Cincom

SUPRA SERVER PDM

Migration Guide
(OS/390 & VSE)

P26-0550-64

SUPRA® Server PDM Migration Guide (OS/390 & VSE)

Publication Number P26-0550-64

© 1992–1994, 1997, 1998, 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This document contains unpublished, confidential, and proprietary information of Cincom.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

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 Migration Guide (OS/390 & VSE)*, P26-0550-64, 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

Contents

About this book

Using this document

This manual is intended for all database personnel involved in the migration process from a previous release of SUPRA Server to SUPRA Release 2.7. This manual provides step-by-step instructions for migrating to SUPRA Release 2.7 from previous releases of SUPRA or from Series 80 TOTAL.

No migration is needed from versions of SUPRA Server Release 2.4 and above.

Document organization

The information in this manual is organized as follows:

Chapter 1—Changes from prior releases
Describes new features in the current release and enhancements from prior releases.

Chapter 2—Migrating to SUPRA Server 2.7 from SUPRA 1.*x*
Provides information for SUPRA 1.*x* users migrating to Release 2.7 of SUPRA.

Chapter 3—Migrating to SUPRA Server 2.7 from Series 80 TOTAL
Provides information for TOTAL users migrating to Release 2.7 of SUPRA.

Appendix A—Using the file convert utility
Provides additional information for using the File Convert utility when upgrading from Series 80 TOTAL to SUPRA.

Appendix B—Relating TOTAL Utility statements to SUPRA UCL
Acquaints you with the SUPRA DBA Utilities Component Language (UCL).

Index

Revisions to this manual

The following changes have been made for this release:

- A chapter listing new features in this release and enhancements from prior releases has been added. This information was previously in the *SUPRA Server Planning Guide*, P26-0422, which has been eliminated. See "Changes from prior releases" on page 15.
- The NORMAL product is no longer distributed. If you use NORMAL, retain your files and previous documentation. References to NORMAL in this document have been deleted.
- Added a note that a partial key with MASK=YES will return an IPAR, under "Updating application programs" on page 45.
- Added new information about the Read Ahead Buffering Facility under "PDM configuration and administration" on page 62.

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 (ȸ)	Indicates a space (blank). The example indicates that four spaces appear between the keywords.	`BEGNȸȸȸȸSERIAL`
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** Bootstrap modules are on the SUPRA LINKLIB library. **VSE** Bootstrap modules are on the base sublibrary.

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 Directory Maintenance, DBA utilities, DBAID, 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 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

Educational material is available from your regional Cincom education department.

1

Changes from prior releases

New features in Release 2.7

Release 2.7 of SUPRA Server PDM support includes the following new features:

- Improved read-only batch performance.
- Additional Options module for setting defaults.

The SUPRA Server documentation series has been revised and reorganized as follows:

- The *SUPRA Server Planning Guide*, P26-0422, has been dropped from the series. Information pertaining to new features and migration has been moved to this manual. Resource requirements have been moved to the *SUPRA Server PDM OS/390 Installation Guide*, P26-0149, and the *SUPRA Server PDM VSE/ESA Installation Guide*, P26-0132.
- The *SUPRA Server Digest*, P26-9065, previously contained information pertaining to SUPRA Server PDM on all platforms (OS/390, VSE, UNIX, and VMS), and also to SUPRA Server with SQL Support. That manual has been divided into separate manuals for each product and operating system. To view the Digest for this product, refer to *SUPRA Server PDM Digest for OS/390 and VSE Systems*, P26-9062.

PDM and directory enhancements from prior releases of SUPRA Server

The following table lists selected enhancements of the SUPRA Server PDM and the SUPRA Server Directory that are still present but not new in SUPRA Server release 2.7. These enhancements were added to the product since Series 80 TOTAL:

- Release 2.7 includes these enhancements to logging and recovery procedures:

  - SUPRA Server provides multiple logical volumes for the System Log. The System Log is blocked and buffered, and I/O to the Log is overlapped. In addition, you can implement the System Log with any of the following access methods, using the device listed:

Access method	Device
BDAM	(disk)
ESDS	(VSAM disk)
BSAM	(disk or tape)
WORK	(tape; under VSE)
OUTPUT	(tape; under VSE)

  - SUPRA Server uses the task log file to provide central task-level recovery . The task log file is blocked and buffered, and I/O to the Log is overlapped.

  - SUPRA Server provides special DML for task restart logic. Detection of task failures is also improved.

  - SUPRA Server supports selective file RECOVER and RESTORE functions.

- SUPRA Server release 2.7 (PDM support) includes these improvements that enhance overall performance and operating efficiency:

  - SUPRA Server provides secondary keys that enable you to index any field in the database.

  - Buffer management is more efficient. SUPRA Server finds available buffers automatically, eliminating the need to search tables.

  - Task management and event-dispatching management are improved. Tasks are dispatched according to priority rather than by the round-robin method used in TOTAL and TIS. I/O operations have the highest priority. Internal tasks have a lower priority, and new DML functions have the lowest priority. In addition, SUPRA Server uses a variable-length list of event control blocks (ECBs) to manage I/O and other events.

  - The queuing scheme for holding records is more efficient.

  - SUPRA Server uses the optimum data structure for its internal tables and lists.

  - SUPRA Server uses fast-path coding, designed so that the most frequently used logic follows the shortest, most direct instructional paths.

  - Improved efficiency for data list processing. All data lists are bound, which means that efficient code is generated to move specific data. The bound code is automatically used again when appropriate.

  - Space management of SUPRA Server native related files has been improved. Instead of being clustered at the beginning of a file, new chains of records are placed at locations throughout the file by means of a hashing algorithm. SUPRA Server no longer uses Cylinder Control Records. There is, therefore, no performance bottleneck caused by tasks holding Cylinder Control Records when adding or deleting records.

  - Almost all SUPRA Server PDM DMLs are fully overlapped. Any task can execute virtually any DML instruction for which resources are available, without regard to commands being executed by other tasks. QUIET and QMARK are the only DML commands that are not overlapped, and they are valid only in a nontask logging environment.

- SUPRA Server provides extended status information for every unsuccessful DML instruction, including the module name and error number. Depending on the type of error, status information can also include other debugging data. The combination of status code, module name, and error number helps you pinpoint the specific error condition and the exact point in the PDM code where the PDM detected the error.

- SUPRA Server provides VSAM support. (Only Primary files can be KSDS VSAM files.)

- System administration tasks have been improved in the following ways:

  - SUPRA Server provides an interactive facility to develop schemas on the Directory. You can also maintain the Directory with batch transactions, and batch reports are available.

  - There is an interactive facility for displaying PDM statistics and commands to manage PDM tasks.

  - SUPRA Server provides run-time statistics for each file in the database.

  - SUPRA Server has improved user exits. Three user-exit programs can be linked with the PDM interface, and eight exit programs can be loaded by the PDM.

- SUPRA Server uses OS/390 cross-memory services to transfer data between the central PDM and interfaces. The CSA is no longer used to transfer data.

- Under VSE/SP and VSE/ESA, SUPRA Server provides cross-address space communication using XPCC.

- For OS/390, additional products are available to relieve storage constraints and to improve data storage in memory. The PDM can allocate storage in extended memory, and PDM buffers can be cached in extended memory.

- SUPRA Server provides support for new devices, including 3390s.

- Provides Y2K compliance.

- Removes some PDM and RDM storage limitations, giving you greater storage flexibility. Note the following:

  - RDM CICS COBOL and PL/I applications and data areas can reside in extended storage.

  - PDM applications can reside in extended storage when you use a central PDM.

  - For CICS RDM, task storage areas and global views can reside in extended storage.

  - SUPRA Server Online Directory Maintenance is divided into smaller load modules, which reduces storage limitations in the CICS region.

OS/390 Major PDM control blocks can be allocated in extended storage.

  - PDM buffers can be cached in extended storage.

  - A memory threshold can be set so that noncritical task memory is freed if PDM exceeds the threshold.

- Improved index read performance.

- Several enhancements simplify system administration tasks under CICS:

  - A command has been added to load and initialize the RDM and has also added a command to delete the RDM and free storage. This feature enables you to refresh RDM code and global views while CICS is active.

  - The CICS Connector now includes support for releasing task resources when the associated CICS terminal is lost.

  - You can now use CEDF to trace RDM and PDM calls under CICS.

- RDM VSAM can now use CICS MRO support (CICS 3.3 and higher).

- RDM now allows you to defer opening VSAM files under CICS.

- PDM secondary keys can be ordered (based on data type) and can be defined as unique or nonunique.

2

Migrating to SUPRA Server 2.7 from SUPRA 1.*x*

Migration process

When migrating from SUPRA 1.*x* to SUPRA 2.x, the migration process includes the following areas:

- Migrating metadata
- Migrating database files
- Migrating your programs
- Making changes to your configuration and system administration procedures

This chapter lists required procedures needed if you are migrating from SUPRA 1.*x*. If you are migrating from SUPRA 216, you may skip this chapter with the exception of “Updating the bootstrap module” on page 24.

If upgrading from SUPRA 216 to later SUPRA releases, directory migration is not required, but updating the bootstrap modules is required.

Migrating metadata includes part or all of the following:

- Expanding your Directory files
- Updating the bootstrap modules
- Unloading the C$ND NORMAL file (SUPRA 1.0, 1.1, or 1.2 users refer to your previous documentation)
- Migrating the Directory files
- Upgrading the Directory schema CSISCH20
- Upgrading user schemas
- Adding the Burrys schema
- Binding logical views
- Updating the secondary key RDM Eligibility flag
- Checking schemas for consistency
- Reallocating and loading the C$ND NORMAL file (SUPRA Server 1.0, 1.1, and 1.2 users refer to your previous documentation)

Migrating your database files requires no changes to database and index files. However, it does involve revising the task log and statistics files.

Migrating your programs includes part or all of these steps:

- Adding or updating the MANTIS cluster
- Updating applications
- Updating user exits

After completing the migration process, you will need to make changes to your configuration and system administration procedures. These include changes to:

- Procedures and samples
- CICS Connector configuration and administration
- RDM configuration and administration
- Directory Maintenance

Migrating metadata

The Directory structure has changed to support enhancements for SUPRA Server Release 2.7. To migrate metadata, you must:

- Expand your Directory files
- Update the bootstrap modules
- Unload the C$ND NORMAL file, if you are migrating from SUPRA Server 1.0, 1.1, or 1.2 (see earlier versions of documentation if this applies to your system)
- Migrate the Directory files
- Upgrade the Directory schemas
- Load the C$ND NORMAL file, if you are migrating from SUPRA Server 1.0, 1.1, or 1.2 (see earlier versions of documentation if this applies to your system)

These upgrade steps use no task or system logging. However, you should cold start your current system to ensure a clean task log file. Also be sure to make back-ups of the Directory and affected files BEFORE, DURING, and AFTER this upgrade. All Cincom-supplied sample members referenced in this process are supplied on the SUPRA installation tape in the MACLIB, JCLLIB and CSIPARM libraries **OS/390** or in the SUPRA base sublibrary **VSE**.

Expanding directory files

During various steps in migration, records will be added to your Directory files.

Before starting, we recommend that you run statistics to determine the amount of available space. If your Directory files are fairly full, you may want to consider expanding them before starting migration. Use sample TXJDSTAT to run Directory Statistics on your files. Refer to the *SUPRA Server PDM and Directory Administration Guide*, P26-2250, for instructions on how to expand the Directory files.

Updating the bootstrap module

The installation tape supplies these new bootstrap modules: the environment descriptions CSTANONE, CSTAREAD, CSTASUPD, CSTATLOG, and CSTAOLDM; the bootstrap schema CSTASCHM; and the validation module CSTAVMOD. These are on your SUPRA LINKLIB library **OS/390** or in the base sublibrary **VSE**. In the migration, you need to run modify schema to tailor the upgraded SUPRA Server 2.7 bootstrap schema for your SUPRA Server 1.*x* Directory and task log file sizes, access and blocking.

Updating the new bootstrap module involves the following steps:

1. Modifying the SUPRA Server bootstrap schema (required).
2. Making environment descriptions available (optional).
3. Rebuilding the validation module (required).

Whenever you tailor (modify) either a bootstrap schema or a bootstrap environment description, you must also rebuild the validation module to validate the new combination of modules. For detailed discussions of bootstrap modules, refer to the *SUPRA Server PDM and Directory Administration Guide,* P26-2250.

Follow these steps to update your bootstrap module:

1. Modify the bootstrap schema. Use the Modify Schema generator to modify the SUPRA Server bootstrap schema (CSTASCHM) to match your SUPRA Server 1.*x* Directory files, as follows:

   a. Create the batch job:

OS/390

Edit JCLLIB member TXJBMODS to create the batch job. The Directory files pointed to by this sample are your SUPRA Server 1.*x* version to be migrated. The CSTASCHM from the ENVLIB= library must be the upgraded SUPRA Server 2.7 version.

The libraries this sample points to are those from the SUPRA Server installation tape. For the two symbolic parameters LINKLIB= and NEWLIB=, you can use either a new library name in the NEWLIB parameter or the same name as in the LINKLIB parameter, causing the job to replace the schema module on the STEPLIB LINKLIB during the link edit. If you use a new library, it is best to put the five supplied bootstrap environment description modules there before rebuilding the validation module.

VSE

Edit TXJBMODS.A to create the batch job.

   b. Perform the following edit:

OS/390

Edit the appropriate MACLIB member MODSC*nnn*, where *nnn* is the last three digits of the unit/device where your SUPRA Server Directory files will reside (MODSC390 for 3390).

VSE

Edit the appropriate TISBUS*nn*, where *nn* is the last two digits of the device type where your SUPRA Server Directory files will reside.

Change the TOTAL-LOGICAL-RECORDS and LOGICAL-BLOCKS-PER-TRACK to match the last modification of your SUPRA Server 1.*x* Directory bootstrap schema.

   c. If your existing Directory files do not use full tracking, specify OLDFILE and TOTAL-LOGICAL-RECORDS statements.

   d. Submit the job and review the output. The return code should be zero. Refer to the *SUPRA Server Messages and Codes Reference Manual (PDM/RDM Support)*, P26-0126, for error message descriptions.

2. Make environment descriptions available. You need to make the five supplied bootstrap environment descriptions available in order to rebuild the validation module. Depending upon the steps you took when modifying the bootstrap schema, do one of the following:

   a. If you linked the modified schema back to the STEPLIB LINKLIB, skip this section and move to the next step to rebuild your validation module.

   b. If you linked the modified schema to a new library, you must link the environment descriptions to that library. Choose one of the following:

      - Copy the five supplied modules: CSTANONE, CSTAREAD, CSTASUPD, CSTATLOG, and CSTAOLDM, to your new LINKLIB.

