDM, wait for the console operator's reply before writing to that file again.
4 4 4	NWATƀƀƀƀ ƀƀƀƀNWAT NWATNWAT	Write to the next logical log volume without waiting for the console operator's reply.

New volume exit input

Register 0	Unpredictable
Register 1	Address of the new volume exit parameter list (see the illustration under "Coding PDM exits" on page 260)
Registers 2–12	Unpredictable
Register 13	Address of an 18 fullword register save area
Register 14	Return address
Register 15	New volume exit module address

The action parameter contains spaces on entry.

New volume exit output

The exit must set Register 15 to 0 or 4, and it must restore all other register contents to their original content.

If the exit modifies the 8-character action parameter, it must leave it with valid contents: spaces and/or valid, correctly positioned 4-character commands. The only valid command is NWAT. Any command present in the action parameter must start in the first or fifth column of the parameter.

General considerations. The following lists the general considerations for coding the new volume exit:

- If the return code in register 15 is not 0 or 4, the PDM terminates abnormally.
- If the return code in register 15 is 0, the action parameter is ignored.
- If the return code in register 15 is 4, and the action parameter is invalid, the PDM terminates abnormally.
- This exit can be specified only in the user Environment Description, not in the boot Environment Description.
- If you are using the optional Buffer Cache Facility (see “Improving database performance” on page 48, “Coding the Buffer Cache Facility input file” on page 95, and “Adding Environment Descriptions” on page 140) and if you have coded your own name for this exit in an Environment Description, you must link edit this exit using the PDMXNVOL entry point name.

7

Monitoring and controlling the PDM

Options

SUPRA Server helps you monitor and control the PDM with the following options:

- Interactive Services
- TIS/XA subsystem operator commands
- Termination of the PDM

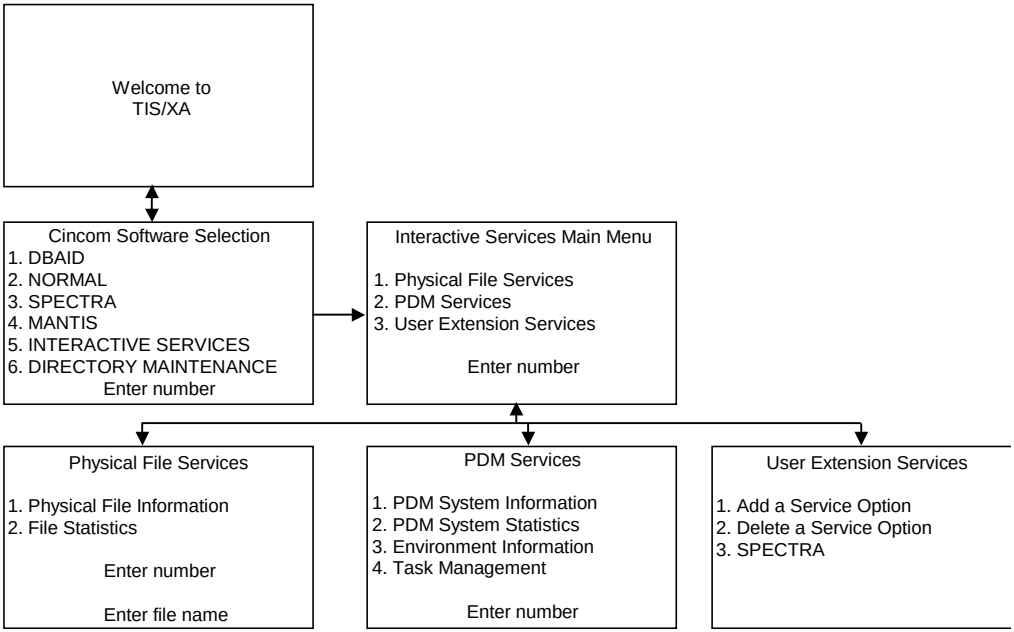
Interactive Services consists of programs to help you maintain the PDM. Its menu allows you to display statistics on physical files, execution statistics such as environment information and information about tasks and interfaces. Interactive Services also allows you to add or delete service programs as necessary.

The TIS/XA subsystem operator commands allow you to assist the PDM in cross-address-space central communications. In other words, by using the subsystem operator commands, you can facilitate communications between a PDM running in central or attached central operating mode and a PDM interface in a different address space. For an overview of PDM operating modes, see “Introduction to the SUPRA Server Directory and Physical Data Manager (PDM)” on page 21.

Termination of the PDM can be controlled or uncontrolled. You can control termination of the PDM by issuing the ENDTO command or by using the PDM Termination Utility.

Using Interactive Services

To use Interactive Services, first sign on to the TIS/XA system. The TIS/XA Software Selection Facility menu gives you access to all of the components of TIS/XA through one screen. While you can sign on to the other components separately, you can sign on to Interactive Services only through the TIS/XA system. The following figure summarizes the screens you must access to invoke Interactive Services:



The first screen welcomes you to TIS/XA.

```
                                                                0110
            ///           ///
         //////         /////
       ////////       ///////
     //////////     /////////   WELCOME TO:
   ///////////    ///////////
  ///////////   /////// TTTTTT IIIIII SSSSSS     // XX   XX  AAAAAA
//////////////  ////////  TT     II   SS        //   XX XX   AA  AA
                          TT     II   SSSSSS   //     XXX    AAAAAA
//////////////  ////////  TT     II       SS  //     XX XX   AA  AA
 //////////////  ////////  TT   IIIIII SSSSSS //     XX   XX  AA  AA
  ///////////  ///////////
    //////////  //////////
      //////// ////////
         /////  //////
           ///  ///          PLEASE ENTER
                             USERID:                          :
                             PASSWORD:                        :
TO BY-PASS SELECTION SCREEN, ENTER INFORMATION:                   :
:
PA1/PA2=EXIT
```

Specify a user ID and password; then press ENTER to display the Software Selection Screen.

You can bypass the TIS/XA Software Selection Facility menu by typing IN/SERVE on the immediate selection line. That action takes you directly to the Interactive Services menu.

If you did not enter IN/SERVE on the previous screen, the Cincom Software Selection Menu screen is displayed. It varies according to the components your company has installed. For example:

```
                                                                  0210
              CINCOM SOFTWARE SELECTION MENU

 ENTER SELECTION INFORMATION:                                        :
:
* TO EXECUTE WITH CURRENT USER-ID, PRESS ENTER.
* TO EXECUTE WITH ALTERNATE USER-ID, PRESS PF2/PF14
* TO VIEW EXPANDED MENU DEFINITION, PRESS PF1/PF13


1. DBAID              11. UTIL/SQL
2. SPECTRA            12. M_SERIES
3. MANTIS             13. CPCS
4. DIRECTORY          14. OPC/SQL
5. IN/SERV
6. RESIGNON
7. CONTROL
8. I/SQL
9. LOAD/SQL
10. EASY/SQL


PA1/PA1=EXIT
PF1/PF13 WITHOUT SELECTION INFORMATION=HELP
```

From this list, you select the option for Interactive Services.

When you select Interactive Services, the Interactive Services Main Menu is displayed.

```
INTERACTIVE SERVICES MAIN MENU              300

    SELECT SERVICE OPTION:

    1. PHYSICAL FILE SERVICES
    2. PDM SERVICES
    3. USER EXTENSION SERVICES









    ENTER OPTION NUMBER :    :




PA2-EXIT
```

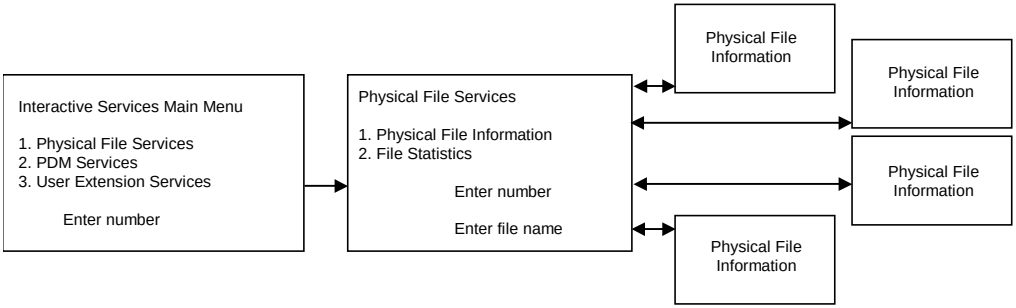
Select the option:

- **Physical File Services.** To view file attributes and file statistics online
- **PDM Services.** To view PDM statistics and environment information and you can manage tasks
- **User Extension Services.** To add your own utility programs to the Interactive Services Main Menu

Selecting Physical File Services

If you select Physical File Services, the series of screens in the following figure are displayed. With this option, you can see the physical file's data set name, type, mode, access method, blocksize, and so on. You can also see statistics on the files, for example, the number of logical reads and writes, physical reads and writes, and forced writes.

To see these statistics, you must have indicated in your active Environment Description that you want the PDM to collect them. You must also have defined a Statistics file in the active schema. You do not need to define a physical file, but you must include the DD statement in the run JCL (//STAT DD DUMMY).



NOTE

The PDM gathers some statistics regardless.

The first screen is the Physical File Services Main Menu.

```
                    PHYSICAL FILE SERVICES
                                                          300B



          1. PHYSICAL FILE INFORMATION
          2. FILE STATISTICS

          ENTER OPTION NUMBER : # :



          ENTER FILE NAME TO REQUEST A KNOWN FILE OR
          LEAVE A BLANK TO CHOOSE FROM A LIST OF FILES  :      :



PA2-EXIT
```

Enter 1 for Physical File Information or 2 for File Statistics. After you select an option, you can enter the name of a file on which you want information or statistics.

When you enter the name of a file, you go directly to the Physical File Information screen or the File Statistics screen.

If you do not enter the name of a file on the Physical File Services screen, a list of your files is displayed.

```
    FILE SELECTION LIST                    520

TYPE "S" NEXT TO THE FILE YOU WANT TO DISPLAY :


. FL01      . ZIPC          . FN06          . FN18          . FN30
. FL02      . SUPP          . FN07          . FN19          . FN31
. FL03      . PART          . FN08          . FN20          . FN32
. FL04      . INVT          . FN09          . FN21          . FN33
. FL05      . WARE          . FN10          . FN22          . FN34
. FL06      . ORDR          . FN11          . FN23          . FN35
. M001      . INVO          . FN12          . FN24          . FN36
. M002      . FN01          . FN13          . FN25          . FN37
. M003      . FN02          . FN14          . FN26          . FN38
. CUST      . FN03          . FN15          . FN27          . FN39
. ADDR      . FN04          . FN16          . FN28          . FN40
. STAT      . FN05          . FN17          . FN29          . FN41




PA2-EXIT   ENTER-PROCESS   PF8-FORWARD
```

Type an S (for Select) next to the file you want; then press ENTER.

Option 1. Displaying Physical File Information

When you select Physical File Information, the following screen is displayed. It shows the name of a file, its type, mode, blocksize, logical record length, and so on. This information comes from the Directory.

```
  PHYSICAL FILE INFORMATION            500


FILE NAME : TLOG :

DDNAME  ................. : DBMSTLOG
DATA SET NAME  .......... :
BUFFER POOL NAME  ........ : BUF4
FILE TYPE  .............. : TASK LOG
FILE MODE  .............. :
OPEN MODE  .............. : SUPD
EUPD OWNER (TASK NAME)  .. :
ACCESS METHOD  .......... : BDAM
CODED FILE  ............. : N
LOGICAL RECORD LENGTH  ... : 1
BLOCKSIZE  .......... ... : 3024
BLOCKS PER TRACK ......... : 6
RECORDS PER BLOCK  ....... : 3024
RECORDS PER FILE (TLR)  .. : 900
CONTROL INTERVAL SIZE  ... : 0
PRIME NUMBER  ........... : 1

PA2-EXIT
```

Option 2. Displaying File Statistics

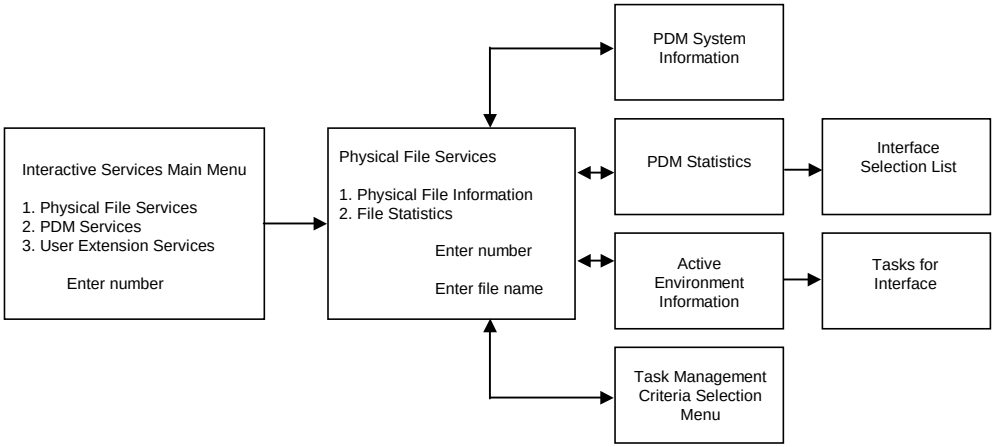
When you select File Statistics, the following screen showing the statistics the PDM continuously gathers for the file is displayed.

```
PHYSICAL FILE STATISTICS                  510


FILE NAME : TLOG :


F1.01  TOTAL LOGICAL READS...................... 2        2
F1.02  TOTAL PHYSICAL READS..................... 2        2
F1.03  TOTAL IN-MEMORY HITS..................... 0        0
F1.04  TOTAL IN-MEMORY HITS ON UPDATED BUFFER... 0        0
F1.04  TOTAL PHYSICAL UPDATES FORCED BY PHYSICAL READ.....0

F2.01  AVG LOGICAL READ PER PHYSICAL READ.............1.00
F2.02  % OF LOGICAL READS WHICH WERE IN-MEMORY HITS...0.00%
F2.03  % OF IN-MEMORY HITS TO AN UPDATED BUFFER.......0.00%
F2.04  % OF PHYSICAL READS FORCING A PHYSICAL UPDATE..0.00%




                             -MORE-
PA2-EXIT
WMM: PF7-UP  PF8-DOWN  PF9-END  PF10-LEFT   PF11-RIGHT  PF12-HOME   001 001
```

Press PF8 to display the next two screens which provide the remaining statistics on the specified file.

Selecting PDM Services

The following figure depicts the order in which the screens are displayed, if you select PDM Services from the Interactive Services Main Menu:



The first screen displayed is the PDM Services menu:

```
                              PDM SERVICES                    300E


            1. PDM SYSTEM INFORMATION
            2. PDM SYSTEM STATISTICS
            3. ENVIRONMENT INFORMATION
            4. TASK MANAGEMENT


            ENTER OPTION NUMBER :   :


PA2-EXIT
```

Enter the number of the option you want.

Option 1. Displaying PDM System Information

Selecting Option 1 on the PDM Services menu displays the PDM Information screen.

```
                              PDM INFORMATION                   600


  PDM NAME : PDMTEST1 :

  PDM TYPE  ........(multitask/single-task)...  multitask
  ATTACHED MODE  ............................  yes
  CENTRAL MODE  .............................  yes
  PDM AND APPLICATION IN SAME ADDRESS SPACE ..  yes









PA2-EXIT
```

Option 2. Displaying PDM System Statistics

When you select Option 2 from the PDM Services menu, the PDM Statistics screen is displayed.

```
PDM STATISTICS FOR PDMTEST1                     610

S1.01  DATE AND TIME STATISTICS WERE LAST RESET  ..  XX/XX/XX 00:00:00

       CURRENT TASKS  ..............................     0
S2.01  TOTAL TASKS  ................................     0
S2.02  MAXIMUM CONCURRENT TASKS  ...................     0

S3.01  CURRENT RECORD HOLDING ENTRIES IN USE  ......     0
S3.02  MAXIMUM RECORD HOLDING ENTRIES USED  ........     0
S3.03  CURRENT MONITOR ENTRIES IN USE  .............     0
S3.04  MAXIMUM MONITOR ENTRIES USED  ...............     0

S4.01  TOTAL READ COMMANDS  ........................     0       0.00%
S4.02  TOTAL UPDATE COMMANDS  ......................     0       0.00%
S4.03  TOTAL ADD AND DELETE COMMANDS  ..............     0       0.00%
S4.04  TOTAL OTHER COMMANDS  .......................     0       0.00%

                                                       -MORE-
PA2=EXIT
WWM: PF7-UP  PF8-DOWN  PF9-END  PF10-LEFT  PF11-RIGHT  PF12-HOME  001 001
```

Press PF8 to display the next two screens which provide the remaining PDM statistics.

```
       CURRENT COMMANDS  ..........................          0
S5.01  TOTAL COMMANDS ISSUED TO THE PDM  ...........          0
S5.02  MAX NUMBER OF COMMANDS AT COMMAND STARTS  ...          0
S5.03  SUM OF COMMANDS AT COMMAND STARTS  ..........          0
S5.04  AVG NUMBER OF COMMANDS AT COMMAND STARTS  ...              0.00

S6.01  TOT ELAPSED TIME ON COMMANDS ISSUED TO PDM ..   00:00:00
S6.02  AVG ELAPSED TIME PER COMMAND ISSUED TO PDM ..   00:00:00
S6.03  MAX ELAPSED TIME FOR ANY COMMAND TO PDM  ....   00:00:00

S7.01  NUMBER OF TIMES PDM WAS INACTIVE  ...........          0
S7.02  AVG NUM OF TIMES PDM WAS INACTIVE PER COMMAND
S8.01  AMOUNT OF TIME PDM WAS ACTIVE  ..............             0.00%
S8.02  AMOUNT OF TIME PDM WAS INACTIVE  ............             0.00%
S8.03  TOTAL PDM TIME  .............................             0.00%
S8.04  AVG AMOUNT OF TIME PDM WAS ACTIVE PER COMMAND

                                                            -MORE-
PA2=EXIT
WWM: PF7-UP  PF8-DOWN  PF9-END  PF10-LEFT  PF11-RIGHT  PF12-HOME  001 001
```

```
S9.01  TOTAL BYTES OF MEMORY (IN K)  ...............    0
S9.02  TOTAL MEMORY USED AT PRESENT TIME (IN K)  ...    0        0.00%
S9.03  MAXIMUM BYTES OF MEMORY USED (IN K)  ........    0        0.00%
S9.04  THRESHOLD MEMORY  ...........................    0%

S10.01  TOTAL BYTES OF XA MEMORY (IN K)............  00:00:00
S10.02  TOTAL XA MEMORY USED AT PRESENT TIME (IN K) . 00:00:00       0.00%
S10.03  MAXIMUM BYTES OF XA MEMORY USED  (IN K) .....        0       0.00%









PA2=EXIT
WWM: PF7-UP  PF8-DOWN  PF9-END  PF10-LEFT  PF11-RIGHT  PF12-HOME  001 001
```

Option 3. Displaying environment information

When you select Option 3 from the PDM Services menu, information about the environment in which the PDM is running is displayed. This information comes from the PDM.

```
                        ACTIVE ENVIRONMENT INFORMATION                  620

LOADED DIRECTORY ENVIRONMENT DESCRIPTION NAME  ....... xxxxxxxx
LOADED DIRECTORY SCHEMA NAME  ....................... xxxxxxxx

STATISTICS  ..........................................        0
TASK LOGGING  ........................................        0
SYSTEM LOGGING  ......................................        0

IMAGE LOGGING  .......................................        0
BEFORE IMAGE LOGGING  ................................        0
AFTER IMAGE LOGGING  .................................        0

ALL SIGNON LOGGING    ................................        0
UPDATE SIGNON LOGGING  ...............................        0

PA2-EXIT                                                     -MORE-
```

Press PF8. The next screen is displayed:

```
                        ACTIVE ENVIRONMENT INFORMATION                  620

ALL FUNCTION LOGGING  ................................
LOADED DIRECTORY SCHEMA NAME  ........................

NUMBER OF FILES IN SYSTEM  ...........................        0
MAXIMUM LOGICAL RECORD LENGTH  .......................        0
MAXIMUM BLOCKSIZE  ...................................        0

MAXIMUM ELEMENTS IN A FILE  ..........................        0
MAXIMUM LINKPATHS IN A FILE  .........................        0
MAXIMUM KEY SIZE  ....................................        0

OPENX OPTION  ........................................

FORCE END OF VOLUME OPTION  ..............................

PA2-EXIT
```

Option 4. Selecting task management

When you select Option 4 from the PDM Services menu, the Task Management-Criteria Selection menu is displayed. Using this screen, you can display information about tasks and interfaces, or purge an inactive task whose interface is no longer connected (Sections A through E).

> **NOTE**
> You cannot purge interfaces; you can purge only the tasks for an interface.

```
TASK MANAGEMENT - CRITERIA SELECTION MENU            630

SPECIFY CRITERIA OR USE DEFAULTS.

ACTION            : DISPLAY :       (DISPLAY/PURGE)
PURGE OPTION      : LIST    :       (LIST/NO LIST)

INTERFACE NAME    : ALL      :      (ALL/NAME)
INTERFACE TYPE    : ALL   :         (ALL/BATCH/CICS/CM)
INTERFACE STATUS  : ALL          :  (ALL/CONNECTED/NOTCONNECTED)

TASK NAME         : ALL      :      (ALL/NAME)
TASK STATUS       : ALL       :     (ALL/ACTIVE/NONACTIVE)

TIME SINCE
LAST COMMIT      :      :          (TIME IN MINUTES)



PA2-EXIT   ENTER-PROCESS
```

The Task Management screen contains default values. You need not change the fields unless you want different options. If you leave all of the defaults as they are, the PDM displays a list of all types of interfaces. To display information or purge tasks, you select options from those displayed in the column on the right of the screen. For example, the options for ACTION are DISPLAY or PURGE. If you try to enter anything else, a message appears in place of the options prompting you to enter a valid option.

The only exception occurs when you enter INTERFACE NAME and TASK NAME. Task Management does not check the validity of interface and task names. If you enter invalid names, they are displayed on the screens where Task Management lists interface and task names.

1. Displaying Interfaces

   To display all of the interfaces, accept the default values shown on the Task Management screen:

```
INTERFACE SELECTION LIST                    1631

TYPE "S" NEXT TO AN INTERFACE NAME TO LIST TASKS BY CRITERIA

INTERFACE    INTERFACE       INTERFACE
NAME         TYPE            STATUS
CINCOMI1     BATCH           CONNECTED
ED24PROC     BATCH           NOT CONNECTED
S INTERF01   CICS            NOT CONNECTED
INTERF02     CICS            CONNECTED



PA2-EXIT   PF8-FORWARD   ENTER-PROCESS
```

   To display a list for a specific type of interface, like CICS, press PA2 to go back to the Task Management screen. There, set INTERFACE NAME to ALL and INTERFACE TYPE to CICS. You can set INTERFACE STATUS to any of the three options. If you want to see the CICS interfaces that are still connected, for example, set INTERFACE STATUS to CONNECTED.

2. Displaying Tasks from the Interface Selection List

   To display tasks from the Interface Selection List, enter S next to an interface name. For example, if you put an S next to INTERF01, the tasks for that interface are displayed:

```
TASKS FOR INTERFACE INTERF01                    1632

TYPE "P" NEXT TO A TASK NAME TO PURGE THE TASK.
TYPE "ALL" TO PURGE ALL TASKS ON THIS SCREEN :     :

TASK      TASK SIGN    TIME SINCE           COMPLETION
NAME      ON STATUS    LAST COMIT            MESSAGE
TSK01      ACTIVE       00:00:01
TSK02      NONACTIVE    01:03:24
TSK03      NONACTIVE    00:25:10
TSK04      ACTIVE       00:00:03
TSK05      NONACTIVE    00:10:35









PA2-EXIT   PF8-FORWARD   ENTER-PROCESS
```

3. Displaying Tasks from the Task Management Screen

   You can also display tasks by supplying the name of the interface on the Task Management screen. For example, if you set INTERFACE NAME to INTERF02, you go directly to the screen called Tasks for Interface INTERF02:

```
TASKS FOR INTERFACE INTERF02                        1632

TYPE "P" NEXT TO A TASK NAME TO PURGE THE TASK.
TYPE "ALL" TO PURGE ALL TASKS ON THIS SCREEN :     :

TASK      TASK SIGN    TIME SINCE           COMPLETION
NAME      ON STATUS    LAST COMIT            MESSAGE
TSK01      NONACTIVE    00:57:03
TSK02      ACTIVE       00:00:02
TSK03      NONACTIVE    00:23:57









PA2-EXIT   PF8-FORWARD   ENTER-PROCESS
```

   If you want to see the inactive tasks for the same interface, go back to the Task Management screen and change TASK STATUS to NONACTIVE. Press ENTER, and the list is displayed on this screen.

4. Purging Tasks from a List of Tasks

   Once you display tasks, you can purge them by typing P next to the task you want to purge. For example, you can use the previous screen, Tasks for Interface INTERF02, to purge tasks. To purge all tasks, type ALL between the colons at the top of the screen.

```
TASKS FOR INTERFACE INTERF02              1632

TYPE "P" NEXT TO A TASK NAME TO PURGE THE TASK.
TYPE "ALL" TO PURGE ALL TASKS ON THIS SCREEN :     :

TASK      TASK SIGN   TIME SINCE           COMPLETION
NAME      ON STATUS   LAST COMIT            MESSAGE
TSK01        ACTIVE     00:00:01
P TSK02      NONACTIVE  01:03:24
P TSK03      NONACTIVE  00:25:10
TSK04        ACTIVE     00:00:03
TSK05        NONACTIVE  00:10:35





PA2-EXIT   PF8-FORWARD   ENTER-PROCESS
```

After you type P next to a task, a short completion message indicates the result of the purge:

```
TASKS FOR INTERFACE INTERF02

TYPE "P" NEXT TO A TASK NAME TO PURGE THE TASK.
TYPE "ALL" TO PURGE ALL TASKS ON THIS SCREEN :     :

TASK        TASK SIGN   TIME SINCE           COMPLETION
NAME        ON STATUS   LAST COMIT            MESSAGE
TSK01        ACTIVE      00:00:01
P TSK02      NONACTIVE   01:03:24            TASK PURGED
P TSK03      NONACTIVE   00:25:10            TASK PURGED
TSK04        ACTIVE      00:00:03
TSK05        NONACTIVE   00:10:35





PA2-EXIT   PF8-FORWARD   ENTER-PROCESS
```

If you type ALL (to purge all tasks), a message appears asking you to confirm the purge:

```
TASKS FOR INTERFACE INTERF02                 1632
TYPE "YES" ON THE LINE BELOW TO CONFIRM PURGE.
TYPE "P" NEXT TO A TASK NAME TO PURGE THE TASK.
TYPE "ALL" TO PURGE ALL TASKS ON THIS SCREEN :     :

TASK      TASK SIGN    TIME SINCE           COMPLETION
NAME      ON STATUS    LAST COMIT            MESSAGE
TSK01      ACTIVE       00:00:01
TSK02      NONACTIVE    01:03:24
TSK03      NONACTIVE    00:25:10
TSK04      ACTIVE       00:00:03
TSK05      NONACTIVE    00:10:35




PA2-EXIT   PF8-FORWARD   ENTER-PROCESS
```

You confirm the purge by typing YES where you entered ALL. After you press ENTER, a message appears next to each task indicating whether it was purged.

```
TASKS FOR INTERFACE INTERF02                 1632

TYPE "P" NEXT TO A TASK NAME TO PURGE THE TASK.
TYPE "ALL" TO PURGE ALL TASKS ON THIS SCREEN : YES :

TASK        TASK SIGN   TIME SINCE           COMPLETION
NAME        ON STATUS   LAST COMIT            MESSAGE
TSK01        ACTIVE       00:00:01
TSK02        NONACTIVE    00:03:24          TASK PURGED
TSK03        NONACTIVE    00:25:10          TASK PURGED
TSK04        ACTIVE       00:00:03
TSK05        NONACTIVE    00:10:35          TASK PURGED





PA2-EXIT   PF8-FORWARD   ENTER-PROCESS
```

5. Purging Tasks from the Task Management Screen

   To purge tasks from the Task Management screen, set the ACTION to PURGE and the PURGE OPTION to NO LIST. Set the rest of the criteria to indicate the tasks you want to purge. To purge all inactive tasks for all CICS interfaces that are not connected, and that have not issued a COMIT in 10 minutes, set the options:

```
TASK MANAGEMENT - CRITERIA SELECTION MENU

SPECIFY CRITERIA OR USE DEFAULTS.

ACTION            : PURGE   :       (DISPLAY/PURGE)
PURGE OPTION      : NO LIST :       (LIST/NO LIST)

INTERFACE NAME    : ALL      :      (ALL/NAME)
INTERFACE TYPE    : CICS. :         (ALL/BATCH/CICS/CM)
INTERFACE STATUS  : NOTCONNECTED :  (ALL/CONNECTED/NOTCONNECTED)

TASK NAME         : ALL      :      (ALL/NAME)
TASK STATUS       : NONACTIVE :     (ALL/ACTIVE/NONACTIVE)

TIME SINCE
LAST COMIT       : 10   :          (TIME IN MINUTES)


PA2-EXIT   ENTER-PROCESS
```

   After you press ENTER, a message asks you to confirm the purge. Confirm by typing YES.

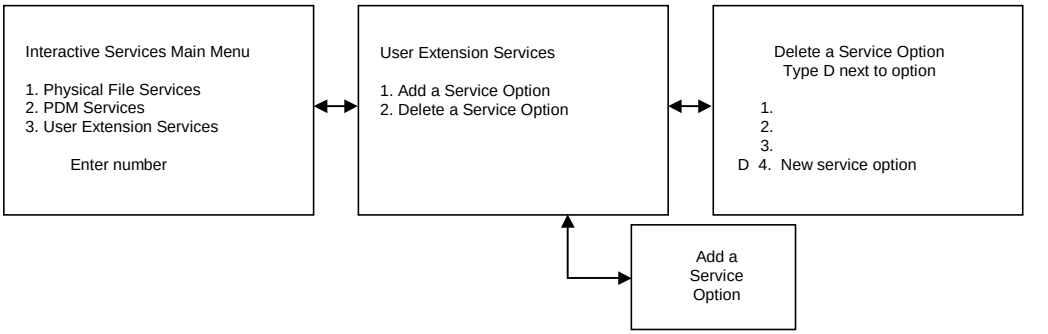
```
TASK MANAGEMENT - CRITERIA SELECTION MENU           630
CSFC0281  TYPE "YES" ON THE LINE BELOW TO CONFIRM PURGE.
SPECIFY CRITERIA OR USE DEFAULTS.

ACTION            : PURGE   :       (DISPLAY/PURGE)
PURGE OPTION      : NO LIST :       (LIST/NO LIST)

INTERFACE NAME    : ALL      :      (ALL/NAME)
INTERFACE TYPE    : ALL   :         (ALL/BATCH/CICS/CM)
INTERFACE STATUS  : NOTCONNECTED :  (ALL/CONNECTED/NOTCONNECTED)

TASK NAME         : ALL      :      (ALL/NAME)
TASK STATUS       : NONACTIVE :     (ALL/ACTIVE/NONACTIVE)

TIME SINCE
LAST COMIT       :  0   :          (TIME IN MINUTES)

PA2-EXIT   ENTER-PROCESS
```

After you type YES, a message indicates whether the tasks were purged.

```
TASK MANAGEMENT - CRITERIA SELECTION MENU            630
ALL TASKS HAVE BEEN PURGED
SPECIFY CRITERIA OR USE DEFAULTS.

ACTION            : PURGE   :       (DISPLAY/PURGE)
PURGE OPTION      : NO LIST :       (LIST/NO LIST)

INTERFACE NAME    : ALL      :      (ALL/NAME)
INTERFACE TYPE    : ALL   :         (ALL/BATCH/CICS/CM)
INTERFACE STATUS  : NOTCONNECTED :  (ALL/CONNECTED/NOTCONNECTED)

TASK NAME         : ALL      :      (ALL/NAME)
TASK STATUS       : NONACTIVE :     (ALL/ACTIVE/NONACTIVE)

TIME SINCE
LAST COMIT       :  0   :          (TIME IN MINUTES)



PA2-EXIT   ENTER-PROCESS
```

Selecting User Extension Services

The following figure depicts the order in which the screens are displayed, if you select User Extension Services from the Interactive Services Main:



The first screen is the User Extension Services menu is displayed. With this screen, add six utility programs to the Interactive Services Main menu. Also, delete utilities to make room for others.

```
USER EXTENSION SERVICES MENU              300


1. ADD A SERVICE OPTION
2. DELETE A SERVICE OPTION

ENTER OPTION NUMBER :     :



PA2-EXIT
```

Option 1	To add a program
Option 2	To delete a program you have already added

Option 1. Adding a Service Option

Selecting Option 1 displays the Add a Service Option screen.

```
 ADD A SERVICE OPTION                          700




ENTER THE FOLLOWING INFORMATION:

SERVICE OPTION TO APPEAR ON THE MENU (DO NOT INCLUDE A NUMBER)
:                                                :


NAME OF PROGRAM :                      :

IS THIS PROGRAM A MANTIS PROGRAM (Y/N) :   :

IF YES, ENTER MANTIS USER ID WHICH CONTAINS THIS PROGRAM
:                                 :



PA2-EXIT    ENTER-PROCESS
```

To add an option, enter the information requested at the prompts:

- **SERVICE OPTION TO APPEAR ON THE MENU.** The option name can be u7p to 41 characters long.
- **NAME OF PROGRAM.** The program name can be up to 20 characters long.
- **MANTIS USER ID.** The ID can be up to 30 characters long.

When you finish entering the information, press ENTER. A completion message is displayed at the top of the screen.

After you add the new service option, it is displayed on the Interactive Services Main menu when you return to it:

```
INTERACTIVE SERVICES MAIN MENU             300

SELECT SERVICE OPTION:

1. PHYSICAL FILE SERVICES
2. PDM SERVICES
3. USER EXTENSION SERVICES
4. USERS FILE STATISTICS









ENTER OPTION NUMBER  :    :





PA2-EXIT
```

Option 2. Deleting a Service Option

Selecting Option 2 from the User Extension Services menu displays the Delete Service Option screen. Enter D (for Delete) next to the option you want to delete. You cannot delete the first three options. If you try, you receive an error message. The following screen deletes an example option:

```
 DELETE SERVICE OPTION                        710



TYPE "D" NEXT TO THE OPTION(S) YOU WANT DELETED:

   1. PHYSICAL FILE SERVICES
   2. PDM SERVICES
   3. USER EXTENSION SERVICES
D  4. USERS FILE STATISTICS









PA2-EXIT   ENTER-PROCESS
```

After you delete an option, it does not appear on the Interactive Services Main Menu when you return to it.