OS/390

Execute the Create Environment Description bootstrap generator five times. Use JCLLIB member TXJBENVD, changing the MACLIB member each time as documented in the sample. The LINKLIB and NEWLIB parameters must both point to your new library containing the schema you modified. The job links the regenerated environment descriptions to this library.

VSE

Execute the Create Environment Description bootstrap generator five times. Use TXJBENVD.A, changing the EXEC PROC=TISBUENVX each time. (The procedure names are TISBUENV1 through TISBUENV5.)

3. Rebuild the validation module. After you have modified the bootstrap schema and possibly regenerated the environment descriptions, execute the following steps to run the Create Validation Module generator and rebuild the validation module.

   a. Create the batch job:

OS/390 Edit JCLLIB member TXJBVMOD to create the batch job. The LINKLIB and NEWLIB parameters must both point to the library containing the SUPRA Server schema you modified and the SUPRA Server environment descriptions. The job links the rebuilt validation module to this library.

VSE Edit TXJBVMOD to create the batch job.

   b. Perform the following:

OS/390 Use MACLIB input member CSTAVMOD as coded. MACLIB combines each of the five distributed bootstrap environment descriptions with your modified bootstrap schema, creating five valid module combinations. The PDM uses the validation module at execution to verify the boot modules named in CSIPARM or UCL.

VSE Use TISBUVAL.PROC as coded.

   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.

   c. Submit the job and review the output; the return code should be zero. Refer to the *SUPRA Server Messages and Codes Reference Manual (PDM/RDM Support)*, P26-0126, for error message descriptions.

Migrating the directory files

The process of migrating your Directory files changes the internal structure of your current Directory files and adds SUPRA Server data. Follow these steps:

1. Back up your Directory files before continuing, in case problems occur and you have to start the process over.
2. Create the following batch job:

OS/390 Create the batch job using the JCLLIB member TXJMIGXX. The MGRPGM symbolic parameter must be tailored to name the correct migration utility program for SUPRA Server 2.7. (If you are migrating from SUPRA release 1.0 through 1.2, use CSMBMIG4 on the first run to migrate the Directories to 1.3.5 level, then use CSMBMIG5 on the second run to migrate to the current SUPRA level.) (If you are migrating from TIS 2.3, you first use CSMBMIG3 from a SUPRA 1.3.5 release to migrate the Directories to 1.3.5 level, then use CSMBMIG5 from the current SUPRA release to upgrade to the current level.) This sample should point to your SUPRA Server Directory files and the libraries from the new SUPRA Server installation tape.

VSE Create the batch job using TXJMIG3X. (If you are migrating from SUPRA release 1.2.5 or 1.2.7, the program name on the EXEC statement should be CSMBMIG4 on the first run, then CSMBMIG5 on the second run. If you are migrating from SUPRA release 1.3.5, the program name on the EXEC statement should be CSMBMIG5, and there will only be one run.)

This job uses CSIPARM member TXPMIGRT, which does not use task logging. DO NOT change the accessed CSIPARM member in order to execute with task logging. If a problem occurs, you should restore from the backup.

3. Submit the job and review the output. The return code should be zero. There should be no messages.
4. Existing secondary keys will be migrated to remain compatible with their current populated attributes. You therefore do not have to repopulate unless you want to change those attributes.
5. Check all user schemas in the migrated Directory for consistency, using check options of all physical and logical entities, before attempting to use the schemas. Do this now with Batch Directory Maintenance, or later with Online Directory Maintenance.
6. Back up your Directory files before continuing.

Upgrading directory schemas

The steps to upgrade the Directory schemas are as follows:

1. Upgrading CSISCH20
2. Upgrading user schemas with CSISCH20 data embedded
3. Embedding CSISCH20 in user schemas
4. Updating the BURRYS schema
5. Binding Logical Views
6. Updating secondary key RDM Eligibility flags
7. Checking schemas for consistency

Upgrading schema CSISCH20

Schema CSISCH20 defines the SUPRA Server Directory files and the TIS/XA Software Selection Facility. These definitions allow you to:

- Run utilities against Directory files.
- Run Directory reports.
- Use the TIS/XA Software Selection Facility.
- Run RDM reports.
- Request metadata through the new Directory logical views using DBAID or SPECTRA.

You can upgrade the CSISCH20 schema on the Directory, or you can completely delete it and re-add it.

After upgrading the CSISCH20 schema on the Directory, you should upgrade any other schemas on your Directory which currently have CSISCH20 embedded.

The following steps use Batch Directory Maintenance. Changes to Batch Directory Maintenance that may affect successful execution include:

- The +SIGNON statement must reference a user defined as a DBA in the directory.
- To blank a field you must use a +NULL statement defining the null character and use that character in the first position of the field to be blanked.

The following steps upgrade schema CSISCH20 to the Directory:

1. (Optional) perform the following:

OS/390

You can edit MACLIB members DIRLOD13, DIRLMENU, and/or DIRLDVAD to relate additional users to desired views. The supplied DIRLOD13, DIRLMENU and DIRLDVAD transactions only relate the logical views to user CSI-DBA (authorized DBA) and CINCOM. By adding additional relate statements now, you can relate the views to the specified users in schema CSISCH20 as well as other schemas which you will upgrade by re-embedding or embedding CSISCH20 (“Embedding schema CSISCH20” on page 35).

VSE

The equivalent procedures for VSE are TISDM13, TISDMMEN, and TISDMDVA.

2. Create the Batch Directory Maintenance job:

OS/390

Edit JCLLIB member TXJBDIRM to create the Batch Directory Maintenance job. This job uses CSIPARM member TXPBSUPD and does not use task logging.

Although these steps process multiple members in one run, you can precede each member by a batch signon transaction (DIRLODSO) and process them as separate jobs.

VSE

Edit TXJBDIRM.A to create the Batch Directory Maintenance job.

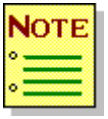
a. Run 1. Upgrade the Directory internal and external definition.

OS/390

Concatenate* the following supplied MACLIB Directory transactions input members:

DIRLODSO	Signs on as authorized User CSI-DBA. User CINCOM must also be present in your Directory.
DIRLOD11	(SUPRA Server 1.0 upgrades only) Upgrades 1.0 to 1.1.
DIRLOD12	(SUPRA Server 1.0 and 1.1 upgrades only) Upgrades 1.1 to 1.2.
DIRLOD13	(For 1.0, 1.1, and 1.2 upgrades) Upgrades 1.2 to 1.3.
DIRLOD22	(For 1.0, 1.1, 1.2 and 1.3 upgrades) Upgrades 1.3 to 2.x.

The DIRLOD*nn* transactions do not delete the schema on your current Directory. They do delete and add each entity they are changing, but any additional data is not lost.

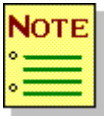


You can concatenate the MACLIB input members for a Batch Directory Maintenance job using the sample JCL in one of two ways. You can either copy all indicated input members into one member and name it in the symbolic MACLIB member parameter, or you can name the DIRLODSO member in the symbolic parameter and use a SYSIN DD override for additional concatenation.

VSE

Run the following procedures to upgrade the Directory internal and external definition:

TISDM11	(SUPRA Server 1.0 upgrades only) Upgrades 1.0 to 1.1.
TISDM12	(SUPRA Server 1.0 and 1.1 upgrades only) Upgrades 1.1 to 1.2.
TISDM13	(For 1.0, 1.1, and 1.2 upgrades) Upgrades 1.2 to 1.3. You must edit this member at statement CG FI C$ND to specify your current physical characteristics of the NORMAL domain file (access method, blocks per track, file size, etc.). Note the new record length of 237.
TISDM22	(For 1.0, 1.1, 1.2 and 1.3 upgrades) Upgrades 1.3 to 2.7.



The TISDM*nn* transactions do not delete the schema on your current Directory. They do delete and add each entity they are changing, but any additional data is not lost.

b. Run 2. Replace the TIS/XA Software Selection Menu and add new metadata views.

OS/390

Concatenate the following members:

DIRLODSO	Signs on.
DIRLDMNU	Deletes the previous version of the TIS/XA Software Selection Facility views and the table. In prior releases, the Batch Directory Maintenance transactions for the Menu were included in the DIRLODRE member.
DIRLMENU	Deletes and adds the new TIS/XA Software Selection Facility logical views to CSISCH20 and the new Menu table to the Directory.
DIRLDVDE	Deletes the Directory views.
DIRLDVAD	Adds the metadata logical views of Directory to CSISCH20 for SPECTRA and DBAID requests.

VSE

Run the following procedures to replace the TIS/XA Software Selection Facility and add new metadata views:

TISDMDMN	Deletes the previous version of the TIS/XA Software Selection Facility views and the table. In prior releases, the Batch Directory Maintenance transactions for the Menu were included in the TISDMNML member.
TISDMMEN	Deletes and adds the new TIS/XA Software Selection Facility logical views to CSISCH20 and the new Menu table to the Directory.
TISDMDVD	Deletes the Directory views.
TISDMDVA	Adds the metadata logical views of Directory to CSISCH20 for SPECTRA and DBAID requests.

These views should be added to the schemas in which they will likely be used.

c. Run 3. (OPTIONAL) Tailor the Directory file size definitions. This step is included in case you have changed the Directory file sizes but have not yet changed the CSISCH20 description.

OS/390

Concatenate the following members:

DIRLODSO Signs on.

DIRCH*nnn* (DIRLODDV for VSAM Directory files) Changes BDAM Directory file definitions and other loaded entities to your requirements. The MACLIB contains a DIRCH*nnn* member for your BDAM device type (where *nnn* is the device type 375 for 3375, 380 for 3380, and 390 for 3390). Use this member to match the DIRLODCM Directory file definitions with the modified bootstrap schema.

VSE

To tailor the Directory file size definition, run this procedure:

TISDMD*nn* (TISDMDVS for VSAM Directory files) Changes Directory file definitions and other loaded entities to your requirements (where *nn* is the last two digits of the device type.)

This match is not strictly necessary, but it causes Directory reports on the Directory files to show the same physical values the PDM actually uses from the bootstrap.

3. Submit the runs separately and review the output when each job ends. The supplied transactions may cause an acceptable return code greater than zero, either by resequencing in Variable Edits (VE) or by trying to delete nonexistent entities. Refer to the *SUPRA Server Messages and Codes Reference Manual (PDM/RDM Support)*, P26-0126, for error message descriptions.

You can make backups after any execution of the job. You should make backups before proceeding to the next step since you are not using task logging.

If you prefer, you can completely delete schema CSISCH20 and re-add it. (Any entities you added to it will be lost.) To do this, replace Run 1 in this section with the following; then perform Runs 2 and 3.

OS/390

Replace Run 1 with the following concatenated MACLIB members:

DIRLODSO	Signs on.
DIRLODRD	Deletes previous definition of the 3 NORMAL schemas (needed for Conceptual schema and domain deletion from the Directory).
DIRLODDE	Deletes existing definition of CSISCH20 and previous Cincom tables and edit masks.
DIRLODCM	Loads SUPRA Server tables and edit masks. Adds schema CSISCH20 and loads Directory file definitions and a READ environment description (SUPRA Server internal schema).
DIRLODLV	Adds Directory reports logical views to CSISCH20.
DIRCH*nnn*	(DIRLODDV for VSAM Directory files) Changes BDAM Directory file definitions and other loaded entities to your requirements. The MACLIB contains a DIRCH*nnn* member for your BDAM device type (where *nnn* is the device type: 375 for 3375, 380 for 3380, or 390 for 3390). Use this member to match the DIRLODCM directory file definitions with the modified bootstrap schema.

This match is not strictly necessary, but it causes Directory reports on the Directory files to show the same physical values the PDM actually uses from the bootstrap.

VSE

Replace Run 1 with the following procedures:

TISDMNMD	Deletes previous definition of the 3 NORMAL schemas (needed for conceptual schema and domain deletion from the Directory).
TISDMDE	Deletes existing definition of CSISCH20 and previous Cincom tables and edit masks.
TISDMSCH	Loads SUPRA Server tables and edit masks. Adds schema CSISCH20 and loads Directory file definitions and a READ environment description (SUPRA Server internal schema) and adds Directory reports logical views to CSISCH20.
TISDMD*nn*	(TISDMDVS for VSAM Directory files) Changes Directory file definitions and other loaded entities to your requirements (where *nn* is the last two digits of the device type.)

This match is not strictly necessary, but it causes Directory reports on the Directory files to show the same physical values the PDM actually uses from the bootstrap.

Upgrading CSISCH20 in user schemas

For schemas that already contain embedded CSISCH20 information, repeat the steps in “Upgrading schema CSISCH20” on page 29, editing the members to replace all occurrences of the schema name CSISCH20 with the selected schema name.

Embedding schema CSISCH20

To embed the SUPRA Server CSISCH20 schema information in user schemas, follow the steps below. This information allows you to perform Directory reporting and use the TIS/XA Software Selection Facility and make Directory metadata queries from the active schema.

For schemas that do not contain embedded CSISCH20 information, follow these steps to add CSISCH20 information:

1. Perform the following edit:

OS/390

Again edit the JCLLIB member TXJBDIRM.

VSE

Edit TXJBDIRM.A.

2. Perform the following:

OS/390

Concatenate the following supplied MACLIB Directory transaction input members:

DIRLODSO	Signs on.
DIRLODCO	Copies all schema CSISCH20 entities (except log files and metadata logical views) into another schema. It first deletes the entities from the target schema. You must edit to replace each occurrence of the names USERSCHM and USERED*nn*, BUF1 and BUF3. The comments in the member describe each variable name. If you wish your schema to contain only the Menu, or only Directory reports, you can comment out unwanted view and file entities.
DIRLDVDE	Deletes Directory views.
DIRLDVCO	*Optional*. Copies the logical views of Directory for metadata requests. You might want to use this only for selected schemas, both for security reasons and for Directory file space considerations. Edit this member to change the schema name.

VSE

Run the following procedures:

TISDMCO	Copies all schema CSISCH20 entities (except log files and metadata logical views) into another schema. It first deletes the entities from the target schema. You must edit to replace each occurrence of the names USERSCHM and USERED*nn*, BUF1 and BUF3. The comments in the member describe each variable name. If you wish your schema to contain only the Menu, or only Directory reports, you can comment out unwanted view and file entities.
TISDMDVD	Deletes Directory views.
TISDMDVC	*Optional*. Copies the logical views of Directory for metadata requests. You might want to use this only for selected schemas, both for security reasons and for Directory file space considerations. Edit this member to change the schema name.

3. If you run this job against a schema that has the previous version of the TIS/XA Software Selection Facility:

OS/390 You will need to concatenate MACLIB member DIRLDMNU with the job.

VSE You will need to run TISDMDMN.PROC.

4. Submit the job and review the output. The supplied transactions may cause an acceptable return code greater than zero due to resequencing in Variable Edits (VE) or deleting nonexistent entities.

5. Perform the following:

OS/390 Rerun the job for each schema you select, changing USERSCHM, USERED*nn*, BUF1 and BUF3 names in DIRLODCO and schema name in DIRLDVCO.

VSE Rerun the job for each schema you select, changing USERSCHM, USERED*nn*, BUF1 and BUF3 names in TISDMCO.PROC and schema name in TISDMDVC.PROC.

6. Back up your Directory files before continuing.

Updating the Burrys schema

The distributed SUPRA Server 2.7 Directory contains a new BURRYSCH and new Burrys database files for educational purposes. You must use these instead of the previous BURRYSCH or files.

The SPECTRA Personal File System (PFS) file has been defined as the LV002 VSAM file to the BURRYS schema.

To upgrade the Directory to the new BURRYSCH, use JCL sample TXJBDIRM and the BURRYSDM MACLIB member. This replaces the schema. Make sure that the new Burrys database files (E$*xx*) have been replaced with the new versions.

The default file definitions on the Burrys schema are distributed for BDAM access method and 3390 device types. If you want to use another access method or device type, follow these steps:

1. Perform the following:

OS/390

Use JCL sample TXJBDIRM and the following members in MACLIB to modify your Burrys schema file definition.

BURRYSVC	Changes your Burrys file access method to VSAM ESDS.
DIRCH*xxx*	Changes the device type your Directory files will reside on. The access method will remain BDAM where *xxx* represents the device type. For example, *xxx* = 350 implies that the files will reside on 3350 DASD.
NMLCH*xxx*	Changes the device type your NORMAL files will reside on. The access method will remain BDAM where *xxx* represents the device type. For example, *xxx* = 350 implies that the files will reside on 3350 DASD.

VSE

Use JCL sample TXJBDIRM.A and the following procedures to modify your Burrys schema file definition.

TISBRVC	Changes your Burrys file access method to VSAM ESDS.
TISDMD*nn*	Changes the device type your Directory files will reside on (where *nn* is the last two digits of the device type).
TISDMN*nn*	Changes the device type your NORMAL files will reside on (where *nn* is the last two digits of the device type).

2. Submit the job and review the output; the return code should be zero. Refer to the *SUPRA Server Messages and Codes Reference Manual (PDM/RDM Support)*, P26-0126, for diagnostic and error message descriptions.

3. After the job completes successfully, check the Burrys and/or CSISCH20 schemas for consistency.

Binding logical views

Use the BIND BOUND command in DBAID to rebind all currently bound views in a schema. This should be done for each schema containing bound logical views. Binding logical views improves performance on the initial access to a view, but is only suggested for production systems where the physical database is stable.

You must rebind all bound views. Optionally, you can also bind any other logical views in your Directory.

We recommend the following procedure:

1. Bind the two views for the TIS/XA Software Selection Facility.

2. Bind the Directory metadata logical views only in the schema you expect to be active when you issue DBAID or SPECTRA requests for Directory information. There are more than 70 metadata views, and they can fill your Directory DATA file (C$-D), especially if they are bound in multiple schemas.

Because binding requires considerable space on the Directory DATA file (C$-D), you need to balance the enhanced speed of opening bound views against the file space required. For information on expanding the Directory files, refer to the *SUPRA Server PDM and Directory Administration Guide*, P26-2250.

Use DBAID to rebind all previously bound logical views (use the BIND BOUND command). You can also use the following DBAID input members to bind supplied logical views:

VSE	OS/390	
TISDBBND	DIRLODDB	Binds the two TIS/XA Software Selection Facility views and NORMAL Design views. On the distributed Directory, these views are bound in two schemas, CSISCH20 and BURRYSCH.
TISDMDVA	DIRLDVAD	Binds the Directory metadata views.
TISBRBD	BURRYSBD	Binds the Burrys views.

To bind Burrys, TIS/XA Software Selection Facility, and Directory metadata views, follow these steps:

1. Back up your Directory first, in case binding fills the DATA file.

2. If you use schemas with task logging, reformat your Task Log File. See "Migrating database files" on page 43 for Task Log File considerations.

3. Edit JCL sample TXJBDAID to create the batch job. This job uses CSIPARM member TXPUTLOG (Burrys schema with task logging). To bind Burrys logical views in the Burrys schema, run the job with the input member BURRYSBD **OS/390** or TISBRBD **VSE**. You can also follow steps 3 and 4 for the Burrys schema.

   To bind views in another schema, change the CSIPARM file from TXPUTLOG to the name of your CSIPARM file. You can bind views only in an active schema having DIR-ACCESS SUPD. You must also add your file JCL statements.

   If you want to bind in schema CSISCH20, create a CSIPARM file as follows:

   ```
   DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTATLOG),
   REALM=(SCHEMA=CSISCH20,ENVDESC=CSIENVRG),END.
   ```

   The CSIPARM file shown here specifies a task logging environment. Be aware that bound views in CSISCH20 will be copied hereafter when you copy the schema or selected logical views, or when you use DIRLODCO and DIRLDVCO **OS/390** or TISDMDVC **VSE**. This requires extra DATA file space.

4. Run the job once with MACLIB input member DIRLODDB **OS/390** or TISDBBND **VSE** to bind the two TIS/XA Software Selection Facility views (first two in the member). Remember that these views are already bound in CSISCH20 and the Burrys schema on the distributed Directory.

5. Run the job again with MACLIB input member DIRLDVBD **OS/390** or TISDBDVB **VSE** to bind the metadata logical views.

General considerations

- If you receive message DERR600CSTA650, the binding filled up your Directory DATA file. In this case, you can either unbind some views or you can go through the process of expanding the Directory.
- Refer to the *SUPRA Server RDM Administration Guide (OS/390 & VSE),* P26-8220, for examples of using batch DBAID. Refer to the *SUPRA Server Messages and Codes Reference Manual (PDM/RDM Support for OS/390 & VSE)*, P26-0126, for error message descriptions.
- To bind the views in a different schema, repeat the steps, changing the CSIPARM and file definitions.
- The only way to unbind is to delete and re-add a logical view. Use DBAID on the affected schema to LIST, REMOVE(Y), SAVE, and PERMIT (to reinstate the User relates) for each view you wish to unbind. Refer to the *SUPRA Server RDM Administration Guide (OS/390 &VSE),* P26-8220, for information about binding views.

Updating secondary key RDM eligibility flags

You should consider the following changes to secondary keys when you migrate to SUPRA Server Release 2.7:

- The secondary key POPULATED flag has been removed.
- A secondary key MAINTENANCE ALLOW attribute has been added. This attribute indicates whether to allow maintenance on a secondary key structure.
- NAVIGATION PRIMITIVE entities have been removed from the Directory.
- A secondary key ELIGIBLE TO RDM attribute has been added. If your RDM views use secondary keys, you need to review and update this attribute. This applies to views using generalized (WHERE) and explicit (VIA/USING) address techniques.

  If the ELIGIBLE TO RDM attribute is set to Y, RDM will try to use the secondary key when opening and binding views. The associated index must be on the active schema and must be populated.

  If the ELIGIBLE TO RDM attribute is set to N, the opening or binding of a view will fail.

Checking schemas for consistency

At this point in the migration, check all entities in each schema on your Directory for consistency. Use Directory Maintenance to check the schemas. Specify Y for all the physical and logical entities to be checked.

Migrating database files

When migrating to SUPRA Server 2.7 from SUPRA Server Releases 1.3.5 or 1.3.6, your database and index files do not require change.

When migrating to SUPRA Server 2.7 from a release prior to SUPRA Server 1.3.5 (SUPRA Server 1.2), depopulate and populate all secondary keys.

You need to reformat your SUPRA Server 1.*x* Task Log File before running task logging with SUPRA Server 2.x. The prerequisite is that your last PDM shutdown was unforced on your previous execution of the PDM. Otherwise, you will need to allocate and format using JCL sample TXJFTLOG to reformat your existing Task Log File.

If you run with a PDM statistics file, it requires a minimum LRECL of 512. Make sure your statistics file meets this requirement.

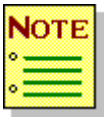
Migrating programs

Migrate your programs to SUPRA Server Release 2.7 by performing the following steps:

1. Add the TIS/XA Software Selection Facility to your MANTIS cluster.
2. Update applications.
3. Update CICS Connector exits.
4. Update PDM interface exits.
5. Update RDM CICS exits.

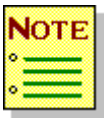
The information in the following sections assumes you are migrating programs from SUPRA Server Release 1.3.5. When migrating from an earlier release of SUPRA, there may be additional considerations.

Adding the TIS/XA Software Selection Facility to your MANTIS cluster



Skip this section if you have already used the MANTIS Install or Reinstall Cluster from the SUPRA Server upgrade installation tape.

If you are a MANTIS user and have not yet reinstalled, follow the instructions below. The MANTIS Cluster must contain the upgraded TIS/XA Software Selection Facility information in order for it to function properly for this release of SUPRA.



Back up your MANTIS Cluster before continuing.

1. User CSI-DBA will be added or replaced in the MANTIS Cluster. If you have CSI-DBA as a user, rename it before you REPRO the Reinstall Cluster.

2. SAVE the MASTER:SIGN_ON and MASTER:TERMINATE MANTIS programs if you will not be using the software menu and wish to retain your existing versions. Otherwise they will be replaced during the reinstall. After the reinstall, you can replace the programs you saved.

3. Check the JCL used to install SUPRA Server in order to locate which file on the tailored tape contains the MANTIS Reinstall Cluster.

4. REPRO the MANTIS Reinstall Cluster from the SUPRA Server installation tape to your existing cluster by using the REPLACE option.

Updating application programs

PDML applications require no coding changes to operate with SUPRA Server Release 2.7. However, you must link edit CICS PDM applications with the 2.7 version of DATBASC. For instructions on link editing CICS PDM applications with the new DATBASC, refer to the *SUPRA Server PDM DML Programming Guide,* P26-4340.

For SUPRA Server Release 2.7, there are several changes to PDM DML:

- The CNTRL command is not supported.
- The LOCK and MNTR end parameter options for READ commands are not supported.
- New statistics have been added to RSTAT.
- There are some new SHOWX data item keywords, and some existing keywords have different meanings and results. These changes could affect any applications you already have that specify a SHOWX command. Review your programs to determine if there are existing SHOWX commands present and update them if necessary. For more information on changes to SHOWX commands, refer to the *SUPRA Server PDM DML Programming Guide,* P26-4340.
- When using READX with the MASKING option, applications will almost certainly notice a change in status returned on the first READX call using BEGN in the qualifier. With prior releases (example: SUPRA 1.3.5) *NXT would usually be returned meaning that the key value was different for the initial key built using the mask. Note that the key returned would still match the mask.

  With SUPRA 2.4 and above, on the first call, **** will be returned in the status field, provided the key returned still matches the mask. *NXT will never be returned as a status on the first call.

  On subsequent calls, the behavior of SUPRA 2.4 and above is consistent with previous releases. If the key returned by the PDM has not changed since the last call, **** is returned . If it has changed and the key returned matches the mask, *NXT is returned. When there are no more keys that match the mask, END. is returned in the qualifier and **** is returned in the status.
- If you have applications which issue READX PDM commands to perform generic reads, you should take these differences into account.

- Statuses returned by READX when partial key options are used have changed: when using partial keys (and MASK=NO) if a record matches the partial key **** will be returned, *NXT is returned when the subsequent record's key differs from the first key returned.
- Note that a partial key with MASK=YES will return an IPAR.
- Only batch mode supports alternate indexes for non-PDM KSDSs.

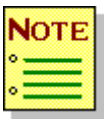
To maintain compatibility for TOTAL statuses and DML, use the PDM and CICS Connector exit programs supplied.

For the PDM interface precommand, postcommand, and DATBASXT exits, use the supplied exit programs CSTK0001, CSTK0002, and CSTK0003. For more information, refer to the *SUPRA Server PDM and Directory Administration Guide*, P26-2250.

You should also use the preprocess and postprocess exit points in the CICS Connector exit CSTXUSRX, which replaces the modified TOTINT and CSTXDBXT exits. (See also the TOTC parameter in the CSTXOPRM macro.) For more information on the CICS Connector exits, refer to the *SUPRA Server CICS Connector Systems Programming Guide*, P26-7452. For more information on the CSTXOPRM macro, refer to the *SUPRA Server PDM and Directory Administration Guide*, P26-2250, and the *SUPRA Server RDM PL/I Programmer's Guide*, P26-8331.

RDML applications do not require changes to operate with SUPRA Server Release 2.7. However, if you use CICS/ESA, you must process COBOL, COBOL II, and PL/1 applications with the new preprocessors and recompile them.

In previous releases of SUPRA, preprocessors generated an ADDRESS CSA command. This command is not valid in CICS/ESA.



If your CICS RDML applications or data areas reside in extended storage, you must link edit them with the new RDM interface CSVNICIC. You can continue to use CSVICICS with applications that reside below the 16 MB line, but we recommend that you link edit these applications with the new CSVNICIC interface as well. For information on link editing CICS RDM applications, refer to the *SUPRA Server RDM COBOL Programmer's Guide*, P26-8330.

Updating exit programs

The exit programs you need to update include all or part of the following:

- CICS Connector exits
- PDM interface exits
- RDM CICS exits
- TISXA Software Selection Facility exits

Most SUPRA Server exits now receive control in AMODE-31. You must verify that your exit programs can operate in this mode. If an exit program cannot operate in this mode, you must either upgrade it or insert code that will switch to 24-bit addressing mode and then restore the addressing mode to AMODE-31 before returning. The following considerations apply to upgrading exit programs:

- In general, CICS macro level requests cannot be issued in AMODE-31. You must add code to switch to AMODE-24 (see the following sample code).
- PDM parameters and data areas can reside above the 16 MB line. If your exit programs access these, the programs must operate in AMODE-31.