Using batch task management

To execute Task Management in a batch environment if CICS is not connected use batch Task Management to terminate an inactive task or a task whose interface is no longer connected.

OS/390 VSE

To execute batch Task Management in OS/390 and VSE, see sample JCL member TXJCACCS.

Batch Task Management executes in the same way as the online version; your keyboard input is the same as your entries on the screens. The following OS/390 JCL and keyboard input illustrate how to execute Task Management, list all interfaces, and back out:

```
//  JOB  . . . etc.
//TISXA  EXEC PGM=TISXB
//STEPLIB   DD  DSN=TISXA.INTERFLM,DISP=SHR
//          DD  DSN=TISXA.LINKLIB, DISP=SHR
//CSIPARM   DD  DSN=TISXA.CSIPARM(TXPCENTR),DISP=SHR
//TERMINAL  DD  SYSOUT=A,DCB=BLKSIZE=133 Simulated screen
//PRINTER   DD  SYSOUT=A,DCB=BLKSIZE=133 Simulated printer
//SYSPRINT  DD  SYSOUT=A,DCB=BLKSIZE=133 Keyboard stmts/errors
//SETPRAY   DD  DISP=SHR,DSN=mantis.cluster
//CSOT      DD  DISP=SHR,DSN=mantis.transfer.cluster
//* define any other user data sets here
//KEYBOARD  DD  *                        Simulated keyboard
CINCOM;CINCOM;IN/SERVE/X;<BLANK=ON>;<ECHO=ON>;
2;
4;
DISPLAY;LIST;ALL;ALL;ALL;ALL;ALL;0;
<PF8>;
<PA2>;
<PA2>;
<PA2>;
<PA2>;
/*
//
```

Line 1	CINCOM;CINCOM—To respond with user ID and password to the Welcome to TIS/XA screen. IN/SERVE—To respond with immediate selection of Interactive Services. X causes Interactive Services not to return to the TIS/XA Software Selection Facility menu upon completion. BLANK=ON—To print blank lines on the screen. ECHO=ON—To display your input onscreen. When ECHO=ON is active, whatever you key in is “echoed” on the display. When you specify ECHO=ON, you must put your first response on the same line.
Line 2–2	To select Option 2, PDM Services, from the Interactive Services Main Menu.
Line 3–4	To select Option 4, Task Management, from the PDM Services screen.
Line 4	DISPLAY;LIST;ALL;ALL;ALL;ALL;ALL;0; To select the options that display a list of all interfaces from the Task Management—Criteria Selection menu. You must list all of these options in a batch environment; the default values are not automatically filled in as they are in the online environment. In addition, you must list them in the same order as they appear on the screen. If you want to change any options, you must list them all again in order.
Line 5-PF8	To scroll through the screens of interfaces on the Interface Selection List. Specify PF8 as many times as necessary to print all of the interfaces.
Line 6-PA2	To return to the Task Management-Criteria Selection menu.
Line 7-PA2	To return to the PDM Services screen.
Line 8-PA2	To return to the Interactive Services Main Menu.
Line 9-PA2	To sign-off Batch MANTIS.

After the batch job executes, the results will be printed. Because ECHO=ON: was specified in line 1, each screen will be printed twice: once as it looks before input and once as it looks after input.

To correct errors in batch task management, you must resubmit the corrected input statements.

If an error occurs, you may see unexpected results on your printout. A single typing error may result in more than one error message. For example, if you typed DSSPLAY instead of DISPLAY on line 4, the next screen on the printout would show an error message and a request to correct the mistake, which cannot be done in the batch environment.

Task Management, however, is looking for DISPLAY spelled correctly, and it will look for a recognizable input. Task Management sees your input, PF8 to scroll forward, on line 5. Because Task Management is looking for a valid entry on line 4, and PF8 is an invalid entry on that line, the printout shows another error message.

Each time you press PF8, the printout shows an error message, until Task Management comes to PA2, which is a valid response on any line.

Using TIS/XA subsystem operator commands (OS/390)

Under the OS/390 operating systems, the TIS/XA subsystem assists the PDM in cross-address-space central communications; that is, communications between a PDM running in central or attached central mode, and a PDM interface in a different address space. In this section, a central PDM means a PDM running in central or attached central mode, and an interface means an interface for a central mode or attached central mode PDM in a different address space.

You enter the TIS/XA subsystem operator commands using the following format:

```
/command[Ø]parameter
```

The slash (/) is the default command prefix character for the TIS/XA subsystem. You set a command prefix character for your subsystem when you install it. For information on installing the TIS/XA subsystem, see “Installing TIS/XA subsystem (OS/390)” on page 53.

Throughout this discussion, the slash (/) represents the command prefix character. When you enter a command, replace the slash (/) with an appropriate command prefix character. For every command except DTISS, you must use the command prefix character of the subsystem(s) you want to affect. You cannot enter any spaces between the command prefix character and the command. Each command (except DTISS) is 1-character long. You can enter a single space or no space between the command and its parameter(s). You can use uppercase or lowercase letters when entering the command.

DISPLAY command

The DISPLAY command displays the names of active central PDMs and the names of interfaces connected to a central PDM.

$$\text{/D[b]}\left[\frac{\textbf{DBM}}{\textbf{dbm} - \textbf{id}[,\textbf{interface} - \textbf{id}]}\right]$$

$$\left[\frac{\textbf{DBM}}{\textbf{dbm} - \textbf{id}}\right]$$

Description *Optional*. Specifies the central PDM to be displayed.

Default DBM

Format 1–8 characters. First character must be alphabetic, $, #, or @. Remaining characters must be alphanumeric, $, #, or @.

Considerations

- When you specify DBM, the subsystem displays all active central PDMs but does not display the name, status, or location of any central PDM interface.
- When you specify *dbm-id*, the subsystem always displays the named active central PDM (if present).
- When you specify *dbm-id* and you do not specify *interface-id*, the subsystem displays the name, status, and location of each active central PDM interface associated with that PDM.
- When you specify both *dbm-id* and *interface-id*, the subsystem displays the status and location of the named active central PDM interface (if present) associated with that PDM; it does not display any other interfaces.

interface-id

Restriction You cannot specify *interface-id* unless you specify *dbm-id*.

Description *Optional*. Specifies the central PDM interface to be displayed.

Format 1–8 characters. First character must be alphabetic, $, #, or @. Remaining characters must be alphanumeric, $, #, or @.

Considerations

- When you specify *interface-id*, the subsystem displays the status and location of the named active central PDM interface (if present) associated with the specified PDM; it does not display any other interfaces.
- When you specify *dbm-id* and you do not specify *interface-id*, the subsystem displays the name, status, and location of each active central PDM interface associated with that PDM.

General considerations

- The command prefix character is shown here as /, but you must use the command prefix character of the subsystem(s) you want to affect. This command has no effect on subsystems with different command prefix characters.
- The number of interfaces displayed reflects the number of interfaces communicating with a PDM. If an interface job ends without signing off or abending, the subsystem detects that the interface is not physically present but is still logically connected to the PDM. In such a situation, this command shows zero interfaces, but an ENDTO command or the PDM Termination utility may still have an ACTV status.

Examples

- To display all active central PDMs for the subsystem(s) whose command prefix is %, enter either:

  ```
  %D
  ```

  or:

  ```
  %DØDBM
  ```

  The subsystem responds:

  ```
  %CSTS310I TISS(SUBA) ACTIVE DBMS(002) INTERFACES(003)
  %CSTS309I CENTR(PRODDBMS) ASID(002) STATUS(COMM RDY)
  %CSTS309I CENTR(TESTDBM) ASID(017) STATUS(NOT RDY)
  ```

  This output reveals the following facts:

  - Two central PDMs and three central PDM interfaces are active and associated with the TIS/XA subsystem named SUBA.
  - The PDM named PRODDBMS is running in address space 2 and is ready to communicate with central PDM interfaces.
  - The PDM named TESTDBM is running in address space 17 and is not ready to communicate with central PDM interfaces.

- To display all connected interfaces for the TESTDBM central PDM for the subsystem(s) whose command prefix is <, enter:

  ```
  <DØTESTDBM
  ```

  The subsystem responds:

  ```
  <CSTS310I TISS(TIS8) ACTIVE DBMS(001) INTERFACES (002)
  <CSTS309I CENTR(TESTDBM) ASID(013) STATUS(COMM RDY)
  <CSTS309I INTER(TESTCM1) ASID(021) STATUS(CONNECT)
  <CSTS309I INTER(TESTREPT) ASID(017) STATUS(CONNECT)
  ```

  This output reveals the following facts (among others):

  - The central PDM named TESTDBM has two active central PDM interfaces associated with it; their names are TESTCM1 and TESTREPT.
  - The interface named TESTREPT is running in address space 17.
  - The interface named TESTREPT is connected; that is, it has successfully initiated communications with the central PDM.

- To display the status of a central PDM interface named PRODCICS associated with a central PDM named PRODDBM and the TIS/XA subsystem(s) under the subsystem(s) whose command prefix is ?, enter:

  ```
  ?DØPRODDBM,PRODCICS
  ```

  The subsystem responds:

  ```
  ?CSTS310I TISS(BETA) ACTIVE DBMS(002) INTERFACES(007)
  ?CSTS309I CENTR(PRODDBM) ASID(017) STATUS(COMM RDY)
  ?CSTS309I TISS(BETA) INTER(PRODCICS) ASID(012)
  STATUS(CONNECT)
  ```

  This output shows that the interface named PRODCICS is connected; that is, it has successfully initiated communications with the central PDM.

- To display the status of a central PDM named TESTDBM3 associated with the TIS/XA subsystem(s) whose command prefix is /, enter:

  ```
  /DØTESTDBM3.
  ```

  The subsystem responds:

  ```
  /CSTS310I TISS(SUBC) ACTIVE DBMS(002) INTERFACES(000)
  /CSTS310I TISS(SUBA) ACTIVE DBMS(001) INTERFACES(000)
  CSTS308I TIS SUBSYSTEM SUBA, NO ACTIVE ENTRY FOUND
  /CSTS309I TIS(SUBC) CENTR(TESTDBM3) ASID(013)
  STATUS(COMM RDY)
  ```

  This output reveals the following facts (among others):

  - There are two active TIS/XA subsystems whose command prefix is / -SUBA and SUBC.
  - There is no active central PDM named TESTDBM3 associated with the TIS/XA subsystem named SUBA.
  - There is an active central PDM named TESTDBM3 associated with the TIS/XA subsystem named SUBC.

DTISS command

The DTISS command displays the 4-character name(s) and command prefix character(s) for all active TIS/XA subsystem(s). It also displays all active central PDMs and central PDM interfaces associated with the subsystem(s). Use this command to determine the command prefix for the subsystem(s).

/DTISS

General considerations

- The command prefix character is shown here as /, but you can use any character. It need not be the same as the command prefix for any installed TIS/XA subsystem.
- You should avoid using a command prefix character for DTISS which is the same as the command prefix character for a non-TIS/XA subsystem. If you use such a character, the DTISS command would still work, but the non-TIS/XA subsystem would attempt to interpret DTISS as one of its commands. This could lead to unintended results.
- The number of interfaces displayed reflects the number of interfaces communicating with a PDM. If an interface job ends without signing off or abending, the subsystem detects that the interface is not physically present but is still logically connected to the PDM. In such a situation, this command shows zero interfaces, but an ENDTO command or the PDM Termination Utility may still have an ACTV status.

Example

- To display all active subsystem(s), enter:

  ```
  ?DTISS
  ```

  The subsystem responds:

  ```
  %CSTS310I TISS(TIS8) ACTIVE DBMS(002) INTERFACES(004)
  ?CSTS310I TISS(SUBA) ACTIVE DBMS(001) INTERFACES(000)
  <CSTS310I TISS(TSTI) DRAINING DBMS(001) INTERFACES(000)
  ```

  Each subsystem's command prefix character precedes the message text. Use that command prefix character to issue subsequent commands to that subsystem. For example, to display the names of PDMs using subsystem TIS8, you would enter:

  ```
  %D
  ```

- To display all active subsystem(s), enter:

  ```
  /DTISS
  ```

  If no subsystems are active, none respond. OS/390 displays a message saying that the command is invalid.

SET command

The SET command cancels a STOP command. If the indicated subsystem(s) are stopped or draining because of a STOP command, the SET command reactivates them so they can accept communications requests.

/T

General consideration

The command prefix character is shown here as /, but you must use the command prefix character of the subsystem(s) you want to affect. This command has no effect on subsystems with different command prefix characters.

Example

To cancel the draining of a subsystem, enter:

```
%T
```

The subsystem responds:

```
%CSTS304I TIS SUBSYSTEM TIS8 IS ACTIVE
```

STOP command

The STOP command takes the subsystem out of service. If any PDMs or interfaces are executing when you issue the command, the subsystem remains available for their use. No new PDMs or interfaces, including warm starts, may be started using this subsystem (except for ENDTO). If no PDMs or interfaces are executing when you issue the command, the subsystem is taken out of service.

/P

General considerations

- The command prefix character is shown here as /, but you must use the command prefix character of the subsystem(s) you want to affect. This command has no effect on subsystems with different command prefix characters.
- If there are active PDMs using this subsystem, the subsystem is not stopped. However, the subsystem is draining, which means interfaces cannot connect to the active PDMs for this subsystem, and no PDMs can begin using the subsystem. When all interfaces have disconnected and all PDMs have terminated, you may reissue the STOP command to the subsystem. You must reissue the STOP command to successfully halt the subsystem.
- After you issue the STOP command, you cannot reconnect interfaces to the PDM.

Examples

- To stop a subsystem with no active PDMs, enter:

  ```
  ?P
  ```

  The subsystem responds:

  ```
  ?CSTS303I TIS SUBSYSTEM TIS9 IS STOPPED.
  ```

- To drain a subsystem with active PDMs, enter:

  ```
  %P
  ```

  The subsystem responds:

  ```
  %CSTS302I TIS SUBSYSTEM TSTA IS DRAINING.
  %CSTS307I TIS SUBSYSTEM TSTA, ACTIVE DBM: TESTDBMA
  ```

- To drain a subsystem with the active PDM, TESTDBMA, enter:

  ```
  %P TESTDBMA
  ```

  The subsystem responds:

  ```
  %CSTS310I TIS(TSTA) DRAINING DBMS(001) INTERFACES(001)
  %CSTS309I CENTR(TESTDBMA) ASID(013) STATUS(COMM RDY)
  %CSTS309I INTER(TESTCICS) ASID(017) STATUS(CONNECT)
  ```

Terminating a multitask PDM

Termination of the PDM can be controlled or uncontrolled. Uncontrolled termination's result from an abend in the system, a hardware failure, and so on. Controlled termination's occur when you terminate the PDM.

Two types of uncontrolled termination or failure can occur during processing:

- Task failure
- System failure

Control the termination of a PDM, operating in central mode, by:

- Issuing the ENDTO command. (For more information, refer to the *SUPRA Server PDM DML Programming Guide (OS/390 & VSE)*, P26-4340.)
- Executing the PDM Termination Utility. (For information on the PDM Termination Utility, refer to the *SUPRA Server PDM DBA Utilities User's Guide (OS/390 & VSE)*, P26-6260.)

Uncontrolled termination

A task failure may occur in one or more tasks (one task or the entire CICS), or a single-application task may fail.

A system failure may occur when anything fails that affects the PDM. In a system failure, the PDM will dump additional information in the form of snap dumps along with the normal system dump. Snap dumps are selective dynamic dumps produced at various points in a run.

Refer to the *SUPRA Server PDM Logging and Recovery Guide (OS/390 & VSE)*, P26-2223, for more information about the symptoms and causes of failures and the procedures you should follow to correct failures under several uncontrolled termination's.

Controlled termination

A controlled termination can be forced or unforced, depending on how you code the FORCE parameter of the ENDTO command. For information on coding the ENDTO command, refer to the *SUPRA Server PDM DML Programming Guide (OS/390 & VSE)*, P26-4340.

If the PDM is attached under CICS, you terminate it with the DISCONNECT operator command. For more information, refer to the *SUPRA Server PDM CICS Connector Systems Programming Guide (OS/390 & VSE)*, P26-7452.

NOTE

A PDM attached under CICS will terminate if CICS terminates.

Monitoring PDM memory usage

Monitor PDM memory usage through the memory monitor facility. For more specific information, refer to “Managing file space” on page 46. The memory usage information reported by this facility is also available with the PDML SHOWX command. For more information on the SHOWX command, refer to the *SUPRA Server PDM DML Programming Guide (OS/390 & VSE)*, P26-4340.

8

Expanding directory and PDM files

Expansion process

Before you begin the expansion process, it is important to back up your Directory or PDM files. If you do not back up your files, recovery from errors during the file-expansion process is impossible.

All procedures and sample members referred to in this chapter are in the SUPRA Server product libraries on your installation tape. Operating system considerations are noted where applicable.

Expanding Directory files

In order to decide whether or not you need to expand your Directory files, you need to determine the current size of your Directory. The size of your Directory is proportional to the number of entities and relationships defined in your Directory files. Use the DBA Utilities File Statistics function to find the available, unused record space. For this purpose, JCLLIB member TXJDSTAT is provided.

The following table provides space estimates for the entities you will be adding. Use these estimates to help you determine whether or not you need to expand your Directory files:

Added entity	DATA	DEF#	NAME	STRU	TEXT (Long text)
For each schema	0	1	1	0	1 record per line
For each file, including BASE.IR	0	2	2	4 for first	1 record per line
For each record code IR	0	1	1	1	1 record per line
For each physical field	1	1	1	2 for first; 1 for subsequent	1 record per line
For each Environment Description	0	1	1	2	1 record per line
For each buffer pool	1	0	0	0	None
For each env-desc to file relationship	0	0	0	2 for first; 1 for subsequent	None

To expand your Directory files to add the entities and relationships you require, follow these steps:

1. Unload the Directory files using the appropriate JCLLIB member TXJDIRUL (OS/390/VSE) or TXJDIRUT (VSE only).

2. Change the SUPRA Server bootstrap schema (CSTASCHM) to the new sizes you need by executing the following steps:

   a. Create a batch job for modifying the bootstrap schema using the JCLLIB member TXJBMODS.

OS/390 For OS/390, edit the Directory files pointed to by this sample to your DSN names.

VSE Under VSE, the member uses procedure TISBUSCH.

Consideration: For the two symbolic parameters LINKLIB= and NEWLIB=, either use a new library name in the NEWLIB parameter or use the same name used in the LINKLIB parameter. If you use a new library, you must put the five supplied bootstrap Environment Description modules there before executing step 3. If you use the same name, the job will replace the schema module on the LINKLIB during the link edit.

   b. Edit the appropriate member: MODSCnnn, where *nnn* is the last three digits of the unit/device where your SUPRA Server Directory files reside (the sample uses MODSC380 for 3380). Change the TOTAL-LOGICAL-RECORDS value for the five Directory files.

   c. Submit the job and review the output; the return code for the Modify step should be zero. Refer to the *SUPRA Server PDM Messages and Codes Reference Manual (RDM/PDM Support for OS/390 & VSE)*, P26-0126, for diagnostic and error message descriptions.

3. Rebuild the SUPRA Server bootstrap validation module (CSTAVMOD) to incorporate the modified schema module with the same Environment Descriptions. This must be done any time you revise a bootstrap schema or Environment Description.

   To rebuild the validation module:

   a. Create a batch job for rebuilding the validation module using JCLLIB member TXJBVMOD. Both the LINKLIB and NEWLIB parameters must point to the library you named in NEWLIB for step 2.

   b. Create five valid module combinations using input member CSTAVMOD as coded. Each of the five distributed bootstrap Environment Descriptions will have the same modified bootstrap schema. The PDM uses the VALMOD at execution to verify the boot modules named in CSIPARM or UCL.

   c. Submit the job and review the output; the return code should be zero. Refer to the *SUPRA Server PDM Messages and Codes Reference Manual (RDM/PDM Support for OS/390 & VSE)*, P26-0126, for diagnostic and error message descriptions.

   d. If you named a new library for NEWLIB, tailor all JCLLIB members that use LINKLIB to reflect the name of the library containing your updated bootstrap modules.

4. Reallocate the Directory files. Do not format them because the next step does this as part of the load.

5. Load the updated Directory files. To do this, use JCLLIB member TXJDIRLD. Be sure the ENVLIB parameter names the library where you have linkedited your newly updated schema, SUPRA Server Environment Descriptions, and new validation module.

   For more information on the Load utility or UCL, refer to the *SUPRA Server PDM DBA Utilities User's Guide (OS/390 & VSE)*, P26-6260.

6. Change the Directory File Definition on the Directory. Cincom supplies the CSISCH20 Schema for Directory reports. CSISCH20 contains the definition of the Directory files. Use Directory Maintenance to tailor the supplied definition to your definition.

OS/390

To change the Directory file definition in OS/390:

Edit the sign-on information in DIRLODSO to include a valid DBA ID and password. Edit the Directory file changes in DIRLODCH to match the output of the Modify Schema program run in step 2. Submit a Batch Directory Maintenance job using DIRLODSO and DIRLOCH as input to put your changes in effect.

VSE

To change the Directory file definition in VSE:

Edit the sign-on information in TISDMCH to include a valid DBA ID and password. Edit the Directory file changes in TISDMCH to match the output of the Modify Schema program run in step 2. Submit the TISDMCH procedure to put your changes in effect.

For more information on batch changes to files, refer to the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261.

7. Verify initialization. Use Directory Maintenance, either boot-driven or Directory-driven, to ensure that the PDM initializes properly. Run a Directory report to ensure the new boot description matches the Directory description.

Expanding PDM files

The steps for expanding PDM files are similar to the steps for expanding Directory files. When you are expanding PDM files, however, it is not necessary to update the bootstrap schema.

The following steps explain how to expand a PDM primary file (LV00). The steps for expanding a PDM related file are the same; however, you should refer to the *SUPRA Server PDM DBA Utilities User's Guide (OS/390 & VSE)*, P26-6260, for differences in the UCL used to unload and load a related file. The JCL members TXJDIRUL and TXJDIRLD refer to both primary and related PDM files.

1. Back up your Directory files and the PDM files you wish to expand, in this case the LV00 file.

2. Create a job to UNLOAD your PDM LV00 file. See JCLLIB members TXJDIRUL and TXJC$NDU for examples.

3. Create an input member to update the LV00 file definition; increase the TOTAL-LOGICAL-RECORDS for the LV00. See JCLLIB member TXJDIRM for JCL to perform Batch Directory Maintenance. For more information, refer to the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261.

4. You can also use Online Directory Maintenance to make this change. However, the Batch Directory Maintenance input member you create will serve as an audit trail of the change. For more information, refer to the *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260.

5. Allocate a new LV00 matching the changed file characteristics you specified in step three above.

NOTE

Do not format the file as the next step does this as part of the load.

6. Create a job to load your LV00 data into the new LV00 file. See JCLLIB members TXJDIRLD and TXJC$NDL for examples.

9

Using the TIS/XA Software Selection Facility

Overview

The TIS/XA Software Selection Facility, a MANTIS application, provides a single entry point to all online Cincom products. The facility consists of programs providing a common menu to select various TIS/XA components to assist you in running TIS/XA components and applications. The TIS/XA Software Selection Facility is easy to use and can be tailored to fit the needs of your environment.

The example screens in this chapter reflect the TIS/XA Software Selection Facility provided for SUPRA Server 2.7 customers. The menu for SUPRA Server 2.1.6 varies slightly. In either case, because the Software Selection Facility is built according to the software available at your installation, your menu may vary from the examples.

This chapter provides information on:

- Using security exits
- Using the available TISXA Software Selection Facility functions, including:
  - Directly accessing facility menu options
  - Signing on to the facility with an alternate user ID and password
  - Moving around the facility using quick exits
  - Changing current user ID and password by re-signing on to the facility
  - Passing user data to MANTIS or CONTROL:
  - Executing MANTIS programs from the facility
  - Executing the facility in batch mode

The programs used by the TIS/XA Software Selection Facility reside on the Master User and the CSI-DBA User. If you are using the facility, you need to ensure that any MANTIS programs your system uses contain the CHAIN to the MASTER:TERMINATE program to exit. If you are not using the facility, replace the distributed version of MASTER:SIGN_ON and MASTER:TERMINATE with your modified version. For information about updating the MANTIS facilities, refer to *AD/Advantage MANTIS Administration OS/390, VSE/ESA,* P39-5005.

Executing the TIS/XA Software Selection Facility

Before you use the TIS/XA Software Selection Facility (SSF), you must:

- In the MASTER:SIGNON program, insert this line before all chains to CONTROL:SIGNON:

  ```
  CHAIN "CSI-DBA:CSFS0000-SIGN-ON-A"
  ```

- In the MASTER:TERMINATE program, insert this line before all chains to CONTROL:SIGNON:

  ```
  CHAIN "CSI-DBA:CSFS0000-TERMINATE-A"
  ```

The sign-on screen is shown below. Enter the user ID and password. Then press ENTER to start SSF.

```
          ////            ////
        //////          /////
     ////////        ////////
    //////////      //////////       WELCOME TO :
   ///////////    ///////////                                        T.M.
  ///////////   /////// TTTTTT  IIIIII  SSSSSS    // XX   XX   AAAAAA
 ///////////  ///////////TT      II    SS       //   XX XX    AA  AA
                         TT      II      SSSS  //     XXX     AAAAAA
///////////  ///////////TT      II        SS //     XX XX    AA  AA
 ///////////  ///////////TT    IIIIII  SSSSSS//     XX   XX   AA  AA
   ///////////  ///////////
     ////////// //////////
       ///////  ////////    PLEASE ENTER
        ////// //////
          ////  //         USER-ID  :                              :
                            PASSWORD :                               :

TO BY-PASS SELECTION SCREEN, ENTER INFORMATION:                     :
:                                                                    :


PA2/PA1=EXIT
```

If the ID and password are valid, the Software Selection menu displays the following information. After three invalid attempts to sign-on, you are forced out of the Software Selection Facility. (See “Using security exits” on page 338 for information about security exits available with the facility.)

To choose a selection from the menu, enter the name or number of the selection on the selection line below Enter Selection Information and press ENTER. For example, to sign on to MANTIS, type 4 on the selection line and press ENTER.

```
                                                                  0210
                 CINCOM SOFTWARE SELECTION MENU

   ENTER SELECTION INFORMATION:                                     :
   :
   * TO EXECUTE WITH CURRENT USER-ID, PRESS ENTER.
   * TO EXECUTE WITH ALTERNATE USER-ID, PRESS PF2/PF14
   * TO VIEW EXPANDED MENU DEFINITION, PRESS PF1/PF13


 1. DBAID              11. UTIL/SQL
 2. SPECTRA            12. M_SERIES
 3. MANTIS             13. CPCS
 4. DIRECTORY          14. OPC/SQL
 5. IN/SERV
 6. RESIGNON
 7. CONTROL
 8. I/SQL
 9. LOAD/SQL
 10. EASY/SQL


PA1/PA1=EXIT
PF1/PF13 WITHOUT SELECTION INFORMATION=HELP
```

Directly accessing menu options

To directly access menu options type the name or number of the desired component at the bottom of the sign-on screen and press ENTER. You must enter the name or number of the selection you want as it is on the menu, for example: MANTIS. When you enter MANTIS, the MANTIS menu is displayed.

```
          ////            ////
         /////          //////
      ////////        ////////
    //////////      //////////       WELCOME TO :
  ////////////    //////////                                        T.M.
 //////////////   /////// TTTTTT  IIIIII  SSSSSS    // XX   XX   AAAAAA
//////////////  //////////TT      II    SS       //   XX XX    AA  AA
                          TT      II      SSSS  //     XXX     AAAAAA
 //////////////  //////////TT      II        SS //     XX XX    AA  AA
  ////////////  //////////TT    IIIIII  SSSSSS//     XX   XX   AA  AA
   ////////////  //////////
     ////////// //////////
       ///////  ////////    PLEASE ENTER
        ////// //////
          //// ///         USER-ID  :                                :
                           PASSWORD :                                 :

 TO BY-PASS SELECTION SCREEN, ENTER INFORMATION:                      :
 : MANTIS                                                              :


 PA2/PA1=EXIT
```

Selecting menu options using alternate user ID and password

To select a menu option with user information that is different from the current user information, enter the name or number of the selection on the selection line below Enter Selection Information and press PF2/PF14.

```
                      CINCOM SOFTWARE SELECTION MENU

     ENTER SELECTION INFORMATION:
     : MANTIS                                                          :
       * TO EXECUT
       * TO EXECUT  | ALTERNATE USER ID PASSWORD & ENGINE SIGN ON ID   |
                    |                                                  |
                    | FOR SELECTION = MANTIS        ENTER:             |
                    |                                                  |
     1. DBAID       | ALTERNATE USER-ID:                                 |
     2. SPECTRA     | :                                   :              |
     3. MANTIS      | ALTERNATE PASSWORD:                                |
     4. DIRECTRY    | :                                   :              |
     5. IN/SERVE    | AND/OR ALTERNATE SUPRA/DRDM KERNEL NAME:           |
     6. RESIGNON    | :          :  <== USE ONLY WITH DRDM SELECTIONS    |
     7. CONTROL     |                                                    |
                    |                                                    |
                    | PRESS ENTER TO EXECUTE          PA2 PA1=EXIT       |
                    |____________________________________________________|

PF1/PF13=HELP        PA2/PA1=EXIT
```

When you press PF2/PF14, a window displays. Enter your alternate user ID and password and press ENTER. Entering an invalid user ID or password three times terminates the current MANTIS session.

Changing current user ID and password

To change your user ID and password before making another selection from the TIS/XA Software Selection Facility menu resigning on to the facility type RESIGNON or the option number (in this example, 7). The selection line below Enter Selection Information is displayed; then press ENTER.

```
                                                                  0210
                CINCOM SOFTWARE SELECTION MENU

  ENTER SELECTION INFORMATION:                                      :
  :
  * TO EXECUTE WITH CURRENT USER-ID, PRESS ENTER.
  * TO EXECUTE WITH ALTERNATE USER-ID, PRESS PF2/PF14
  * TO VIEW EXPANDED MENU DEFINITION, PRESS PF1/PF13


 1. DBAID              11. UTIL/SQL
 2. SPECTRA            12. M_SERIES
 3. MANTIS             13. CPCS
 4. DIRECTORY          14. OPC/SQL
 5. IN/SERV
 6. RESIGNON
 7. CONTROL
 8. I/SQL
 9. LOAD/SQL
 10. EASY/SQL


PA1/PA1=EXIT
PF1/PF13 WITHOUT SELECTION INFORMATION=HELP
```

The TIS/XA Software Selection Facility will return you to the initial sign-on screen.

Using quick exits

To make a quick exit from the TIS/XA Software Selection Facility sign-on screen or Software Selection menu, follow the selection name or number with /X on the “Enter Selection Information” line. When you exit from your selection, you return to the point where you invoked the TIS/XA Software Selection Facility, without displaying the menu.

Passing user data to MANTIS or CONTROL

When you use the TIS/XA Software Selection Facility, you can pass user data to a MANTIS or CONTROL: selection. The facility invokes MANTIS or CONTROL: using the current user ID and password and passes the user data to the selection as unmodified data in the COMNDATA field.

To pass user data to MANTIS or CONTROL:, enter the selection name and user data on the facility sign-on screen or at the menu. You must separate the selection name and user data with a comma or period. For example, to pass ZZZZ to MANTIS, type MANTIS,ZZZZ on the selection line and press ENTER.

To pass data to a selection and use the quick exit option, separate the /X and user data with a comma or period. For example, type MANTIS/X.ZZZZ on the selection line and press ENTER.

For information about passing user data to MANTIS programs, see *AD/Advantage MANTIS Administration OS/390, VSE/ESA*, P39-5005. For information about passing user data to CONTROL: programs, see the appropriate module manual.

Executing MANTIS programs

The TIS/XA Software Selection Facility uses the MANTIS feature of modifying the MASTER:SIGN ON and MASTER:TERMINATE programs in order to function properly. In order to allow your MANTIS application programs to return directly to the facility after execution, code CHAIN MASTER:TERMINATE as the exit point.

Executing menu options in batch mode

Execute the following TIS/XA Software Selection Facility menu options in batch mode, by:

- MANTIS
- Interactive Services
- Resign-on

To execute these options with UPDATE or READ-ONLY access, however, with READ-ONLY access you cannot execute applications that update MANTIS files.

The batch TIS/XA Software Selection Facility accepts user ID and password data using the PARM operand on the EXEC statement. For sample JCL to execute the facility in batch mode, see your sample JCL for the appropriate sample members.

The following screen shows the menu for batch mode applications:

```
           CINCOM SOFTWARE SELECTION MENU

ENTER SELECTION NAME OR NUMBER:
:                                                                    :
* TO EXECUTE WITH CURRENT   USER-ID, PRESS ENTER.
* TO EXECUTE WITH ALTERNATE USER-ID, PRESS PF2/PF14.

1. MANTIS
2. INTACTIV
3. RESIGNON









PF1/PF13=HELP        PA2/PA1=EXIT
```

Using security exits

The TIS/XA Software Selection Facility provides six user security exits:

Exit	Function
MASTER:CSFS0000 SIGNON EXIT	Checks SSF sign-on information.
MASTER:CSFS0000 MENU EXIT	Allows a tailored menu for each user.
MASTER:CSFS0000 VALUSER EXIT	Checks user information.
MASTER:CSFS0000 VALSELECT EXIT	Checks selection information.
MASTER:CSFS0000 VALALT EXIT	Checks ALT-USER information.
MASTER:CSFS0000 FAILSIGN EXIT	Expands security breach checking.

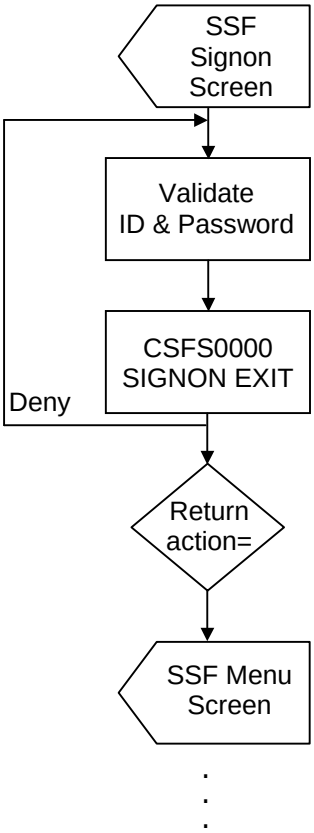
These exits are discussed in the following sections.