For general OS/390/XA program conversion requirements, refer to IBM documentation. An example of code that will switch operation to AMODE-24 and then restore the original addressing mode appears below:

```
************************************************************
*               CODE TO SWITCH TO AMODE-24                 *
************************************************************
       LA   R9,LABEL2         SETUP FOR RETURN TO ORIGINAL
       LA   R14,LABEL1        SETUP FOR SWITCH TO AMODE-24
       BSM  R9,R14            CHANGE TO AMODE-24, SAVE AMODE
       DC   H'0'
LABEL1 DS 0H
       ... your existing code which must switch to AMODE-24
       ... is here.
************************************************************
*               CODE TO RESTORE AMODE                      *
************************************************************
       BSM  0,R9              RESTORE AMODE
       DC   H'0'
LABEL2 DS   0H
```

Updating CICS Connector exit programs

All CICS Connector exit programs may be invoked in AMODE-31. Review your exit programs and update them if necessary. For more information on using the exits, refer to the *SUPRA Server CICS Connector Systems Programming Guide,* P26-7452.

- The DATBASC exit CSTXUDAT operates the same way as it did for SUPRA Server Release 1.3.5. However, if the return code in register 15 is not zero, control passes to the return address in register 14, and the PDM call does not occur.

This exit will operate in the addressing mode of the application calling DATBASC.

- The task start user ID exit CSTXUTID is now called by the CICS Connector only at the start of the task. Its primary purpose is to allow you to inspect and change the task ID. You can also use this exit to change the synchronization options for a task.
- The CSTXDBXT exit is obsolete. Its functions are performed by separate pre-DML and post-DML exits. To invoke these exits, refer to the TPRE and TPOS parameters in the CSTXOPRM macro.
- The CSTXKCCX exit is obsolete. The Task Detach user exit in CSTXUSER now performs this exit's functions. To invoke the Task Detach user exit, refer to the TDUE parameter in the CSTXOPRM macro.

Updating PDM interface exit programs

PDM interface exits are invoked in the address mode of the application that issues the DML. For more information on using the exits, refer to the *SUPRA Server PDM and Directory Administration Guide*, P26-2250.

- PDM pre- and post-command interface exits, CSTK0001 and CSTK0002, are invoked in the address mode of the application that issues the DML. You should review your exit programs and update them if necessary.

- The DATBASXT exit has not changed for SUPRA Server Release 2.7. However, it is invoked in the address mode of the application issuing the DML. Review your exit programs and update them if necessary.

- The EIB address is passed to PDM interface exits in order to support CICS command level requests in the exit. For more information, refer to the parameter address lists for PDM interface exits in the *SUPRA Server PDM and Directory Administration Guide*, P26-2250.

Updating PDM exit programs

OS/390

For SUPRA Server Release 2.7, all PDM exits are invoked in AMODE-31. Review your exit programs and update them if necessary. If you use the SUPRA Server 2.7 buffer cache facility, all PDM names in the Environment Description must be updated to "CSTTEXIT". If you want to use other exits with CSTTEXIT, you must link edit the exits with CSTTEXIT and use the following entry point names:

- 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 additional information about the buffer cache facility, refer to the *SUPRA Server PDM and Directory Administration Guide*, P26-2250.

VSE

For SUPRA Server Release 2.7, the PDM and all PDM exits are invoked in AMODE-24.

Updating RDM CICS exit programs

RDM CICS exits (CSVX*xxxx*) are invoked in AMODE-24, although the parameters passed to the exits may be 31-bit addresses. Review your exit programs and upgrade them if necessary. For more information on using the exits, refer to the *SUPRA Server RDM Administration Guide,* P26-8220.

Updating TIS/XA Software Selection Facility exit programs

User exit programs for the TIS/XA Software Selection Facility (SSF) have changed as follows for SUPRA Server Release 2.7:

- Renamed existing exits to conform to Cincom standards.
- Improved documentation accompanying the exits.
- Added new exits.
- Moved MASTER programs SIGN ON and TERMINATE from the MASTER user to the CSI-DBA user to enhance security.

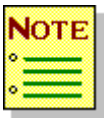
The following table shows changes to individual exits. Upgrade your exit programs as necessary. For more information about SSF, refer to the *SUPRA Server PDM and Directory Administration Guide*, P26-2250.

Exit name	Old exit name	Changes/functionality
CSFS0000 SIGNON EXIT	VALIDATE EXIT	CONTINUE and UPDATE functionality has been removed.
CSFS0000 VALUSER EXIT	-	Replaces functionality of CONTINUE and UPDATE in old VALIDATE EXIT, allowing you to restrict user access to specific selections on specific databases.
CSFS0000 VALALT EXIT	-	Replaces functionality of CONTINUE and UPDATE for an alternate user, allowing you to restrict an alternate user's access to specific selections.
CSFS0000 MENU EXIT	-	Allows you to restrict access to selections by site, terminal location, and so on.
CSFS0000 VALUSER EXIT	-	Replaces functionality of CONTINUE and UPDATE in old VALIDATE EXIT, allowing you to restrict user access to specific selections on specific databases.
CSFS0000 VALSELECT EXIT	SELECTION EXIT	Adds default and alternate database names when parameters are passed. (Applies only to SQL databases at this time.)
CSFS0000 FAILSIGN EXIT	FAILED SIGNON	User documentation has been added.

Making changes to the configuration and system administration

After the migration process, you must consider changes or updates to your system configuration and system administration in the following areas:

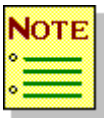
- Procedures and samples
- CICS Connector configuration and administration
- RDM configuration and administration
- Directory Maintenance
- PDM configuration and administration



The changes and updates listed assume you are migrating from SUPRA Server Release 1.3.5. When migrating from a prior release of SUPRA, there may be additional changes in configuration and system administration you need to consider.

Procedures and samples

SUPRA Server Release 2.7 has revised catalogued procedures, sample JCL, and sample input. They now contain documentation on how to use them. You should use the new JCL samples and procedures, or upgrade your JCL to reflect these changes.



For CICS execution JCL, all RDM modules are now loaded from libraries defined by the DFHRPL DD statement.

CICS Connector configuration and administration

The changes to the CICS Connector configuration include changes to:

- Storage requirements
- Initialization
- Operator control table
- CICS resource definition
- Termination

Storage requirements

In prior releases, you specified storage requirements in the OPER CONNECT OPER command. For SUPRA Server Release 2.7, the CICS Connector calculates storage requirements automatically. In addition, the CICS Connector will now try to extend storage if necessary. For more information, refer to the MEME parameter of the CSTXOPRM macro in the *SUPRA Server PDM and Directory Administration Guide,* P26-2250.

Previously, memory was allocated from CICS DSA and associated with the Support Services program (TISI transaction). The Support Services program is not present in Release 2.7. Storage is now allocated as follows:

- OS/390 GETMAIN for OS/390 CICS
- CICS dynamic storage area for CICS/ESA

You should take this change into account when you determine your OSCOR and DSASZE parameters. For more information on estimating required storage, refer to the *SUPRA Server OS/390 Installation Guide*, P26-0149, or the *SUPRA Server VSE Installation Guide*, P26-0132.

CICS Connector initialization

In earlier releases of SUPRA, you could start the CICS Connector with the PLT initialization program CSTXPLTI. This program issued a START command for the OPER transaction, which was queued by CICS until the CICS initialization was complete.

It is now recommended that you use the sequential (BSAM) device as a simulated terminal to issue the OPER CONNECT command to initialize the CICS Connector, start RDM if necessary (OPER RDM START), and issue any other transactions that need to be done at CICS startup. Although you can use a new PLT initialization program, CSTXPLCI, which will issue an OPER CONNECT, this method is no longer recommended because the new CICS releases now run all PLTI programs concurrently.

For more information on starting the CICS Connector, refer to the *SUPRA Server CICS Connector Systems Programming Guide*, P26-7452.

Changes to the Operator Control Table

The following table shows new or changed parameters of the Operator Control Table, which is defined by the CSTXOPRM macro. For more information on the CSTXOPRM macro, refer to the *SUPRA Server PDM and Directory Administration Guide*, P26-2250.

Parameter	Description/controls
MEME	Specifies the maximum number of additional memory blocks that may be acquired if memory allocation is exceeded.
IFRTRY	Specifies the number of one-second delays to issue while waiting to free task context for an “in-flight” PDM request.
TRACE1-3	Defines the trace IDs for starting and stopping commands and for new trace entries.
TOTC	Converts TOTAL commands to SUPRA Server equivalents.
PUSHPOP	Handles the PUSH/POP option for CICS commands.
IMMEDRS	Handles the RESET/SINOF of a task when it detaches, instead of scheduling an RSSO transaction.
OCPTRTM	Specifies the transaction ID for CSTXTMSG, used to notify the operator that a task is being reset or signed off (RESET/SINOF).
USERTID	Defines the transaction ID for the user initialization program.
MAXPACK	Defines the maximum packet size for a parameter (VSE only).
MESSQID	Defines the queue ID name for the audit log.
TFUL, CFUL, ICOR	Defines number of retries for statuses.
TDUE, TPOP, TPOS, TPRE, TPRP, TSYN, TUID	Specifies user exit names. CSTXUSER supplies the default logic for all of these exit points.
SYNC	Handles the task synchronization option.

CICS resource definition

With CICS/ESA, you must define some resources with RDO. Installation tapes now provide sample CICS table input and sample off-line RDO transactions. Separate samples are provided for ESA/CICS and CICS/VSE.

For more information on CICS resource definition, refer to the *SUPRA Server CICS Connector Systems Programming Guide*, P26-7452, and the *IBM Installation Primer* that you receive with your installation tape.

Termination

In prior releases, you could terminate the CICS Connector with the PLT termination program CSTXPLTT. This program has been replaced by the PLT termination program CSTXPLDT. For more information on terminating the CICS Connector, refer to the *SUPRA Server CICS Connector Systems Programming Guide,* P26-7452.

RDM configuration and administration

Changes to RDM configuration include changes to the following:

- Storage requirements
- Initialization
- The RDM options module
- CICS resource definition
- Termination

Storage requirements

- In prior releases, RDM task context was allocated in “slots.” The number and size of the slots allocated was determined by the C$VUWORK macro, as assembled and link edited in the module CSVUCICS. In SUPRA Server Release 2.7, you allocate task context in a separate “heap” and “stack.” You specify heaps and stacks in the options module, CSVOOPTM. For more information on RDM storage and the options module, refer to the *SUPRA Server RDM Administration Guide,* P26-8220.

  - Heaps are used to store view context, and can be allocated above the 16 MB line. Heap storage is required for every RDM CICS transaction and is retained until the transaction signs off the RDM. All heap storage is allocated during initialization of the RDM. The size and number of heaps is determined by the options module CSVOOPTM. If more tasks sign on to RDM than there are available heaps, heap storage rolls to auxiliary temporary storage (for pseudoconversational tasks).

  - Using Release 1.3.5 slot sizes as a guideline, you should allocate approximately 75–80% of the slot size to the new heap size. Since you can allocate heap storage above the 16 MB line, you may want to use more heaps than the number of slots you specified with Release 1.3.5 to reduce rolling (see above).

  - Stacks are used to store task context for the Pascal routines of RDM. Stacks are always allocated below the 16 MB line. Stack storage is required for every RDM CICS task. The task acquires stack storage when it enters the RDM and releases the stack storage when it leaves the RDM. Stacks are no longer rolled to CICS auxiliary storage. (Rolling stack context to CICS auxiliary storage is not necessary because stacks no longer contain task context that is required across RDML calls.) The size of stacks is determined by the options module CSVOOPTM.

  - Using SUPRA Server Release 1.3.5 slot sizes as a guideline, you should allocate approximately 25–30% of the slot size to the new stack size.

- The global view pool size is also defined in the options module CSVOOPTM. Since the pool can be allocated above the 16 MB line, you may wish to use a larger pool size than in Release 1.3.5 so that more views can remain in memory. After global views are opened, SUPRA Server releases all unused memory in the pool.

- For SUPRA Server Release 2.7, if you run the PDM in central or attached mode, you can allocate RDM CICS above or below the 16 MB line. This option is possible because of OS/390/XA extended storage support in the SUPRA Server 2.7 PDM. The options module CSVOOPTM defines whether storage is allocated above or below the 16 MB line.

- In SUPRA Server Release 1.3.5, RDM storage is acquired with OS/390 GETMAIN requests. This storage is allocated from the CICS OSCOR area, as specified in the SIT. In Release 2.7, RDM storage is acquired as follows:

  - Stacks are allocated from the CICS dynamic storage area.

  - Heaps and the global view pool are acquired with OS/390 GETMAIN requests. When you define storage below the 16 MB line, these are allocated from the OSCOR area. With CICS/ESA, OSCOR is obsolete, and the available space will be the size of your CICS region, minus the DSASZE operand you specify in the SIT.

RDM initialization

In prior releases of SUPRA, you could initialize the RDM with the PLT initialization program CSTXRDIN. You may now use the sequential (BSAM) device as simulated terminals to issue the OPER RDM START command.

The RDM options module

The RDM macro C$VUWORK is obsolete. A new options module, CSVOOPTM, defines the RDM CICS storage configuration. The macro C$VOOPTM generates CSVOOPTM. The following table compares the operands used to supply RDM run-time options in the new macro C$VOOPTM with those in the obsolete macro C$VUWORK. For more information on the options module, refer to the *SUPRA Server RDM Administration Guide,* P26-8220.

C$VOOPT operand	C$VUWORK operand	Comment/description
CICS=YES	CICS=YES	Specifies CICS environment.
GETMAIN	none	Specifies whether GETMAINS will be above or below the 16 MB line.
GLOBSIZ	none	Specifies global view pool size.
HEAP	#SLOT#	Specifies maximum number of heaps.
HEAPSZ	SLOTSZ	Specifies heap size.
RDMUSR	none	Specifies maximum number of concurrent RDM users that may be signed on.
RPTSIZE	RPTSIZE	Specifies online DBAID report size (max. increased from 31K to 63K).
STACKSZ	SLOTSZ	Specifies the size of RDM stacks.
SYNCTYP	none	Specifies the type of CICS sync pointing.
TCISIZE	none	Specifies CICS temporary storage control interval size.
TSROLL	none	Specifies temporary storage roll destination.

CICS resource definition

With CICS/ESA you must define some resources with RDO. Your installation tape provides sample CICS table input and sample off-line RDO transactions. Separate samples are provided for CICS/ESA and CICS/VSE.

For more information on CICS resource definition, refer to the *SUPRA Server CICS Connector Systems Programming Guide*, P26-7452, and the Installation Primer that you receive with the installation tape.

Termination

You can now use the OPER RDM STOP or OPER RDM FORCE commands to terminate RDM, release all RDM storage, and delete all RDM modules. For more information, refer to the *SUPRA Server CICS Connector Systems Programming Guide*, P26-7452.

Changes to Directory Maintenance

Changes to Directory Maintenance include the following:

- The station priority attribute in the Directory component description no longer specifies the priority of Directory Maintenance in the CICS environment. If you need to set the priority, you can define a priority for the transaction resource for Directory Maintenance.
- A REORGANIZE secondary key utility has been added which allows you to rebuild an existing secondary key tree structure without accessing the primary or related file.
- The DEACTIVATE/REACTIVATE secondary keys utility has been removed from Directory Maintenance.
- There are changes to Batch Directory Maintenance statements which may affect your SUPRA Server 1.3.5 Batch Directory Maintenance transactions. For more information on these changes, refer to the *SUPRA Server Directory Batch User's Guide*, P26-1261.
- +SIGNON statement must reference a user defined as a DBA in the directory.
- To blank a field you must use a +NULL statement defining the null character and use that character in the first position of the field to be blanked.

Review your applications to determine if these changes will affect your applications. For more information, refer to the *SUPRA Server Directory Batch User's Guide,* P26-1261.

PDM configuration and administration

Changes to PDM configuration and administration affect storage configuration and initialization.

For SUPRA Server, PDM applications and data areas can reside in extended storage.

For SUPRA Server, there are a number of storage and performance enhancements:

OS/390

Some control blocks have been allocated in extended storage to reduce memory requirements below the 16 MB line. Extended storage is indicated with the new XAMEM parameter of the CSIPARM file. For additional information about extended storage and the CSIPARM file, refer to the *SUPRA Server PDM and Directory Administration Guide,* P26-2250.

OS/390

The buffer cache facility improves PDM performance by using an extended storage cache to buffer PDM files. A new input file, PDMXIN, controls the initialization and operation of the buffer cache facility. For additional information about the buffer cache facility, refer to the *SUPRA Server PDM and Directory Administration Guide,* P26-2250.

- You can define a memory cleanup threshold in the Environment Description. When the amount of PDM memory in use exceeds this threshold, the PDM attempts to free noncritical task control blocks. For additional information about the memory cleanup threshold, refer to the *SUPRA Server Directory Batch User's Guide,* P26-1261, or the *SUPRA Server Directory Online User's Guide,* P26-1260.

- New attributes are available for suppressing task and system logging for particular files. For additional information, refer to the Environment Description, PDM file relationship in the *SUPRA Server Directory Batch User's Guide,* P26-1261, or the *SUPRA Server Directory Online User's Guide*, P26-1260.

- 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. New parameters in the CSIPARM file govern the operation of this feature. This performance feature is active unless the user specifically requests otherwise. The STMT interfaces will require three times the RDBUFFER value in additional memory in order to process. For additional information about the Read Ahead Buffering Facility, refer to the *SUPRA Server PDM and Directory Administration Guide*, P26-2250.

3

Migrating to SUPRA Server 2.7 from Series 80 TOTAL

When migrating from Series 80 TOTAL to SUPRA Server Release 2.7, the process includes the following areas:

- Migrating metadata
- Migrating your database files
- Migrating your programs

Skip this chapter if you are not now operating under Series 80 TOTAL.

Migrating metadata includes the following steps:

1. Expanding Directory files
2. Converting DDL transactions
3. Embedding schema CSISCH20 in your user schemas
4. Binding logical views in your user schemas

Migrating database files involves converting them to SUPRA Server native format via the File Conversion utility.

Migrating your programs includes these steps:

1. Tailoring user applications
2. Adding or updating the MANTIS Cluster
3. Upgrading Utility Control Language

After you have migrated to SUPRA Server Release 2.7, you will need to make some changes in your configuration and system administration procedures. These changes include:

- Changes in procedures and sample JCL
- Changes in CICS Connector configuration and administration
- Changes in PDM initialization and termination
- Using utilities with different file formats

Migrating metadata

Series 80 TOTAL uses a DBMOD to define a database schema. With SUPRA, the input to generate a TOTAL DBMOD is translated into a schema definition on the Directory. To translate DBMOD input, you must:

- Expand your Directory files
- Convert DDL transaction statements

Since these steps use no task or system logging, you should make backups of Directory and database files before, during, and after this upgrade. All Cincom-supplied sample members referenced in the migration process are supplied on the SUPRA Server installation tape in the MACLIB, JCLLIB, UCLCODE, and CSIPARM libraries **OS/390**, and in the base sublibrary **VSE**.

Expanding directory files

It is important that you make sure the supplied Directory files contain sufficient unused record space for the addition of all your TOTAL database descriptions. Depending on how large your TOTAL DBMODs are, you may need to expand the SUPRA Server Directory file size definitions in the bootstrap schema.

Before starting migration, we recommend that you run statistics to determine the amount of available space. If your Directory files are fairly full, you may want to consider expanding them prior to starting migration. Use sample TXJDSTAT to run Directory Statistics on your files. Refer to the *SUPRA Server PDM and Directory Administration Guide,* P26-2250, for instructions on how to expand the Directory files and how to estimate space requirements for adding new entities.

Converting DDL transaction statements

The SUPRA Server database is defined when you convert your TOTAL DDL transaction statements (DBMODs) into SUPRA Server Batch Directory Maintenance (BDM) transaction statements. When applied to the Directory, the converted DDL transaction statements create the internal schema. The internal schema defines your database and is similar to the CSISCH20 and BURRYSCH schemas, but without logical view external schema information.

To estimate the amount of space needed on the DDLSPUN output file for the generated Batch Directory Maintenance (BDM) transaction images, use the following DDL conversion information:

DDL input	Convert operation	Estimated no. of 80-byte BDM images produced
BEGIN-DATA-BASE-GENERATION	Add a Schema	8 (+1 for each line of long text)
DATA-BASE-NAME=	Add an Environment Description	9 (+1 for each line of long text)
CTLX:, OPTIONS:, JCL=	Ignored	None
LOGGING:	Add a System Log File description	27
IOAREA= in DBMOD prologue	Add a Buffer Pool	1 for each statement
BEGIN-*xxxxxx*-DATA-SET and physical statements*	Add a File with defaults, Change File	10 (+1 for each line of long text)
IOAREA= (within a file definition)	Add File/Buffer Pool relationship in Environment Description	3 (+1 for each line long text)
MASTER-DATA PRIMARY-DATA BASE-DATA	Add a BASE. Internal Record	1 (+1 for each line of long text)
Key, linkpath, data or filler element	Add a Physical Field	10 (+1 for each line of long text)
RECORD-CODE=	Add a record code Internal Record	8 (+1 for each line of long text)
CONTROL-INTERVAL, CYLINDER-LOAD-LIMIT, DISK-EXTENTS	Ignored	None
END-*xxxxxxx*-DATA-SET	End of file data	7
END-DATA-BASE-GENERATION	End of this schema	1

* The physical statements are not optimized or overridden as in DBGEN; they are taken literally. Change to match DBGEN listing.

In addition, your +SIGNON and +run options for RELATE and CHECK cause additions to the output file. The following list summarizes the number of BDM transactions produced by the DDL Convert utility:

- Each Schema—9 (+ *n* long text) for BEGIN and END
- Each LOGGING—27 for the SLOG File and Physical Field description
- Prologue IOAREAs—1 (+ *n* long text) for each pool
- Each Environment Description—9 (+ *n* long text) for the definition
- Each File—17 (+ *n* long text) for BEGIN and END, plus:

  for the Buffer Pool the file is related to
  (+ *n* long text) for the BASE Internal Record
  (+ *n* long text) for each Physical Field
  (+ *n* long text) for each Internal Record in a coded file (those not in DDL are not defined)

Using the Convert utility

To convert your DDL statements, run the Convert utility separately for each DDL statement which references a DBMOD. You can create a separate schema for each DBMOD or one large schema containing all of your DBMODs.

- To create multiple schemas, follow the Convert utility instructions in this section. Since you may have several DBMODs describing a number of files, the process must be performed multiple times.
- To create one schema containing all of your transaction statements, follow the Convert utility instructions in this section, but add each DBMOD definition to the first schema by using the run options in “Coding run options” on page 72.

The Convert utility works by reading a MACLIB member containing the DDL input statements used for DBGEN. Follow these steps:

1. Before you begin, make sure you have allocated enough space on the Directory files to hold all of the data describing your TOTAL files (see "Expanding directory files" on page 65). If the unused record space on the Directory is not sufficient, the third step (adding the schema definition) will abend.

2. Edit JCLLIB member TXJDDL80 to create the batch job. Refer to the symbolic DDLSPUN parameter which describes the output file.

   If you are using the +REPLACE DDL Convert run option anywhere in your input stream, add the Directory file DD statements to the job stream.

3. Prepare an input stream containing:

   a. A sign-on statement beginning in position 1:

      ```
      +SIGNONbCSI-DBA(blanks through position 39)CSI-DBA
      ```

      DDL Convert echoes the sign-on statement as the first transaction on your output file.

OS/390

Alternatively, you can concatenate MACLIB member DIRLODSO ahead of your output file as described in this section.

   b. A series of DDL Convert run option transactions (+keyword) if you do not want the defaults.

   c. A +SCHEMA-NAMEb/b/*xxxxxxxx* transaction. If you do not code a 1- to 8- character name, the name SCHEM001 is assigned.

   d. A +ENVDESC-NAMEb/b/*xxxxxxxx* transaction. If you do not code a 1- to 8- character name, the default name ENVDN001 is used.

   e. Your DDL statement stream. See the table under "Converting DDL transaction statements" on page 65 for a list of certain DBMOD Prologue and File Physical Description statements that will be ignored by the convert program, and for those that are treated differently than by DBGEN.

4. Submit the job and review the output. The return code should be zero. If it is not, look for syntax errors in the DDL statements that would not convert. The message 'CSMU852E DDL INPUT STATEMENT NOT RECOGNIZED. CARD IGNORED.' immediately follows the statement in error.

   Tailor the Convert utility output to ensure that the DDL statements have been properly converted. Add new BDM statements. For commands and parameters used in defining the BDM transactions, refer to the *SUPRA Server Directory Batch User's Guide,* P26-1261.

   Check the following areas of your converted transaction statements:

   a. Ensure that all DDL statements have been converted to the respective BDM (Batch Directory Maintenance) transactions add or change statements. This includes logical data, physical data and the naming data for entities.

   b. Verify that other transactions have been generated according to your +run options, including the +SCHEMA-NAME and +ENVDESC-NAME statements, environment description/file relationships, and schema consistency checks.

   c. In the generated environment description, the file-open-mode transaction for all files is READ, the PDM access mode is RDONLY, and the Directory access is READ.

   d. When you load the converted transaction statements to the Directory, the SUPRA Server defaults (already defined in your new Directory) will be used for all required fields in the batch transactions that do not contain a value. Many SUPRA Server required fields are not available from your DDL.

   e. For example, check for blank VSAM file values in the CG FI transactions and check for character or numeric data format for a physical field (element) in the AD PF transactions. Enter values if you do not want the default to be used.

   f. To find the default values that will be used in the various transactions such as FI and PF, refer to the *SUPRA Server Directory Online User's Guide,* P26-1260, or the *SUPRA Server Directory Batch User's Guide,* P26-1261.

   Your DDL values are taken literally, so check the generated physical descriptions of your files against their actual DBMOD values (record length, records-per-block, blocks-per-track, total logical records, etc.). You will find these in the CG FI transactions.

During the subsequent job of loading to the Directory, any file with a record length less than 21 is flagged as an error (see the discussion on file conversion in "Migrating database files" on page 84). If you need to add filler fields, you can adjust the Directory description of the file's physical fields, logical record length, and blocking now using the generated transactions, or you can adjust it later with Online Directory Maintenance.

Any variable file with coded records has been defined as a file with a BASE Internal Record and the various 'code' Internal Records. However, if the file has any record codes not defined in the DDL, they are also not defined to the Directory. You can add AD IR transactions if you want to define them.

If you did not have the LOGGING: statement in your DDL, you can add a System Log File definition to your schema. You can also add the Task Log File and Statistics File definitions (which were not needed in DDL). You can copy all three of these definitions from the MACLIB member DIRLODCM **OS/390** or TISDMSCH **VSE**. Then tailor the Task Log File definition to match its block size definition on the boot schema listing.

5. Make a copy of the generated READ environment description transactions, and rename and revise it for other modes. Generally, you need one with file-open-mode of NONE to use with SUPRA Server utilities (UNLOAD, LOAD, STATS, etc.), and another one with file-open-mode of SUPD to use for production.

   The environment descriptions will require a PDM access mode of UPDATE. The SUPD environment description needs Directory access of SUPD. You can revise the options to show statistics 'yes', task logging 'yes', and any desired system logging options. Also, add relate transactions for the files if 'yes' (along with the copied relates of database files).

OS/390 Review the DIRLODRI MACLIB member for a sample of an AD ED (add environment description) using logging and update mode.

VSE Review the TISDMNML procedure for a sample of an AD ED (add environment description) using logging and update mode.

6. Update the Directory using Batch Directory Maintenance. This process creates the internal schema. Follow these steps:

   a. Edit the JCL sample TXJBDIRM. This job uses CSIPARM member TXPBSUPD and does not use task logging.

OS/390

For the accessed MACLIB input member, comment out all TXI*xxxxx* members and use the name of the output file from the previous step. If you did not put a sign-on transaction in your output file, concatenate* the MACLIB member DIRLODSO ahead of the file.

   b. Submit the job and review output; the return code should be zero. Refer to the *SUPRA Server Messages and Codes Reference Manual (PDM/RDM Support)*, P26-0126, for error message descriptions.

      You may have messages denoting inconsistent Internal Records, File definitions, and Schema. You can correct this now with further BDM change transactions, or use Online Directory Maintenance display and change facilities. The schema will need to pass the consistency check before the PDM can use it (by your naming it in a CSIPARM file or UCL program).

   c. Back up your Directory files before continuing.

To concatenate MACLIB input members for a Batch Directory Maintenance job, use the sample JCL in one of two ways. Either copy all indicated input members into one member and name it in the symbolic MACLIB member parameter, or name the DIRLODSO member in the symbolic parameter and use SYSIN DD concatenation as shown in JCLLIB member TXJMIGDR.

Coding run options

You must code the run options for the Convert utility if you do not want to use the defaults. The default options are as follows (there is no default +SIGNON statement):

+PRINTBDM	+NOTEXT
+SIMULATE	+RELATE
+NOSEQUENCE	+CHECK
+ADD	

If you do not want the default options, place the appropriate alternate statements at the beginning of your DDL input deck. Alternatively, you can place these statements within the statements for one DBMOD before a MASTER, PRIMARY, or BASE-DATA statement and after the END-DATA statement. They are ignored if placed within the data portion of a file description.