CSFS0000 SIGNON EXIT

This exit allows the checking of user information at sign-on. SSF passes the user ID and password to the exit, and the exit issues one of the following:

- **ACCEPT.** (Default) Allows user to continue and see the SSF main screen.
- **DENY.** Does not allow user to continue. Displays an error message on the sign-on screen, and increases the strike counter by one. When the strike counter reaches three, SSF terminates and sends the user to CSFS0000 FAILSIGN EXIT.

The following figure shows the SIGNON EXIT process:



CSFS0000 MENU EXIT

This exit allows menu selections to be eliminated from the main SSF menu based on several criteria.

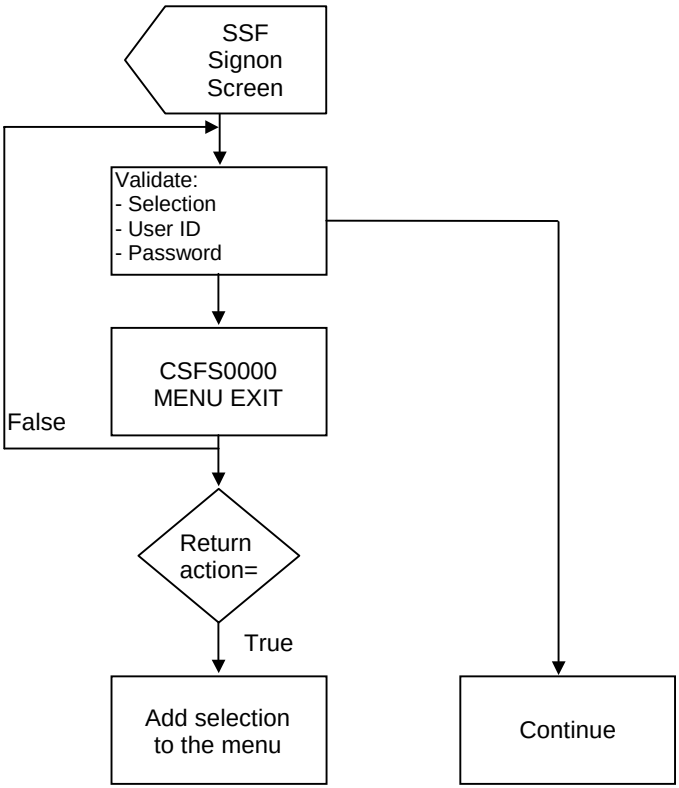
The following parameters are passed to the exit:

- Selection from the main SSF menu screen
- User ID entered at the sign-on screen
- User password entered at the sign-on screen

CSFS MENU EXIT returns the following:

- **TRUE.** (Default) Allows user to see this selection on the SSF main menu.
- **FALSE.** Does not allow user to see this selection on the SSF main menu.

The following figure shows the MENU EXIT process:



CSFS0000 VALUSER EXIT

This exit allows checking of the user information, selection, and data engine name after a selection has been made and before alternate user information is collected.

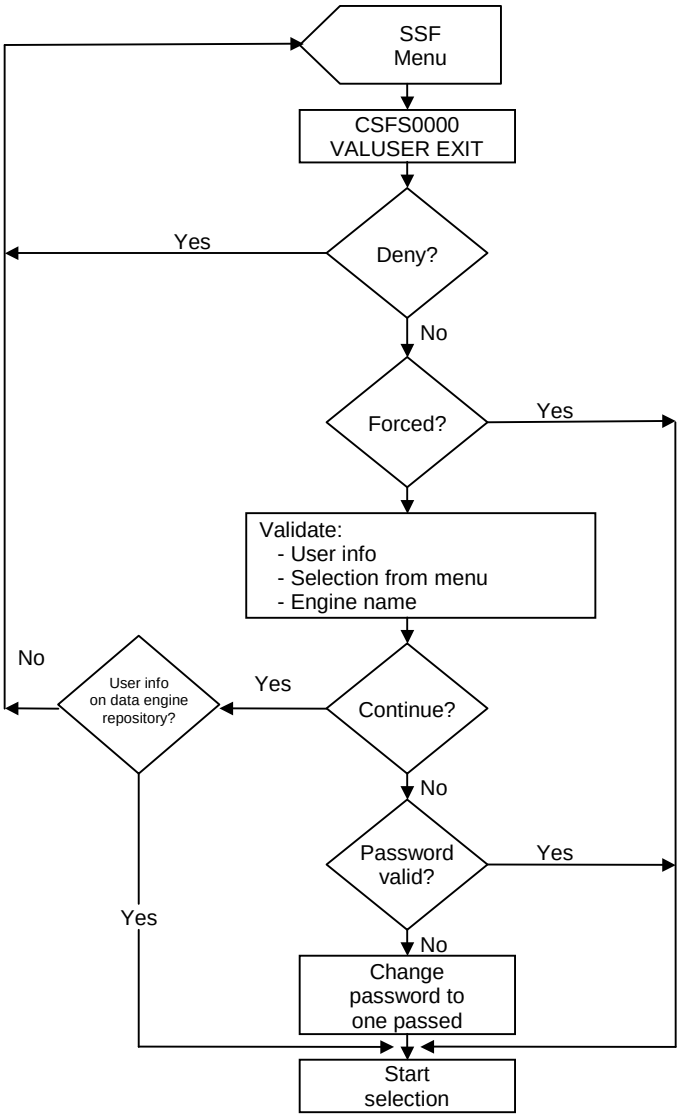
SSF provides the following input:

- The selection from the main menu
- The engine name associated with the selection from the menu
- The user ID entered at the sign-on screen
- The password entered at the sign-on screen

The exit issues one of the following:

- **CONTINUE.** (Default) The validation processing continues. If the user information is found on the data engine repository, it allows the selection to start on the named data engine. This is used only by the RDML data engine type. Other engines have no actions in the current release, and an error message appears on the SSF main screen.
- **OTHERWISE.** Does not allow the user to start the selection on the named data engine. Displays an error message on the SSF main menu.
- **DENY.** Does not allow the user to start the named data engine. Displays an error message on the SSF main menu.
- **FORCE.** Allows the user to start the selection on the named data engine. Does not check the data engine's metadata. (SSF checks for its needs.)
- **UPDATE.** The validation process continues. If the entered password does not match the password found on the data engine repository, the password on the data repository is changed to matched the one passed, and the selection starts on the named data engine. If the two passwords are the same, processing continues and allows the selection to start. This is used only by the RDML data engine type. Other engines have no actions in the current release, and an error message appears on the SSF main screen.

The following figure shows the VALUSER EXIT process:



CSFS0000 VALSELECT EXIT

This exit is automatically invoked for each selection before the selection is started and either permits or denies the user permission to execute the option.

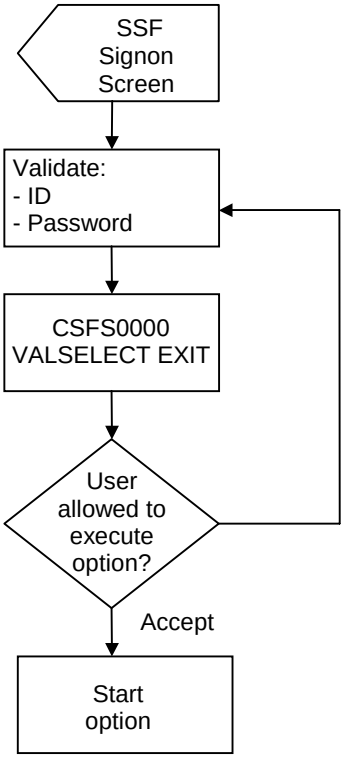
SSF passes the following to the exit:

- The selection from the SSF main menu
- The engine name associated with the selection from the menu
- The engine name entered at the alternate user information screen
- The user ID
- The user password
- The user ID entered at the alternate information screen
- The user password entered at the alternate information screen

The exit issues one of the following:

- **ACCEPT.** Permits the option to begin
- **DENY.** Does not permit the option to begin and displays an error message on the SSF main menu

The following figure shows the VALSELECT EXIT process:



CSFS0000 VALALT EXIT

This exit allows checking of the user information, alternate user information, selection, and data engine name before starting a selection using the alternate user information.

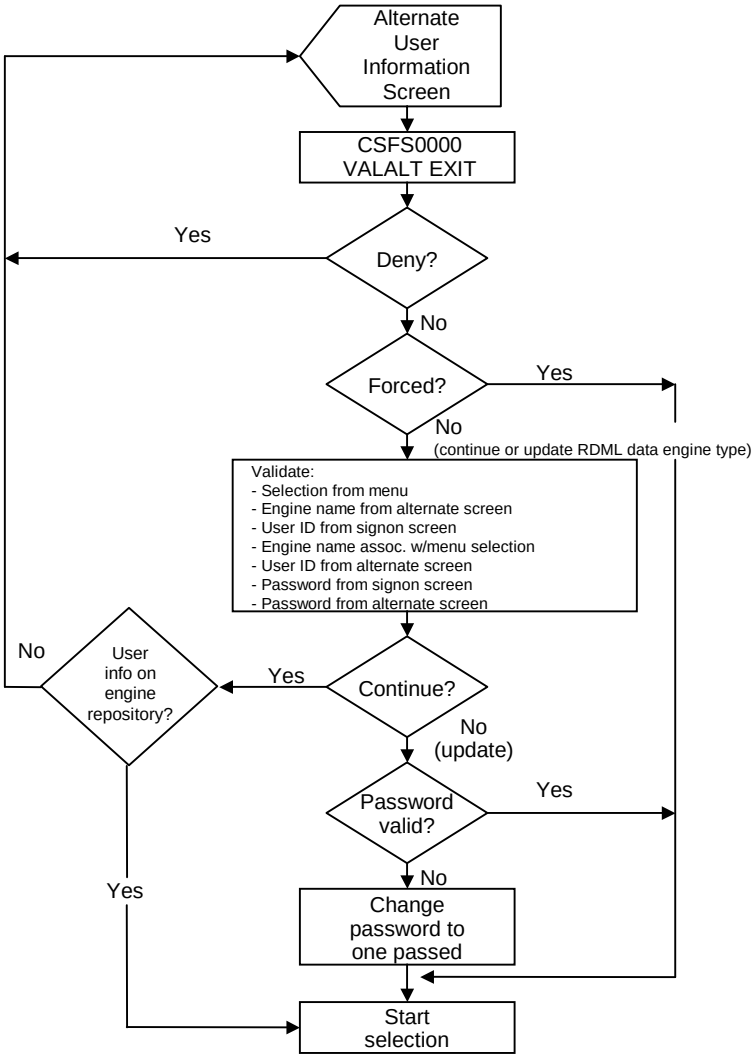
SSF passes the following information to the exit:

- Engine name associated with the selection from the main menu
- Engine name entered at the alternate user information screen
- User ID entered at the sign-on screen
- Password entered at the sign-on screen
- User ID entered at the alternate user information screen
- Password entered at the alternate user information screen

The exit issues one of the following:

- **CONTINUE.** (Default) Continues the validation process. If the alternate user information is found on the data engine repository named, allows the selection to start on the named data engine. CONTINUE is used only by the RDML data engine type. Other engine types have no actions in the current release, and an error message is displayed on the alternate user screen.
- **OTHERWISE.** Does not allow the user to start the selection on the named data engine. Displays an error message on the alternate user screen.
- **DENY.** Does not allow the user to start the named data engine with the alternate user information entered. Displays an error message on the alternate user screen. The strike counter is incremented by one. When the strike counter is greater than three, SSF terminates, and the user is sent to CSFS0000 FAILDSIGN EXIT.
- **FORCE.** Allows the user to start the selection on the named data engine. Does not check the data engine's metadata. (SSF checks for its needs.)

- **UPDATE.** The validation process continues. If the alternate password does not match the password found on the data engine repository, the password on the data repository is changed to match the one passed, and the selection starts on the named data engine. If the two passwords are the same, processing continues and allows the selection to start. UPDATE is used only by the RDML data engine type. Other engine types have no actions in the current release, and an error message is displayed on the alternate user screen.

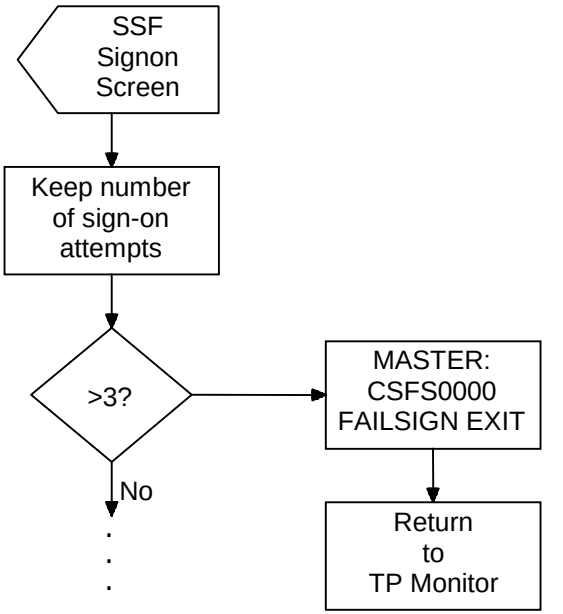


CSFS0000 FAILSIGN EXIT

This exit is invoked after a user makes three unsuccessful sign-on attempts from the initial or alternate user sign-on screen.

SSF passes the number of sign-on attempts to the exit. If the number is greater than three, the exit invokes the master CSFS0000 FAILSIGN EXIT.

The following figure shows the FAILSIGN EXIT process:



10

Generating Directory reports

Overview

Directory reports are available to assist in monitoring the structure and performance of the Directory. Use comprehensive retrieval to generate Directory reports, or the DBA can create reports using RDM views (see "Using RDM views to create Directory reports" on page 419).

Directory reports can be run in their entirety, or use trigger records to:

- Include or exclude a Directory category or information about any of the following Directory entities:
  - Schema
  - Conceptual Schema
  - User
  - File
  - Security Group
- Control the content and format of the report as follows:
  - Select information processed after a date
  - Specify a sort sequence
  - Display long or short text
  - Print full or condensed version

Generating reports

The following procedures will generate Directory reports.

OS/390

Using JCL sample member TXJDIRPT. To include trigger records in line or from a partitioned data set member, use a partitioned data set member, define the data set using these values:

```
DCB=(RECFM=F,LRECL=80,BLKSIZE=multiple of 80)
```

You must supply a trigger sign-on record with a valid user ID and password . See “Using user sign-on” on page 366 for the sign-on record format.

If you want to generate a report using only the default values for trigger records, include only a SIGNON trigger record followed by a /* immediately after the //RPTDIR.RPTR DD * statement.

VSE

Use the JCL sample TXJDIRPT.

Choose a selection from the menu screen below:

```
-------  TIS/XA SELECTION MENU  ---------  mm/dd/yy  hh:mm:ss A DISK  72%
INPUT THE FUNCTION NUMBER THEN PRESS ENTER OR A PF KEY
====>  7
1)  SUPRA SET-UP                      11)  BATCH MANTIS
2)  BATCH DIRECTORY MAINTENANCE       12)  SPECTRA
3)  DBA UTILITIES                     13)  START CENTRAL PDM
4)  EXECUTION STATISTICS              14)  STOP CENTRAL PDM
5)  START RDM APPLICATION             15)  SOFTWARE SELECTION FACILITY
6)  COMPREHENSIVE RETRIEVAL           16)  BATCH SOFTWARE SEL. FACILITY
7)  DIRECTORY REPORT                  17)  RDM PREPROCESSORS
8)  DBAID                             18)  LOGICAL VIEW DIRECTORY REPORTS
9)  NORMAL                            19)  COMMUNICATIONS MONITOR
10)  MANTIS                           20)  COMMUNICATIONS MONITOR UTILITIES





1= HELP    2=        3= END      4= RETURN    5=       6= RETV
7=         8=        9= RUN     10=          11=      12=
```

Then the Directory Reports menu displays:

```
-------  DIRECTORY REPORTS   ---------     mm/dd/yy  hh:mm:ss A DISK  72%
INPUT THE FUNCTION NUMBER THEN PRESS ENTER OR A PF KEY
====>
1)  DIRECTORY CONCEPTUAL SCHEMA
2)  DIRECTORY FULL DEFINITION
3)  DIRECTORY NAME
4)  DIRECTORY SCHEMA STRUCTURE
5)  DIRECTORY PHYSICAL STRUCTURE
6)  DIRECTORY FULL TEXT REPORT
7)  DBA USER REFERENCE
8)  DIRECTORY EXTERNAL FIELDS WHERE USED
9)  DIRECTORY NAME CROSS REFERENCE
10)  DIRECTORY SECURITY GROUP REPORT





1= HELP    2=        3= END      4= RETURN    5=       6= RETV
7=         8=        9= RUN     10=          11=      12=
```

You must use a trigger file for Directory Reports containing at least a SIGNON statement. The system issues a FILEDEF with the name RPTR INPUT * for the trigger file. To override this, issue a FILEDEF for the RPTR ddname in a user EXEC. The TIS/XA Selection Facility searches all accessed disks in the defined disk search order until it finds the file RPTR INPUT.

2. To select a particular report, for example the Directory Full Definition report, type 2 next to the command arrow and press ENTER. TSF displays the Input Parameters Edit screen for the Directory Full Definition report:

```
-------  INPUT PARAMETER EDIT  --------    mm/dd/yy  hh:mm:ss A DISK  72%
PLEASE EDIT DESIRED FIELD THEN PRESS A PF KEY OR ENTER
====>
MODULE NAME ==>   DIRDEFNP
CSIPARM MEMBER      ==>
OUTPUT DESTINATION  ==>  DISK
INPUT SOURCE        ==>  DISK
INPUT FILE NAME     ==>  SUPRABLIB MACLIB DIRDEFNP
USER EXEC(S)        ==>
USER LOADLIB        ==>  STMTLIB   SUPRALIB
               ==>
SORT DISK MODE      ==>  A
OS PARM STRING      ==>  <--N/A-->





1= HELP     2=        3= END      4= RETURN    5=       6= RETV
7= XED LIST 8=        9= RUN     10=          11=      12= EXIT
```

Notice that many of the input fields contain values. TIS/XA displays the values you entered the last time you generated this report. Examine the displayed values to ensure they are correct. The field OS PARM STRING is not used and the field USER EXEC(S) is used only when you run the Directory reports in single-task mode. Make any needed corrections and press PF9 to run the report.

The following sections discuss using trigger records to customize Directory reports. The table in the next section, “Using trigger records,” lists the Directory reports and shows whether you can use trigger records for selection.

Using trigger records

To use more than one trigger record for a report, always supply the SIGNON trigger record (see “Using user sign-on” on page 366). Within each group of trigger records, the Select/Suppress print option must remain constant. Do not select and suppress the print option for the same trigger record. The first page of each report generated using trigger records lists the trigger records used and a validity indicator.

Report name	Catg.	Schema name	CS name	User name	Maint. date	Sort	Text	File	SG	Cond SG	Sign-on
DBA User Ref. (DIRUSERP)	No*	No	No	Yes	No	No	Yes	No	No	No	Reqd
Conceptual Schema Structure (DIRCSSTP)	Yes	No	Yes	No	No	No	No	No	No	No	Reqd
Ext. Fields Where Used (DIRXFWUP)	No	Yes	No	No	Yes	No	No	No	No	No	Reqd
Full Def. (DIREFNP)	Yes	Yes	No	No	Yes	No	Yes	Yes	No	No	Reqd
Full Text Description (DIRTEXTP)	Yes	Yes	No	Yes	Yes	No	Yes	No	No	No	Reqd
Name Cross Ref. (DIRXREFP)	Yes	Yes	No	Yes	Yes	No	Yes	No	No	No	Reqd
Name (DIRNAMEP)	No	No	No	No	No	Yes	No	No	No	No	Reqd
Physical Structure (DIRSTRUP)	No	No	No	No	No	No	No	No	No	No	Reqd
Schema Structure (DIRSCSTP)	Yes	Yes	No	No	No	No	No	No	No	No	Reqd
Security Grp. (DIRSECGP)	Yes	No	No	Yes	No	No	No	No	Yes	Yes	Reqd

* Except special categories DB and EU

Using category selection

This trigger records, selects, or suppresses the category indicated by the specified category code. For the DBA User Reference Report, the trigger record selects or suppresses Logical View, Relation, and Security Group information.

Positions 1–9 CATEGORY=

Description *Required*. Must be coded as shown.

Positions 10–11 cc

Description *Required*. Specifies the category code to be selected or suppressed.

Options

AS	Access Set
AT	Attribute
CD	Directory Component Description
CS	Conceptual Schema
DB	Includes Logical View, Relation, and Security Group information from the DBA User Reference Report
DM	Domain
ED	Environment Description
EM	Edit Mask
EU	Excludes Logical View, Relation, and Security Group information from the DBA User Reference Report
FI	File
FK	Foreign Key
IR	Internal Record
KC	Key Code

Options (*cont.*)	LG	Log Group
	LV	Logical View
	NP	Navigation Primitive
	PF	Physical Field
	PR	Procedure
	RE	Relation
	RW	Reserved Word
	SC	Schema
	SK	Secondary Key
	SG	Security Group
	TA	Tables
	US	User
	XF	External Field

Consideration For the DBA User Reference Report, you can omit this parameter. The report then prints as if you coded DB.

Positions 13–21 *PRINT* $\left\{\frac{\textbf{YES}}{\textbf{NO}}\right\}$

Description *Optional*. Specifies print selection or suppression.

Default PRINT=Yes

Example Submit the following trigger record to print all entities within the Logical View category:

```
00000000011111111112222222222333333333344444444445555555555666666667777777778
12345678901234567890123456789012345678901234567890123456789012345890123456789 0
                                    CATEGORY=LV PRINT=YES
```

NOTE

Not all categories are available for every report. The following table shows the categories to select for each report that uses category selection trigger records. When you submit more than one trigger record, the first trigger record determines the print option for all categories.

Directory report	Valid categories
DBA User Reference	DB, EU(DIRUSERP)
Conceptual Schema Structure (DIRCSSTP)	CS, DM
Full Definition (DIRDEFNP)	AS, AT, CD, CS, DM, ED, EM, FI, FK, IR, KC, LV, NP, PF, PR, RE, RW, SC, SG, SK, TA, US, XF
Full Text Description (DIRTEXTP)	AS, AT, CD, CS, DM, ED, EM, FI, FK, IR, LV, NP, PF, PR, RE, RW, SC, SG, TA, US, XF
Name Cross Reference (DIRXREFP)	AS, AT, CD, CS, DM, ED, EM, FI, FK, IR, KC, LV, NP, PF, PR, RE, RW, SC, SK, TA, US, XF
Schema Structure (DIRSCSTP)	ED, FI, LV, XF
Security Group (DIRSECGP)	SG, US

Using schema name selection

This trigger record selects or suppresses the specified schema. You can enter up to 64 of these trigger records for one report.

Positions 1–3 SC=

Description *Required*. Must be coded as shown.

Positions 4–33 entity name

Description *Required*. Specifies the schema to be selected or suppressed.

Format 1–8 alphanumeric or special characters (#, $, and @)

Positions 39–47 PRINT= $\left\{\frac{\textbf{YES}}{\textbf{NO}}\right\}$

Description *Optional*. Specifies print selection or suppression.

Default PRINT=Yes

Example Submit the following trigger record to print all entities within the DEMOSCH Schema:

```
0000000001111111111222222222233333333334444444444555555555566666666677777777778
1234567890123456789012345678901234567890123456789012345678901234589012345678 90
                    SC=DEMOSCH                              PRINT=YES
```

Using Conceptual Schema name selection

This trigger record selects or suppresses the specified Conceptual Schema. You can enter up to 64 of these trigger records for a Directory Conceptual Schema Structure Report (DIRCSSTP).

Positions 1–3 CS=

Description *Required*. Must be coded as shown.

Positions 4–33 entity name

Description *Required*. Specifies the Conceptual Schema to be selected or suppressed.

Format 1–30 alphanumeric or special characters (#, $, and -)

Positions 39–47 PRINT= {**YES** / **NO**}

Description *Optional*. Specifies print selection or suppression.

Default PRINT=YES

Options

PRINT=YES	Print this entity
PRINT=NO	Do not print this entity

Example Submit the following trigger record to print all entities within the BURRY- CONCEPTUAL-SCHEMA Conceptual Schema:

```
000000000111111111122222222223333333333444444444455555555556666666667777777778
123456789012345678901234567890123456789012345678901234567890123458901234567890
                    CS=BURRY-CONCEPTUAL-SCHEMA            PRINT=YES
```

Using user name selection

This trigger record selects or suppresses all information related to the specified User. You can enter up to 64 of these trigger records for one report.

Positions 1–3 US=

Description *Required*. Must be coded as shown.

Positions 4–33 user name

Description *Required*. Specifies the User to be selected or suppressed.

Format 1–30 alphanumeric characters

Positions 39–47 PRINT= $\left\{\begin{matrix}\textbf{YES}\\ \hline \textbf{NO}\end{matrix}\right\}$

Description *Optional*. Specifies print selectivity or suppression.

Default PRINT=YES

Example Submit the following trigger record to print all entities related to the user SMITH:

```
000000000111111111122222222223333333333444444444455555555556666666667777777778
123456789012345678901234567890123456789012345678901234567890123458901234567890
                    US=SMITH                              PRINT=YES
```

Using maintenance date selection

This trigger record selects all Directory information that has been added or updated using Directory Maintenance on or after the specified date.

Positions 1–17 MAINTENANCE AS OF

Description *Required*. Must be coded as shown.

Positions 19–24 mmddyy

Description *Required*. Specifies the date.

Format 6 numeric characters representing the month, day, and year

Example Submit the following trigger record to print all Directory information processed by Directory Maintenance on or after February 1, 1990:

```
00000000011111111112222222222333333333344444444445555555555666666667777777778
12345678901234567890123456789012345678901234567890123456789012345890123456789 0
MAINTENANCE AS OF 020190
```

Using sort option

This trigger record indicates whether to sort information on the Directory Name Report (DIRNAMEP) by definition number, entity name, or qualifier definition number. If you omit this trigger record, information will be presented in definition number sequence.

Positions 1–5 SORT=

Description *Required*. Must be coded as shown.

Positions 6–9 sort option

Description *Required*. Specifies the report sequence.

Options

DEFN	Definition number
NAME	Entity name
QUAL	Qualifier definition number

Example Submit the following trigger record to sort all entities by entity name:

```
00000000011111111112222222222333333333344444444445555555555666666667777777778
12345678901234567890123456789012345678901234567890123456789012345890123456789 0
SORT=NAME
```

Using text selection

This trigger record selects or suppresses the text related to entities on the specified report.

Positions 1–9 TEXTOPTS=

Description *Required*. Must be coded as shown.

Positions 10–11 **{ST | LT}**

Restriction LT is valid only for the Directory Full Text Description Report (DIRTEXTP).

Description *Required*. Indicates whether to print short text or long text.

Options

ST	Print any short text.
LT	Do not print text.

Consideration If you omit this trigger record for the Directory Full Text Description Report, both long text and short text is printed.

Positions 13–21 PRINT= **{YES | NO}**

Description *Optional*. Specifies print selection or suppression.

Default YES

Example Submit the following trigger record to print any short text on the specified report:

```
0000000001111111111222222222233333333334444444444555555555566666666677777777778
1234567890123456789012345678901234567890123456789012345678901234589012345678 90
TEXTOPTS=ST PRINT=YES
```

Using file selection

This trigger record identifies the files to be included on the Directory Full Definition Report (DIRDEFNP). If you omit this trigger record, all files are included on the report.

Positions 1–5 FILE=

Description *Required*. Must be coded as shown.

Positions 6–9 file option

Description *Required*. Specifies the files to be selected.

Default ALL.

Options

ALL	All files
DIRM	Only Directory files (the System Log file, the Task Log file, the Statistics file and all files that begin C$-)
USER	All files except Directory files

Example Submit the following trigger record to print only the Directory files and the Internal Records qualified by the Directory files:

```
0000000001111111111222222222233333333334444444444555555555566666666677777777778
1234567890123456789012345678901234567890123456789012345678901234589012345678 90
CATEGORY=FI
FILE=DIRM
CATEGORY=IR
```

General considerations

- If you include more than one File Selection trigger record per report, only the last one is used.
- Use of this trigger record also affects subreports about entities that are qualified by files (Internal Records, Secondary Keys, Key Codes, and Physical Fields).

Using Security Group selection

This trigger record selects or suppresses all information related to the specified Security Group. You can enter up to 64 of these trigger records for one report.

Positions 1–3 SG=

Description *Required*. Must be coded as shown.

Positions 4–34 Security Group name

Description *Required*. Specifies the Security Group to be selected or suppressed.

Format 1–30 alphanumeric or special characters (#, $, -)

Positions 39–47 PRINT= $\left\{\begin{matrix}\textbf{YES}\\ \hline \textbf{NO}\end{matrix}\right\}$

Description *Optional*. Specifies print selection or suppression.

Default PRINT=YES

Example Submit the following trigger record to print all entities related to the Security Group TESTSG:

```
000000000111111111122222222223333333333444444444455555555556666666667777777778
123456789012345678901234567890123456789012345678901234567890123458901234567890
SG=TESTSG                              PRINT=YES
```

Using condense option

This trigger record specifies whether the expanded or condensed version of the Directory Security Report (DIRSECGP) is to be printed. If you omit this trigger record, the expanded report is produced.

Positions 1–9 CONDENSE=

Description *Required*. Must be coded as shown.

Position 10 condense option

Description *Required*. Indicates whether to produce the condensed report.

Default N

Options Y Print condensed report.

N Print expanded report.

Example Submit the following trigger record to print the condensed version of the Directory Security Report:

```
00000000011111111112222222222333333333344444444445555555555666666667777777778
12345678901234567890123456789012345678901234567890123456789012345890123456789 0
CONDENSE=Y
```

Using user sign-on

This trigger record controls access to the Directory Security Report to protect against unauthorized use of Security Group data. This record must be the first trigger record specified. When you use this trigger record, the system validates your user ID and password, checks that you are related to the Logical View for the report, and verifies that you are a DBA. If any of these checks fail, the system terminates the report and Security Group output.

NOTE

A user SIGNON trigger record is required for every Directory report.

Positions 1–7 +SIGNON

Description *Required*. Must be coded as shown.

Positions 9–38 user ID

Description *Required*. Identifies the User submitting the report.

Format 1–30 alphanumeric or special characters (#, $, -)

Positions 40–69 password

Description *Required*. Specifies the password of the User submitting the report.

Format 1–30 alphanumeric characters

Example If your user ID is SMITH and your password is M9923, submit the following trigger record to produce the Directory Security Report:

```
0000000001111111111222222222233333333334444444444555555555566666666677777777778
1234567890123456789012345678901234567890123456789012345678901234589012345678 90
+SIGNON SMITH                          M9923
```

Using Directory reports

The Directory reports are as follows:

- DBA User Reference Report
- Directory Conceptual Schema Structure Report
- Directory External Fields Where Used Report
- Directory Full Definition Report
- Directory Full Text Description Report
- Directory Name Cross Reference Report
- Directory Name Report
- Directory Physical Structure Report
- Directory Schema Structure Report
- Directory Security Group Report

Each of these reports shows standard header information, as shown in the following example.

```
CINCOM SYSTEMS, INC.          DIRUSERP - CBA USER REFERENCE      JUN. O3. 1997 16:32 24 PAGE:
                                  TIS/XA DIRECTORY
```

Using the DBA User Reference Report

The DBA User Reference Report (DIRUSERP) lists user descriptions and attribute data alphabetically by user ID. You can submit trigger records to select or suppress specified users and to print any short text associated with each user.

The DBA, RDM users, and Comprehensive Retrieval users use this report to review the logical views, relations, Security Groups, procedures, and associated information available to each user.

Multiple pages are produced for each user. The first page provides user identification information, procedures available to the user, along with their type flag and associated logical view name. Subsequent pages are produced for each logical view and relation.

The logical view information includes the qualifying schema, the access sets related to the logical view, the related external fields, and the physical fields, internal records, and files related to each external field. The relation information includes the qualifying conceptual schema, the associated domains, the attributes making up each relation, and whether the attributes are excluded from the user.

You may not want RDM or Comprehensive Retrieval users to have the logical view, relation, and Security Group information. To suppress printing of this information, enter CATEGORY=EU in positions 1–11. To print the complete report, enter CATEGORY=DB in positions 1–11 or omit this trigger record.

The following example shows the DBA User Reference Report without logical view, relation, and Security Group information (code EU). The next example shows the complete DBA User Reference Report (code DB). The number of pages produced depends on the amount of information available to the user.