Code all run option statements to begin with the + sign in position 1. Any other data on the same statement as a run option keyword are ignored.

1. Printing the output file

+PRINTBDM

Description *Optional*. Prints a list of every Batch Directory Maintenance image written to the output file. The printed image is preceded by the literal 'BDM CARD GENRTD'.

2. Determining output file space before writing

+SIMULATE

Description *Optional*. Executes the utility without writing to the output file. During simulated conversion, the number of potential BDM images is counted and printed at end of job. This gives you the size needed for allocating the file.

Consideration Although you do not write to the SYSPUNCH data set, the DD statement must be present.

3. Checking DDL sequence numbers

+SEQUENCE
+NOSEQUENCE

Description *Optional*. Determines whether to check the input DDL statements on positions 73–80 for ascending

Blank sequence numbers are ignored.

4. Adding new entities or replacing existing entities

+ADD
+REPLACE

Description *Optional*. Determines whether to use an add or replace function for entities following the option.

Options

ADD	An add for each entity is generated without checking the Directory for a duplicate. During the subsequent Directory Maintenance job, if the entity already exists on the Directory, a message is generated.
REPLACE	The Directory is examined before generating the BDM transactions for a schema or file. If they do not exist, an add is generated. If a schema does exist, naming data is generated; if a file does exist, a delete then an add for the file are generated.

Considerations

- To get maximum benefit from the REPLACE option, we recommend that you convert multiple DBMODs in separate jobs. Otherwise, the Directory would not yet contain the entities from the previous DBMOD.
- For the REPLACE option, DATBAS is used in the DDL Convert utility to invoke the PDM to read the Directory; therefore, your Directory files must be defined in the JCL.

5. Saving DDL comments as long or short text

+NOTEXT
+LONGTEXT
+SHORTEXT

Description *Optional*. Determines whether to save comments in the DBMOD input statements as Directory Short Text or Long Text transactions for an entity (edit output to ensure text is for correct entity).

Options

NOTEXT	No DDL comments are saved.
LONGTEXT	Comments following the DDL statement up to position 72, and on subsequent lines up until the next DDL statement, are saved as Long Text for the entity.
SHORTEXT	Comments following the DDL statement up to position 72 are saved as Short Text for the entity.

6. Creating file relationships

+RELATE
+NORELATE

Description *Optional*. Determines whether to generate the relationship from the environment description entity to a file in the DBMOD. If you use the default, all files are related to the preceding +ENVDESC environment description.

Options

RELATE	The relate for a file is generated.
NORELATE	The file is related to the generated environment description entity.

Consideration The naming data relationships are automatically generated (physical field/file, file/schema, etc.).

7. Generating consistency check commands

+CHECK
+NOCHECK

Description *Optional*. Determines whether to generate consistency checks for added or replaced schema data. If you use NOCHECK, you will still need to consistency check the schema before it can be used.

Naming DDL statements

You must follow the naming guidelines given below for DDL statement conversion.

1. Naming the schema

+SCHEMA-NAME { **New-schema-name** | **Existing-schema-name** }

Description *Optional*. Determines what schema will contain the subsequent DDL statement transactions.

Default SCHEM001 (up to SCHEM032 for multiple DBMODS in same run)

Options

New-schema-name	A new schema entity is generated.
Existing-schema-name	Use this option with the +REPLACE option when you wish to add to an existing schema (already on Directory from a previous run).

Considerations

- Naming data with this schema name is added to all subsequent DDL statements until another BEGIN-DATA-BASE-GENERATION statement is encountered.
- When you have multiple DBMODS (with different files) in the same run, you can duplicate the +SCHEMA-NAME for each to put them in the same schema. However, use different environment description names.

2. Naming the environment description

+ENVDESC-NAME { **New - name** | **Existing - name** }

Description *Optional*. Determines what environment description will contain the subsequent DDL statement transactions.

Default DATA-BASE-NAME=*xxxxxxxx* value, or, if not present, ENVDN001 (up to ENVDN200 for multiple DBMODS in same run)

Options

New-name A new environment description entity for this schema is generated.

Existing-name Use this option with the +REPLACE option when you wish to add to an existing environment description (already on Directory from a previous run).

Consideration When you name data or a relate with this environment description, the name you use is added to all subsequent DDL statements.

Embedding schema CSISCH20 in your user schemas

Schema CSISCH20 defines the SUPRA Server Directory files and the TIS/XA Software Selection Facility. These definitions allow you to:

- Run utilities against Directory files
- Run Directory reports
- Use the TIS/XA Software Selection Facility
- Run RDM reports
- Request metadata through the new Directory logical views using DBAID or SPECTRA

To perform the previous tasks with your user schemas, you will need to embed the CSISCH20 information in the schemas. Follow these steps to embed CSISCH20:

1. Edit the JCL sample TXJBDIRM.

2. Perform the following:

OS/390

Concatenate the following supplied MACLIB Directory transaction input members (see footnote in "Upgrading schema CSISCH20" on page 29):

DIRLODSO	Signs on.
DIRLODCO	Copies all schema CSISCH20 entities (except log files and metadata logical views) into another schema. It first deletes the entities from the target schema. You must edit to replace each occurrence of the names USERSCHM and USERED*nn*, BUF1 and BUF3. The comments in the member describe each variable name. If you wish your schema to contain only the Menu, or only Directory reports, you can comment out unwanted view and file entities.
DIRLDVDE	Deletes Directory views.
DIRLDVCO	*Optional*. Copies the logical views of Directory for metadata requests. You might want to use this only for selected schemas, both for security reasons and for Directory file space considerations. Edit this member to change the schema name.

VSE

Run the following members:

TISDMCO	Copies all schema CSISCH20 entities (except log files and metadata logical views) into another schema. It first deletes the entities from the target schema. You must edit to replace each occurrence of the names USERSCHM and USERED*nn*, BUF1 and BUF3. The comments in the member describe each variable name. If you wish your schema to contain only the Menu, or only Directory reports, you can comment out unwanted view and file entities.
TISDMDVD	Deletes Directory views.
TISDMDVC	*Optional*. Copies the logical views of Directory for metadata requests. You might want to use this only for selected schemas, both for security reasons and for Directory file space considerations. Edit this member to change the schema name.

3. If you run this job against a schema that has the previous version of the TIS/XA Software Selection Facility, you will need to concatenate MACLIB member DIRLDMNU **OS/390** or TISDMDMN **VSE** with the job.

4. Submit the job and review the output. The supplied transactions may cause an acceptable return code greater than zero due to resequencing in Variable Edits (VE) or deleting nonexistent entities.

5. Rerun the job for each schema as follows:

OS/390 Rerun the job for each schema you select, changing USERSCHM, USERED*nn*, BUF1 and BUF3 names in DIRLODCO and schema name in DIRLDVCO.

VSE Rerun the job for each schema you select, changing USERSCHM, USERED*nn*, BUF1 and BUF3 names in TISDMCO and schema name in TISDMDVC.

6. Back up your Directory files before continuing.

7. Check all entities in each schema on your Directory for consistency, using Directory Maintenance. (Specify Y for all the physical and logical entities to be checked.)

Binding logical views

Binding logical views improves performance on the initial access to the view. You can bind the following views supplied with SUPRA:

- TIS/XA Software Selection Facility
- Directory metadata
- Burrys

Since binding requires considerable space on the Directory DATA file (C$-D), you must weigh the advantage of enhanced speed when bound views are opened against the file space required to bind them. There are more than 70 Directory metadata views, which can fill up your Directory DATA file, especially if you bind them in multiple schemas.

OS/390

Use the following DBAID input members to bind supplied logical views:

DIRLODDB	Binds the two TIS/XA Software Selection Facility views. On the distributed Directory, these views are bound in two schemas, CSISCH20 and BURRYSCH.
DIRLDVAD	Binds the Directory metadata views.
BURRYSBD	Binds the Burrys views.

VSE

Use the following DBAID input members to bind supplied logical views:

TISDBBND	Binds the two TIS/XA Software Selection Facility views. On the distributed Directory, these views are bound in two schemas, CSISCH20 and BURRYSCH.
TISDMDVA	Binds the Directory metadata views.
TISBRBD	Binds the Burrys views.

To bind Burrys, TIS/XA Software Selection Facility, and Directory metadata views, follow these steps:

1. Back up your Directory first, in case binding fills the DATA file.

2. Edit JCL sample TXJBDAID to create the batch job. This job uses CSIPARM member TXPUTLOG (Burrys schema with task logging). To bind Burrys logical views in the Burrys schema, run the job with the input member BURRYSBD OS/390 or TISBRBD VSE. You can also follow steps 3 and 4 for the Burrys schema.

   To bind views in another schema, change the CSIPARM file from TXPUTLOG to the name of your CSIPARM file. You can bind views only in an active schema having DIR-ACCESS SUPD. You must also add your file JCL statements.

   If you want to bind in schema CSISCH20, create a CSIPARM file as follows:

   ```
   DIRECTORY=(SCHEMA=CSTASCHM,ENVDESC=CSTATLOG),
   REALM=(SCHEMA=CSISCH20,ENVDESC=CSIENVRG),END.
   ```

   The CSIPARM file shown here specifies a task logging environment. Be aware that the bound views in CSISCH20 will be copied hereafter when you copy the schema or selected logical views, or when you use DIRLODCO and DIRLDVCO OS/390 or TISDMCO and TISDMDVC VSE. This requires extra DATA file space.

3. Run the job once with MACLIB input member DIRLODDB OS/390 or TISDBBND VSE to bind the two TIS/XA Software Selection Facility views (first two in the member). Remember that these views are already bound in CSISCH20 and the Burrys schema on the distributed Directory.

4. Run the job again with MACLIB input member DIRLDVBD OS/390 or TISDBDVB VSE to bind the metadata logical views.

General considerations

- If you receive message DERR600CSTA650, the binding filled up your Directory DATA file. In this case, you can either unbind some views or you can go through the process of expanding the Directory.
- Refer to the *SUPRA Server PDM RDM Administration Guide (OS/390 &VSE),* P26-8220, for examples of using batch DBAID. Refer to the *SUPRA Server PDM Messages and Codes Reference Manual (PDM/RDM Support)*, P26-0126, for error message descriptions.
- To bind the views in a different schema, repeat the steps, changing the CSIPARM and file definitions.
- The only way to unbind is to delete and re-add a logical view. Use DBAID on the affected schema to LIST, REMOVE(Y), SAVE, and PERMIT (to reinstate the User relates) for each view you wish to unbind. Refer to the *SUPRA Server PDM RDM Administration Guide (OS/390 & VSE),* P26-8220, for information about binding views.

Migrating database files

You do not need to make any changes to your TOTAL database files to operate SUPRA Server Release 2.7. Release 2.7 supports TOTAL files, converted TOTAL files, and native SUPRA Server files. It is not necessary to define the file format to SUPRA. The SUPRA PDM determines whether the file is in TOTAL or SUPRA Server format when the file is opened.

You can convert your TOTAL database files to SUPRA Server native format using the File Convert utility. The File Convert utility does not affect your data records but converts free space records and the file lock record. For more information on the File Convert utility, see "Using the File Convert utility" on page 101.

You will also need to reformat your task log file before running task logging with SUPRA Server. The prerequisite is that your last PDM shutdown was unforced on the previous execution of the PDM. Otherwise, to reformat your existing task log file, you will need to allocate and format using JCLLIB member TXJFTLOG.

Before you begin, back up your database files in case of conversion problems requiring a re-execution. Also, be aware that once files are successfully converted to SUPRA Server native format, they cannot be reconverted to TOTAL format.

Follow these steps to convert your database files:

1. Increase file record length. The File Convert utility cannot convert a file having a record length less than 21 bytes to SUPRA Server native format. Under TOTAL, variable files are a minimum of 21 bytes but master files can be shorter. However, all SUPRA Server native files must be a minimum of 21 bytes.

   You must therefore increase the record size of any file less than 21 bytes before converting it. This involves re-DBGEN, file unload and load (TOTAL utilities), and either using the new DDL source for re-executing the DDL Convert step, or applying Directory Maintenance to the present Directory file description. A revised record length might also affect some applications.

All BDAM and ESDS primary and related files for SUPRA Server format must have a minimum record length of 21 bytes. VSAM KSDS files must be minimum 21 + key length + key displacement.

2. Edit JCL sample TXJCNV80 to create a batch job which runs the File Convert utility for TOTAL files.

3. Edit MACLIB input member TXICNV80 for input control statements. The input member should contain:

```
DBMOD=name
FILES=filenam1[,...,filenamn],END.
  .
  .
  .
FROM=TOTAL
TO=TIS-2-NATIVE
[IGNORE-FREE-CHECK]
DBMOD=name
FILES=filenam1[,...,filenamn],END.
  .
  .
  .
FROM=TOTAL
TO=TIS-2-NATIVE
[IGNORE-FREE-CHECK]
```

   For a full description of input control statements, see “Relating TOTAL utility statements to SUPRA UCL” on page 109.

   You can speed up the process of file conversion by including multiple DBMODs in the same job.

4. Submit the job and review the output; the return code should be zero. If the file conversion was successful, the next to the last message in the output is, “CSTU072I FILE CONVERT COMPLETED. ALL FILES CONVERTED.” Refer to the *SUPRA Server Messages and Codes Reference Manual (PDM/RDM Support),* P26-0126, for error message descriptions.

Considerations

- As the File Conversion utility executes, it prints a message as each file is started and completed. If a file starts but does not complete, you must restore the file from a system back-up, since conversion has only partially completed. Also, if a file is started and you receive a message that the file must be restored, restore the file from a system back-up. See “Using the File Convert utility” on page 101 for further file conversion considerations.

- At some point after your schema describing the files is consistent, you can use the SUPRA DBA utilities to print File Statistics on selected database files in your schema, and compare them with your TOTAL File Statistics. You can compare active records, linkages, chain migration, and so on.

Migrating programs

To migrate your programs from Series 80 TOTAL to SUPRA Server Release 2.7, you must:

1. Upgrade user applications
2. Upgrade exit programs
3. Link edit applications
4. Upgrade MANTIS TOTAL applications
5. Upgrade Utility Control Language programs

Updating user applications

SUPRA Server is compatible with your existing TOTAL application programs. However, there are several compatibility considerations to bear in mind when upgrading to SUPRA, including those for specific commands such as the SHOWX, for general features such as data list processing, and for error status codes such as *CCR.