DBA User Reference Report (User Information)

```
CINCOM SYSTEMS, INC.   DIRUSERP - CBA USER REFERENCE   AUG. O3. 1997 16:32 24 PAGE:

                                TIS/XA DIRECTORY
  USER:  EDUC-DBA
                            USER NAME:  CINCOM SYSTEMS, INC.  LANGUAGE:   ENGLISH

                            ADDRESS-1:  2300 MONTANA AVENUE   DBA OPTIONS:  Y
                            ADDRESS-2:  CINCINNATI
                            ADDRESS-3:     OHIO 45211
                            PHONE #:    662-2300
                            SHORT TEXT: ** NO SHORT TEXT **
```

DBA User Reference Report (Logical View Information)

```
CINCOM SYSTEMS, INC.   DIRUSERP - DBA USER REFERENCE   AUG. O3. 1997 16:32 24 PAGE:

                                TIS/XA DIRECTORY
USER:  EDUC-DBA

RELATION:  BRANCH          FOR CONCEPTUAL SCHEMA:  BURRYS-CONCEPTUAL - SCHEMA

                       ATTRIBUTE  AT EXCLUDED   DOMAIN

BRANCH-ADOR                       DM-ADDRESSES
BRANCH-CITY                       DM-CITIES
BRANCH-DEL-ROUTE                  DM-ROUTES
BRANCH-NAME                       DM-NAMES
BRANCH-NO                         DM-BRANCH-IDENTIFIERS
BRANCH-REGION                     DM-REGION-  IDENTIFIERS
BRANCH-SLS-QUOTA                  DM-MONIES
BRANCH-STATE                      DM-STATES
BRANCH-STF-QUOTA                  DM-QUANTITIES
BRANCH-ZIPCODE                    DM-ZIPCODES
USER:  EDUC-DBA
RELATION:  CUSTOMER            FOR CONCEPTUAL SCHEMA:  BURRYS-CONCEPTUAL - SCHEMA
                       ATTRIBUTE  AT EXCLUDED   DOMAIN
BRANCH-ADOR                       DM-ADDRESSES
BRANCH-CITY                       DM-CITIES
BRANCH-DEL-ROUTE                  DM-ROUTES
BRANCH-NAME                       DM-NAMES
BRANCH-NO                         DM-BRANCH-IDENTIFIERS
BRANCH-REGION                     DM-REGION-  IDENTIFIERS
BRANCH-SLS-QUOTA                  DM-MONIES
BRANCH-STATE                      DM-STATES
BRANCH-STF-QUOTA                  DM-QUANTITIES
BRANCH-ZIPCODE                    DM-ZIPCODES

USER:  EDUC-DBA

RELATION:  GROUP               FOR CONCEPTUAL SCHEMA:  BURRYS-CONCEPTUAL - SCHEMA
                       ATTRIBUTE  AT EXCLUDED   DOMAIN
GROUP-CODE                        DM-GROUP-IDENTIFIERS
GROUP-DESC                        DM-DESCRIPTIONS
USER:  EDUC-DBA
RELATION:  INVLINE             FOR CONCEPTUAL  SCHEMA :  BURRYS-CONCEPTUAL - SCHEMA
                       ATTRIBUTE  AT EXCLUDED   DOMAIN
```

Using the Directory Conceptual Schema Structure Report

The Directory Conceptual Schema Structure Report (DIRCSSTP) produces three subreports that describe the structure of each conceptual schema:

- Listing the schemas related to the conceptual schema
- Listing the relations related to the conceptual schema along with the attributes and relationships for each relation
- Listing all domains in the Directory

The following example labeled “Directory Conceptual Schema Structure Report (Related Schemas)” shows the subreport listing all schemas related to this conceptual schema. The next example, labeled “Directory Conceptual Schema Structure Report (Related Relations),” shows the subreport listing each relation and its attributes, its related attributes, primary keys, and foreign keys. The third example, labeled “Directory Conceptual Schema Structure Report (Domains),” shows the subreport listing all domains in the Directory and their attributes along with the qualifying relation and conceptual schemas for the attributes.

You can submit Conceptual Schema Name Selection trigger records to select or suppress specified conceptual schemas (See “Using Directory reports” on page 367). You can submit Category Selection trigger records to select or suppress the Related Schemas and Relations subreport (category=CS) or the Domain subreport (category=DM) (See “Generating reports” on page 350).

You cannot use any other category selection records for this report.

Directory Conceptual Schema Structure Report (Related Schemas)

```
CINCOM SYSTEMS, INC. DIRCSSTP - DIRECTORY CONCEPTUAL SCHEMA STRUCTURE REPORT  JUNE 27, 1997 16:29:36 PAGE:1

                                    TIS/XA DIRECTORY
                                    CONCEPTUAL SCHEMA STRUCTURE FOR BURRYS-CONCEPTUAL-SCHEMA

SCHEMAS RELATED TO CONCEPTUAL SCHEMA BURRYS-CONCEPTUAL-SCHEMA  :
                                                      BURRYSCH
```

Directory Conceptual Schema Structure Report (Related Relations)

```
CINCOM SYSTEMS, INC.  DIRCSSTP - DIRECTORY CONCEPTUAL SCHEMA STRUCTURE REPORT  JULY 27, 1997 16:29:36 PAGE: 2

                                    TIS/XA DIRECTORY
                                    CONCEPTUAL SCHEMA STRUCTURE FOR BURRYS-CONCEPTUAL-SCHEMA

RELATION:  BRANCH

  RELATION TYPE             I - INDEPENDENT ENTITY
  SUBTYPE                   NO
  EXTENSION                 NO
  ORDERED PRIMARY KEY       YES
  NUMBER OF TUPLES          418
  DD-NAME                   E$BR
  DATA SET NAME             SUPRA.E$BR
  FILE NAME                 E$BR
  CONSISTENT                YES
                                 ATTRIBUTES:        FOR RE:  BRANCH
                            ATTRIBUTE LENGTH DATA FORMAT DOMAIN      RELATED EXT. FLD   SCHEMA
BRANCH-NO                   00004 C - CHARACTER     DM-BRANCH-IDENTIFIERS   BRANCH-NO  BURRYSCH
BRANCH-NAME                 00020 C - CHARACTER     DM-NAMES BRANCH-NAME BURRYSCH
BRANCH-ADDR                 00020 C - CHARACTER     DM-ADDRESSES BRANCH-ADDR BURRYSCH
BRANCH-CITY                 00013 C - CHARACTER     DM-CITIES    BRANCH-CITY BURRYSCH
BRANCH-STATE                00002 C - CHARACTER     DM-STATES    BRANCH-STATE   BURRYSCH
BRANCH-ZIPCODE              00005 Z - CHARACTER     DM-ZIPCODES  BRANCH-ZIPCODE BURRYSCH
BRANCH-REGION               00003 C - CHARACTER     DM-REGION-IDENTIFIERS   BRANCH-REGION  BURRYSCH
BRANCH-DEL-ROUTE            00002 C - CHARACTER     DM-ROUTES    BRANCH-ROUTE   BURRYSCH
BRANCH-SLS-QUOTA            00009 Z - ZONED DECIMAL DM-MONIES    BRANCH-SLS-QUOTA   BURRYSCH
BRANCH-STF-QUOTA            00005 Z - ZONED DECIMAL DM-QUANTITIES   BRANCH-STF-QUOTA   BURRYSCH
                            PRIMARY KEY:                 FOR RE:  BRANCH
                            ATTIBUTES     DOMAIN
                                 BRANCH-NO          DM-BRANCH-IDEBTIFIERS
                            RELATED FOREIGN KEYS    QUALIFYING RELATION

                                 CUT-AT             INVOICE
                                 DISTRIBUTED-TO     MANIFEST
                                 OPENED-CREDIT-ACCOUNT-AT    CUSTOMER
                                 STORED-AT          STOCK
```

```
CINCOM SYSTEMS, INC. DIRCSSTP - DIRECTORY CONCEPTUAL SCHEMA STRUCTURE REPORT  JULY 27, 1997 16:29:36 PAGE: 2

                                    TIS/XA DIRECTORY
                                    CONCEPTUAL SCHEMA STRUCTURE FOR CSI-SP-NORMAL-CS


RELATION:  CSI-SP-ATTRIBUTE (CONT'D)

  RELATED FOREIGN KEYS          QUALIFYING RELATION

    CSI-SP-ATTRIBUTE -1            CSI-SP-FK-AT-RELSHIP

FOREIGN KEYS:                             FOR RE:  CSI-SP-ATTRIBUTE

  FOREIGN KEY:  CSI-SP-DM-AT-RELSHIP

    PRIMARY KEY RELATION        CSI-SP-DM-AT-RELSHIP
    FOREIGN KEY TYPE            I  -  IGNORE

    FOREIGN KEY ATTRIBUTES      DOMAIN

                           DM-DEFNO                    CSI-SP-DM-DEFNO
                           DEFNO                       CSI-SP-AT-DEFNO

  FOREIGN KEY:  CSI-SP-DOMAIN

    PRIMARY KEY RELATION        CSI-SP-DOMAIN
    FOREIGN KEY TYPE            I  -  IGNORE

  FOREIGN KEY ATTRIBUTES             DOMAIN

                           DM-DEFNO                    CSI-SP-DM-DEFNO

  FOREIGN KEY:  CSI-SP-RE-AT-RELSHIP

    PRIMARY KEY RELATION        CSI-SP-RE-AT-RELSHIP
    FOREIGN KEY TYPE            I  -  IGNORE

    FOREIGN KEY ATTRIBUTES      DOMAIN

                           RE-DEFNO           CSI-SP-RE-DEFNO
                           DEFNO              CSI-SP-AT-DEFNO
```

Directory Conceptual Schema Structure Report (Domains)

```
CINCOM SYSTEMS, INC.  DIRCSSTP - DIRECTORY CONCEPTUAL SCHEMA STRUCTURE REPORT JUNE 27, 1997 16:29:36 PAGE: 2

                                        TIS/XA DIRECTORY

                                        REPORT FOR ALL DOMAINS IN THE DIRECTORY

  DOMAIN   CSI-SP-AT-DELETE-INDICATOR

                            FUNCTION      STRING
                            UNIT
                            DATA FORMAT        C  -  CHARACTER
                            LENGTH        1
                            NUMBER OF DECIMAL PLACES    0
                            SIGNED OPTION NO
                            NULLS ALLOWED OPTION   NO
                            NULL VALUE
                            DEFAULT VALUE
                            RETRIEVAL VALIDATION OPTION NO
                            VALIDATION OPTION
                            VALIDATION MINIMUM
                            VALIDATION MAXIMUM
                            VALIDATION TABLE NAME
                            VALIDATION EXIT
                            RELATED ATTRIBUTES FOR CS:   CSI-SP-NORMAL-CS
                                 DEL-SWITCH             FOR RE:  CSI-SP-ATTRIBUTE
  DOMAIN   CSI-SP-AT-MOVED-INDICATOR

                            FUNCTION      STRING
                            UNIT
                            DATA FORMAT        C  -  CHARACTER
                            LENGTH        1
                            NUMBER OF DECIMAL PLACES    0
                            SIGNED OPTION NO
                            NULLS ALLOWED OPTION   NO
                            NULL VALUE
                            DEFAULT VALUE
                            RETRIEVAL VALIDATION OPTION NO
                            VALIDATION OPTION
                            VALIDATION MINIMUM
                            VALIDATION MAXIMUM
                            VALIDATION TABLE NAME
                            VALIDATION EXIT
                            RELATED ATTRIBUTES FOR CS:   CSI-SP-NORMAL-CS
                                 MOVE          FOR RE:  CSI-SP-ATTRIBUTE
```

```
CINCOM SYSTEMS, INC. DIRCSSTP - DIRECTORY CONCEPTUAL SCHEMA STRUCTURE REPORT JUNE 27, 1997 16:29:36 PAGE: 2

                                          TIS/XA DIRECTORY

                                          REPORT FOR ALL DOMAINS IN THE DIRECTORY

  DOMAIN   CSI-SP-AT-NAME

                            FUNCTION      STRING
                            UNIT
                            DATA FORMAT        C  -  CHARACTER
                            LENGTH        30
                            NUMBER OF DECIMAL PLACES    0
                            SIGNED OPTION NO
                            NULLS ALLOWED OPTION    NO
                            NULL VALUE
                            DEFAULT VALUE
                            RETRIEVAL VALIDATION OPTION NO
                            VALIDATION OPTION
                            VALIDATION MINIMUM
                            VALIDATION MAXIMUM
                            VALIDATION TABLE NAME
                            VALIDATION EXIT

                            RELATED ATTRIBUTES FOR CS:   CSI-SP-NORMAL-CS

                                 NAME     FOR RE:  CSI-SP-ATTRIBUTE
  DOMAIN   CSI-SP-AT-REVIEW-INDICATOR

                            FUNCTION      STRING
                            UNIT
                            DATA FORMAT        C  -  CHARACTER
                            LENGTH        1
                            NUMBER OF DECIMAL PLACES    0
                            SIGNED OPTION NO
                            NULLS ALLOWED OPTION    NO
                            NULL VALUE
                            DEFAULT VALUE
                            RETRIEVAL VALIDATION OPTION NO
                            VALIDATION OPTION
                            VALIDATION MINIMUM
                            VALIDATION MAXIMUM
                            VALIDATION TABLE NAME
                            VALIDATION EXIT

                            RELATED ATTRIBUTES FOR CS:   CSI-SP-NORMAL-CS
                                     REVIEWED           FOR RE:  CSI-SP-ATTRIBUTE
```

Using the Directory External Fields Where Used Report

The Directory External Fields Where Used Report (DIRXFWUP) lists external fields within a schema in alphabetical order. The associated physical field, internal record, and file are shown next to each external field along with the logical views that use the field. Use this report to determine whether an external field is in a specified schema and, if so, check how the external field is used. You can submit trigger records for schema selection and maintenance date selection. The following example shows a Directory External Fields Where Used Report.

```
CINCOM SYSTEMS, INC.      DIRXFWUP - DIRECTORY XF WHERE USED      JUNE 27, 1997 16:29:36 PAGE: 2

                                    TIS/XA DIRECTORY
                                                MAINTENANCE AS OF 00/00/00
                                    EXTERNAL FIELDS FOR BURRYS SCHEMA

EXTERNAL FIELDS           PHYSICAL FIELD   INT REC  FILE   REFERENCED IN LOGICAL VIEW(S):

CSI-DIR-COPY-USER-SCHEMA    COPYSCHEM  BASE COPY DIRECTORY-COPY

CSI-DIR-DA-AS-LINE-NUMBER   C$-D14LN   14   C$-D DIRECTORY-COPY
                                                DIRECTORY-SC-STRU-REPORT
                                                DIRECTORY-CS-STRU-REPORT
                                                DIRECTORY-DEFINITION
                                                DIRECTORY-USER-REFERENCE

CSI-DIR-DA-AS-TEXT          C$-D14TL   14   C$-D DIRECTORY-COPY
                                                DIRECTORY-SC-STRU-REPORT
                                                DIRECTORY-CS-STRU-REPORT
                                                DIRECTORY-DEFINITION
                                                DIRECTORY-USER-REFERENCE
```

Using the Directory Full Definition Report

The Directory Full Definition Report (DIRDEFNP) generates subreports by category and for all entities within the category. You can generate subreports for any of the following categories (shown with their trigger record selection code). You can enter Category Selection trigger records for any of the categories shown as subreports.

Access Set (AS)	Internal Record (IR)
Component Description (CD)	Relation (RE)
Environment Description (ED)	Secondary Key (SK)
Foreign Key (FK)	Conceptual Schema (CS)
Logical View (LV)	Edit Mask (EM)
Procedure (PR)	File (FI)
Schema (SC)	Key Code (KC)
User (US)	Physical Field (PF)
Attribute (AT)	Reserved Word (RW)
Domain (DM)	Tables (TA)
External Field (XF)	

Each subreport lists all of the entity names within the category in alphabetical order plus the associated attribute data. The attribute data consists of any information entered during the maintenance session in which the entity was created or last updated. Optionally, you can include any short text for an entity. This report gives you a record of the state of the Directory at a given time and allows you to validate the integrity of the Directory.

The following examples show the Directory Full Definition report:

General considerations

- You can submit trigger records for category selection, schema name selection, maintenance date selection, short text selection, and file selection.
- You can submit the Text Selection trigger record only in conjunction with another type of trigger record.
- The report for the Environment Description category includes buffer pool information.

Directory Full Definition Report (Access Sets)

```
CINCOM SYSTEMS, INC.        DIRXFWUP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                        TIS/XA DIRECTORY
                                                   MAINTENANCE AS OF 00/00/00
                            DIRECTORY DEFINITION FOR ACCESS SETS

ACCESS SET NAME:  ASD-ADD-INVOICE FOR SCHEMA:  BURRYSCH

ACCESS SET TYPE                       L - LUV
CONSISTENT                            YES
GENERATION INDICATOR                  NO

LAST UPDATE:  10:34:21  11/15/92  V:  USER:  CSI-DBA

ACCESS STATEMENTS:          0100 KEY  INVOICE-NO
                                      0200 INVOICE-SLSPRSN
                                      0300 INVOICE-TOTAL
                                      0400 INVOICE-BRANCH
                                      0500 INVOICE-DATE
                                      0600 INVOICE-CUSTOMER
                                      0700 KEY INVLINE-PRODUCT
                                      0800 INVLINE-QNTY
                                      0900 INVLINE-PRICE
                                      1000 PRODUCT-DESC
                                      1100 ACCESS INVC   WHERE INVOICE-NO      = INVOICE-NO    ALLOW INSERT
                                      1200 ACCESS INVL   WHERE INVLINE-INVOICE = INVOICE-NO    ALLOW INSERT
                                      1300 ACCESS PROD ONCE  WHERE PRODUCT-CODE    = INVLINE-PRODUCT

SHORT TEXT:   *NONE*
```

Directory Full Definition Report (Attributes)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                   TIS/XA DIRECTORY
                                                 MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR ATTRIBUTES

AT NAME:  ADD-UPDATE      RELATION:  CSI-SP-SCHEMA  CONC. SCHEMA:  CSI-SP-NORMAL-CS

                          LAST UPDATE:  11:42:07  11/11/92  V:  0  USER:  CSI-DBA

                          DOMAIN                    CSI-SP-TIME-STAMP
                          PRIMARY KEY               NO
                          FUNCTION                  STRING
                          UNIT
                          DATA FORMAT               C - CHARACTER
                          EXTERNAL LENGTH           16
                          NUMBER OF DECIMAL PLACES  0
                          SIGNED OPTION             NO
                          LENGTH OF EDITED FIELD    0
                          TRANSLATE/EDIT SWITCH
                          TRANSLATE/EDIT NAME
                          PRINT HEADING             ADD: DATE & TIME
                          DATA TRANSLATION EXIT

SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*
AT NAME:  ADD-UPDATE      RELATION:  CSI-SP-SCHEMA  CONC. SCHEMA:  CSI-SP-NORMAL-CS

                          LAST UPDATE:  11:43:07  11/11/92  V:  0  USER:  CSI-DBA

                          DOMAIN                    CSI-SP-TIME-STAMP
                          PRIMARY KEY               NO
                          FUNCTION                  STRING
                          UNIT
                          DATA FORMAT               C - CHARACTER
                          EXTERNAL LENGTH           16
                          NUMBER OF DECIMAL PLACES  0
                          SIGNED OPTION             NO
                          LENGTH OF EDITED FIELD    0
                          TRANSLATE/EDIT SWITCH
                          TRANSLATE/EDIT NAME
                          PRINT HEADING             ADD: DATE & TIME
                          DATA TRANSLATION EXIT
SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Component Descriptions)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                TIS/XA DIRECTORY
                                              MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR DIRECTORY COMPONENT DESCRIPTIONS

                          CD NAME:  CD#CSXM0000

                              LAST UPDATE:  14:13:02  11/11/92  V:  1  USER:  CSI-DBA

                              STATION PRIORITY            0
                              CONSOLE OPTION              NO
                              SECURITY CONSOLE OPTION     YES
                              SECURITY TIMEOUT INTERVAL   600
                              PASSWORD CHECK OPTION       YES
                              MASTER DBA USER-ID          CSI-DBA
                              OK VERIFICATION MESSAGE     YES
                              DIRECTORY NAME              TIS/XA DIRECTORY
                              INSTALLATION NAME

                              SHORT TEXT:  DEFINED COMPONENT DESCRIPTION
```

Directory Full Definition Report (Conceptual Schemas)

```
CINCOM SYSTEMS, INC.        DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                  TIS/XA DIRECTORY
                                                 MAINTENANCE AS OF 00/00/00
                            DIRECTORY DEFINITION FOR CONCEPTUAL SCHEMAS

                            CS NAME:  BURRYS-CONCEPTUAL-SCHEMA

                            LAST UPDATE:  10:52:32  11/15/92  V:  0011  USER:  CSI-DBA

                                 RELEASE LEVEL      002.000
                                 CONSISTENT              YES

                                 SHORT TEXT:  NONE
*-------------------------------------------------------------------------------------------------------*
                            CS NAME:  CSI-DEFAULT-CS

                            LAST UPDATE:  11:03:30  11/12/92  V:  0000  USER:  **CSI-DIRECTORY-CREATE-USER**

                                 RELEASE LEVEL      002.000
                                 CONSISTENT              YES

                                 SHORT TEXT:  CONCOM DEFINED DEFAULT CONCEPTUAL SCHEMA
*-------------------------------------------------------------------------------------------------------*
                            CS NAME:  CSI-SP-NORMAL-CS

                            LAST UPDATE:  09:32:00  11/11/92  V:  0016  USER:  CSI-DBA

                                 RELEASE LEVEL      002.000
                                 CONSISTENT              YES

                                 SHORT TEXT:  NONE
*-------------------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Domains)

```
CINCOM SYSTEMS, INC.        DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                  TIS/XA DIRECTORY
                                                 MAINTENANCE AS OF 00/00/00
                            DIRECTORY DEFINITION FOR DOMAINS

                            DM NAME:  CSI-DEFAULT-DM

                            LAST UPDATE:  17:15:25  11/10/92  V:  0   USER:  **CSI-DIRECTORY-CREATE-USER**

                            FUNCTION                    STRING
                            UNIT
                            DATA FORMAT                 C - CHARACTER
                            LENGTH                      1
                            NUMBER OF DECIMAL PLACES    0
                            SIGNED OPTION               NO
                            NULLS ALLOWED OPTION        NO
                            NULL VALUE

                            DEFAULT VALUE
                            RETRIEVAL VALIDATION OPTION NO
                            VALIDATION OPTION
                            VALIDATION MINIMUM
                            VALIDATION MAXIMUM
                            VALIDATION TABLE NAME
                            VALIDATION EXIT

                            SHORT TEXT:  CINCOM DEFINED DEFAULT DOMAIN

*-----------------------------------------------------------------------------------------------------------*

                            DM NAME:  CSI-SP-AT-DEFNO

                            LAST UPDATE:  11:40:59  11/11/92  V:  0     USER:  CSI-DBA

                            FUNCTION                    NUMBER
                            UNIT
                            DATA FORMAT                 B - BINARY
                            LENGTH                      4
                            NUMBER OF DECIMAL PLACES    0
                            SIGNED OPTION               YES
                            NULLS ALLOWED OPTION        NO
                            NULL VALUE

                            DEFAULT VALUE
                            RETRIEVAL VALIDATION OPTION NO
                            VALIDATION OPTION
                            VALIDATION MINIMUM
                            VALIDATION MAXIMUM
                            VALIDATION TABLE NAME
                            VALIDATION EXIT

                            SHORT TEXT:  NONE
*-----------------------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Edit Masks)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                TIS/XA DIRECTORY
                                          MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR EDIT MASKS

                              EDIT              EDIT MASK
                              MASK     LENGTH       CHARACTERS

                                A         17        9999999999999999-

                          LAST UPDATE:  09:34:07  11/10/92  V:  0     USER:  CSI-DBA

                          SHORT TEXT:  NUMERIC FIELD EDIT INCLUDING MINUS SIGN
*-------------------------------------------------------------------------------------------------*

                                B         17        ZZZZZZZZZZZZZZZ9-

                          LAST UPDATE:  09:34:08  11/10/92  V:  0     USER:  CSI-DBA

                          SHORT TEXT:  ZERO SUPPRESSED FIELD TO 1 POSITION W/MINUS
*-------------------------------------------------------------------------------------------------*

                                C         20        ZZZ.ZZZ.ZZZ.ZZZ.ZZ9-

                          LAST UPDATE:  09:34:09  11/10/92  V:  0     USER:  CSI-DBA

                          SHORT TEXT:  ZERO SUPPRESSED TO 1POS. W/MINUS & COMMAS
*-------------------------------------------------------------------------------------------------*

                                D          8        99/99/99

                          LAST UPDATE:  09:34:11  11/10/92  V:  0     USER:  CSI-DBA

                          SHORT TEXT:  STANDARD DATE FORMAT
*-------------------------------------------------------------------------------------------------*

                                E         22        ZZ.ZZZ.ZZZ.ZZZ.ZZZ.99-

                          LAST UPDATE:  09:34:12  11/10/92  V:  0     USER:  CSI-DBA

                          SHORT TEXT:  ZERO SUPPRESSED FIELD W/2 DECIMAL POSITION
*-------------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Environment Descriptions)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                TIS/XA DIRECTORY
                                              MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR ENVIRONMENT DESCRIPTIONS

                          ED NAME:  BURRYENV        SCHEMA:  BURRYSCH

                          LAST UPDATE:  14:56:25  08/10/92  V:  11     USER:  CSI-DBA

                          MAXIMUM CONNECTED INTERFACES     1
                          MAXIMUM CONNECTED THREADS        1
                          MAXIMUM SIGNED ON TASKS          1
                          DBMS STORAGE ALLOCATION          700K
                          MAXIMUM HELD RECORDS             1000
                          LOG OPTIONS                      NNNN
                          TASK LOG OPTION                  NO
                          END LOG OPTION                   F - FORCE END OF VOLUME
                          OPENX OPTION                     P - PROCESS
                          DBMS CONSOLE OPTION              YES
                          STATISTICS INDICATOR             NO
                          FILE ACCESS MODE                 UPDATE
                          DIRECTORY FILE ACCESS METHOD     READ

                          BATCH DELAY TIME                 60
                          TP-MONITOR DELAY TIME            5
                          FUNC. INPUT EXIT NAME
                          OPEN EXIT NAME
                          CLOSE EXIT NAME
                          READ EXIT NAME
                          WRITE EXIT NAME
                          CHECK EXIT NAME
                          SHUTDOWN PASSWORD                **NONE**
                          LOCAL SITE NAME
                          BOUND NAME                       BURRYENN
                          GENERATION INDICATOR             NO
                          LOG WRITE EXIT NAME
                          NEW VOLUME EXIT NAME

                          SHORT TEXT:  NONE
*-------------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Environment Descriptions/Buffer Pool Information)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                             TIS/XA DIRECTORY
                                          MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR ENVIRONMENT DESCRIPTIONS

                          BUFFER POOLS FOR ENVIRONMENT DESCRIPTION: BURRYENN FOR SCHEMA:  BURRYSCH

                              BUFFER-POOL              BFXA
                              BUFFER-SIZE              4096
                              DIR. BUFFER COUNT        2
                              SEQ. BUFFER COUNT        0
                              SEQ. THREAD COUNT        0
                              GENERATION INDICATOR     NO

                              SHORT TEXT:  NONE

                              BUFFER-POOL              BFXF
                              BUFFER-SIZE              4096
                              DIR. BUFFER COUNT        2
                              SEQ. BUFFER COUNT        0
                              SEQ. THREAD COUNT        0
                              GENERATION INDICATOR     NO

                              SHORT TEXT:  NONE

                              BUFFER-POOL              BFXP
                              BUFFER-SIZE              4096
                              DIR. BUFFER COUNT        2
                              SEQ. BUFFER COUNT        0
                              SEQ. THREAD COUNT        0
                              GENERATION INDICATOR     NO

                              SHORT TEXT:  NONE
```

Directory Full Definition Report (External Fields)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                    TIS/XA DIRECTORY
                                                 MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR EXTERNAL FIELDS

                          XF NAME:  BRAN-FOR-MANF   SCHEMA:  BURRYSCH

                               LAST UPDATE:  10:31:28   11/15/92   V:  0  USER:  CSI-DBA

                               DATA FORMAT              C - CHARACTER
                               FUNCTION                 STRING
                               UNIT
                               EXTERNAL LENGTH          4
                               NUMBER OF DECIMAL PLACES 0
                               SIGNED OPTION            NO
                               LENGTH OF EDITED FIELD   00000
                               TRANSLATE/EDIT SWITCH
                               TRANSLATE/EDIT NAME
                               RECORD CODE              ALL
                               PRINT HEADING
                               DATA TRANSLATION EXIT
                               GENERATION INDICATOR     NO

                               SHORT TEXT:  NONE
*-----------------------------------------------------------------------------------------------*
                          XF NAME:  BRAN-ADDR   SCHEMA:  BURRYSCH

                               LAST UPDATE:  10:31:12   11/15/92   V:  0  USER:  CSI-DBA

                               DATA FORMAT              C - CHARACTER
                               FUNCTION                 STRING
                               UNIT
                               EXTERNAL LENGTH          20
                               NUMBER OF DECIMAL PLACES 0
                               SIGNED OPTION            NO
                               LENGTH OF EDITED FIELD   00000
                               TRANSLATE/EDIT SWITCH
                               TRANSLATE/EDIT NAME
                               RECORD CODE              ALL
                               PRINT HEADING
                               DATA TRANSLATION EXIT
                               GENERATION INDICATOR     NO

                               SHORT TEXT:  BRANCH'S STREET ADDRESS
*-----------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Files)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                TIS/XA DIRECTORY
                                          MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR FILES

                          FI NAME:  C$-#   SCHEMA:  BURRYSCH

                              LAST UPDATE:  10:06:56   11/15/92   V:  0  USER:  CSI-DBA

                              DDNAME                C$TDEF#
                              DATA SET NAME                TIS.C$TDEF#
                              FILE TYPE                    P - PRIMARY
                              ACCESS METHOD                BDAM
                              FILE DEVICE TYPE             3380
                              FILE DEVICE ASSIGNMENT       000
                              LOGICAL RECORD LENGTH        374
                              TOTAL LOGICAL RECORDS        18249
                              TOTAL TRACKS                 166
                              RECORDS PER BLOCK/CI         11
                              BLOCKS PER TRACK             10
                              VSAM CONTROL INTERVAL        0
                              TOTAL VSAM CONTROL INTERVAL  0

                              SITE EXIT NAME
                              SITE TABLE NAME
                              CODED RECORD INDICATOR       YES
                              DOS INDICATOR                YES
                              OLD FILE INDICATOR           NO
                              CONSISTENT                   YES
                              GENERATION INDICATOR         NO
                              BLOCKSIZE                    4114
                              RECORDS IN PHYSICAL CYL      1650
                              PRIME NUMBER                 18233
                              CONTROL KEY LENGTH           4
                              CONTROL KEY DISPL            8
                              RECORD CODE DISPL            12
                              NUMBER OF ELEMENTS           449
                              NUMBER OF LINKPATHS          0

                              SHORT TEXT:  LOADED DEFINITION NUMBER FILE (PRIMARY)
*-------------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Foreign Keys)

```
CINCOM SYSTEMS, INC.  DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                  TIS/XA DIRECTORY
                                                 MAINTENANCE AS OF 00/00/00
                      DIRECTORY DEFINITION FOR FOREIGN KEYS

                      FK NAME: ASSIGNED-TO RELATION: BRANCH CONC. SCHEMA: BURRYS-CONCEPTUAL-SCHEMA

                      LAST UPDATE:  10:52:12   11/15/92   V:  0000  USER:  CSI-DBA

                              PRIMARY KEY RELATION        REGION
                              FOREIGN KEY TYPE         A - ASSIGNED
                              CLUSTERED OPTION         N
                              CHAINED                  N

                              SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*
                      FK NAME: BOUGHT-ON RELATION: POLINE CONC. SCHEMA: BURRYS-CONCEPTUAL-SCHEMA

                      LAST UPDATE:  10:52:19   11/15/97   V:  0000  USER:  CSI-DBA

                              PRIMARY KEY RELATION        PO
                              FOREIGN KEY TYPE         E - ESSENTIAL
                              CLUSTERED OPTION         N
                              CHAINED                  N

                              SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*
                      FK NAME: CAN-BUY-FROM RELATION: VS-NO CONC. SCHEMA: BURRYS-CONCEPTUAL-SCHEMA

                      LAST UPDATE:  10:52:18   11/15/92   V:  0000  USER:  CSI-DBA

                              PRIMARY KEY RELATION        SUPPLIER
                              FOREIGN KEY TYPE         E - ESSENTIAL
                              CLUSTERED OPTION         N
                              CHAINED                  N

                              SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Internal Records)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                TIS/XA DIRECTORY
                                          MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR INTERNAL RECORDS

                          IR NAME:  BASE   FOR FILE: CS-#  FOR SCHEMA:  BURRYSCH

                          LAST UPDATE:  10:06:56   11/15/92   V:  2     USER:  CSI-DBA

                              GET VALID                   YES
                              BEFORE GET EXIT
                              AFTER GET EXIT
                              REPLACE VALID               NO
                              BEFORE REPLACE EXIT
                              AFTER REPLACE EXIT
                              INSERT VALID                NO
                              BEFORE INSERT EXIT
                              AFTER INSERT EXIT
                              DELETE VALID                NO
                              BEFORE DELETE EXIT
                              AFTER DELETE EXIT
                              INTERNAL RECORD LENGTH      374
                              CONSISTENT                  YES
                              GENERATION INDICATOR        NO

                              SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*
                          IR NAME:  BASE   FOR FILE: CS-0  FOR SCHEMA:  BURRYSCH
                          LAST UPDATE:  10:10:46   11/15/92   V:  2     USER:  CSI-DBA

                              GET VALID                   YES
                              BEFORE GET EXIT
                              AFTER GET EXIT
                              REPLACE VALID               NO
                              BEFORE REPLACE EXIT
                              AFTER REPLACE EXIT
                              INSERT VALID                NO
                              BEFORE INSERT EXIT
                              AFTER INSERT EXIT
                                      DELETE VALID        NO
                              BEFORE DELETE EXIT
                              AFTER DELETE EXIT
                              INTERNAL RECORD LENGTH      98
                              CONSISTENT                  YES
                              GENERATION INDICATOR        NO

                              SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Key codes)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                    TIS/XA DIRECTORY
                                                MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR KEY CODES

                          KC NAME:  BASE   SK NAME:  E$BRSKO1  FI NAME:  E$BR   SC NAME:  BURRYSCH

                          LAST UPDATE:  15:57:42   11/15/92   V:  0000  USER:  CSI-DBA

                              KEY CODE LENGTH          0004
                              GENERATION IND           NO

                              SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*

                          KC NAME:  BASE   SK NAME:  E$CUSKO1  FI NAME:  E$CU   SC NAME:  BURRYSCH

                          LAST UPDATE:  15:57:50   11/15/92   V:  0000  USER:  CSI-DBA

                              KEY CODE LENGTH          0004
                              GENERATION IND           NO

                              SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*

                          KC NAME:  BASE   SK NAME:  E$CUSKO2  FI NAME:  E$CU   SC NAME:  BURRYSCH

                          LAST UPDATE:  15:57:54   11/15/92   V:  0000  USER:  CSI-DBA

                              KEY CODE LENGTH          00020
                              GENERATION IND           NO

                              SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*

                          KC NAME:  BASE   SK NAME:  E$SKSKO1  FI NAME:  E$SK   SC NAME:  BURRYSCH

                          LAST UPDATE:  15:58:02   11/15/92   V:  0000  USER:  CSI-DBA

                              KEY CODE LENGTH          0004
                              GENERATION IND           NO

                              SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Logical Views)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                TIS/XA DIRECTORY
                                              MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR LOGICAL VIEWS

                          LV NAME:  ADD-INVOICE     SCHEMA:  BURRYSCH

                          LAST UPDATE:  10:37:31   11/15/92   V:  0     USER:  CSI-DBA

                              CONSISTENT                      YES
                              GENERATION INDICATOR            NO
                              PUBLIC VIEW INDICATOR           NO
                              ONLINE QUERY INDICATOR          YES
                              BATCH QUERY INDICATOR           YES
                              COMP. RETRIEVAL INDICATOR       YES
                              GENERALIZED UPDATES INDICATOR   YES
                              LOGICAL USER VIEW IND.          YES
                              SITE TABLE NAME

                              SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*
                          LV NAME:  BILL        SCHEMA:  BURRYSCH

                          LAST UPDATE:  10:37:41   11/15/92   V:  0     USER:  CSI-DBA

                              CONSISTENT                      YES
                              GENERATION INDICATOR            NO
                              PUBLIC VIEW INDICATOR           NO
                              ON-LINE QUERY INDICATOR         YES
                              BATCH QUERY INDICATOR           YES
                              COMP. RETRIEVAL INDICATOR       YES
                              GENERALIZED UPDATES INDICATOR   YES
                              LOGICAL USER VIEW IND.          YES
                              SITE TABLE NAME

                              SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*
                          LV NAME:  BRAN        SCHEMA:  BURRYSCH

                          LAST UPDATE:  10:35:27   11/15/92   V:  0     USER:  CSI-DBA

                              CONSISTENT                      YES
                              GENERATION INDICATOR            NO
                              PUBLIC VIEW INDICATOR           NO
                              ON-LINE QUERY INDICATOR         YES
                              BATCH QUERY INDICATOR           YES
                              COMP. RETRIEVAL INDICATOR       YES
                              GENERALIZED UPDATES INDICATOR   YES
                              LOGICAL USER VIEW IND.          YES
                              SITE TABLE NAME

                              SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Physical Fields)

```
CINCOM SYSTEMS, INC.  DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                  TIS/XA DIRECTORY
                                                 MAINTENANCE AS OF 00/00/00
                        DIRECTORY DEFINITION FOR PHYSICAL FIELDS

                        PF NAME: *C$-#001 FOR INTERNAL REC. BASE. FOR FILE: C$-# FOR SCHEMA: BURRYSCH

                        LAST UPDATE:  10:07:01   11/15/92   V:  1     USER:  CSI-DBA

                                 FUNCTION                    STRING
                                 UNIT OF FUNCTION
                                 DATA FORMAT                 C - CHARACTER
                                 FIELD LENGTH                28
                                 NUMBER OF DECIMAL PLACES    0
                                 SIGNED FIELD                NO
                                 KEY REFERBACK DEFN #        0
                                 LINKPATH TYPE               P - PRIMARY
                                 SEQUENCE FIELD
                                 SEQUENCE FIELD TYPE
                                 NULLS ALLOWED OPTION        NO
                                 NULL VALUE

                                 DEFAULT VALUE
                                 RETRIEVAL VALIDATION OPTION NO
                                 VALIDATION OPTION
                                 VALIDATION MINIMUM
                                 VALIDATION MAXIMUM
                                 VALIDATION TABLE NAME
                                 VALIDATION EXIT
                                 DISPLACEMENT IN RECORD      16
                                 FIELD TYPE                  FILLER FIELD
                                 GENERATION INDICATOR        NO

                                 SHORT TEXT:  ***FILLER***

*-----------------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Procedures)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                TIS/XA DIRECTORY
                                          MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR PROCEDURES

                          PROCEDURE:  CSI-DEFAULT-PR

                          LAST UPDATE: 17:15:18  11/15/92  V: 0  USER:  **CSI-DIRECTORY-CREATE-USER**

                              USER NAME                CSI-DEFAULT-US
                              LAST COMPILE TIME        00:00:00
                              LAST COMPILE DATE        00/00/0000
                              LANGUAGE                 C - COBOL
                              PROCEDURE TYPE           L - LUV

                              SHORT TEXT:  CINCOM DIRECTORY DEFAULTS - PROCEDURES

*-----------------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Relations)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                    TIS/XA DIRECTORY
                                                 MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR RELATIONS

                          RE NAME:  BRANCH :CONCEPTUAL SCHEMA:  BURRYS-CONCEPTUAL-SCHEMA

                          LAST UPDATE:  10:51:58   11/15/92   V:  1     USER:  CSI-DBA

                              RELATION TYPE            I - INDEPENDENT ENTITY
                              NUMBER OF TUPLES         418
                              SUBTYPE                  NO
                              EXTENSION                NO
                              ORDERED PRIMARY KEY      Y
                              DONAME                   E$BR
                              DATA SET NAME            SUPRA.E$BR
                              FILE NAME                E$BR
                              CONSISTENT               YES

                              SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*
                          RE NAME:  CSI-DEFAULT-RE  :CONCEPTUAL SCHEMA:  CSI-DEFAULT-CS

                          LAST UPDATE: 17:15:27  11/15/92  V: 1 USER: **CSI-DIRECTORY-CREATE-USER**

                              RELATION TYPE            I - INDEPENDENT ENTITY
                              NUMBER OF TUPLES         10000
                              SUBTYPE                  NO
                              EXTENSION                NO
                              ORDERED PRIMARY KEY      Y
                              DONAME
                              DATA SET NAME
                              FILE NAME
                              CONSISTENT               NO

                              SHORT TEXT:  CINCOM DEFINED DEFAULT RELATION

*-----------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Reserved Words)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                TIS/XA DIRECTORY
                                          MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR RESERVED WORDS

                          AFTER

                          LAST UPDATE: 16:44:07 11/15/92 V: 0 USER: **CSI-DIRECTORY-CREATE-USER**

                          SHORT TEXT:  CHANGE LOCATION OR ADD A NEW FIELD TO REPORT
*-------------------------------------------------------------------------------------------------*
                          AND

                          LAST UPDATE: 16:44:22 11/15/92 V: 0 USER: **CSI-DIRECTORY-CREATE-USER**

                          SHORT TEXT:  SELECTS DATA WHEN TWO VALUE CONDITION IS MET
*-------------------------------------------------------------------------------------------------*
                          ARE

                          LAST UPDATE: 16:44:36 11/15/92 V: 0 USER: **CSI-DIRECTORY-CREATE-USER**

                          SHORT TEXT:  USED TO TELL QUERY WHAT CONDITION MUST BE MET
*-------------------------------------------------------------------------------------------------*
                          ARENT

                          LAST UPDATE: 16:44:52 11/15/92 V: 0 USER: **CSI-DIRECTORY-CREATE-USER**

                          SHORT TEXT:  USED TO TELL QUERY WHAT CONDITION MUST BE MET
*-------------------------------------------------------------------------------------------------*
                          AS

                          LAST UPDATE: 16:45:08 11/15/92 V: 0 USER: **CSI-DIRECTORY-CREATE-USER**
                          SHORT TEXT:  OVERRIDE DEFAULT COLUMN HEADINGS FOR REQUEST
*-------------------------------------------------------------------------------------------------*
                          ASCENDING

                          LAST UPDATE: 16:45:23 11/15/92 V: 0 USER: **CSI-DIRECTORY-CREATE-USER**

                          SHORT TEXT:  TELLS QUERY WHICH ORDER VALUES ARE TO APPEAR
*-------------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Schemas)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                TIS/XA DIRECTORY
                                              MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR SCHEMAS

                          SC NAME:  BURRYSCH

                          LAST UPDATE:  14:57:07   01/27/93   V:  0016    USER:  CSI-DBA

                              BOUND NAME               BURRYSCH
                              RELEASE LEVEL            002.000
                              CONSISTENT               YES
                              GENERATION INDICATOR     NO

                              SHORT TEXT:  SERVICE LEVEL -- BURRYSCH  SL2101

*-------------------------------------------------------------------------------------------*
                          SC NAME:  CSIDEFSC

                          LAST UPDATE: 17:14:27 11/15/92 V: 0000 USER:**CSI-DIRECTORY-CREATE-USER**

                              BOUND NAME               CSIDEFSC
                              RELEASE LEVEL            002.000
                              CONSISTENT               NO
                              GENERATION INDICATOR     NO

                              SHORT TEXT:  CINCOM DIRECTORY DEFAULTS - SCHEMA

*-------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Secondary Keys)

```
CINCOM SYSTEMS, INC.    DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                  TIS/XA DIRECTORY
                                                 MAINTENANCE AS OF 00/00/00
                        DIRECTORY DEFINITION FOR SECONDARY KEYS

                        SK NAME:  CDFISK00 FOR FI:  CDFI          FOR SC:  CSIDEFSC

                        LAST UPDATE: 17:15:42 01/27/93 V: 0000 USER: **CSI-DIRECTORY-CREATE-USER**

                              % INDEX FI BLOCKS USED   00000
                              DAMAGED SK ACTION        A - ABEND
                              SECONDARY KEY LENGTH     00000
                              POPULATED                NO
                              CONSISTENT               NO
                              GENERATION INDICATOR     NO

                              SHORT TEXT:  CINCOM DEFINED DEFAULT SECONDARY KEY
*-----------------------------------------------------------------------------------------------*
                         SK NAME:  E$BRSK01   FOR FI:  E$BR         FOR SC:  BURRYSCH

                         LAST UPDATE:  15:57:42   01/29/93   V:  0000    USER:  CSI-DBA

                              % INDEX FI BLOCKS USED   00000
                              DAMAGED SK ACTION        A - ABEND
                              SECONDARY KEY LENGTH     00004
                              POPULATED                YES
                              CONSISTENT               YES
                              GENERATION INDICATOR     NO

                              SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*
                         SK NAME:  E$CUSK01   FOR FI:  E$CU         FOR SC:  CSIDEFSC

                         LAST UPDATE:  15:57:50   01/29/93   V:  0000    USER:  CSI-DBA

                              % INDEX FI BLOCKS USED   00000
                              DAMAGED SK ACTION        A - ABEND
                              SECONDARY KEY LENGTH     00004
                              POPULATED                YES
                              CONSISTENT               YES
                              GENERATION INDICATOR     NO

                              SHORT TEXT:  *NONE*
*-----------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Tables)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                TIS/XA DIRECTORY
                                              MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR TABLES

                          TABLE:  CSIA

                          LAST UPDATE:  09:32:42   01/27/93   V:  1       USER:  CSI-DBA

                              SEQ# CODE   TRANSLATED VALUE

                              0100 00.SYSTEM INTERNAL DATA
                              0200 01.RESERVED WORD
                              0300 02.COMPONENT DESCRIPTION
                              0400 03.EDIT MASK
                              0500 04.TABLE
                              0600 05.EXPRESSION EQUATION
                              0700 06.SCHEMA
                              0800 07.ENVIRONMENT DESCRIPTION
                              0900 08.CONCEPTUAL SCHEMA
                              1000 09.FILE
                              1100 10.DOMAIN
                              1200 11.LOGICAL VIEW
                              1300 12.PHYSICAL FIELD
                              1400 13.EXTERNAL FIELD
                              1500 14.ACCESS SET
                              1600 15.PROCEDURE
                              1900 18.RELATION
                              2000 19.ATTRIBUTE
                              2100 20.PRIMARY KEY
                              2200 21.FOREIGN KEY
                              2300 22.SECONDARY KEY
                              2400 23.NAVIGATION PRIMITIVE
                              2500 24.KEY CODE
                              2600 25.SECURITY GROUP

                              SHORT TEXT:  CATEGORY CODE SELECTIONS
*-------------------------------------------------------------------------------------------------*
```

Directory Full Definition Report (Users)

```
CINCOM SYSTEMS, INC.      DIRDEFNP - DIRECTORY FULL DEFINITION      JUNE 27, 1997 16:29:36 PAGE: 1

                                TIS/XA DIRECTORY
                                          MAINTENANCE AS OF 00/00/00
                          DIRECTORY DEFINITION FOR USERS

                          USER-ID:  **PUBLIC**

                          LAST UPDATE: 17:14:28 01/27/93 V: 0 USER: **CSI-DIRECTORY-CREATE-USER**

                              USER NAME:            PUBLIC USER
                              ADDRESS-1:
                              ADDRESS-2:
                              ADDRESS-3:
                              PHONE NUMBER:
                              PASSWORD:
                              DBA OPTION:           N
                              LANGUAGE:             ENGLISH

                              SHORT TEXT:  DEFINED PUBLIC USER
*-------------------------------------------------------------------------------------------*
                          USER-ID:  CINCOM

                          LAST UPDATE: 17:14:30 01/27/93 V: 0 USER: **CSI-DIRECTORY-CREATE-USER**

                              USER NAME:            CINCOM SYSTEMS, INC
                              ADDRESS-1:            2300 MONTANA
                              ADDRESS-2:            CINCINNATI, OHIO 45211
                              ADDRESS-3:
                              PHONE NUMBER:         513 662-2300
                              PASSWORD:
                              DBA OPTION:           N
                              LANGUAGE:             ENGLISH

                              SHORT TEXT:  DEFINED CINCOM USER
*-------------------------------------------------------------------------------------------*
                          USER-ID:  CSI-DBA

                          LAST UPDATE: 17:14:31 01/27/93 V: 0 USER: **CSI-DIRECTORY-CREATE-USER**

                              USER NAME:            CINCOM SYSTEMS, INC
                              ADDRESS-1:            2300 MONTANA
                              ADDRESS-2:            CINCINNATI, OHIO 45211
                              ADDRESS-3:
                              PHONE NUMBER:         513 662-2300
                                      PASSWORD:

                              DBA OPTION:           Y
                              LANGUAGE:             ENGLISH

                              SHORT TEXT:  DEFAULT DBA USER
*-------------------------------------------------------------------------------------------*
```

Using the Directory Full Text Description Report

The Directory Full Text Description Report (DIRTEXTP) lists each entity name in alphabetical order along with the entity qualifiers. For example, a physical field is shown along with its associated internal record, file, and schema. Subsequent lines of information for each entity show the associated short text and/or long text.

You can submit trigger records for category selection, schema name selection, text selection, and maintenance date selection. If you do not submit a trigger record for text selection, both short text and long text are printed. The following figure shows the Directory Full Text Description Report.

Directory Full Text Description Report

```
CINCOM SYSTEMS, INC.   DIRTEXTP - DIRECTORY FULL DESCRIPTION     JUNE 27, 1997 16:29:36 PAGE: 3

                    TIS/XA DIRECTORY
           TEXT OPT:  LONG               MAINTENANCE AS OF 00/00/00
                    TEXT DESCRIPTION FOR BURRYSCH

FILE                LT: END LONG TEXT

               C$NR         FOR SC:  BURRYSCH

                    LT:  100 THE C$NR FILE IS A PRIMARY FILE WHICH CONTAINS ALL RELATIONS.
                    LT: END LONG TEXT

               C$NV         FOR SC:  BURRYSCH

                    LT:  100 THE C$NV FILE IS A PRIMARY FILE WHICH CONTAINS ALL DATA VIEWS.
                    LT: END LONG TEXT

               C$N1         FOR SC:  BURRYSCH

                    LT:  100 THE C$N1 FILE IS A RELATED FILE ON WHICH IS RECORDED ALL CONCEPTUAL
                    LT:  200 SCHEMA TO RELATION RELATIONSHIPS. IT IS PRIMARILY USED TO LIST ALL
                    LT:  300 RELATIONS WITHIN A PARTICULAR CONCEPTUAL SCHEMA.
                    LT: END LONG FILE

               C$N2         FOR SC:  BURRYSCH

                    LT:  100 THE C$N2 FILE IS A RELATED FILE ON WHICH ARE RECORDED ALL RELATION
                    LT:  200 TO ATTRIBUTE RELATIONSHIPS. IT IS USED PRIMARILY FOR LISTING ALL
                    LT:  300 ATTRIBUTES WITHIN A RELATION.
                    LT: END LONG TEXT

               C$N3         FOR SC:  BURRYSCH

                    LT:  100 THE C$N3 FILE IS A RELATED FILE ON WHICH ARE RECORDED ALL ATTRIBUTE
                    LT:  200 TO DOMAIN RELATIONSHIPS. IT IS USED PRIMARILY TO RESTRICT DELETION OF
                    LT:  300 DOMAINS WHICH ARE STILL ASSOCIATED WITH ATTRIBUTES.
                    LT: END LONG FILE

               C$N4         FOR SC:  BURRYSCH

                    LT:  100 THE C$N4 FILE IS A RELATED FILE ON WHICH ARE RECORDED ALL SOURCE
                    LT:  200 RELATION TO FOREIGN KEY TO TARGET RELATION RELATIONSHIPS. IT IS USED
                    LT:  300 PRIMARILY TO ENHANCE PERFORMANCE DURING DELETION OF ALL FOREIGN KEYS
                    LT:  400 ASSOCIATED WITH EITHER A SOURCE OR TARGET RELATION.
                    LT: END LONG TEXT
```

```
CINCOM SYSTEMS, INC.    DIRTEXTP - DIRECTORY FULL DESCRIPTION     JUNE 27, 1997 16:29:36 PAGE: 3

                    TIS/XA DIRECTORY
            TEXT OPT:  LONG               MAINTENANCE AS OF 00/00/00
                    TEXT DESCRIPTION FOR BURRYSCH

FILE            C$N5         FOR SC:  BURRYSCH

                    LT:  100 THE C$N5 FILE IS A RELATED FILE WHICH CONTAINS ALL SOURCE ATTRIBUTE
                    LT:  200 TO FOREIGN KEY TO TARGET ATTRIBUTE RELATIONSHIPS.
                    LT:  END LONG TEXT

                C$N6         FOR SC:  BURRYSCH

                    LT:  100 THE C$N6 FILE IS A RELATED FILE ON WHICH ARE RECORDED ALL EXTERNAL
                    LT:  200 SCHEMA TO DATA VIEW RELATIONSHIPS. IT IS PRIMARILY USED TO LIST ALL
                    LT:  300 DATA VIEWS IN AN EXTERNAL SCHEMA.
                    LT:  END LONG TEXT

                C$N7         FOR SC:  BURRYSCH

                    LT:  100 THE C$N7 FILE IS A RELATED FILE ON WHICH ARE RECORDED ALL DATA VIEW
                    LT:  200 TO DATA ITEM RELATIONSHIPS. IT IS PRIMARILY USED TO LIST ALL DATA
                    LT:  300 ITEMS IN A DATA VIEW.
                    LT:  END LONG FILE

                COPY         FOR SC:  BURRYSCH

                    LT:  NONE

                E$BR         FOR SC:  BURRYSCH

                    LT:  NONE

                E$CU         FOR SC:  BURRYSCH

                    LT:  NONE

                E$IL         FOR SC:  BURRYSCH

                    LT:  NONE
```

Using the Directory Name Cross Reference Report

The Directory Name Cross Reference Report (DIRXREFP) lists all entity names in alphabetical order. For each name, the report identifies six items that collectively provide a history of the entities' creation:

- Category of the entity name.
- Levels of qualification for the name. (The information provided for a physical field name is the internal record, file, and schema to which it belongs.)
- Maintenance version number of the last update.
- Date the name was created or last updated.
- Time of day the name was created or last updated.
- User ID for the session in which the name was created or last updated.

This report provides three lines of data for each entity. The first line shows the entity name, category, and qualifiers. The second line shows the maintenance information. The third line shows the user ID.

Use this report to determine whether an entity is on the Directory and, if so, to check the qualification and the current version number of the entity. The following example shows the Directory Name Cross Reference Report:

NOTE

You can submit five types of trigger records: Category Selection, Schema Name Selection, User Name Selection, Maintenance Date Selection, and Short Text Selection. You can use any of the valid category codes for category selection (see "Using category selection" on page 354).

Directory Name Cross Reference Report

```
CINCOM SYSTEMS, INC.      DIRXREFP - DIRECTORY CROSS REFERENCE     JUNE 27, 1997 16:29:36 PAGE: 3
                                        TIS/XA DIRECTORY
                                     MAINTENANCE AS OF 00/00/00

NAME                CATEGORY   QUALIFICATION AND UPDATE INFORMATION

C$-#                FILE   FOR SC:  BURRYSCH
                    MOD LVL:  4    LAST UPDATE:  11/15/92  10:06
                    USER:  CSI-DBA

  SHORT TEXT:  LOADED DEFINITION NUMBER FILE (PRIMARY)

C$-D                FILE   FOR SC:  BURRYSCH
                    MOD LVL:  4    LAST UPDATE:  11/15/92  10:10
                    USER:  CSI-DBA

  SHORT TEXT:  DATA FILE.

C$-N                FILE   FOR SC:  BURRYSCH
                    MOD LVL:  4    LAST UPDATE:  11/15/92  10:06
                    USER:  CSI-DBA

  SHORT TEXT:  LOADED NAME FILE (PRIMARY)

C$-S                FILE   FOR SC:  BURRYSCH
                    MOD LVL:  4    LAST UPDATE:  11/15/92  10:11
                    USER:  CSI-DBA

  SHORT TEXT:  STRUCTURE FILE

C$-T                FILE   FOR SC:  BURRYSCH
                    MOD LVL:  4    LAST UPDATE:  11/15/92  10:11
                    USER:  CSI-DBA

  SHORT TEXT:  VARIABLE FILE FOR TEXT

C$NA                FILE   FOR SC:  BURRYSCH
                    MOD LVL:  4    LAST UPDATE:  11/15/92  10:12
                    USER:  CSI-DBA

SHORT TEXT:  THE C$NA FILE IS A PRIMARY FILE WHICH CONTAINS ALL ATTRIBUTES.

C$NC                FILE   FOR SC:  BURRYSCH
                    MOD LVL:  4    LAST UPDATE:  11/15/92  10:11
                    USER:  CSI-DBA

  SHORT TEXT:  THE C$NC FILE IS A PRIMARY FILE WHICH CONTAINS ALL CONCEPTUAL SCHEMAS.

C$ND                FILE   FOR SC:  BURRYSCH
                    MOD LVL:  4    LAST UPDATE:  11/15/92  10:12
                    USER:  CSI-DBA

  SHORT TEXT:  THE C$ND FILE IS A PRIMARY FILE WHICH CONTAINS ALL DOAMINS.
```

Using the Directory Name Report

The Directory Name Report (DIRNAMEP) lists all entity names and provides the qualifier's definition number, the definition number of each entity, and the category of each entity.

For example, a physical field is shown with its definition number and the definition number of the qualifying internal record; an internal record is shown with its definition number and the definition number of the qualifying file.

Global names within the Directory are not qualified and cannot be duplicated. They include schema names, Directory component descriptions, user names, edit masks, table names, and all reserved words.

You can sort information on this report by entity name, definition number, or qualifier. You can use a trigger record to specify the sort sequence; other trigger records cannot be used for this report.

This report provides a record of all entity definitions along with the category and qualifier for each. The following example shows the Directory Name Report.

Directory Name Report

```
CINCOM SYSTEMS, INC.    DIRNAMEP - DIRECTORY NAME REPORT     JUNE 27, 1997 16:29:36 PAGE: 3

  SORT OPT:  DEFN       TIS/XA DIRECTORY MAINTENANCE AS OF 00/00/00

NAME              QUALIFIER         DEFN        CATEGORY

DEFINITION NO. OF CONTROL RECORD      0 SYSTEM INTERNAL DATA
AFTER                   100 RESERVED WORD
AND                     101 RESERVED WORD
ARE                     102 RESERVED WORD
ARENT                   103 RESERVED WORD
AS                      104 RESERVED WORD
ASCENDING               105 RESERVED WORD
AT                      106 RESERVED WORD
AVERAGE                 107 RESERVED WORD
AVG                     108 RESERVED WORD
BEFORE                  109 RESERVED WORD
BETWEEN                 110 RESERVED WORD
BIGGER                  111 RESERVED WORD
BY                      112 RESERVED WORD
BYE                     113 RESERVED WORD
CANCEL                  114 RESERVED WORD
CHANGES                 115 RESERVED WORD
COLUMN                  116 RESERVED WORD
COUNT                   117 RESERVED WORD
DEFINE                  118 RESERVED WORD
DEL                     119 RESERVED WORD
DELETE                  120 RESERVED WORD
DESCENDING              121 RESERVED WORD
DIFFERENT               122 RESERVED WORD
DISPLAY                 123 RESERVED WORD
DIVIDE                  124 RESERVED WORD
DIVIDED                 125 RESERVED WORD
DO                      126 RESERVED WORD
DOES                    127 RESERVED WORD
DOESNT                  128 RESERVED WORD
DONT                    129 RESERVED WORD
END                     130 RESERVED WORD
EQ                      131 RESERVED WORD
EQUAL                   132 RESERVED WORD
EQUALS                  133 RESERVED WORD
EXPRESSION              134 RESERVED WORD
EXPRESSIONS             135 RESERVED WORD
EXPRS                   136 RESERVED WORD
FINISHED                137 RESERVED WORD
FROM                    138 RESERVED WORD
GE                      139 RESERVED WORD
GEO                     140 RESERVED WORD
GO                      141 RESERVED WORD
GREATER                 142 RESERVED WORD
GT                      143 RESERVED WORD
HELP                    144 RESERVED WORD
HIGHER                  145 RESERVED WORD
IDENTITY                146 RESERVED WORD
IF                      147 RESERVED WORD
IS                      148 RESERVED WORD
ISNT                    149 RESERVED WORD
                  USER:  CSI-DBA
```

Using the Directory Physical Structure Report

The Directory Physical Structure Report (DIRSTRUP) provides information on all entities contained within the Directory with the following considerations:

- Entities are listed in alphabetical order.
- Multiple lines of data may be shown for an entity.
- The first line of data for each entity shows the entity name, definition number, qualifier definition number, and linkpath information.
- Subsequent lines of data for an entity may provide the category type, internal record code, and parent-child relationship information.

You cannot use trigger records for this report. You can use this report as a reference source for the complete Directory structure. The following example shows the Directory Physical Structure Report.

Directory Physical Structure Report

```
CINCOM SYSTEMS, INC.  DIRSTRUP - DIR. PHYSICAL STRUCTURE   JUNE 27, 1997 16:29:36 PAGE: 2

                            TIS/XA DIRECTORY MAINTENANCE AS OF 00/00/00
NAME             NAME  QUAL    LKID    LINK    LKST  LINK  LKWU  LINK  LKDA  LINK  LKIT
             LINK
             DEF#      DEF#    BEG.    END BEG.   ENDBEG.  ENDBEG.  ENDBEG.  END

*COPY007      2704      2688                    16656 14456        1569  1569

*COPY006      2703      2688                    12919 12919        12664 12664

*RPTR001      3347      3344                    14579 14579         1320  1320

*RPTR002      3350      3344    21196   21196   21160 21196  4431  4431        11373 11373

 PHYSICAL FIELD   HD PARENT-CODE(DEF#): 12(3350)
  PHYSICAL FIELD  DT CHILD-CODE(DEF#):  12(3351)
  PHYSICAL FIELD                12(3352)

*RPTR003      3351      3344                    21159 21159        11074 11074

A              276         1                                 10743 10743

ADD-DATE      1887      1883                    24360  5606

ADD-DATE      1940      1936                    24360 25507

ADD-INVOICE   4140      2171    7412    7412    7383  7412   4536 26345

   LOGICAL VIEW   HD PARENT-CODE(DEF#): 11(4140)

     ACCESS SET   DT CHILD-CODE(DEF#):  14(4110)

BRANCH            4197  4170    8427    8421    8440  8421  26352 11409

  RELATION    HD PARENT-CODE(DEF#): 18(4197)

  PRIMARY KEY DT CHILD-CODE(DEF#):  20(4196)

  RELATION    HD PARENT-CODE(DEF#): 18(4197)

  INTERNAL RECORD DT CHILD-CODE(DEF#):  16(3823)

RELATION      HD PARENT-CODE(DEF#): 18(4197)

  FOREIGN KEY DT CHILD-CODE(DEF#):  21(4227)

  RELATION    HD PARENT-CODE(DEF#): 18(4197)
  ATTRIBUTE   DT CHILD-CODE(DEF#):  19(4251)
  ATTRIBUTE                19(4252)
  ATTRIBUTE                19(4253)
  ATTRIBUTE                19(4254)
  ATTRIBUTE                19(4255)
  ATTRIBUTE                19(4256)
  ATTRIBUTE                19(4257)
  ATTRIBUTE                19(4258)
  ATTRIBUTE                19(4259)
  ATTRIBUTE                19(4260)
```

```
CINCOM SYSTEMS, INC.  DIRSTRUP - DIR. PHYSICAL STRUCTURE    JUNE 27, 1997 16:29:36 PAGE: 3

                           TIS/XA DIRECTORY MAINTENANCE AS OF 00/00/00
NAME              NAME  QUAL    LKID    LINK    LKST  LINK  LKWU  LINK  LKDA  LINK  LKIT
             LINK
             DEF#      DEF#    BEG.    END BEG.   ENDBEG.  ENDBEG.  ENDBEG.  END

BRANCH-ADDR  4016      2171                    27403 26130        12686 12686

BRANCH-ADDR  4253      4197    27402   27402   27403 27402 18402  8430

  ATTRIBUTE  HD PARENT-CODE(DEF#): 13(4253)

  EXTERNAL FIELD  DT CHILD-CODE(DEF#):  13(4016)

  ATTRIBUTE  HD PARENT-CODE(DEF#): 13(4256)

  EXTERNAL FIELD  DT CHILD-CODE(DEF#):  13(4019)

BURRYENV     4009      2171     468     443     513   443  26122 26122  209   177   287
              236

  ENV DESCRIPTION HD PARENT-CODE(DEF#): 07(4006)

    FILE     DT CHILD-CODE(DEF#):  09(3372)
    FILE                  09(3381)
    FILE                  09(3366)
    FILE                  09(3414)

BURRYENV     4009      2171     468     443     513   443  26122 26122  209   177   287
              236

  ENV DESCRIPTION HD PARENT-CODE(DEF#): 07(4006)

  SCHEMA     DT CHILD-CODE(DEF#):  06(2171)

CONCEPTUAL SCHEMA HD PARENT-CODE(DEF#): 06(4170)

  RELATION   DT CHILD-CODE(DEF#):  18(4195)
  RELATION                18(4197)

BURRYSCH     2171         1    26369   25042   26409 25042 11406 11406       13426 13437

  SCHEMA     HD PARENT-CODE(DEF#): 06(2171)
  EXTERNAL FIELD  DT CHILD-CODE(DEF#):  13(2172)
  EXTERNAL FIELD               13(2173)
  EXTERNAL FIELD               13(2174)
```

Using the Directory Schema Structure Report

The Directory Schema Structure Report (DIRSCSTP) produces four subreports describing the full explosion of the structure of each schema for:

- **Environment Descriptions.** Includes attribute data, associated buffer pool information, and Environment Description-to-file relationship data ("Directory Schema Structure Report (Environment Description)" on page 410).

- **Files.** Lists all files and their attributes, internal records, key code, secondary key, and physical fields and some attribute data including length, type, and number of decimal places. This subreport also lists related external fields ("Directory Schema Structure Report (File)" on page 411).

- **External Fields.** Lists all external fields within the schema in alphabetical order ("Directory Schema Structure Report (External Field)" on page 412).

- **Logical Views.** Lists all logical views within the schema, in alphabetical order, along with their associated external fields, Environment Description, users and their attribute data, access sets, procedures, and all associated relationship ("Directory Schema Structure Report (Logical View)" on page 413).

You must submit Schema Name Selection trigger records to select or suppress specified schemas. You can also submit Category Selection trigger records using the category code ED, FI, XF, or LV to select or suppress a particular subreport. You cannot use any other category codes for this report.