The differences between TOTAL DML and SUPRA Server DML include:

- SUPRA Server may return the HELD status
- The TFUL status has been enhanced
- SUPRA Server may return the EMBR status
- SINON/SINOF statuses have changed
- The data list format has changed
- The *CODE=*xx* element has changed
- CLOSX has changed
- SETID/SETIV are not supported in SUPRA Server
- The CONTROL, LOCK, and MONITOR commands are no longer supported in SUPRA Server

This section presents two tables summarizing these and other compatibility considerations.

The following table alphabetically lists the compatibility considerations for general topics and specific commands.

Command or topic	Consideration
*CODE=*xx*	TOTAL allows *CODE=*xx* as the first element of a data list. For SUPRA, the *CODE=*xx* keyword must be preceded by the keyword **CODE** or *COMMON* if it is the first word in a data list.
CLOSX	The results of the CLOSX command with the file option ALL depend on the Directory access method in the active user environment description. You define the Directory access method through Batch or Online Directory Maintenance.
coded records	You cannot read records from a coded file or add records to the file unless all the record codes for the file are defined on the user schema.
data list	When upgrading from TOTAL, make certain that the optional commas in the data list *always* or *never* appear. If you place commas between some items in the data list, you must place commas between all items in the data list.
OPENX	The results of the OPENX command with the file option ALL. Depend on the directory access method in the active user environment description. You define the directory access method through Batch or Online Directory Maintenance. When upgrading from TOTAL, the optional commas in the realm must always or never appear.
RDNXT/ FINDX	SKIP in the RDNXT/FINDX related file qualifiers of BEGNSKIPSERIAL or *rrrr*SKIPSERIAL is no longer valid. SKIP is ignored by SUPRA Server 1.*x*, and processing is the same as BEGNb*bbb*SERIAL and *rrrrbbbb*SERIAL. Unused records are automatically skipped.
READR	In TOTAL, if you issue a READR against a related file whose primary file is closed, you always receive a status code of FNOP. For SUPRA Server 1.*x* you receive a status code of **** whenever the primary record's linkpath does not need to be accessed for the read.
SHOWX	When converting from TOTAL or TIS, you can take advantage of the new SHOWX command to change your error checking logic. Change it to issue a SHOWX each time you receive an error status code, except on a SINON.
SINON/ SINOF	For batch applications running in central operating mode, you may have to modify the SINON/SINOF logic. When an application performs a SINON to the SUPRA Server database, it is assigned a TASK ID which stays signed on until a SINOF command is issued. Any additional SINONs issued before a SINOF produce a return code of ACTV. Transferring control to another program or ending a program does not force a sign-off. Unless you issue a SINOF for a TASK ID, a dangling task results. To resolve a dangling task, you can use the RESTART= keyword in the CSIPARM file to generate a SINOF for the task. Alternatively, you can use the Task Management option online (under Interactive Services) to delete the dangling task. Refer to the *SUPRA Server PDM and Directory Administration Guide*, P26-2250, for information on the CSIPARM file and task management.
SETID/ SETIV	The SETID and SETIV commands are not supported in SUPRA.

TOTAL issues several 4-character status codes which are either not used in SUPRA or are changed to a different status code. The following table shows the old status code, a meaning, and the corresponding status code returned by SUPRA. The SUPRA status code is shown without a new meaning because a particular 4-character status code may be set by more than one command within SUPRA. Refer to the *SUPRA Server PDM Messages and Codes Reference Manual (PDM/RDM Support)*, P26-0126, for a complete description of SUPRA Server status codes.

The terminology for files has changed in SUPRA. A master file is now called a primary file, and a variable file is now called a related file.

Old error code	Meaning	Corresponding new error code
*BCR	Bad Cylinder Control Record. A bad cylinder control record was found. The record has been added to another cylinder.	No longer used
*CCR	Cylinder Control Record. Another task has processed a cylinder control record with an access mode of RECOVR.	No longer used
*FUL	File Full. There is no space available on this file for another add after this one.	FULL
*ICH	Invalid Chain. This status code has one of the following meanings: 1. An invalid RRN has been found in a synonym chain. 2. An RRN addressed a related record with a record code not defined for the specified linkpath.	ICHN
*IVL	Invalid Element. The specified element (data item) exists but is not valid for a particular record code.	IPAR
*LOD	File Loaded Beyond Limit. The file contains records beyond cylinder load limit. Subsequent addition may still be done until the physical load limit is reached (at which time a FULL status is returned).	No longer used
*NUL	Null Record. A blank record has been retrieved.	ICHN
*PNR	Possible Null Record. The record appears to be blank (probably a nondata record or a cylinder control record).	ICHN0042

Old error code	Meaning	Corresponding new error code
ACTV	Task Active. This task is already signed on.	FUNC
BKEY	Blank Key. A control key contains blanks.	BCTL
DEAD	The record is being used by another task.	HELD
DEAD	Waiting for this held record would cause a deadly embrace.	EMBR
ENTF	Data Name Not Found. A data name in the command parameter list has been incorrectly specified, or the requested element does not exist in the Database Descriptor Module.	IPAR
EUPD	Exclusive Update. For ADD-M, ADDVA, ADDVB, ADDVC, ADDVR, DEL-M, DELVD, WRITM and WRITV commands, another task has opened the file for exclusive update.	FNAV
EUPD	For ADDVA, ADDVB, ADDVC, ADDVR and DELVD commands, another task has opened for exclusive update of a file associated with a key in the data area.	FNAV
EUPD	For ADDVR and DELVD another task has opened for exclusive update of a file associated with a key in the object record.	FNAV
EUPD	For OPENX, you are attempting to open a file in either the SUPD or EUPD mode, and the file is open for exclusive update by another task.	OERR
EUPD	For CLOSX, you are attempting to close a file that is open for exclusive update by another task.	CERR
FATL	Fatal Error. An unrecoverable error occurred while maintaining linkpaths.	No longer used
FNOP	File Not Open. No task has opened the file named in the file parameter.	FNAV
FTYP	File Type Invalid. An attempt was made to process a primary file when the function (READV) is valid only for a related file (and vice versa).	FNTF
IARG	Invalid Argument. An argument field has improper punctuation.	IPAR
IEOJ	Invalid End of Job. Another task has reserved the file. The file was left open and locked.	No longer used

Old error code	Meaning	Corresponding new error code
ISIZ	I/O Area Too Long. The I/O area is longer than the maximum length allowed (OS/390 only).	No longer used
IUPD	Intended Update. An OPENX function could not be performed because the file has been opened with intention to update but has not been locked.	OERR
IVDL	Invalid Data List. A data list bound for one file is used in a command to access a different file.	IPAR
IVEL	Invalid Element. The specified data element name exists but is not valid for a particular record code. This could occur when reading down a base linkpath and encountering a record code for which a specified element is not valid.	IPAR
LOCK	Locked File. This status indicates that one of the following conditions exists:	
	1. Another copy has already locked the file.	OERR
	2. A program, previously using the database, aborted and left the file opened and locked.	OERR
	3. There was an attempt to update a file opened in READ mode.	FNAV
LSIZ	Log Size Error. The QMARK or MARKL data-area size exceeds the size of the log record defined in the Database Descriptor Module (DBMOD).	No longer used
MFNF	Master File Not Found. The master file was not found in the DBMOD.	FNTF
MLNF	Master Link Not Found. The file (as specified by the mmmm in the linkpath (*mmmm*LK*xx*) parameter) is not a master file.	FNTF
NACT	Not Active. The task is not active, or it has signed off and has not signed on again.	FUNC
NATH	Not Authorized. For the ADD-M, ADDVA, ADDVB, ADDVC, ADDR, DEL-M, DELVD, WRITM and WRITV commands, the file parameter specifies a Directory file. Only Cincom products are permitted to update Directory files. For the CLOSX and OPENX commands, the file specified in the realm parameter is a Directory file or the CTLX file.	No longer used

Old error code	Meaning	Corresponding new error code
NLOG	No Logging. Logging is not active so the system logging command cannot be executed.	FUNC
NMLK	No Master Link. The linkpath contained in the variable file was not found in the corresponding master file.	IVLK
NOIO	No I/O Buffer Available. No I/O buffer was allocated for a file.	No longer used
NSMR	No Secondary Master Record. The master record does not exist for one of the secondary linkpaths.	MRNF
READ	For ADD-M, ADDVA, ADDVB, ADDVC, ADDVR, DEL-M, DELVD, WRITM and WRITV commands, the task was attempting to update a file that is open in READ mode.	FNAV
READ	For ADDVA, ADDVB, ADDVC and ADDVR commands, the task was attempting to update a master file associated with a key in the data area but the file is open in READ mode.	FNAV
READ	For ADDVA, ADDVB, ADDVC, ADDVR and DELVD commands, the task was attempting to update a file associated with the linkpath parameter, but the file is open in READ.	FNAV
READ	For ADDVR and DELVD commands, the task was attempting to update a master file associated with a key in the record, but the file is open in READ mode.	FNAV
READ	For an OPENX (IUPD, SUPD, EUPD), updates could not be performed because the file is open in READ mode.	OERR
RSRV	Reserved. An open SUPD was not performed because the file is already opened for SUPD.	OERR
SHRE	File Already in Shared Mode. An OPENX for EUPD cannot be performed because the file is already open for SUPD.	OERR
SNTF	Segment Not Found. A segment (data) name is incorrect or does not exist in DBMOD.	IPAR
VFNF	Variable File Not Found. The file named in the file parameter was misspelled or does not exist in the DBMOD.	FNTF

Old error code	Meaning	Corresponding new error code
VLNF	Variable Link Not Found. The linkpath named in the linkpath parameter is misspelled or does not exist in the DBMOD.	IPAR
Not Used	For a SINON, the PDM is already connected to as many interfaces as possible or as many tasks as possible.	TFUL
Not Used	For a SINON, the PDM is already connected to an interface with the same name.	DUPT OS/390
Not Used	The CSIPARM file is incorrect.	PARM
Not Used	For a SINON, when the database has been restarted from the task log file.	RSTR
Not Used	There is an error in processing an OPENX.	OERR
Not Used	There is an error in processing a CLOSX.	CERR

To maintain compatibility for TOTAL statuses and DML, use the PDM and CICS Connector exit programs supplied.

For the PDM interface precommand, postcommand and DATBASXT exits, use the supplied exit programs CSTK0001, CSTK0002, and CSTK0003. For more information on these exits, refer to the *SUPRA Server PDM and Directory Administration Guide,* P26-2250.

You should also use the preprocess and postprocess exit points in the CICS Connector exit CSTXUSRX, which replaces the modified TOTINT and CSTXDBXT exits. (See also the TOTC parameter in the CSTXOPRM macro.) For more information on the CICS Connector exits, refer to the *SUPRA Server CICS Connector Systems Programming Guide*, P26-7452. For more information on the CSTXOPRM macro, refer to the *SUPRA Server PDM and Directory Administration Guide,* P26-2250.

The following exit programs are provided with SUPRA to help provide parameter list and status compatibility:

- The PDM interface precommand exit program CSTK0001. This exit program is called once per DML command, before the command is passed to the PDM. You can use this exit to modify parameters in the DML.

  As furnished, this exit program will not perform any function. You must remove branch statements to perform the functions that are currently coded or insert your own code to alter the incoming DML command. The routines coded are for compatibility with TOTAL and perform the following four functions:

  - Verify and correct the use of RLSE for the end parameter of certain DML commands.

  - Correct data lists generated by Cincom products that are invalid with SUPRA.

  - If the function is a SETID or SETIV, update the status to successful completion (stars) and return the function to the application.

  - If the function is RESET or COMIT, change the third parameter to an acceptable value for SUPRA.

- The PDM interface postcommand exit program CSTK0002. This exit is called once per DML command, after the command is completed by the PDM. You can use this exit to maintain status compatibility.

  As furnished, this exit does not perform any function. You must remove branch statements to perform the functions that are currently coded or insert your own code to alter the incoming DML command. The routines coded are for compatibility with TOTAL and perform the following four functions:

  - If the status is ICHN and this is a READV or READR command, and the end parameter is RLSE, change the status to *PNR.

  - If the status is MRNF and the command is ADDVA, ADDVB, or ADDVR OF DELVD, and the first four bytes of the extended status do not equal the first four bytes of the linkpath name, set the status to NSMR.

  - If the status is ACTV and the command is SINON, change the status to ****.

    If the status is NACT and the command is SINOF, change the status to ***.

  - If the status is not in the internally coded table to suppress, print the extended status on the job log.

Updating exit programs

The DATBASXT exit is available under SUPRA, as it is in Series 80 TOTAL.

The following Series 80 exits are obsolete in SUPRA:

- The Task Detach exit CSTKCEXT. This exit functionality is implemented in the Task Detach user exit point in CSTXUSER. Use this exit point to detect tasks that have not committed resources.
- The Emergency Restart exit CSITBEXT.
- The Dynamic Backout exit CSIDBEXT.

The following exits are new for SUPRA:

- The PDM interface precommand exit CSTK0001. (See "Updating user applications" on page 86 for a description of this exit.) A sample exit program is provided with SUPRA.
- The PDM interface postcommand exit CSTK0002. (See "Updating user applications" on page 86 for a description of this exit.) A sample exit program is provided with SUPRA.

Link editing applications

You need to relink applications to include the SUPRA Server version of DATBAS. If relinking is prohibitive, you may create an alias of TOTALINT for the module CSTEDBMI or copy and rename CSTEDBMI as TOTALINT. Your existing DATBAS loads a module named TOTALINT but the relink will cause it to load CSTEDBMI instead.

For CICS applications, you must link edit the applications with the SUPRA Server supplied DATBASC.

Do not just rename CSTEDBMI without copying it first since various SUPRA Server components use this module name. Any maintenance applied to CSTEDBMI must also be applied to the aliased or renamed TOTALINT module.

Updating MANTIS TOTAL applications

MANTIS can access many different file structures, including TOTAL files. You can use the MANTIS TOTAL Logical View Facility to prototype logical views before using RDM. Use the following coding conventions to facilitate migration of TOTAL applications to RDM:

- Group all TOTAL views at the beginning of the program, possibly with a comment.
- Isolate file navigation from the program's main logic. To do this, do not code all the GET statements in the program mainline. Instead, code a DO statement to a subroutine that navigates the database and accesses the required data (simulating RDM navigation).
- Test the TOTAL views and navigation subroutines. Your Directory DBA then uses this as a prototype for building SUPRA RDM logical views.
- When SUPRA RDM logical views are created, the DBA informs the programmers of their authorized logical views. The programmers then upgrade their programs to:
  - Remove the TOTAL statements and replace them with the appropriate VIEW statements.
  - Replace the DO statement with the appropriate RDM statement: GET, UPDATE, INSERT, or DELETE.
  - Retest the program.