Directory Schema Structure Report (Environment Description)

```
CINCOM SYSTEMS, INC.    DIRSCSTP - DIR. PHYSICAL STRUCTURE   JUNE 27, 1997 16:29:36 PAGE: 2

                            TIS/XA DIRECTORY
                            SCHEMA STRUCTURE FOR BURRYSCH

STRUCTURE SUBREPORT FOR ENVIRONMENT DESCRIPTIONS

ENV. DESC.:  BURRYENN

                        MAX. CONNECTED INTERF   1
                        MAX. CONNECTED THREADS  1
                        MAX. SIGNED-ON TASKS    1
                        DBMS STORAGE ALLOCATION 700K
                        MAXIMUM HELD RECORDS    1000
                        LOG OPTIONS NNNN
                        TASK LOG OPTIONS        NO
                        END LOG OPTIONS         F - FORCE END OF VOLUME
                        OPENX OPTION            P - PROCESS
                        DBMS CONSOLE OPTION     YES
                        STATISTICS INDICATOR    NO
                        ACCESS MODE UPDATE
                        DIRECTORY ACCESS METHOD READ

                        FUNC. INPUT EXIT NAME
                        OPEN EXIT NAME
                        CLOSE EXIT NAME
                        READ EXIT NAME
                        WRITE EXIT NAME
                        CHECK EXIT NAME
                        SHUTDOWN PASSWORD       NONE
                        LOCAL SITE NAME
                        BOUND NAME  BURRYENN
                        BATCH DELAY TIME        60
                        TPMON DELAY TIME        5
                        GENERATION INDICATOR    NO

                        BUFFER POOL:  BFXA
                        BUFFER POOL SIZE:       4096
                        DIRECT BUFFER COUNT:    2
                        SERIAL BUFFER COUNT:    0
                        SERIAL THREAD COUNT:    0
                        GENERATION INDICATOR:   NO

                        BUFFER POOL:  BFXF

                        BUFFER POOL SIZE:       4096
                        DIRECT BUFFER COUNT:    2
                        SERIAL BUFFER COUNT:    0
                        SERIAL THREAD COUNT:    0
                        GENERATION INDICATOR:   NO
```

Directory Schema Structure Report (File)

```
CINCOM SYSTEMS, INC. DIRSCSTP - DIRECTORY SCHEMA STRUCTURE  JUNE 27, 1997 16:29:36 PAGE: 1

                            TIS/XA DIRECTORY

                            SCHEMA STRUCTURE FOR BURRYSCH

STRUCTURE SUBREPORT FOR FILES

FILE:  C$-#

                  DDNAME                    C$TDEF#
                  DATA SET NAME             TIS.C$TDEF#
                  FILE TYPE                 P - PRIMARY
                  FILE ACCESS METHOD        BDAM
                  FILE DEVICE TYPE          3380
                  FILE DEVICE ASSIGNMENT    000
                  LOGICAL RECORD LENGTH     374
                  TOTAL LOGICAL RECORDS     18249
                  TOTAL TRACKS              166
                  RECORDS PER BLOCK/CI      11
                  BLOCKS PER TRACK          10
                  VSAM CONTROL INTERVAL     0
                  TOTAL VSAM CNTRL INTVL    0

                  SITE EXIT NAME
                  SITE TABLE NAME
                  CODED RECORD INDICATOR    YES
                  DOS INDICATOR             YES
                  OLD FILE INDICATOR        NO
                  CONSISTENT                YES
                  GENERATION INDICATOR      NO
                  BLOCKSIZE                 4114
                  RECORDS IN PHYSICAL CYL.  1650
                  PRIME NUMBER              18233
                  CONTROL KEY LENGTH        4
                  CONTROL KEY DISPL.        8
                  RECORD CODE DISPL.        12
                  NUMBER OF ELEMENTS        449
                  NUMBER OF LINKPATHS       0

                  IR:   BASE    LENGTH:  374    GEN. IND.:  NO

                        LVL  PHY. FLD   LENGTH  TYPE      DECIMALS  GEN. IND. EXTERNAL FIELDS

                        00  C$-#ROOT     8  C - CHARACTER 0  NO  CSI-DIR-DN-ROOT

                        00  C$-#CTRL     4  B - BINARY    0  NO  CSI-DIR-DN-DEFN

                        00  C$-#CODE     2  C - CHARACTER 0  NO  CSI-DIR-DN-CATG-CODE
                                                                  CSI-DIR-DN-CATG-TRANS

                        00  C$-#NAMQ    34  C - CHARACTER 0  NO  CSI-DIR-DN-NAME-QUAL

                        00  C$-#NAME    30  C - CHARACTER 0  NO  CSI-DIR-DN-NAME

                        00  C$-#BNAM     2  B - BINARY    0  NO  CSI-DIR-DN-BINARY-NAME
```

Directory Schema Structure Report (External Field)

```
CINCOM SYSTEMS, INC. DIRSCSTP - DIRECTORY SCHEMA STRUCTURE  JUNE 27, 1997 16:29:36 PAGE: 1

                            TIS/XA DIRECTORY

                            SCHEMA STRUCTURE FOR BURRYSCH

STRUCTURE SUBREPORT FOR EXTERNAL FIELDS

EXTERNAL FIELD    GEN. IND.     RELATED ATTRIBUTE     RELATION
BRAN-FOR-MANF          NO       **NONE**
BRANCH-ADDR            NO       BRANCH-ADDR           BRANCH
BRANCH-CITY            NO       BRANCH-CITY           BRANCH
BRANCH-DEL-ROUTE       NO       BRANCH-DEL-ROUTE      BRANCH
BRANCH-NAME            NO       BRANCH-NAME           BRANCH
BRANCH-NO              NO       BRANCH-NO             BRANCH
BRANCH-REGION          NO       BRANCH-REGION         BRANCH
BRANCH-SLS-QUOTA       NO       BRANCH-SLS-QUOTA      BRANCH
BRANCH-STATE           NO       BRANCH-STATE          BRANCH
BRANCH-STF-QUOTA       NO       BRANCH-STF-QUOTA      BRANCH
BRANCH-ZIPCODE         NO       BRANCH-ZIPCODE        BRANCH
CSI-DIR-COPY-CARD      NO       **NONE**
CSI-DIR-COPY-DBA-ID    NO       **NONE**
CSI-DIR-COPY-DEFN-KEY  NO       **NONE**
CSI-DIR-COPY-FUNCTION  NO       **NONE**
```

Directory Schema Structure Report (Logical View)

```
CINCOM SYSTEMS, INC. DIRSCSTP - DIRECTORY SCHEMA STRUCTURE  JUNE 27, 1997 16:29:36 PAGE: 1

                            TIS/XA DIRECTORY

                            SCHEMA STRUCTURE FOR BURRYSCH

STRUCTURE SUBREPORT FOR LOGICAL VIEWS

LOGICAL VIEW:  ADD-INVOICE CONT'D

  USER:       EDUC-DBA

              CINCOM SYSTEMS, INC       BATCH DELETE OPTION:    YES
              2300 MONTANA AVENUE       ONLINE DELETE OPTION:   YES
              CINCINNATI                BATCH SAVE OPTION:      YES
              OHIO  45211               ONLINE SAVE OPTION:     YES
                                        BATCH DEFINE QUERIES:   YES
                                        ONLINE DEFINE QUERIES:  YES
                                        BATCH EXECUTE QUERIES:  YES
                                        ONLINE EXECUTE QUERIES: YES
                                        ONLINE SUBMIT OPTION:   NO
                                        DEFINE COMP. RETRIEVAL: YES
                                        EXECUTE COMP. RETRIEVAL:   YES
                                        DEFINE GEN. UPDATES: YES
                                        EXECUTE GEN. UPDATES:   YES
                                        DEFINE LUV APPLICATION: YES
                                        EXECUTE LUV APPLICATION:   YES
  ACCESS SET: ASD-ADD-INVOICE   GEN. IND.:  NO

              0100 KEY  INVOICE-NO
              0200      INVOICE-SLSPRSN
              0300      INVOICE-TOTAL
              0400      INVOICE-BRANCH
              0500      INVOICE-DATE
              0600      INVOICE-CUSTOMER
              0700 KEY  INVLINE-PRODUCT
              0800      INVLINE-QNTY
              0900      INVLINE-PRICE
              1000      PRODUCT-DESC
              1100 ACCESS INV   WHERE INVOICE-NO      =  INVOICE-NO   ALLOW INSERT
              1200 ACCESS INVL  WHERE INVLINE-INVOICE =  INVOICE-NO   ALLOW INSERT
              1300 ACCESS PROD ONCE WHERE PRODUCT-CODE=  INVLINE-PRODUCT
```

Using the Directory Security Group Report

The Directory Security Group Report (DIRSECGP) produces two subreports that describe Security Group-to-user relationships.

- Lists each Security Group along with related Maintenance Restrictions and users ("Directory Security Group Report (Security Groups)" on page 415).
- Lists each user along with attribute data and related Security Groups ("Directory Security Group Report (Users)" on page 416).

You can submit trigger records for category selection, Security Group selection, and user selection. The only valid category selections are SG, which indicates the first subreport, and US, which indicates the second subreport.

The DBA familiar with Security Group structure can submit a trigger record with CONDENSE=Y in positions 1-10 to produce a condensed version of either subreport. The condensed subreports are shown in "Condensed Directory Security Group Report (Security Groups)" on page 417 and "Condensed Directory Security Group Report (Users)" on page 418. If you omit the trigger record, the complete report is produced.

To produce this report, you must submit the User Sign-on trigger record before any other trigger records. Use of this trigger record protects against unauthorized use of this report. When you use this trigger record, the system validates your user ID and password, checks that your ID is related to the logical view for the report, and checks that you are a DBA. If any of these checks fail, the system terminates the report and Security Group output.

Directory Security Group Report (Security Groups)

```
CINCOM SYSTEMS, INC. DIRSECGP - DIRECTORY SECURITY GROUP REPORT JUNE 27, 1997 16:29:36
  PAGE: 6

                           TIS/XA DIRECTORY

SECURITY GROUP SUBREPORT


  SECURITY GROUP:  EDUCSG (CONT'D)

             MAINTENANCE RESTRICTION:  PERMSCSD

                 PERMIT/DENY OPTION    PERMIT
                 CATEGORY CODE
                 COMMAND CODE          SD
                 SUBCOMMAND/SUBCATEGORY

                 NAMING DATA CATEGORY  SC
                 ENTITY NAME 1         CSISCH20
                 ENTITY NAME 2
                 ENTITY NAME 3
                 ENTITY NAME 4

             MAINTENANCE RESTRICTION:  PERMSCST

                 PERMIT/DENY OPTION    PERMIT
                 CATEGORY CODE
                 COMMAND CODE          ST
                 SUBCOMMAND/SUBCATEGORY

                 NAMING DATA CATEGORY  SC
                 ENTITY NAME 1         CSISCH20
                 ENTITY NAME 2
                 ENTITY NAME 3
                 ENTITY NAME 4

  SECURITY GROUP:  EDUCSG

             RELATED USERS:

                 USER-ID       USER NAME

                 EDUC-DBA      CINCOM SYSTEMS, INC.
                 STUDENT       CINCOM SYSTEMS, INC.
```

Directory Security Group Report (Users)

```
CINCOM SYSTEMS, INC.    DIRSECGP - DIRECTORY SECURITY GROUP REPORT JUNE 27, 1997 16:29:36
  PAGE: 6

                           TIS/XA DIRECTORY
USER SECURITY GROUP SUBREPORT

  USER-ID:  **PUBLIC**

              USER NAME     PUBLIC USER
              ADDRESS 1
              ADDRESS 2
              ADDRESS 3
              PHONE NUMBER
              DBA OPTION        N
              LANGUAGE

              RELATED SECURITY GROUPS FOR USER **PUBLIC**

                  SECURITY GROUP    NO. OF MAINT. RESTRICTIONS

                  CSISG001          3

  USER-ID:  CINCOM

              USER NAME     CINCOM SYSTEMS, INC
              ADDRESS 1     2300 MONTANA
              ADDRESS 2     CINCINNATI, OHIO 45211
              ADDRESS 3
              PHONE NUMBER  513 662-2300
              DBA OPTION    N
              LANGUAGE

              RELATED SECURITY GROUPS FOR USER CINCOM

                  SECURITY GROUP    NO. OF MAINT. RESTRICTIONS

                  CSISG001          3
  USER-ID:  CSI-DBA
              USER NAME     CINCOM SYSTEMS, INC
              ADDRESS 1     2300 MONTANA
              ADDRESS 2     CINCINNATI, OHIO 45211
              ADDRESS 3
              PHONE NUMBER  513 662-2300
              DBA OPTION    Y
              LANGUAGE
```

Condensed Directory Security Group Report (Security Groups)

```
CINCOM SYSTEMS, INC. DIRSECGP - DIRECTORY SECURITY GROUP REPORT JUNE 27, 1997 16:29:36
  PAGE: 6

                           TIS/XA DIRECTORY

SECURITY GROUP SUBREPORT (CONDENSED)


  SECURITY GROUP:  CSISG001 NO. OF MRS:  3

    MAINTRESTR    PE/DE CATG    COMD    SUBC    NCTG   ENTITY NAMES

              DENYMR    DENY    MR               1>
                                            2>
                                            3>
                                            4>

              DENYSG    DENY    SG               1>
                                            2>
                                            3>
                                            4>

              DENYUS    DENY    US               1>
                                            2>
                                            3>
                                            4>
  SECURITY GROUP: CSISG001

RELATED USERS:
                  USER-ID       USER NAME

                  **PUBLIC**        PUBLIC USER
                  CINCOM        CINCOM SYSTEMS, INC
                  CSI-DEFAULT-US
                  DEMO
                  EDUCATION     STUDENTS IN EDUCATION
                  STUDENT-1     CINCOM SYTEMS, INC.
                  STUDENT-2     CINCOM SYTEMS, INC.
                  STUDENT-3     CINCOM SYTEMS, INC.
                  STUDENT-4     CINCOM SYTEMS, INC.
                  STUDENT-5     CINCOM SYTEMS, INC.
                  STUDENT-6     CINCOM SYTEMS, INC.
                  STUDENT-7     CINCOM SYTEMS, INC.
                  STUDENT-8     CINCOM SYTEMS, INC.
                  STUDENT-9     CINCOM SYTEMS, INC.
```

Condensed Directory Security Group Report (Users)

```
CINCOM SYSTEMS, INC. DIRSECGP - DIRECTORY SECURITY GROUP REPORT JUNE 27, 1997 16:29:36
  PAGE: 1

                            TIS/XA DIRECTORY

USER SECURITY GROUP SUBREPORT (CONDENSED)

  USER-ID:  **PUBLIC**  USER NAME:  PUBLIC USER DBA?  N

              CSISG001              3

  USER-ID:  CINCOM      USER NAME:  CINCOM SYSTEMS, INC  DBA?  N

              CSISG001              3

  USER-ID:  CSI-DBA     USER NAME:  CINCOM SYSTEMS, INC  DBA?  Y

              CSISG001

  USER-ID:  CSI-DEFAULT-US  USER NAME:      DBA?  N

              CSISG001              3

  USER-ID:  DEMO        USER NAME:      DBA?  N

              CSISG001              3

USER-ID:  EDUC-DBA      USER NAME:CINCOM SYSTEMS, INC DBA?  Y

              EDUCSG                14

  USER-ID:  EDUCATION   USER NAME:STUDENTS IN EDUCATION  DBA?  N

              CSISG001              3

  USER-ID:  STUDENT     USER NAME:CINCOM SYSTEMS, INC DBA?  Y

              EDUCSG                14

  USER-ID:  STUDENT-1   USER NAME:  PUBLIC USER DBA?  N

              CSISG001              3

  USER-ID:  STUDENT-2   USER NAME:  PUBLIC USER DBA?  N

              CSISG001              3
```

Using RDM views to create Directory reports

As an alternative to the Directory reports and subreports Cincom provides, you may use the RDM views provided with Directory Maintenance to create customized Directory reports (see the following table). Each view contains an explanation of its use. You use the RDM views to create SPECTRA processes. An example SPECTRA process is given at the end of this section. The Batch Directory Maintenance member to load the RDM Directory views is DIRLDVAD in the product MACLIB.

You should not use the Design Directory views, which begin with GET, INSERT, DELETE, or UPDATE. These views are specialized and would ruin the Directory if used for general purposes.

View	Description
CSI-DIR-REL-XF-AT	Displays relationships between external fields (XF) and attributes (AT).
CSI-DIR-REL-XF-US	Displays relationships between external fields (XF) and users (US).
CSI-DIR-REL-RE-SC	Displays relationships between relations (RE) and schemas (SC).
CSI-DIR-REL-RE-IR	Displays relationships between relations (RE) and internal records (IR).
CSI-DIR-REL-RE-US	Displays relationships between relations (RE) and users (US).
CSI-DIR-REL-PF-PF	Displays relationships between physical fields (PF) and physical fields (PF).
CSI-DIR-REL-LV-XF	Displays relationships between logical views (LV) and external fields (XF).
CSI-DIR-REL-LV-US	Displays relationships between logical views (LV) and users (US).
CSI-DIR-REL-LV-ED	Displays relationships between logical views (LV) and Environment Descriptions (ED).
CSI-DIR-REL-LV-AS	Displays relationships between logical views (LV) and access sets (AS).
CSI-DIR-REL-KC-PF	Displays relationships between key codes (KC) and physical fields (PF).

View	Description
CSI-DIR-REL-FK-AT	Displays relationships between foreign keys (FK) and attributes (AT).
CSI-DIR-REL-PK-AT	Displays relationships between primary keys (PK) and attributes (AT).
CSI-DIR-REL-ED-FI	Displays relationships between Environment Descriptions (ED) and files (FI).
CSI-DIR-REL-US-SG	Displays relationships between users (US) and Security Groups (SG).
CSI-DIR-REL-AT-US	Displays relationships between attributes (AT) and users (US).
CSI-DIR-USER	Displays fixed attribute data for user (US).
CSI-DIR-LT-USER	Displays long text data for user (US).
CSI-DIR-PROCEDURE	Displays fixed attribute data for procedure (PR).
CSI-DIR-LT-PROCEDURE	Displays long text data for procedure (PR).
CSI-DIR-VD-TABLES	Displays variable attribute data for tables (TA).
CSI-DIR-TABLES	Displays fixed attribute data for tables (TA).
CSI-DIR-LT-TABLES	Displays long text data for tables (TA).
CSI-DIR-RESERVED-WORD	Displays fixed attribute data for reserved word (RW).
CSI-DIR-LT-RESERVED-WORD	Displays long text data for reserved word (RW).
CSI-DIR-EDIT-MASK	Displays fixed attribute data for edit mask (EM).
CSI-DIR-LT-EDIT-MASK	Displays long text data for edit mask (EM).
CSI-DIR-DIR-COMP-DESC	Displays fixed attribute data for Directory component description (DC).
CSI-DIR-LT-DIR-COMP-DESC	Displays long text data for Directory component description (DC).
CSI-DIR-SECONDARY-KEY	Displays fixed attribute data for Secondary Key (SK).
CSI-DIR-LT-SECONDARY-KEY	Displays long text data for Secondary Key (SK).

View	Description
CSI-DIR-SCHEMA	Displays fixed attribute data for schema (SC).
CSI-DIR-LT-SCHEMA	Displays long text data for schema (SC).
CSI-DIR-PHYSICAL-FIELD	Displays fixed attribute data for physical field (PF).
CSI-DIR-LT-PHYSICAL-FIELD	Displays long text data for physical field (PF).
CSI-DIR-KEY-CODE	Displays fixed attribute data for key code (KC).
CSI-DIR-LT-KEY-CODE	Displays long text data for key code (KC).
CSI-DIR-INTERNAL-RECORD	Displays fixed attribute data for internal record (IR).
CSI-DIR-LT-INTERNAL-RECORD	Displays long text data for internal record (IR).
CSI-DIR-FILE	Displays fixed attribute data for file (FI).
CSI-DIR-LT-FILE	Displays long text data for file (FI).
CSI-DIR-ENVIRON-DESC	Displays fixed attribute data for Environment Description (ED).
CSI-DIR-LT-ENVIRON-DESC	Displays long text data for Environment Description (ED).
CSI-DIR-BUFFER-POOL	Displays fixed attribute data for buffer pool (BP).
CSI-DIR-LOGICAL-VIEW	Displays fixed attribute data for logical view (LV).
CSI-DIR-LT-LOGICAL-VIEW	Displays long text data for logical view (LV).
CSI-DIR-EXTERNAL-FIELD	Displays fixed attribute data for external field (XF).
CSI-DIR-LT-EXTERNAL-FIELD	Displays long text data for external field (XF).
CSI-DIR-VD-ACCESS-SET	Displays variable attribute data for access set (AS).
CSI-DIR-ACCESS-SET	Displays fixed attribute data for access set (AS).
CSI-DIR-LT-ACCESS-SET	Displays long text data for access set (AS).

View	Description
CSI-DIR-RELATION	Displays fixed attribute data for relation (RE).
CSI-DIR-LT-RELATION	Displays long text data for relation (RE).
CSI-DIR-PRIMARY-KEY	Displays fixed attribute data for primary key (PK).
CSI-DIR-FOREIGN-KEY	Displays fixed attribute data for foreign key (FK).
CSI-DIR-LT-FOREIGN-KEY	Displays long text data for foreign key (FK).
CSI-DIR-DOMAIN	Displays fixed attribute data for domain (DM).
CSI-DIR-LT-DOMAIN	Displays long text data for domain (DM).
CSI-DIR-CONCEP-SCHEMA	Displays fixed attribute data for conceptual schema (CS).
CSI-DIR-LT-CONCEP-SCHEMA	Displays long text data for conceptual schema (CS).
CSI-DIR-ATTRIBUTE	Displays fixed attribute data for attribute (AT).
CSI-DIR-LT-ATTRIBUTE	Displays long text data for attribute (AT).
CSI-DIR-SECURITY-GROUP	Displays fixed attribute data for Security Group (SG).
CSI-DIR-LT-SECURITY-GROUP	Displays long text data for Security Group (SG).
CSI-DIR-MAINT-RESTRICTION	Displays fixed attribute data for Maintenance Restriction (MR).

Example SPECTRA process

This process will generate a hard copy report of the Directory metadata for the given Conceptual Schema and Internal Schema.

```
PROCESS NAME: CSI-MIG-MRP-CS

 CSG-PROCESS-LANGUAGE-STATEMENT ENGLISH
 *  TO EXECUTE THIS PROCESS, SUPPLY A CONCEPTUAL SCHEMA NAME AND A
 *  SCHEMA NAME, E.G.,
 *     GO CSI-MIG-MRP-CS BURRYS-CONCEPTUAL-SCHEMA BURRYSCH
 GET CS-NAME, SC-NAME FROM TERMINAL

 SET       X = "================================="
 LINESIZE 132
 GET FROM CSI-DIR-RELATION
     WHERE RE-CONCEPTUAL-SCHEMA = CS-NAME
     ORDER BY RE-RELATION

 TITLE NODEFAULT
     LEFT DATE3 " " TIME12
   CENTER "DIRECTORY METADATA MIGRATION REPORT "
          ";FOR CONCEPTUAL SCHEMA: " TRIM(RE-CONCEPTUAL-SCHEMA)
          ", RELATED TO SCHEMA: " SC-NAME@1..8 ";;"
   RIGHT "PAGE " PAGECOUNT

 PRINT RE-RELATION@1..18        AS "RELATION"
       RE-RELATION-TYPE         AS "RELATION TYPE"
       RE-SUBTYPE               AS "SUBTYPE"
       RE-EXTENSION             AS "EXTENSION"
       RE-NUMBER-OF-TUPLES      AS "NUMBER OF TUPLES"
       RE-DDNAME                AS "DD-NAME"
       RE-DATA-SET-NAME         AS "DATA SET NAME"
       RE-FILE-NAME             AS "FILE NAME"

 GET FROM CSI-DIR-ATTRIBUTE
     WHERE AT-CONCEPTUAL-SCHEMA = RE-CONCEPTUAL-SCHEMA
       AND AT-RELATION          = RE-RELATION

 GET FROM CSI-DIR-DOMAIN
     WHERE DM-DOMAIN            = AT-DOMAIN
 PRINT AT-ATTRIBUTE@1..18       AS "ATTRIBUTE"         ONCE
       AT-PRIMARY-KEY           AS "PRIMARY;KEY"       ONCE
       DM-DOMAIN@1..18          AS "DOMAIN"            ONCE
       DM-DATA-FORMAT@5..18     AS "DATA FORMAT"       ONCE
       DM-LENGTH                AS "LENGTH"            ONCE
       DM-NUMBER-DECIMAL-PLACES AS "DECIMAL;PLACES"    ONCE
       DM-NULLS-ALLOWED-OPTION  AS "NULLS;ALLOWED"     ONCE
       DM-VALIDATION-OPTION     AS "VALIDATION;OPTION" ONCE

 GET FROM CSI-DIR-REL-XF-AT
     WHERE XF-AT-CONCEPTUAL-SCHEMA = AT-CONCEPTUAL-SCHEMA
       AND XF-AT-RELATION          = AT-RELATION
       AND XF-AT-ATTRIBUTE         = AT-ATTRIBUTE
       AND XF-AT-SCHEMA            = SC-NAME
 PRINT XF-AT-EXTERNAL-FIELD     AS "EXTERNAL FIELD"
```

```
WHEN CSI-DIR-ATTRIBUTE FINISHED (

     SPACE 1
     MESSAGE X & X & X & X
     SPACE 1
     MESSAGE "FOREIGN KEY                    " & " "
             "TARGET RELATION                " & " "
             "TYPE"                            & " "
             "SOURCE ATTRIBUTE               "

     MESSAGE "-----------------------------" & " "
             "-----------------------------" & " "
             "----"                            & " "
             "-----------------------------"

GET FROM CSI-DIR-FOREIGN-KEY
    WHERE FK-CONCEPTUAL-SCHEMA = RE-CONCEPTUAL-SCHEMA
      AND FK-RELATION          = RE-RELATION

WHEN FK-FOREIGN-KEY CHANGES (

     GET FROM CSI-DIR-REL-FK-AT
         WHERE FK-AT-CONCEPTUAL-SCHEMA = FK-CONCEPTUAL-SCHEMA
           AND FK-AT-RELATION          = FK-RELATION
           AND FK-AT-FOREIGN-KEY       = FK-FOREIGN-KEY
     MESSAGE FK-FOREIGN-KEY              & " "
             FK-PRIMARY-KEY-RELATION     & "  "
             FK-FOREIGN-KEY-TYPE         & "   "
             FK-AT-ATTRIBUTE             & "     " )

WHEN CSI-DIR-FOREIGN-KEY MISSING (MESSAGE "*** NO FOREIGN KEYS EXIST ***" ))
WHEN RE-RELATION CHANGES (
     SPACE 1
     MESSAGE X & X & X & X
     SPACE 1
     MESSAGE "RELATED USERS"
     MESSAGE "-----------------------------"

     GET FROM CSI-DIR-REL-RE-US
         WHERE RE-US-CONCEPTUAL-SCHEMA = RE-CONCEPTUAL-SCHEMA
          AND RE-US-RELATION          = RE-RELATION

     MESSAGE RE-US-USER

WHEN CSI-DIR-REL-RE-US MISSING (

     MESSAGE "*** NO USERS ARE RELATED ***" ))
```

The following steps explain the preceding example SPECTRA process:

1. Read in CS, SC from input stream.
2. Initialize a literal to create the underline used to separate the report sections, ‘X = “==========” .
3. Set the report width to 132 characters.
4. Read in fixed attribute data about the entity relation, where RE-CONCEPTUAL- SCHEMA is equal to the given CS, sorting according to RE-RELATION.
5. If new RE-RELATION:
   a. Print date, time, report header, Conceptual Schema, and Physical Schema at the top of a new page.
   b. Print the following values next to the corresponding headings: (See previous example output report)

      RE-RELATION (Will be truncated to 18 characters)
      RE-RELATION-TYPE
      RE-SUBTYPE
      RE-EXTENSION
      RE-ORDERED-PRIMARY-KEY
      RE-NUMBER-OF-TUPLES
      RE-DDNAME
      RE-DATA-SET-NAME
      RE-FILE-NAME

c. Read in fixed attribute data about the entity attribute, where: AT-CONCEPTUAL-SCHEMA is equal to RE-CONCEPTUAL-SCHEMA and AT-RELATION is equal to RE-RELATION.D. Do for each AT-ATTRIBUTE:

   i. Read in fixed attribute data about the entity Domain, where DM-DOMAIN = AT-DOMAIN.

   ii. Print the following values under the corresponding headings (see sample output report):

   AT-ATTRIBUTE (Will be truncated to 18 characters)
   AT-PRIMARY-KEY
   AT-DOMAIN (Will be truncated to 18 characters)
   DM-DATA-FORMAT (Will be a substring from positions 5 to 18)
   DM-LENGTH
   DM-NUMBER-DECIMAL-PLACES
   DM-NULLS-ALLOWED-OPTION
   DM-VALIDATION-OPTION

   iii. Read in data about existing relationships between external fields and attributes.

   iv. Print external field information under corresponding header.

e. Print section divider lines.

f. Print headers for foreign key, target relation, type, and source attribute columns.

g. Read in fixed attribute data about the entity foreign key.

h. If new foreign key:

   i. Read in fixed attribute data about the relationship between old foreign key and attribute.

   ii. Print old foreign key, primary key relation, type and attribute under the corresponding headers.

i. If no foreign key records found, print appropriate message under headers.

j. Print headers for related users and DBA option.

k. Read in fixed attribute data about the relationship between relations and users.

l. Print users and DBA option under corresponding headers.

m. If no user record exists, print appropriate message under headers.

11

Routing console messages (OS/390)

Routing process

You can route console messages in any SUPRA Server component that uses the WTO CLM or CSEOCONS. To control the console message destination, assemble the macro CSIMCS and link edit the resulting module (MCSCODES). Next, you need to update the linkdecks to include the modules CSEOCONS and MCSCODES.

The following table shows the components that allow you to tailor console message destination using MCSCODES. It shows the specific component, any linkdecks for the component, and the resulting composite module name.

Component	Linkdeck names	Module names
PDM-Single-task	CSTL000C	CSTADBMS
PDM-Multitask	CSTL000D	DBMS
PDM Interface -STST	CSTN998Q	CSTESTST
-STMT	CSTN998R	CSTESTMT
-CICS	CSTN998S	CSTXCSMT
-MTMT	CSTN998T	CSTEDCMT
-CSIPARM I/O	CSTL997C	CSTH050C
PDM Descriptor Generators	CSTM004D	CSTGVDBD (Create VALMOD)
	CSTM006D	CSTGMSCH (Modify SCHEMA)
	CSTM007D	STGGENV (Create ENVDESC)
Batch Directory Maintenance	LKBDIRMT	CSMBDIRM
Online Directory Maintenance	LKCDIRMT	CSMCDIRM (For CICS only)
DDI Program	LKCSMDDR	CSMYDDI
DDI Program (V2; HDMP:PDM)	LKCSMDD2	CSMYDDI2
DDL Convert	LKDDLCVT	CSMUDCVT
Migration Program	LKMIGUTL	CSMBMIGU
Migration Program 3	LKMIGUT3	CSMBMIG3
Migration Program 4	LKMIGUT4	CSMBMIG4
Migration Program 5	LKMIGUT5	CSMBMIG5
Migration Program 6	LKMIGUT6	CSMBMIG6
DBA Utilities	CSUZMEX	CSUXSTAT
	CSUZMTM	CSUTTERM
	CSUZMUTN	CSUNUTIL
	CSUZMUTO	CSUOUTIL
Relational Data Manager	CSVODBA	CSVODBA
SPECTRA	None	None
MANTIS	None	None

12

PDM Options Module

Replacement for optional fixes

The PDM Options Module is a replacement for optional fixes that controlled the way the PDM handled the features described below. If you previously applied any of the optional fixes that controlled these features, you will no longer need to apply them as of SUPRA 2.7. Instead you would use the OPTIONS module.

This module also introduces options and requirements that are new with SUPRA 2.7.

Assembling the C$THOPTN macro

The macro C$THOPTN builds the PDM Option module CSTHOPTN. To assemble this macro, see CSTHOPTN, a member on the distributed source statement library for the CSTHOPTN assembly default used in creating the distributed CSTHOPTN module.

Macro C$THOPTN is located in MACLIB for OS/390, and C$THOPTN.A and CSTHOPTN.A in VSE base library.

Assemble with selected options as documented below, and link CSTHOPTN into your base library. Link CSTHOPTN in AMODE24/RMODE24. For the OS/390 CICS Interface, CSTHOPTN must be in one of the STEPLIB concatenated authorized libraries.

Macro call format:

```
****** ***************************** Top of Data ******************************
 000001       C$THOPTN TYPE=TABLE,                                             X
 000002             BBPSLOG=N,            BYPASS SLOG BP DEALLOC ON SWAP       X
 000003             BDTDL=N,                 BYPASS DTDL CREATION              X
 000004             RECST=1000,              NON-TLR RECORD STEAL COUNT        X
 000005             USTATX=N,                USER STATUS EXIT (CSTHUSXT)       X
 000006             RDAH=Y,                  READ AHEAD BUFFERING FACILITY     X
 000007             PSWRD=nnnnnnnn,          PASSWORD                          X
 000008             BEGDAT=20yymmdd,         BEGIN DATE (PASSWORD)             X
 000009             ENDDAT=20yymmdd,         END DATE (PASSWORD)               X
 000010             WARDAY=60,               WARNING DAYS BEFORE EXPIRATION    X
 000011             BLDSTFILE=Y,             BUILD FILE C.B. AT INIT., ST      X
 000012             BLDMTFILE=Y              BUILD FILE C.B. AT INIT., MT      X
 000013       SPACE 2
 000014       C$THOPTN TYPE=DSECT
 000015       SPACE 2
 000016       END

 ****** **************************** Bottom of Data ****************************
```

Available options

BBPSLOG

$$\left[\textbf{BBPSLOG} = \left\{\begin{matrix}\underline{\textbf{N}} \\ \textbf{Y}\end{matrix}\right\},\right]$$

Description *Optional.* Prevents releasing the memory used for a buffer pool when the system log file swaps to another volume.

Default N Releases memory and then acquires more.

Options Y Prevents releasing memory.

Format 1 alphanumeric characters

Consideration When the option is selected, relate all the system log files in a log group to the same buffer pool.

This option is valid for the PDM only. Select by placing a **Y** in place of the **N**. If selected you must ensure that all the system log files in a log group are related to the same buffer pool. Its purpose is to circumvent a possible ICOR when the system log swaps to another volume. Normally, the file is closed, which releases the buffer pool. If PDM storage is fragmented, the opening of the new volume, which sets up a new buffer pool, could result in an ICOR status.

BDTDL

$$\left[\mathbf{BDTDL} = \left\{\begin{matrix}\underline{\mathbf{N}} \\ \mathbf{Y}\end{matrix}\right\},\right]$$

Description *Optional.* Prevents creating control blocks that relate data lists to individual tasks.

Default N Builds control blocks to associate data list information to each task that uses the data list.

Options Y Prevents building data list control blocks per task.

Format 1 alphanumeric character

Consideration The option can affect performance and memory usage. If the user has few data lists, which are used by many users, selecting Y may be beneficial.

This option is valid for the PDM only. Select by placing a **Y** in place of the **N**. If selected, this may affect performance, and will reduce the PDM memory usage. If many varieties of data lists exist in applications, the option may degrade performance. If DBIO routines are utilized with fixed data lists that all applications use, this option may increase performance.

BEGDAT

[BEGDAT=20*yymmdd*,]

Description	*Required*. The begin date is generated by Cincom and is set in the options module on your installation tape. The begin date is part of the password generation algorithm and is valid when run on or after the date specified, and on or before the end date specified. The date must match the date used by Cincom to generate the password. If you change the CPU on which SUPRA is executing, a different password is required. The Password is valid for the CPU it was configured for and from the start date until the end date that is specified below.	
Default	20*yymmdd*	The begin date used to generate the password above.
Options	20*yymmdd*	New date used in password generation based on new dates and CPU.
Format	8 numeric characters that represent the year, month and date.	
Consideration	This keyword is valid for the PDM. If a valid password is not present, the SUPRA database will not initialize For the OS/390 CICS Interface, CSTHOPTN must be in one of the STEPLIB concatenated authorized libraries.	

BLDMTFILE

$$\left[\textbf{BLDMTFILE} = \left\{\frac{\textbf{Y}}{\textbf{N}}\right\},\right]$$

Description *Optional*. The default causes system initialization to access the Directory to build the control blocks for the user data base files defined in the environment descriptor in the multi- task PDM. If the user has large numbers of data files and does not do automatic opens, but only opens files as they are needed, changing this option to N, can shorten the time for initialization and use less memory since it only will build the data file control blocks it needs. This replaces Optional FIX 950270 in SUPRA 2.6 If this fix was applied to your SUPRA 2.6 system then you should change the value to N.

Default Y build all data file control blocks listed in the environment descriptor during initialization.

Options N build the data file control block only when an open for that file is requested.

Format 1 alphabetic character

Consideration The option can affect performance and memory usage in the PDM. Leave the default, Y, if you open all your environment descriptor files at initialization. Mantis Binding will not work with this applied.

If you use MANTIS binding and have chosen BLDSTFILE=N, then an FNTF0106CSTA360D will occur, which causes a TVI.

BLDSTFILE

$$\left[\textbf{BLDSTFILE} = \left\{\frac{\textbf{Y}}{\textbf{N}}\right\},\right]$$

Description *Optional*. The default causes system initialization to access the Directory to build the control blocks for the user data base files defined in the environment descriptor in the single task PDM. If the user has large numbers of data files and does not do automatic opens, but only opens files as they are needed, changing this option to N, can shorten the time for initialization and use less memory since it only will build the data file control blocks it needs. This replaces Optional FIX 950270 in SUPRA 2.6 If this fix was applied to your SUPRA 2.6 system then you should change the value to N.

Default Y build all data file control blocks listed in the environment descriptor during initialization.

Options N build the data file control block when an open for that file is requested.

Format 1 alphabetic character

Consideration The option can affect performance and memory usage in the PDM. Leave the default, Y, if you open all your environment descriptor files at initialization. Mantis Binding will not work with this applied.

If you use MANTIS binding and have chosen BLDSTFILE=N, then an FNTF0106CSTA360C will occur, which causes a TVI.

ENDDAT

[ENDDAT=20*yymmdd*,]

Description *Required*. The end date is generated by Cincom and is set in the options module on your installation tape. The end date is part of the password generation algorithm and is valid when run on or before the date specified, and on or after the begin date specified. The date must match the date used by Cincom to generate the password. If you change the CPU on which SUPRA is executing, a different password is required. The Password is valid for the CPU it was configured for and from the start date until the end date specified.

Default 20*yymmdd* The end date used to generate the password above.

Options 20*yymmdd* New date used in password generation based on new dates and CPU.

Format 8 numeric characters that represent the year, month and date.

Consideration This keyword is valid for the PDM. If a valid password is not present, the SUPRA database will not initialize For the OS/390 CICS Interface, CSTHOPTN must be in one of the STEPLIB concatenated authorized libraries.

PSWRD

[PSWRD=*nnnnnnnnn*,]

Description *Required*. The password is generated by Cincom and is set in the options module on your installation tape. If you change the CPU on which SUPRA is executing, a different password is required. The Password is valid for the CPU it was configured for and from the start date until the end date that is specified below.

Default *nnnnnnnn* The password at installation. Do not use the exit.

Options *mmmmmmmm* New password based on new dates and CPU.

Format 8 numeric characters

Consideration This keyword is valid for the PDM. If a valid password is not present, the SUPRA database will not initialize. For the OS/390 CICS Interface, CSTHOPTN must be in one of the STEPLIB concatenated authorized libraries. The keywords: PSWRD, BEGDAT, ENDDAT and WARDAY are all part of the Security password protection for SUPRA. All the keywords must have valid values as supplied by CINCOM for your CPU and be used within the dates specified.

RDAH

$$\left[\mathbf{RDAH} = \left\{\begin{matrix}\underline{\mathbf{Y}} \\ \mathbf{N}\end{matrix}\right\},\right]$$

Description *Optional.* Executes readahead buffering performance option.

Default N Sets default to not execute performance option.

Options Y Performance option is in effect.

Format 1 alphanumeric character

Consideration The option can affect performance and memory usage in the batch interface either positively or negatively. CSIPARM values are used to tune this feature and can override this default.

Readahead is a performance feature that is new with SUPRA release 2.7. It uses resources in the STMT interfaces and may prove to be a feature that the user wishes to turn off. Rather than having to change all the CSIPARM files to indicate that the feature is not wanted, the user can set the option to **N** in the CSTHOPTN module. This will change the default for Readahead to No. If the feature is desired for some heavy read-only applications, the CSIPARM can be used to turn it back on.

RECST

$$\left[\textbf{RECST} = \left\{\begin{matrix}\underline{\textbf{1000}} \\ \textbf{\textit{nnnnn}}\end{matrix}\right\},\right]$$

Description *Optional.* Establishes the record stealing count when executing the PDM in Non-TLR mode. When this value is exceeded, the requesting task can steal the held record.

Default 1000 Sets default to steal after 1000 requests.

Options *nnnnn* 1-32,767.

Format 1-5 numeric characters

Consideration *nnnnn* cannot exceed 32,767

This option is valid for the PDM only. Change the count by replacing the **1000** with the 1 to 5 digit value of your choice, not to exceed 32,767. Valid only while running the PDM in Non-TLR Mode. Establishes the record stealing count, which when exceeded, will allow the requesting task to steal the held record.

USTATX

$$\left[\textbf{USTATX} = \left\{\begin{matrix}\underline{\textbf{N}} \\ \textbf{Y}\end{matrix}\right\},\right]$$

Description *Optional.* Adds an additional user exit for the purpose of viewing the extended status in the PDM or in the Interface.

Default N Do not use the exit.

Options Y Use the exit.

Format 1 alphabetic character

Consideration This option is valid for the PDM, and the PDM Interfaces. It functions the same way as the extended status function of the PDM Interface Post Command Exit. Select by placing a Y in place of the N. If selected, the User Status Exit (CSTHUSXT) must be available for the PDM, or the PDM will abend with CSTA360T and CSTA670T messages during initialization. If CSTHUSXT is not available on the PDM Interface side, this option will be ignored, and processing will continue normally, without the exit. For the OS/390 CICS Interface, CSTHOPTN and CSTHUSXT must be in one of the STEPLIB concatenated authorized libraries.

WARDAY

[WARDAY=60]

Description	*Required.* The value represents the number of days before the end date that a message will begin to be generated. The message will indicate that the password is expiring on the end date listed above. A new password will be required at the time of the end date.	
Default	60	The number of days prior to the end date that you want to be notified about the expiration of your password.
Options	*nn*	1-60.
Format	2 numeric characters that represent the number of days warning you wish to receive before your password expires.	
Consideration	This keyword is valid for the PDM. If you specify 0, the default of 60 will be used. The password expiring message will appear on the console at initialization time starting nn days prior to the end date.	

A

Using SUPRA Server sample procedures, macros, schemas, linkdecks, and sample source members

Samples provided

SUPRA Server includes the following:

- Sample JCL members and procedures to execute all SUPRA Server components
- Macros to generate linkdecks for Batch and CICS Directory Maintenance, to provide defaults for Comprehensive Retrieval RDL programs, to provide defaults for the Directory and the PDM, to build the CICS operator control parameter table, and to provide routing and descriptor codes for console messages
- Macros to generate CICS Connector user exit parameter lists and work areas
- Sample bootstrap and environment schemas
- Sample linkdecks
- Sample source members for exit programs, options, macro assemblies, and Directory transactions

Using SUPRA Server sample procedures and JCL

SUPRA Server includes sample JCL members and procedures to execute all components. The JCL samples are located in the SUPRA Server JCL library (JCLLIB). The procedures used in the sample JCL are located in the SUPRA Server procedure library (PROCLIB). The libraries contain index members providing a short description of each sample. The index member for the JCLLIB is TXJ$INDX. The index member for the PROCLIB is TIS$PRC.

Consider the following:

- Sample JCL members and procedures contain comments describing the symbolic parameters.
- Sample input members are located in the SUPRA Server MACLIB; the index member for input samples is TXI$INDX. Sample UCLCODE members are located in the SUPRA Server UCLCODE library; the index member is TXU$INDX.

Using SUPRA Server macros

SUPRA Server macros are located in the SUPRA Server macro library (MACLIB). Index member $INDMSU2 provides a short description of each macro. This appendix provides usage information for the following macros:

- **C$MDMLNK.** Generates linkdecks for Batch and CICS Directory Maintenance.
- **C$THOPTN.** Provides the ability to set default values to control the execution of the PDM. (See the previous section above)
- **C$SAWORK.** Provides default values to be used in all Comprehensive Retrieval RDL programs.
- **C$TXCMUE.** Common user exit PARM and work area.
- **C$TXMROT.** Defines MRO Remote Transactions for CSTXOPER to CSTXLST1.
- **C$TXTDUE.** Task Detach User Exit PARM and work area.
- **C$TXTPOP.** Post-Processing User Exit PARM and work area.
- **C$TXTPOS.** Post-DML User Exit PARM and work area.
- **C$TXTPRE.** Pre-DML User Exit PARM and work area.
- **C$TXTPRP.** Pre-Process User Exit PARM and work area.
- **C$TXTSYN.** User Synchronization Exit PARM and work area.
- **C$TXUTID.** User Task ID Exit PARM and work area.
- **C$XCDFLT.** Provides defaults for the Directory and the PDM.
- **CSTXOPRM.** Builds the CICS operator Control Parameter table. (See *SUPRA Server PDM CICS Connector Systems Programming Guide (OS/390 & VSE)*, P26-7452.
- **CSIMCS.** Provides routing and/or descriptor codes for console messages.

Assembling the C$MDMLNK macro

The C$MDMLNK macro generates linkdecks for batch and CICS Directory Maintenance by generating linkage editor Input Control statements to define the number of load modules you will need. Default linkage-editor-input streams, LKBDIRMT for batch, and LKCDIRMT for CICS are supplied for installation. To assemble the C$MDMLNK macro, substitute your ENVIRN, NUMLM, and OPSYS parameters to produce a new customized linkdeck. Then execute the linkage editor using this new linkdeck as input.

This macro generates up to twelve load modules. Use this macro only if you want to change the default values.

C$MDMLNK ENVIRN= { **BATCH** | **CICS** } **X**

,NUMLM=*n* **X**

,OPSYS= { **OS** | **DOS** }

ENVIRN= $\left\{\begin{matrix}\textbf{BATCH}\\ \textbf{CICS}\end{matrix}\right\}$

Description *Required*. Indicates whether you are running batch or CICS.

NUMLM=*n*

Description *Required*. Specifies the number of load modules.

Options 1–12

Considerations

- In a CICS environment, the recommended number of load modules is twelve. One is invalid for CICS. The LKCDIRMT member on the installation library contains the twelve-load module, CICS Directory Maintenance linkdeck.
- In a batch environment, the recommended number of load modules is one for OS/390 and twelve for VSE. Two are invalid for OS/390. The LKBDIRMT member on the VSE installation library contains the twelve-load module, Batch Directory Maintenance linkdeck; the LKBDIRMT linkdeck member on the OS/390 installation library contains the single-load module, Batch Directory Maintenance.

OPSYS= {**OS** | **DOS**}

Description *Required*. Indicates the operating system where you use these linkdecks.

General consideration

If you reassemble the C$MDMLNK macro for a new load-module configuration, replace the distributed LKBDIRMT or LKCDIRMT linkdecks with your new, customized linkdecks. After you link edit Directory Maintenance using your new linkdeck(s), you will have one- to twelve-load modules as follows (depending on the considerations shown for the C$MDMLNK macro):

Batch output	CICS output
CSMBDIRM	CSMCDIRM
CSMBDIR1	CSMCDIR1 (OS/390 only)
CSMBDIR2 (OS/390 only)	CSMCDIR2
CSMCDIR3	CSMCDIR3
CSMBDIR4	CSMCDIR4
CSMBDIR5	CSMCDIR5
CSMBDIR6 (VSE only)	CSMCDIR6 (VSE only)
CSMBDIR7	CSMCDIR7
CSMBDIR8	CSMCDIR8
CSMBDIR9	CSMCDIR9
CSMBDIRA	CSMCDIRA
CSMBDIRB	CSMCDIRB

Assembling the C$SAWORK macro

The C$SAWORK macro supplies Comprehensive Retrieval with the dynamically loaded CSSWORK module containing system and default values for various Report Definition Language (RDL) statements. If you want to change any values, reassemble the CSSWORK source module with your parameters and link the resulting object module as the load module for CSSWORK.

You can override some of these parameters during a run by using the associated RDL statement in your program as indicated in the following descriptions:

$$\left[\mathbf{DATE} = \left\{\frac{\mathbf{A}}{\mathbf{E}}\right\}\right]$$ **X**

$$\left[\mathbf{,EXIN} = \left\{\frac{\mathbf{IPT}}{\mathbf{nnn}}\right\}\right]$$ **X**

$$\left[\mathbf{,EXINBLK} = \left\{\frac{\mathbf{80}}{\boldsymbol{nnnnn}}\right\}\right]$$ **X**

$$\left[\mathbf{,EXINREC} = \left\{\frac{\mathbf{80}}{\boldsymbol{nnnnn}}\right\}\right]$$ **X**

$$\left[\mathbf{,EXOT} = \left\{\frac{\mathbf{002}}{\mathbf{nnn}}\right\}\right]$$ **X**

$$\left[\mathbf{,LINESZ} = \left\{\frac{\mathbf{132}}{\mathbf{nnn}}\right\}\right]$$ **X**

$$\left[\mathbf{,MAXERR} = \left\{\frac{\mathbf{0}}{\boldsymbol{nnnnn}}\right\}\right]$$ **X**

$$\left[\mathbf{,MINBUF} = \left\{\frac{\mathbf{1504}}{\boldsymbol{nnnnn}}\right\}\right]$$ **X**

$$\left[,\mathbf{MINREC} = \left\{\begin{matrix}\underline{\mathbf{250}}\\ \mathit{nnnnn}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{PAGESZ} = \left\{\begin{matrix}\underline{\mathbf{58}}\\ \mathit{nnnnn}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{PARM} = \left\{\begin{matrix}\underline{\mathbf{024}}\\ \mathbf{nnn}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{PARMLNG} = \left\{\begin{matrix}\underline{\mathbf{80}}\\ \mathit{nnn}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{PRIN} = \left\{\begin{matrix}\underline{\mathbf{001}}\\ \mathbf{nnn}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{RPTR} = \left\{\begin{matrix}\underline{\mathbf{IPT}}\\ \mathbf{nnn}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{RPTRBLK} = \left\{\begin{matrix}\underline{\mathbf{80}}\\ \mathit{nnnnn}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{RPTRREC} = \left\{\begin{matrix}\underline{\mathbf{80}}\\ \mathit{nnnnn}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{SOCAREA} = \left\{\begin{matrix}\underline{\mathbf{25}}\\ \mathit{nnnnnnnnnn}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{SOCLCB} = \left\{\begin{matrix}\underline{\mathbf{022}}\\ \mathbf{nnn}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{SOCLST} = \left\{\begin{matrix}\underline{\mathbf{LST}}\\ \mathbf{nnn}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\text{SOCSYS} = \left\{\begin{matrix}\underline{\text{023}}\\ \text{nnn}\end{matrix}\right\}\right] \qquad \text{X}$$

$$\left[,\text{SORTCOR} = \left\{\begin{matrix}\underline{\text{0}}\\ \textit{nnnnnnnnnn}\end{matrix}\right\}\right] \qquad \text{X}$$

$$\left[,\text{SORTNAM} = \text{sortname}\right] \qquad \text{X}$$

$$\left[,\text{SORTWK} = \left\{\begin{matrix}\underline{\text{4}}\\ \text{n}\end{matrix}\right\}\right] \qquad \text{X}$$

$$\left[,\text{WORKFLD} = \left\{\begin{matrix}\underline{\text{20}}\\ \text{nnnnnnnn}\end{matrix}\right\}\right] \qquad \text{X}$$

$$\left[\text{DATE} = \left\{\begin{matrix}\underline{\text{A}}\\ \text{E}\end{matrix}\right\}\right]$$

Description *Optional.* Indicates which date format Comprehensive Retrieval is to use on its headings and on your report if you use $NDATE.

Default A

Options A American format (mm/dd/yy)

E European format (dd/mm/yy)

$$\left[,\text{EXIN} = \left\{\begin{matrix}\underline{\text{IPT}}\\ \text{nnn}\end{matrix}\right\}\right]$$

Description *Optional.* Indicates the VSE “SYSNO” number for the EXIN input file.

Default IPT

Format 3 alphanumeric characters

Consideration VSE only

$$\left[\text{,EXINBLK} = \left\{\begin{matrix}\underline{\textbf{80}} \\ \textbf{\textit{nnnnn}}\end{matrix}\right\}\right]$$

Description *Optional.* Indicates the default block size of any file named in an EXTRACT:OPTIONS:INPUT= statement.

Default 80

Format 1–5 numeric characters

Consideration You may override the default block size when you code your INPUT= statement for EXIN. If you do so, any other INPUT= statements will use the override default value.

$$\left[\text{,EXINREC} = \left\{\begin{matrix}\underline{\textbf{80}} \\ \textbf{\textit{nnnnn}}\end{matrix}\right\}\right]$$

Description *Optional.* Indicates the default record size of any file named in an EXTRACT:OPTIONS:INPUT= statement.

Default 80

Format 1–5 numeric characters

Consideration You may override the default record size when you code your INPUT= statement for EXIN. If you do so, any other INPUT= statements will use the override default value.

$$\left[\text{,EXOT} = \left\{\begin{matrix}\underline{\textbf{002}} \\ \textbf{nnn}\end{matrix}\right\}\right]$$

Description *Optional.* Indicates the VSE “SYSNO” of the data file.

Default 002

Format 3 numeric characters

Consideration VSE only

[,LINESZ = {132 / *nnn*}]

Description *Optional.* Indicates the default number of print positions per print line that you want on every report.

Default 132

Options 1–132

Consideration You may override this value for a particular report with a LINESIZE= statement following your REPORT=*xx* statement in PRINT.OPTIONS.

[,MAXERR = {0 / nnnnn}]

Description *Optional.* Indicates maximum number of system execution time errors that are normally permitted.

Default 0

Format 1–5 numeric characters

[,MINBUF = {1504 / *nnnnn*}]

Description *Optional.* Indicates the default block size for the EXOT data file and for an EXTRACT:OPTIONS:OUTPUT= statement.

Default 1504

Format 1–5 numeric characters

Consideration You may override the default block size with an OUTPUT= statement for EXOT. If you do, any other OUTPUT= statements will use the override default value.

$$\left[\text{,MINREC} = \left\{\begin{matrix}\underline{\textbf{250}} \\ \textbf{\textit{nnnnn}}\end{matrix}\right\}\right]$$

Description *Optional.* Indicates the default record size for the EXOT data file and for an EXTRACT:OPTIONS:OUTPUT= statement.

Default 250

Format 1–5 numeric characters

Consideration You may override the default record size with an OUTPUT= statement for EXOT. If you do, any other OUTPUT= statements will use the override default value.

$$\left[\text{,PAGESZ} = \left\{\begin{matrix}\underline{\textbf{58}} \\ \textbf{\textit{nnnnn}}\end{matrix}\right\}\right]$$

Description *Optional.* Indicates the default number of print lines per page that you want for every report.

Default 58

Format 1–5 numeric characters

Consideration You may override the value for a particular report with a PAGESIZE= statement following your REPORT=*xx* statement in PRINT.OPTIONS.

$$\left[\text{,PARM} = \left\{\begin{matrix}\underline{\textbf{024}} \\ \textbf{nnn}\end{matrix}\right\}\right]$$

Description *Optional.* Indicates the VSE “SYSNO” of the parm input file.

Default 024

Format 3 numeric characters

Consideration VSE only

[,PARMLNG = {80 / *nnn*}]

Description *Optional.* Indicates the default maximum record and block size of the one-record PARM file.

Default 80

Format 1–3 numeric characters

Options 1 to 256

[,PRIN = {001 / nnn}]

Description *Optional.* Indicates the VSE “SYSNO” number of the data file.

Default 001

Format 3 numeric characters

Consideration VSE only

[,RPTR = {IPT / nnn}]

Description *Optional.* Indicates the VSE “SYSNO” number for the RPTR input file.

Default IPT

Format 3 alphanumeric characters

Consideration VSE only

[,RPTRBLK = {80 / *nnnnn*}]

Description *Optional.* Indicates the default block size of the Directory Reports Selection file.

Default 80

Format 1–5 numeric characters

$$\left[\text{,RPTRREC} = \left\{\begin{matrix}\underline{\mathbf{80}} \\ \textit{nnnnn}\end{matrix}\right\}\right]$$

Description *Optional.* Indicates the default record size of the Directory Reports Selection file.

Default 80

Format 1–5 numeric characters

$$\left[\text{,SOCAREA} = \left\{\begin{matrix}\underline{\mathbf{25}} \\ \textit{nnnnnnnnnn}\end{matrix}\right\}\right]$$

Description *Optional.* Indicates the maximum amount of storage in 1K blocks (1,024K) Comprehensive Retrieval can obtain in either the Extract or Print Phase for building work tables. Storage is freed between phases.

Default 25

Format 1–10 numeric characters

Consideration You may override this value with a RUN.OPTIONS: SOCAREA statement.

$$\left[\text{, SOCLCB} = \left\{\begin{matrix}\underline{\mathbf{022}} \\ \mathbf{nnn}\end{matrix}\right\}\right]$$

Description *Optional.* Indicates the VSE SYSNO of the SOCLCB work file.

Default 22

Format 3 numeric characters

Consideration VSE only

$$\left[\text{, SOCLST} = \left\{\begin{matrix}\underline{\text{LST}} \\ \text{nnn}\end{matrix}\right\}\right]$$

Description *Optional.* Indicates the VSE SYSNO of the printer file.

Default LST

Format 3 alphanumeric characters

Consideration VSE only

$$\left[\text{, SOCSYS} = \left\{\begin{matrix}\underline{\text{023}} \\ \text{nnn}\end{matrix}\right\}\right]$$

Description *Optional.* Indicates the VSE SYSNO of the SOCSYS work file.

Default 23

Format 3 numeric characters

Consideration VSE only

$$\left[\text{,SORTCOR} = \left\{\begin{matrix}\underline{\text{0}} \\ \textit{nnnnnnnnnn}\end{matrix}\right\}\right]$$

Description *Optional.* Indicates the amount of storage in 1K blocks (1,024 bytes) that can be used during the Sort phase.

Default 0

Format 1–10 numeric characters

Considerations

- A default value of zero allows the sort phase to use all storage that is not reserved by the Comprehensive Retrieval driver and system tasks.
- Comprehensive Retrieval clears all its tables and releases the associated storage before giving control to the Sort.

SORTNAM=*sortname*

Description *Optional.* Indicates the Sort program to be used by Comprehensive Retrieval.

Format Name of the sort program

Consideration If you do not specify a Sort program name, Comprehensive Retrieval will use the program IERRCO00.

[, SORTWK = {4 / n}]

Description *Optional.* Indicates the number of Sort work files. *N* must be greater than 0 but less than 5.

Default 4

Format 1 numeric character

Consideration VSE only

[, WORKFLD = {20 / *nnnnnnnn*}]

Description *Optional.* Indicates the number of 8-byte $XXXX fields to be allocated for user-created system data fields.

Default 20

Format 1–10 numeric characters

Considerations

- Macro expansion will always add 20 to this default value to create the standard System Defined Labels. You may enter zero if none of your RDL programs will create any $ labels.
- If any one of your RDL programs requires more than 20 $ labels, increase this value.

General considerations

- When the EXOT data file goes to the Print Phase in the same run, its buffer and file characteristics will automatically be used for the Print Input Data file, even if changed with an OUTPUT=EXOT statement. You do not need a PRINT.OPTIONS:INPUT= statement.
- When the EXOT or other PRINT type file is input to Print-only in a later run, the Print Phase uses the default EXOT MINBUF and MINREC size in CSSWORK. You would need a PRINT.OPTIONS:INPUT statement only if different from the default.
- When PARMLNG, MINBUF and MINREC, EXINBLK and EXINREC, or RPTRBLK and RPTRREC are set to default zero in CSSWORK, Comprehensive Retrieval will obtain file characteristics from the JCL or FILEDEF.

Assembling the C$XCDFLT macro

Assemble the C$XCDFLT macro to regenerate the CSXSDMOD phase. The installation tape supplies the default object CSXSCDFT.

$$\mathbf{C\$XCDFLT}\left[\mathbf{DATEFMT}=\left\{\frac{\mathbf{A}}{\mathbf{E}}\right\}\right] \qquad \mathbf{X}$$

$$\left[\mathbf{DSECT}=\left\{\frac{\mathbf{NO}}{\mathbf{YES}}\right\}\right] \qquad \mathbf{X}$$

$$\left[\mathbf{,LINSBDY}=\left\{\frac{\mathbf{59}}{\boldsymbol{nnn}}\right\}\right] \qquad \mathbf{X}$$

$$\left[\mathbf{,LINSNCH}=\left\{\frac{\mathbf{8}}{\mathbf{6}}\right\}\right] \qquad \mathbf{X}$$

$$\left[\mathbf{,LOGDATE}=\left\{\frac{\mathbf{NO}}{\mathbf{YES}}\right\}\right] \qquad \mathbf{X}$$

$$\left[\mathbf{,LOGJOB}=\left\{\frac{\mathbf{NO}}{\mathbf{YES}}\right\}\right] \qquad \mathbf{X}$$

$$\left[\mathbf{,LOGTIME}=\left\{\frac{\mathbf{NO}}{\mathbf{YES}}\right\}\right] \qquad \mathbf{X}$$

$$\left[\mathbf{,PAGECHR}=\left\{\frac{\mathbf{6B}}{\mathbf{4B}}\right\}\right] \qquad \mathbf{X}$$

$$\left[\mathbf{,PRINTLU}=\left\{\frac{\mathbf{SYSLST}}{\boldsymbol{name}}\right\}\right] \qquad \mathbf{X}$$

$$\left[\mathbf{,PUNCHLU}=\left\{\frac{\mathbf{SYSPCH}}{\boldsymbol{name}}\right\}\right] \qquad \mathbf{X}$$

$$\left[\mathbf{,READLU}=\left\{\frac{\mathbf{SYSIPT}}{\boldsymbol{name}}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\text{PRINTDD}=\left\{\begin{matrix}\textit{name}\\ \underline{\text{SYSPRINT}}\end{matrix}\right\}\right]$$ X

$$\left[,\text{PUNCHDD}=\left\{\begin{matrix}\textit{name}\\ \underline{\text{SYSPUNCH}}\end{matrix}\right\}\right]$$ X

$$\left[,\text{READDD}=\left\{\begin{matrix}\underline{\text{SYSIN}}\\ \textit{name}\end{matrix}\right\}\right]$$ X

$$\left[\begin{matrix}\text{SRTAQT1}=(\text{C}|\text{'C'X'DD'},\ldots,\text{C}|\text{'C'}|\text{X'DD'})\\ .\\ .\\ .\\ \text{SRTAQT9}=(\text{C}|\text{'C'X'DD'},\ldots,\text{C}|\text{'C'}|\text{X'DD'})\end{matrix}\right]$$ X

$$\left[,\text{SRTCORE}=\left\{\begin{matrix}\underline{\text{MAX}}\\ \text{int}\textit{eger}\end{matrix}\right\}\right]$$ X

$$[,\text{SRTMDST}=\text{CONSOLE}]$$ X

$$\left[,\text{SRTMSEV}=\left\{\begin{matrix}\underline{\text{CRITICAL}}\\ \text{ALL}\\ \text{NONE}\end{matrix}\right\}\right]$$ X

$$\left[,\text{SRTNAME}=\left\{\begin{matrix}\underline{\text{SORT}}\\ \textit{name}\end{matrix}\right\}\right]$$ X

$$[,\text{SYSTEM}=\text{DOS}]$$ X

$$\left[,\text{TIMECHR}=\left\{\begin{matrix}\underline{\text{7A}}\\ \text{7B}\end{matrix}\right\}\right]$$ X

$$\left[,\text{TIMECHR}=\left\{\begin{matrix}\underline{\text{7A}}\\ \text{7B}\end{matrix}\right\}\right]$$ X

DATEFMT= {**A** / **E**}

Description *Optional.* Indicates whether to print the date in American or European format.

Default A

Options A American format (mm/dd/yy)

E European format (dd/mm/yy)

Consideration In American format, month prints first (NOV. 12, 1997). In European format, day prints first (12 NOV., 1997).

DSECT= {**NO** / **YES**}

Restriction VSE only

Description *Optional.* Indicates whether to generate a DSECT or CSECT for the defaults module.

Default NO

Consideration If you code DSECT=YES, the C$XCDFLT macro generates only the DSECT without the defaults module.

LINSBDY= {**59** / ***nnn***}

Description *Optional.* Indicates the maximum number of lines per body of output page.

Default 59

Options 1–999

LINSNCH= {**8** | **6**}

Description *Optional.* Indicates the lines per inch on output.

Default 8

Options 8 or 6

LOGDATE= {**NO** | **YES**}

Restriction VSE only

Description *Optional.* Indicates whether to print the date on each console message.

Default NO

LOGJOB= {**NO** | **YES**}

Restriction VSE only

Description *Optional.* Indicates whether to print the job name on each console message.

Default NO

LOGTIME= {**NO** | **YES**}

Restriction VSE only

Description *Optional.* Indicates whether to print the time on each console message.

Default NO

PAGECHR= {**6B** | **4B**}

Description *Optional.* Indicates the page number punctuation mark. This must be a valid hexadecimal character representation.

Default 6B

Options 6B The page numbers display as 12,345

4B The page numbers display as 12.345

PRINTLU= {**SYSLST** | ***name***}

Restriction VSE only

Description *Optional.* Indicates the Printer file name logical unit.

Default SYSLST

Format Use standard IBM naming conventions. Must be in the form SYS*xxx*, where *xxx* is any valid VSE assign.

PUNCHLU= {**SYSPCH** | ***name***}

Restriction VSE only

Description *Optional.* Indicates the Punch file name logical unit.

Default SYSPCH

Format Use standard IBM naming conventions. Must be in the form SYS*xxx*, where *xxx* is any valid VSE assign.

READLU= { **SYSIPT** | ***name*** }

Restriction VSE only

Description *Optional.* Indicates the Input file name logical unit.

Default SYSIPT

Format Use standard IBM naming conventions. Must be in the form SYS*xxx*, where *xxx* is any valid VSE assign.

PRINTDD= { ***name*** | **SYSPRINT** }

Restriction OS/390 only

Description *Optional.* Indicates the Printer file name logical unit.

Default SYSPRINT

Format Use standard IBM naming conventions.

PUNCHDD= { ***name*** | **SYSPUNCH** }

Restriction OS/390 only

Description *Optional.* Indicates the Punch file name logical unit.

Default SYSPUNCH

Format Use standard IBM naming conventions.

READDD= { **SYSIN** | ***name*** }

Restriction OS/390 only

Description *Optional.* Indicates the Input file name logical unit.

Default SYSIN

Format Use standard IBM naming conventions.

SRTAQT1=(C|'C'X'DD',...,C|'C'|X'DD')
.
.
.
SRTAQT9=(C|'C'X'DD',...,C|'C'|X'DD')

Restriction OS/390 and VSE only

Description *Optional.* Indicates Sort Translate table entries.

SRTCORE= { **MAX** | **integer** }

Description *Optional.* Indicates the amount of storage available for Sort.

Default MAX

Options MAX or a positive integer

SRTMDST=CONSOLE

Description *Optional.* Indicates the Sort Message destination.

Format Code exactly as shown.

SRTMSEV= { **CRITICAL** | **ALL** | **NONE** }

Description *Optional.* Indicates Sort Message severity.

Default CRITICAL

Options CRITICAL

ALL

NONESRTNAME= {SORT / *name*}

Description *Optional.* Indicates Sort program name.

Default SORT

Format 1–8 alphanumeric character Sort program name

SYSTEM=DOS

Restriction VSE only

Description *Conditional.* Required for VSE. Indicates the system you are running.

Format Code exactly as shown.

TIMECHR= {7A / 4B}

Description *Optional.* Indicates the time display punctuation mark. This must be a valid hexadecimal character representation.

Default 7A

Options

7A The time displays as 12:47:19

4B The time displays as 12.47.19

General considerations

- The following example shows the JCL needed to create and link edit a new default module:

```
// EXEC      PROC=assigns
// OPTION    CATAL,NODECK
   PHASE     CSXSDMOD,*
// EXEC      ASSEMBLY
      C$XCDFLT  DATEFMT=E,PAGECHR=4B,SYSTEM=DOS
      END
/*
// EXEC      LNKEDT
//
```

- Sample JCL for assembling and link editing the default module is supplied on the SUPRA Server installation tape in member TXJASMDF.

Assembling the CSTXOPRM macro

The Operator Control Parameter Table macro CSTXOPRM builds the CICS operator control parameter table CSTXOTBL. To assemble this macro, see CSTXOTBL, a member on the distributed source statement library for the CSTXOPRM assembly default used in creating the distributed CSTXOTBL module. Also see the JCL sample TXJAOPRM.

This macro has three formats:

- Format 1 is required to define the default CONNECT values.
- Format 2 is used to group files.
- Format 3 is used to define files.

Format 1 (define default CONNECT values):

CSTXOPRM TABLE	**X**
$\left[\text{,IFRTRY} = \left\{\begin{matrix}\underline{5} \\ nnn\end{matrix}\right\}\right]$	**X**
$\left[\text{,IMMEDRS} = \left\{\begin{matrix}\underline{N} \\ Y\end{matrix}\right\}\right]$	**X**
$\left[\text{,MAXPACK} = \left\{\begin{matrix}\underline{0} \\ nnnnn\end{matrix}\right\}\right]$	**X**
$\left[\text{,MEME} = \left\{\begin{matrix}\underline{5} \\ nn\end{matrix}\right\}\right]$	**X**
$\left[\text{,MESSQID} = \left\{\begin{matrix}\underline{MUSR} \\ xxxx\end{matrix}\right\}\right]$	**X**
$\left[\text{,OCPTRID} = \left\{\begin{matrix}\underline{OPER} \\ xxxx\end{matrix}\right\}\right]$	**X**

$$\left[,\mathbf{OCPTRTM}=\left\{\begin{matrix}\underline{\mathbf{TMSG}}\\ \mathit{xxxx}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{PUSHPOP}=\left\{\begin{matrix}\underline{\mathbf{N}}\\ \mathbf{Y}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[\mathbf{SYNC}=\left\{\begin{matrix}\underline{\mathbf{N}}\\ \mathbf{U}\\ \mathbf{C}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{TASKS}=\left\{\begin{matrix}\underline{\mathbf{3}}\\ \mathit{nnnnn}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{TDUE}=\left\{\begin{matrix}\underline{\mathbf{NO}}\\ \mathbf{mod}\,\mathit{name}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{TPOP}=\left\{\begin{matrix}\underline{\mathbf{NO}}\\ \mathbf{mod}\,\mathit{name}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{TPOS}=\left\{\begin{matrix}\underline{\mathbf{NO}}\\ \mathbf{mod}\,\mathit{name}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{TPRE}=\left\{\begin{matrix}\underline{\mathbf{NO}}\\ \mathit{modname}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{TPRP}=\left\{\begin{matrix}\underline{\mathbf{NO}}\\ \mathit{modname}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$$\left[,\mathbf{TSYN}=\left\{\begin{matrix}\underline{\mathbf{NO}}\\ \mathit{modname}\end{matrix}\right\}\right] \qquad \mathbf{X}$$

$\left[,\text{TUID} = \left\{\begin{matrix}\underline{\text{NO}} \\ \textit{modname}\end{matrix}\right\}\right]$	X
$\left[,\text{TFUL} = \left\{\begin{matrix}\underline{30} \\ \textit{nnnnn}\end{matrix}\right\}\right]$	X
$\left[,\text{CFUL} = \left\{\begin{matrix}\underline{30} \\ \textit{nnnnn}\end{matrix}\right\}\right]$	X
$\left[,\text{ICOR} = \left\{\begin{matrix}\underline{5} \\ \textit{nnnnn}\end{matrix}\right\}\right]$	X
$\left[,\text{THREADS} = \left\{\begin{matrix}\underline{2} \\ \textit{nnnnn}\end{matrix}\right\}\right]$	X
$\left[,\text{TOTC} = \left\{\begin{matrix}\underline{\text{N}} \\ \text{Y}\end{matrix}\right\}\right]$	X
$\left[,\text{TRACE1} = \left\{\begin{matrix}\underline{5} \\ \textit{nnn}\end{matrix}\right\}\right]$	X
$\left[,\text{TRACE2} = \left\{\begin{matrix}\underline{6} \\ \textit{nnn}\end{matrix}\right\}\right]$	X
$\left[,\text{TRACE3} = \left\{\begin{matrix}\underline{7} \\ \textit{nnn}\end{matrix}\right\}\right]$	X
$\left[,\text{USERTID} = \left\{\begin{matrix}\underline{\text{NO}} \\ \textit{xxxx}\end{matrix}\right\}\right]$	X

TABLE

Description *Required*. Identifies this as defining the default CONNECT values.

Consideration You can code only one TABLE entry for CSTXOTBL in a source module. The TABLE entry must be the first entry coded and must precede the GROUP and FILE entries.

IFRTRY= {5 | *nnn*}

Description *Optional.* Specifies the number of one-second delays to issue while waiting to free task context for an inflight PDM request.

Default 5

Options 1–2, 5, 149, 483, 647

Considerations

- IFRTRY should be set to the maximum return time for a single DML.
- You can increase the default to allow a long-running DML (waiting on HELD status) to complete.

IMMEDRS= {N | Y}

Description *Optional.* Handles the RESET/SINOF of a task when it detaches instead of scheduling an RSSO transaction.

Default N

Consideration If RSSO notification to other components is not needed, change to Y to avoid scheduling RSSO tasks.

MAXPACK= {**0** / ***nnnnn***}

Restriction VSE only

Description *Optional.* Documents the maximum packet size for a parameter.

Default 0

Options 1–37766

MEME= {**5** / ***nn***}

Description *Optional.* Specifies the maximum number of additional memory blocks that can be acquired if memory allocation is exceeded during the connect.

Default 5

Options 1–99

Consideration This should be large enough to allow for some expansion but small enough to prevent excessive memory allocation without warning. In VSE environments, consider MAXPACK times the number of concurrent tasks doing DML divided by the block size (65504).

MESSQID= {**MUSR** / ***xxxx***}

Description *Optional.* Defines the queue ID in your DCT table for the audit log.

Default MUSR

Format 4 alphanumeric characters

OCPTRID= {**OPER** | ***xxxx***}

Description *Optional.* Specifies the transaction ID for the operator control program.

Default OPER

Format 4 alphanumeric characters

Consideration Must be the same as the TRANSID parameter on the PCT entry for program CSTXOPER.

OCPTRTM= {**TMSG** | ***xxxx***}

Description *Optional.* Specifies the transaction ID for CSTXTMSG and is used to notify the operator that a task is being reset or signed off (RESET/SINOF).

Default TMSG

Format 4 alphanumeric characters

Consideration The trans definition must reference CSTXTMSG or a custom user message program.

PUSHPOP= {**N** | **Y**}

Description *Optional.* Handles the PUSH/POP option for CICS commands. It allows internally generated EXEC CICS requests to be bracketed with EXEC CICS PUSH HANDLE-CICS POP HANDLE condition requests.

Default N

Consideration You may need this if user exits set handle conditions.

SYNC= { **N** / **U** / **C** }

Description *Optional.* Handles the task synchronization option.

Default N

Options N No

U User defined. Activates CSTXOSY1 or CSTXUSY3 to allow the user to control sync actions.

C Prior compatibility. SUPRA commit generates a CICS sync point, and a CICS sync point rollback generates a SUPRA reset.

Considerations

- Should be set to the minimum required by your resources.
- Set to C only to keep CICS resources in recoverable sync with SUPRA PDM resources.

TASKS= { **3** / ***nnnnn*** }

Description *Optional.* Specifies the default number of CICS tasks that may be signed on to the PDM at any time.

Default 3

Format 1–32,767

Considerations

- This is the default value for the TASKS parameter on the CONNECT operator control command.
- This affects the number of bytes of virtual storage used. The more tasks specified, the more memory required.

TDUE= { **NO** | ***modname*** }

Description *Optional.* Specifies the user exit name. CSTXUSER supplies the default logic for this exit point.

Default NO

Considerations

- The modname cannot be NO.
- The modname is CSTXUSER unless you want to separate the exits into different modules, or you have existing modules you want to use.

TPOP= { **NO** | ***modname*** }

Description *Optional.* Specifies the user exit name. CSTXUSER supplies the default logic for this exit point.

Default NO

Considerations

- The modname cannot be NO.
- The modname is CSTXUSER unless you want to separate the exits into different modules, or you have existing modules you want to use.

TPOS= { **NO** | ***modname*** }

Description *Optional.* Specifies the user exit name. CSTXUSER supplies the default logic for this exit point.

Default NO

Considerations

- The modname cannot be NO.
- The modname is CSTXUSER unless you want to separate the exits into different modules, or you have existing modules you want to use.

TPRE= { **NO** | ***modname*** }

Description *Optional.* Specifies the user exit name. CSTXUSER supplies the default logic for this exit point.

Default NO

Considerations

- The modname cannot be NO.
- The modname is CSTXUSER unless you want to separate the exits into different modules, or you have existing modules you want to use.

TPRP= { **NO** | ***modname*** }

Description *Optional.* Specifies the user exit name. CSTXUSER supplies the default logic for this exit point.

Default NO

Considerations

- The modname cannot be NO.
- The modname is CSTXUSER unless you want to separate the exits into different modules, or you have existing modules you want to use.

TSYN= { **NO** | ***modname*** }

Description *Optional.* Specifies the user exit name. CSTXUSER supplies the default logic for this exit point.

Default NO

Considerations

- The modname cannot be NO.
- The modname is CSTXUSER unless you want to separate the exits into different modules, or you have existing modules you want to use.

TUID= { **NO** | ***modname*** }

Description *Optional.* Specifies the user exit name. CSTXUSER supplies the default logic for this exit point.

Default NO

Considerations

- The modname cannot be NO.
- The modname is CSTXUSER unless you want to separate the exits into different modules, or you have existing modules you want to use.

TFUL= { **30** | ***nnnnn*** }

Description *Optional.* Defines the number of retries (with a one-second delay) after a TFUL status.

Default 30

Format 1–32767

Consideration Tasks and threads can be set to average values and retry values can be set high to handle spikes in demand.

CFUL= { **30** | ***nnnnn*** }

Description *Optional.* Defines the number of retries (with a one-second delay) after a CFUL status.

Default 30

Format 1–32767

Consideration Tasks and threads can be set to average values and retry values can be set high to handle spikes in demand.

ICOR= {**5** / ***nnnnn***}

Description *Optional.* Defines the number of retries (with a one-second delay) after an ICOR status.

Default 5

Format 1–32767

Consideration To correct ICOR problems, a reanalysis of resource requirements is usually required.

THREADS= {**2** / ***nnnnn***}

Description *Optional.* Specifies the default number of threads between the CICS connector and the PDM. This is the default for the THREADS parameter on the CONNECT operator control command.

Default 2

Options 1–32767

TOTC= {**N** / ***Y***}

Description *Optional.* Provides TOTAL compatibility for SUPRA DML.

Default N

Consideration TPRP=CSTXUSER and TPOP=CSTXUSER must be coded.

TRACE1= $\left\{\begin{matrix}\underline{\mathbf{5}} \\ \textbf{\textit{nnn}}\end{matrix}\right\}$

Description *Optional.* Specifies the trace ID for PDML commands on entry to the interface.

Default 5

Format 1–199

Considerations

- If you do not code TRACE1, tracing is turned off.
- If you supply a value for TRACE1, tracing is turned on. TRACE2 and TRACE3 entries will be generated automatically using defaults unless you supply values for them.
- Do not set TRACE1 to zero.
- SUPRA Server is delivered with tracing turned off in the default module CSTXOTBL.

TRACE2= $\left\{\begin{matrix}\underline{\mathbf{6}} \\ \textbf{\textit{nnn}}\end{matrix}\right\}$

Description *Optional.* Specifies the trace ID for a PDML command status upon exit from the interface.

Default 6

Format 1–199

TRACE3= $\left\{\begin{matrix}\underline{\mathbf{7}} \\ \textbf{\textit{nnn}}\end{matrix}\right\}$

Description *Optional.* Specifies the trace ID for a new pseudo conversational task associated with a PDM task signed on by a prior task (CICS Task 1 signs on, processes, and passes a next transid to CICS Task 2. When CICS Task 2 starts, the TRACE3 ID displays).

Default 7

Format 1–199

USERTID= { **NO** | ***xxxx*** }

Description Defines the transaction ID for the user initialization program.

Default NO

Format 4 alphanumeric characters

Considerations

- A transaction ID cannot be NO.
- You can use this to cause an EXEC CICS start transaction ID (*xxxx*) to start a user program that requires a connected PDM.
- User initialization is started as soon as OPER CONNECT gets a good status from PDML CNNCT.
- You can use this user initialization program to do any initialization that must wait until the connect is completed.

Format 2 (group files):

CSTXOPRM	**GROUP,GROUPNM=*gggggggg***

GROUP

Description *Required*. Identifies this as defining the name of the group of files.

Format Enter as shown.

GROUPNM=*gggggggg*

Description *Required*. Specifies the name of the group. This value is used in the GROUP parameter of the Operator Control OPEN and CLOSE commands to specify a group of files to be opened or closed.

Format 1–8 alphanumeric characters

Considerations

- Each group in the CSTXOTBL must have a unique name.
- A GROUP statement must precede the FILE statements in the group.

Format 3 (define files):

CSTXOPRM FILE	**X**
[,CLOSMODE = {COMP / PART}]	**X**
,FILENM=*ffff*	**X**
[,OPNMODE = {SUPD / IUPD / READ}]	**X**
[LABELN=*ffff*]	**X**

FILE

Description *Required*. Identifies this as defining the file to be opened or closed.

CLSMODE= {COMP / PART}

Description *Optional.* Specifies the mode to be used when the file is closed. The file is closed as part of the file group.

Default COMP

Options

COMP	Use the operating system close. The file is unlocked if necessary. The file mode is reset to null. NO task may access the file except to open it.
PART	Do not use the operating system close. The file is unlocked by CLOSX (nontask-logging) or COMMIT/SIGN-OFF (task-logging), if necessary. The file mode is reset to IUPD. Any task may read the file, but no task may update the file.

FILENM=*ffff*

Description *Required*. Specifies the name of a PDM file. The same file may appear in more than one group.

Format 4 alphanumeric characters

Consideration This value must be the name of a file defined on the Directory.

OPNMODE= { **SUPD** | **IUPD** | **READ** }

Description *Optional.* Specifies the mode to be used when the file is opened. The file is opened as part of the file group.

Default SUPD

Options

SUPD	Marks the file for shared updating. The file is locked and may be updated by all tasks using the same copy of the PDM.
IUPD	Marks the file for read only. The file need not be locked. The file may be read by all tasks but not updated by any task.
READ	Marks the file for read only. The file cannot be locked. The file may be read by any task but not updated.

LABELN=*ffff*

Description *Optional.* Specifies the name of a label used in the CSTXOFLE CSECT expansion of the CSTXOPRM file format.

Default Value of FILENM parameter

Format 4 alphanumeric characters

Consideration If the FILENM parameter does not define a valid Assembler label, use this parameter to provide a valid pseudonym.

Assembling the CSIMCS macro

If you have Multiple Console Support (MCS), you may wish to use routing and/or descriptor codes for console messages (WTO and WTOR). Set MCS routing and descriptor codes by compiling a module using the CSIMCS macro. This creates a module with an entry point of MCSCODES. Include this module in the link edit for the desired component. These codes would then be used for all WTO and WTOR messages issued by the component.

The following components can use MCSCODES: PDM, PDM Interfaces, Directory Maintenance, DDL Convert, and the Descriptor Generators:

NOTE

The default route codes for SUPRA Server if you do not supply the MCSCODES module are:

- WTO: No default. Uses operating system defaults assigned at SYSGEN time.
- WTOR: 1—Master Console Action.

For more information on console messages (WTO and WTOR), refer to the IBM manual *OS/VS OS/390 Supervisor Services and Macro Instructions,* GC28-0628.

[*label*] CSIMCS	**[DESC = (n,n,…n)]**	**X**
	[,ROUTCDE = (n,n,n…n)]	**X**

label

Restriction OS/390 only

Description *Optional.* The label is not referred to in any way.

DESC = (n,n,n...n)

Restriction OS/390 only

Description *Optional.* Identifies the message descriptor code to assign to the message.

Default 0

Options

0	No Descriptor Code
1	System Failure
2	Immediate Action Required
3	Eventual Action Required
4	System Status
5	Immediate Command Response
6	Job Status
7	Application Program/Processor
8	Out-Of-Line Message
9	Operator Request
10	Dynamic Status Displays

Considerations

- Descriptor codes 1–6 are mutually exclusive; codes 7–10 can be assigned in combination with other codes.
- If you specify multiple codes, enclose them in parentheses and separate them by commas; otherwise, the parentheses are optional.

ROUTCDE = (*n*,*n*,*n*...*n*)

Restriction OS/390 only

Description *Optional.* Assigns the routing code to the message.

Default 0

Options

0	No route code	7	Unit Record Pool
1	Master Console Action	8	Teleprocessing Control
2	Master Console Information	9	System Security
3	Tape Pool	10	System Error/Maintenance
4	Direct Access Pool	11	Programmer Information
5	Tape Library	12	Emulators
6	Disk Library		

Consideration If you specify multiple codes, enclose them in parentheses and separate them by commas; otherwise, the parentheses are optional.

Using the SUPRA Server bootstrap and Directory schemas

This section describes the supplied SUPRA Server bootstrap schema and Environment Descriptions, the bootstrap validation module (VALMOD), and the Directory schemas and Environment Descriptions.

Using the SUPRA Server bootstrap schema

The SUPRA Server bootstrap schema consists of the bootstrap schema, bootstrap Environment Descriptions and the bootstrap validation module. Sample information on each is provided below.

- **Bootstrap Schema.** The bootstrap schema is named CSTASCHM. CSTASCHM describes the five Directory files and the Task Log file. The product library on the installation tape includes a tailored member containing statements for modifying CSTASCHM. Do not modify CSTASCHM unless you need to change the physical characteristics of these files; for example, if you need to increase the size of any of these files.

- **Bootstrap Environment Descriptions.** The following alphabetical list describes each bootstrap Environment Description provided and names any CSIPARM file samples that use it. The name of each Environment Description is also the name of the product library member on the SUPRA Server maclib (MACLIB) containing the statements for creating it. All these bootstrap Environment Descriptions are related to the schema CSTASCHM.

  - CSTANONE—Useful for executing the SUPRA DBA Utilities in a bootstrap-only (non-Directory driven) environment with no task logging.

  - CSTAOLDM—Useful for running the SUPRA PDM in a bootstrap-only teleprocessing environment with task logging. Causes the Directory files to be opened for update.

  - CSTAREAD—Useful for running the SUPRA PDM with no task logging. Causes the Directory files to be opened for read only.

  - CSTASUPD—Useful for running the SUPRA PDM in a bootstrap-only teleprocessing environment with no task logging. Causes the Directory files to be opened for update.

  - CSTATLOG—Useful for running the SUPRA PDM in a directory-driven environment with task logging.

- **Bootstrap Validation Module.** The MACLIB member containing input statements for the sample VALMOD is named CSTAVMOD. It specifies the sample schema and the sample Environment Descriptions associated with the schema, as follows:

```
SCHEMA=CSTASCHM
ENVDESC=CSTANONE
ENVDESC=CSTAOLDM
ENVDESC=CSTAREAD
ENVDESC=CSTASUPD
ENVDESC=CSTATLOG
```

  Whenever you add, delete or revise a bootstrap schema or Environment Description, you must recreate and relink your VALMOD. See “Defining the operational environment” on page 115 for information on modifying your bootstrap environment.

Using SUPRA Server Directory schemas

Cincom provides two Directory schemas and their associated Environment Descriptions. For information on creating, modifying and using Directory schemas and Environment Descriptions, refer to the *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260, or the *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261.

- **Directory Schema CSISCH20.** Schema CSISCH20 describes the Directory files and is related to the following Environment Descriptions:
  - CSIENVRN—Useful for running Directory or RDM reports. Relates only the Directory files. Causes the Directory files to be opened for read only.
  - CSIENVRG—Useful for running the SUPRA PDM with task logging against the Directory. Relates the Directory files. Causes the Directory files to be opened for update. This Environment Description is the most commonly used with CSISCH20.
- **Directory Schema BURRYSCH.** Schema BURRYSCH describes the Burrys database files and is related to the following Environment Descriptions:
  - BURRYENV—Useful for running the SUPRA PDM with task logging against the Burrys database. This is the most common environment when using the Burrys schema with SUPRA.
  - BURRYENN—Useful for running the DBA utilities against the Burrys database without task logging. The files are initially unopened.
  - BURRYNLG—Useful for running the SUPRA PDM with no task logging against the Burrys database. Causes the Directory files to be opened for update. Useful for migrating the SPECTRA personal file data set LV00.
  - BURRYERD—Useful for executing read-only functions of the SUPRA PDM with no task logging against the Burrys database. Prevents the files from being opened for update.

Using SUPRA Server linkdecks

SUPRA Server linkdecks are located on the SUPRA Server macro library (MACLIB). An index member, $INDLSU2 provides a short description of each linkdeck, as well as link edit attributes and the load module name.

Using SUPRA Server sample source members

Sample source members for exit programs, options macro assemblies, and directory transactions are located in the SUPRA Server macro library (MACLIB). An index member,

$INDSSU2 provides a short description of each member.

B

Using the trace and debug facilities

Trace and debug facilities

Debug and trace facilities are available for the Physical Data Manager (PDM), DBA utilities, Relational Data Manager (RDM) applications, SPECTRA, and MANTIS applications.

At times you may need debug output to assist your Technical Service Center in resolving a usage or production problem. To provide PDM debugging information, use the CSIPARM file to enable debug and trace facilities. Code the DEBUG parameter on the CSIPARM file to produce console messages for internal errors. Due to the volume of messages displayed, code YES for this parameter only when using the PDM in test mode. You may also specify the TRACESIZE parameter defining the size of the PDM Internal Trace table for PDM debugging. See “Initializing the PDM” on page 63 for information on CSIPARM parameter values.

To receive additional information for debugging purposes, use normal system dumps and snap dumps. Snap dumps are selective dynamic dumps produced at various points in a PDM run. The PDM issues snap dump requests in various places in its abend processing. Under OS/390, each snap has a snap title and snapped data. The snapped data is output to a PDMSNAP data set. You must include a PDMSNAP DD card in the JCL for the PDM job step.

VSE does not support snap titles; therefore, the title is snapped before the data. The PDUMP facility does not determine the output destination; VSE determines the destination.

For information on providing Utility debugging information, refer to the *SUPRA Server PDM DBA Utilities User's Guide (OS/390 & VSE)*, P26-6260.

For information on providing RDM and DBAID debugging information, refer to the *SUPRA Server PDM RDM Administration Guide (OS/390 & VSE)*, P26-8220.

For information on providing SPECTRA debugging information, refer to SPECTRA documentation.

For information on providing MANTIS debugging information, refer to MANTIS documentation.

Index

A

abend, avoiding 162
ACCESS-METHOD statement 173
ACCESS-MODE statement 195
active schema maintenance
  disabling 42
  enabling 39, 40
AFTER statement 196
allocating files 153
alternate ID, accessing menu options with 334
APF-authorized library
  defining 55
  name 61
ATTACH parameter
  for OS/390 69
  for VSE/AF 2.1 71
  in CSIPARM file 37
attached central operating mode 37
attached operating mode 37
Attacher program 37
audit trails, in Batch Directory Maintenance 38

B

batch task management 306
BBPSLOG option 431
BDTDL option 432
BEFORE statement 197
BEGDAT option 433
BLDMTFILE option 434
BLDSTFILE option 435
BLOCKSIZE statement 174
bootstrap environment description
  cold starting the PDM 64
  modifying 189
  using 23
bootstrap environment modules, description of 66
bootstrap schema
  modifying 163
  modules 66
  samples 487
  using the distributed sample 487
bootstrap validation module
  description of 66
  distributed sample 488
buffer cache facility
  check exit 272
  coding input file 95
  command initialization exit 265
  improving database performance 50
  logical writing exit 276
  new volume exit 278
  open exit 268
  write exit 271
buffer pools
  adding 147
  defining 50
BUFFER statement 198
BURRYSCH, Directory schema 489

C

C$MDLNK macro, assembling 446
C$SAWORK macro, assembling 449
C$STHOPTN macro 430
  BBPSLOG option 431
  BDTDL option 432
  BEGDAT option 433
  BLDMTFILE option 434
  BLDSTFILE option 435
  ENDDAT option 436
  PSWRD option 437
  RDAH option 438
  RECST option 439
  USTATX option 440
  WARDAY option 441
C$XCDFLT macro, assembling 460
central operating mode 36
check exit, coding 272
close exit, coding 273
COBOL application programs 33
cold starting, PDM system requirements 64
command initialization exit, coding 264

conceptual schema
  adding 106
  creating 104
  data 26
  hierarchy of 105
CONNECTED-INTERFACES
    statement 199
CONNECTED-THREADS
    statement 200
consistency checking, schema
    151
console messages (OS/390),
    routing 427
CONTROL, passing data to 336
CONTROL-INTERVAL-SIZE
    statement 175
create environment description
  ACCESS-MODE statement 195
  AFTER statement 196
  BEFORE statement 197
  BUFFER statement 198
  CONNECTED-INTERFACES
    statement 199
  CONNECTED-THREADS
    statement 200
  considerations for 193
cross-address-space
    communications 36
CSIPARM file
  creating 93
  example input 91
  OS/390 parameters 69
  used in initialization 68
  VSE parameters 71
CSISCH20, Directory schema
    489
CSMT interface 35
CSTEDBMI program 101
CSTN999*x* link decks 102
CSTXOPRM macro, assembling
    468

D

data
  maintained by Directory 25
  system 27
database
  active schema maintenance 39
  adding files 118
  controlling 43
  defining files 23, 110
  Directory Maintenance 38
  external schema 114
  maintaining files 43
  memory allocation 49
  monitoring 43
  performance, improving 48
  Secondary Keys 113
  statistics 48
  SUPRA, defining 103
  user environment files,
    descriptions 23
  user schemas files,
    descriptions 23
  utilities, using 48
DATBASXT exit
  coding 255
  function 251
DBA User Reference Report,
    using 368
DBM parameter
  for OS/390 69
  for VSE/AF 2.1 71
DDL convert utility 44
DDNAME statement 176
debug facilities 491
DEBUG parameter
  coding 76
  for OS/390 69
  for VSE/AF 2.1 71
derived views
  creating 114
  defining
    using DBAID 33
    using Directory Maintenance
      33
DEVADDR statement 176
DEVICE statement 177
Directory
  categories, hierarchy of 28
  defining to PDM 160
  DISPLAY command, TIS/XA
    subsystem 310
  DTISS command, TIS/XA
    subsystem 315
  entities 28
  file space estimates 324
  file types 24
  files, expanding 324
  maintaining data 25
  maintenance 38, 44
  metadata, example SPECTRA
    report 423

Directory (*cont.*)
  operations 21
  overview 23
  parameter
    coding 77
    for VSE/AF 2.1 71
    function of 65
  RDM views 419
  reports
    conceptual schema structure 370
    creating using RDM views 419
    DBA User Reference 368
    External Fields Where Used 375
    Full Definition 376
    Full Text Description 399
    generating 349
    Name 404
    Name Cross Reference 402
    parameter, coding 77
    Physical Structure 406
    Schema Structure, using 409
    Security, using 414
    standard header information 367
    using 367
  SET command, TIS/XA subsystem 317
  SPECTRA process, example 423
  STOP command, TIS/XA subsystem 318
  trigger records. *See* trigger records
  updating 38
  using RDM views to create 419
Directory parameter (OS/390) 69
DML parameters (OS/390) 266
DRDM 35
DTFNAME statement 178

E

END parameter, coding 90
ENDDAT option 436
END-FILE statement 178
END-SCHEMA statement 178
entities 28
entity maintenance, performing 28
environment descriptions
  adding 140
  relating to files 144
exits
  PDM interface, coding
    DATBASXT 251, 255
    general 251
    post-command (CSTEXT02) 251, 258
    precommand (CSTEXT01) 251, 253
  PDM, coding
    check 272
    close 273
    command initialization 264
    logical write 274
    open 268
    read 269
    write 270
  TIS/XA security
    FAILSIGN 338, 348
    MENU 338, 340
    SIGNON 338
    VALALT 338, 346
    VALSELECT 338, 344
    VALUSER 342
Exits
  TIS/XA security
    VALUSER 338
expanding files
  Directory 324
  PDM 328
extended storage support 49
external field 26
external schema
  adding 114
  creating 114
  data 26

F

failed sign-on exit 348
file compatibility options 43
file space 46, 324
FILE statement, for schema 179
files
  allocating 153
  database, defining 110
  Directory, expanding 324
  formatting 154
  opening 23, 154
  PDM, expanding 328

files (*cont.*)
  primary 111
  related 112
  system log. *See* system log file
  task log. *See* Task Log file
  types in directory 24
  types in PDM 31
fixes, optional 429
Full Definition reports
  Access Sets 377
  Attributes 378
  Component Description 379
  Conceptual Schema 380
  Domains 381
  Edit Masks 382
  Environment Description/Buffer Pool 384
  Environment Descriptions 383
  External Fields 385
  Files 386
  Foreign Keys 387
  Internal Records 388
  Key Codes 389
  Logical Views 390
  Physical Fields 391
  Procedures 392
  Relations 393
  Reserved Word 394
  Schemas 395
  Secondary Keys 396
  Tables 397
  Users 398
Full Text Description report 399

H

held records, specifying 49
hierarchy of Directory categories 26, 28

I

I/O buffers 147
ID, user
  alternate 334
  changing 335
index files
  adding 132
  maintaining 46
initialization parameters
  coding 73
  defining 64
  function of 68
  selecting for PDM 68
initialization parameters. *See* also CSIPARM file parameters
initializing
  PDM 63
  TIS/XA subsystem (OS/390) 53
input specifications
  for Directory files 164
  for Statistics file 170
  for System Log file 168
  for Task Log file 166
  modify schema program 164
installation of TIS/XA subsystem (OS/390) 53
interactive services
  function of 279
  menus in 280
  using 44, 280
Inter-Directory copy utility 44
interface exits, PDM. *See* exits
INTERFACE parameter
  coding 78
  for OS/390 69
  for VSE/AF 2.1 71
interfaces
  defining 100
  link decks 102
internal schema
  adding 109
  creating 107
  data 25
  defining 116
IPL 53, 54

K

Key, Secondary. *See* Secondary Key

L

link decks
  CSTN999*x* 102
  interface 102
  location 490
load modules, Directory 38
Log Groups, adding 149
logging, SUPRA 44

logical write exit, coding 274
LOGICAL-BLOCKS-PER-TRACK statement 180

M

MACHID parameter
  for OS/390 69
  for VSE/AF 2.1 71
macros, assembling
  C$MDMLNK 446
  C$SAWORK 449
  C$THOPTN 430
  C$XCDFLT 460
  CSTXOPRM 468
Maintenance, Active Schema. *See* Active Schema Maintenance
MANTIS
  executing from TIS/XA software selection facility 336
  passing data to 336
  RDM interface 33
  using with TIS/XA software selection facility 329
MASTER
  SIGN_ON program, coding for TIS/XA software selection facility 330
  TERMINATE program, coding for TIS/XA software selection facility 330
MAXARG parameter
  coding 79
  for OS/390 69
  for VSE/AF 2.1 71
MAXIO parameter
  coding 80
  for OS/390 69
  for VSE/AF 2.1 71
MAXPACKET parameter
  coding 81
  for OS/390 69
  for VSE/AF 2.1 71
MAXUSER parameter, increasing 55, 58
memory
  allocating for PDM 49
  requirements, establishing in Directory 38
MEMORY statement 209
menu options, batch mode 337
messages, console 427
modify schema program
  description of 160
  example 186
modify schema statements
  ACCESS-METHOD 173
  BLOCKSIZE 174
  CONTROL-INTERVAL-SIZE 175
  DDNAME 176
  description 172
  DEVADDR 176
  DEVICE 177
  DTFNAME 178
  END-FILE: 178
  END-SCHEMA 178
  FILE 179
  LOGICAL-BLOCKS-PER-TRACK 180
  OLDFILE 181
  SCHEMA 182
  TOTAL-LOGICAL-RECORDS 183
  UNIT 184
monitoring the database 43
multitask PDM, description 34

N

Name Cross Reference report, Directory 402
Name report, Directory 404
new volume exit, coding 277

O

OLDFILE statement 181
open exit, coding 268
operating modes, PDM 36
operational environment
  Buffer Pools 147
  environment description 140
  index files 132
  Log Groups 149
  primary files 117
  related files 117
  System file 138
  System Log 138
  Task Log 138
Options module 429

P

PAGESIZE parameter
coding 82
for OS/390 69
for VSE/AF 2.1 71
PATH parameter
coding 36, 82
for OS/390 69
for VSE/AF 2.1 71
PDM. *See* Physical Data Manager (PDM)
Physical Data Manager (PDM)
accessing through Directory maintenance 30
check exit 272
cold starting 64
command initialization exit 264
DATBASXT exit 255
defining tasks, threads, and interfaces 99
Directory, defining 160
exit input parameters 262
exits 249
file definitions 30
files
expanding 328
types 31
files, buffering above the 16-megabyte line. *See* buffer cache facility
initializing 63
interactive services 280
interface exits 251
logical write exit 274
multitask 31
new volume exit 277
open exit 268
operating boot-driven 65
operating directory-driven 65
operating modes 36
overview of 30
post-command exit 258
precommand exit 253
primary file (LV00), expanding 328
read exit 269
related to RDM 33
related to SPECTRA 33
services
displaying information 290
screens in 289
selecting 289
single task 34
SQL tasks 35
tasks 35, 99
termination utility 279
threads 99
versions of 34
warm-starting 67
write exit 270
physical file services
displaying file statistics 288
displaying physical file information 287
screens in 284
selecting 284
Physical Structure report, Directory 406
PL/1 application programs 33
post-command exit
coding 258
function of 251
precommand exit
coding 253
function of 253
PRIMAP parameter
coding 83, 84, 85
for OS/390 69
for VSE/AF 2.1 71
primary files
adding 117
and related files, relationship 32
considerations for 111
defining 110
PSWRD option 437

Q

quick exits, TIS/XA software selection facility 335

R

RDAH option 438
RDM. *See* Relational Data Manager (RDM)
read exit, coding 269

REALM parameter
  coding 86
  for OS/390 69
  for VSE/AF 2.1 71
  function of 66
REALMAP parameter
  for VSE/AF 2.1 71
RECST option 439
related files
  adding 117
  and primary files, relationship 32
  defining 110
Relational Data Manager (RDM)
  relationship to PDM 33
  views, using to create Directory reports 419
relational database
  model, designing 103
  structure 23
relational entitles, defining on Directory 103
RELMAP parameter
  coding 87
  for OS/390 69
reports
  Directory. *See* Directory, reports
  Full Definition. *See* Full Definition reports
RESTART parameter
  coding 88
  for OS/390 69
  for VSE/AF 2.1 71
routing console messages 428
RSTAT command, issuing 48

S

samples, location of 490
SCHEMA statement
  bootstrap schema 163
  modify schema 182
schema structure report, Directory 409
schema. *See* conceptual schema, external schema, and internal schema
Secondary Key
  defining 110
  functions of 110
  maintenance of 113
  populating 158
security
  exits, in TIS/XA software selection facility 338
  reports, Directory 414
SHOWX command 48
single-task operating mode 36
single-task PDM 34
source members, location of samples 490
SPECTRA
  application program 33
  process, example of Hardcopy report 423
START operator command, initializing subsystem 57
starting the PDM
  cold 64
  warm 67
statistics file
  adding 138
  running 48
STST interface 34
subsystem (OS/390), TIS/XA, installing 53
system data 27
system log file
  adding 138
  cold starting the PDM 64
system logging 45

T

Task Log File (TLF)
  adding 138
  cold starting the PDM 64
task logging 45
task management
  displaying interfaces 293
  displaying tasks 295
  purging tasks 297
  selecting 293
tasks, defining 99
terminating, multitask PDM 320
threads, defining 99
TIS/XA
  installing subsystem (OS/390) 53
  multitask interface library 102

TIS/XA (*cont.*)
software selection facility
accessing menu options 333
changing current ID and password 335
executing menu options in batch mode 337
exits, security 338
functions of 329
menu, bypassing 281
passing data to MANTIS or CONTROL: 336
security exits 338
selecting menu options with alternate ID and password 334
using 330
subsystem (OS/390) 36
command 53, 57
initializing at IPL 54
initializing by a start operator command 57
subsystem modules, moving 56, 58
subsystem operator commands (OS/390)
coding considerations 310
DISPLAY 309
DTISS 315
functions of 279
SET 317
STOP 318
TISSID parameter
coding 89
for OS/390 69
for VSE/AF 2.1 71
specifying in the CSIPARM file 56, 61
TLF. *See* Task Log File (TLF)
TOTAL-LOGICAL-RECORDS statement 183
TP monitor, warm-connecting PDM with 67
trace facility 491
TRACESIZE parameter
coding 90
for OS/390 69
for VSE/AF 2.1 71
trigger records
category selection 354
conceptual schema name selection 358
condense option 365
file selection 363
for Directory reports 349, 352, 366
maintenance date selection 360
schema name selection 357
security group selection 364
sort option 361
text selection 362
use of by report selection type 353
user name selection 359
user sign-on selection 366

U

UNIT statement 184
user access, authorizing 103
user data 27
user extension services
adding a service option 303
deleting a service option 303
selecting 302
user ID
alternate 334
changing 335
user schema, initialization parameters 68
USTATX option 440
utilities
using database 43
using Directory Maintenance 44

W

WARDAY option 441
warm-starting the PDM 67
write exit, coding 270

X

XAMEM parameter
coding 90
for OS/390 69