Updating utility control language programs

Series 80 TOTAL includes a series of file utility programs for which you code control card input. In SUPRA, there is one utility program driver, and you select each utility and its parameters by using Utility Control Language (UCL) programs.

"Relating TOTAL utility statements to SUPRA UCL" on page 109 compares the two methods to help you relate TOTAL coding techniques to SUPRA UCL. For more information on SUPRA Server utilities, refer to the *SUPRA Server DBA Utilities User's Guide*, P26-6260.

Changes to Directory Maintenance

Changes to Directory Maintenance include the following:

- The station-priority attribute in the Directory component description no longer specifies the priority of Directory Maintenance in the CICS environment. If you need to set the priority, you can define a priority for the transaction resource for Directory Maintenance.
- A REORGANIZE secondary key utility has been added to allow you to rebuild an existing secondary key tree structure without accessing the primary or related file.
- The DEACTIVATE/REACTIVATE secondary key utility has been removed from Directory Maintenance.
- There are changes to Batch Directory Maintenance statements which may affect the Batch Directory Maintenance transactions.
- The +SIGNON statement must reference a user defined as a DBA in the Directory.
- To blank a field, you must use a +NULL statement to define the null character. Use the null character in the first position of the field to be blanked.

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

Making changes to configuration and system administration

You must consider changes to your system configuration and system administration procedures in the following areas:

- Procedures and sample JCL
- CICS Connector configuration and administration
- PDM initialization and termination
- Using utilities with different file formats

Procedures and sample JCL

SUPRA Server provides catalogued procedures and sample JCL. These procedures and samples contain documentation on how to use them. You should use the new JCL samples or upgrade your existing JCL to reflect the changes. If you upgrade your JCL, the following considerations apply:

- Remove all file JCL statements for the CTLX file. SUPRA Server does not use it. Instead, add a CSIPARM file JCL statement to each job that uses the Directory files. We have supplied various CSIPARM members for the upgrade, but you need to create your own for production. Refer to the *SUPRA Server PDM and Directory Administration Guide*, P26-2250, for information about the CSIPARM file.
- Add the Directory file JCL statements to all batch jobs that once used TOTAL. Add or revise Task Log, System Log, and Statistics file JCL statements, as applicable.
- Review the JCL sample TXJBDIRM as a guide for the required load libraries.

CICS Connector configuration and administration

The SUPRA Server CICS Connector differs from the TOTAL CICS interface in the following areas:

- Defining options
- CICS resource definition
- Storage allocation and requirements
- Initialization
- Commands
- Termination

For more information on these areas, refer to the *SUPRA Server CICS Connector Systems Programming Guide*, P26-7452.

PDM initialization and termination

SUPRA Server provides the ability to automatically open files during PDM initialization. You control this ability via Directory Maintenance, File/Environment Description Relate function.

SUPRA Server automatically closes open files at PDM termination.

Consideration

Files that are open to the PDM must not be updated, reloaded, or deleted by anything outside the PDM.

Using utilities with different file formats

You can execute all SUPRA Server utilities with SUPRA native files. However, you cannot execute all of them with TOTAL or SUPRA Server converted files. If you use the following utilities with a TOTAL or SUPRA Server converted file, the results are unpredictable, and you may damage the file:

- Expand
- Modify
- Unlock

If you use the following utilities with a TOTAL or SUPRA Server converted file, the results are unpredictable, but you will not damage the file:

- File Statistics
- Print
- Review

If you use the following utilities with a Series 80 file, the results will be as follows:

- Format—Creates an empty SUPRA Server native file (not a TOTAL or SUPRA Server converted file).
- Version 1 Load—Creates a SUPRA Server native file. TOTAL cannot use the file, and the Unload function cannot convert it back to TOTAL format. To process TOTAL or SUPRA Server converted files, use the Version 2 Unload and Load functions.

You can run the following utilities with a TOTAL file or a SUPRA Server converted file:

- Version 1 Unload (The Version 1 Load utility creates only SUPRA Server native files)
- Version 2 Load
- Version 2 Unload
- Version 2 Insert Linkpath

A

Using the File Convert utility

This appendix provides additional information for using the File Convert utility when upgrading from Series 80 TOTAL to SUPRA. The File Convert utility revises free space records and the file lock record. It is intended for use only during the conversion process and is not needed once all files are in SUPRA Server native format.

Delaying file conversion

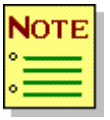
You can delay conversion of your database files by using the SUPRA Server file compatibility feature for TOTAL. If you do delay database file conversion, performance is degraded and you cannot take advantage of SUPRA Server features

Do not mix SUPRA Server and non-SUPRA Server format database files without extremely close control, as the integrity of the files can be jeopardized. The compatibility feature is not intended for mixed files, but to allow you to convert files gradually, reducing the initial impact of upgrading to SUPRA.

File conversion options

The File Convert utility provides four conversion options which are detailed below. In order to use options 2, 3, or 4, your TOTAL DBMOD must be available and must match the files being converted. The File Convert utility operates with one or more DBMODs.

- Convert from TOTAL to SUPRA Server format, *not* allowing for conversion back to TOTAL file format. The resulting SUPRA Server native files do not contain CCRs, so they cannot be converted backward.



This method is the one we recommend for file conversion.

```
TOTAL    -------------->SUPRA Native
```

- Convert TOTAL files to SUPRA Server converted format, allowing for conversion back to TOTAL file format. When converting to converted format, a variable or related file must have a minimum of 25 bytes record length.

```
TOTAL     ------------->SUPRA Converted
```

- Convert from SUPRA Server converted format to SUPRA Server native, not allowing for conversion back to converted format or to TOTAL format. Again, the resulting SUPRA Server native files cannot be converted back to TOTAL format since they have no CCR information.

```
SUPRA Converted  ------>SUPRA Native
```

- Convert a converted file back to TOTAL format.

```
SUPRA Converted  ------>TOTAL
```

Control record format

The control records for the File Convert utility use the following rules:

- Only one keyword can appear on each control record. It can begin anywhere on the record.
- The keyword parameter must immediately follow the keyword (no blanks).
- The first blank after the keyword and parameter terminates processing of the record.
- If a keyword occurs more than one time in the input stream, the last occurrence is used. An exception to this is the FILES statement.
- Code an asterisk (*) in column one to indicate a comment.
- Columns 73–80 are ignored.
- The TIS-2-keyword implies SUPRA.

[DBMOD=*dbmod*]

FILES=*filename1*[,...,*filenamen*],END.

$$\mathbf{FROM} = \left\{ \begin{array}{l} \mathbf{TOTAL} \\ \mathbf{TIS\text{-}2\text{-}CONV} \end{array} \right\}$$

$$\mathbf{TO} = \left\{ \begin{array}{l} \mathbf{TOTAL} \\ \mathbf{TIS\text{-}2\text{-}NATIVE} \\ \mathbf{TIS\text{-}2\text{-}CONV} \end{array} \right\}$$

[IGNORE-FREE-CHECK]

DBMOD=*dbmod*

Restriction *Required* for conversion from or to TOTAL.

Description *Conditional*. Names the DBMOD of the files you are converting (TOTAL itself is not needed).

Format 8 characters

Consideration You can process more than one DBMOD in a run.

FILES=*filename1*[,...,*filenamen*],END.

Description *Required*. Names the files to be converted.

Format 4-character file names, separated by commas, terminated with ,END.

Considerations

- The FILES statement cannot cross record boundaries. The ",END." keyword must appear on the same record with no intervening blanks. However, you can code multiple FILES statements, for example:

```
FILES=PRI1,PRI2,END.
FILES=REL1,END.
FILES=REL2,END.
```

- Files going to native format must be at least 21 bytes long; those variable/related files going to converted format must be at least 25 bytes.

FROM = { TOTAL | TIS-2-CONV }

Description *Required*. Specifies the current file structure of files to be converted.

Options

TOTAL	File is in TOTAL format.
TIS-2-CONV	File is in SUPRA Server converted format (was previously converted from TIS 1.*x* or TOTAL to SUPRA Server converted).

Considerations

- Note that you cannot convert using FROM=native to any other format.
- See Considerations under TO parameter for valid combinations of FROM and TO parameters.

TO = { **TOTAL** | **TIS-2-NATIVE** | **TIS-2-CON** }

Description *Required*. Specifies the type of file structure you want when execution is complete.

Options

TOTAL	File will be TOTAL format.
TIS-2-NATIVE	File will be SUPRA Server native format.
TIS-2-CONV	File will be SUPRA Server converted format (able to be reconverted backward, or forward to native).

Consideration Valid combinations for the FROM and TO parameters are:

FROM=	TO=
1. TOTAL	TIS-2-CONV
2. TOTAL	TIS-2-NATIVE
3. TIS-2-CONV	TIS-2-NATIVE
4. TIS-2-CONV	TOTAL

If you use combination 1, 3, or 4, you must maintain your TOTAL DBMODs so the files being converted match after conversion.

If you use combination 2, you need not maintain the DBMOD or Directory after the conversion.

IGNORE-FREE-CHECK

Description *Optional*. Directs the utility to bypass the check for free space records of variable/related files. The utility checks for all blanks beginning with position 5 in each record.

Consideration Use this keyword with caution. Code it when:

- You have attempted a conversion and received message CSTU086E, which means that the free space check will not work.
- You are positive that some nonblank data exists beyond byte 4 of any in-use related file record. This could conceivably happen for a related (TOTAL variable) file having a key 1–4 bytes long, followed by a blank linkpath and blank data fields. Such a file cannot be converted properly, since free space records cannot be detected.

General considerations

- As the File Convert utility executes, it prints a message as each file is started and completed. If a file starts but does not complete, you must restore the file from a system backup, since conversion will be only partially completed. Also, if a file is started and you receive a message that the file must be restored, restore the file from a system backup.
- When using the SUPRA Server file compatibility feature for TOTAL variable files, higher rates of chaining may occur across cylinder boundaries. Use the File Statistics function of the SUPRA DBA Utilities at regular intervals to monitor chaining rates for compatible files. If problem areas appear, use TOTAL unload and reload utilities for more efficient file organization.

File Convert utility errors

The information and error messages you may receive during utility execution are explained in the *SUPRA Server Messages and Codes Reference Manual (PDM/RDM Support)*, P26-0126. If control card coding errors occur, the first 15 bytes of the incorrect keyword and its parameter are displayed. The keyword is displayed even when the error is only in its parameter. If a file list contains an invalid file name, only the incorrect file name is displayed.

You may receive the following return codes after execution of the File Convert utility:

0 All files converted successfully.

4 Some files converted successfully; others were not started.

8 No files were converted.

12 Conversion of some files was started but could not be completed; restore the files from system backup.

B

Relating TOTAL utility statements to SUPRA UCL

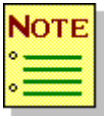
This appendix acquaints you with the SUPRA DBA Utilities Component Language (UCL). Under Series 80 TOTAL, you had a series of file utility programs for which you coded control card input. For SUPRA, there is one utility program driver, for which you code control and selected utility information using UCL statements. This appendix compares the two methods, but does not actually teach UCL. Refer to the *SUPRA Server DBA Utilities User's Guide*, P26-6260, for detailed descriptions and usage of UCL and the associated utilities.

Briefly, you write a UCL program by coding a CONTROL statement and associated parameters followed by one or FUNCTION portions. The functions supersede the old Unload, Load, Format, File Statistics, and so on.

In the CONTROL section, you name a schema and environment description instead of a DBMOD. The bootstrap entities are taken from the CSIPARM file named for the run. The CSIPARM file REALM parameter is ignored unless the FUNCTION is EXPAND. A skeleton UCL program looks like this:

```
CONTROL (BEGIN)
   ENV-DESC (user environment description)
   SCHEMA (user schema)
     .   output listing parameters
     .   sort program if unload, file stats
     .   output data-file description if unload
     .   system log file description if Recover, Restore, Log
  Print
   (the three log FUNCTIONs require omission of above schema and
  env-desc)
       FUNCTION (utility)
.
         .   parameters
.
       FUNCTION (utility)
.
         .   parameters
.
CONTROL (END)
```

The following tables compare Series 80 TOTAL utility statements to SUPRA UCL. The old utility statements are presented at the left and the comparable new UCL statements are presented at the right.



Do not attempt to code a UCL program entirely from the tables. They are only a guide for familiarization.

FORMAT, EXPAND, RESET

Format utility	CONTROL UCL	FUNCTION UCL
⊬FORMAT⊬dbmod	ENV-DESC (*ed-name*) SCHEMA (schema)	
ffff[,...,*ffff*]END.		FUNCTION (FORMAT) FILE (ALL) or multiple FILE (name) or FILE (*ffff*[,...,*ffff*])
⊬EXPAND⊬dbmod	ENV-DESC (*ed-name*) SCHEMA (new schema)	
vvvv[,...,*vvvv*]END.		The CSIPARM REALM names the old schema. FUNCTION (EXPAND) FILE (ALL) or multiple FILE (name) or FILE (*ffff*[,...,*ffff*])
⊬RESET⊬dbmod	Not supported	

INSERT LINKPATH

Insert utility	CONTROL UCL	FUNCTION UCL
⊬INSERT⊬	There is no Insert function in UCL. Insert functions are performed by SUPRA Server 2.2 utilities. You can, however, use the UCL MODIFY function as follows:	
DBMOD=dbmod	ENV-DESC (*ed-name*) SCHEMA (new schema)	
		FUNCTION (MODIFY), FILE (one primary file) QUALIFIER (DIRECT) KEY
FILES=({**ALL.** | **mmmm**[**...mmmm**]**.**}**)**		
		(key-value) RECORD (ALL) ELEMENT (linkpath) Change the linkpath to rrns or blanks
,CLEARLKS=(*mmmm*LK*xx*[...mmmmLKxx].)		
,END.		DATA (.*nnnnnnnn*END.)

UNLOAD

Unload utility	CONTROL UCL	FUNCTION UCL
OLD-DBNAME=dbmod	ENV-DESC (*ed-name*) SCHEMA (old-schema)	
,NEW-DBNAME=dbmod	Obsolete	
SORTNAME=name	Not supported	
[TEST=YES]	Not supported	
	DATA-FILE description SUMMARY-DATA (*xxxx*) LIST (ALL or NONE)	
V-E:*vvvv*[...*vvvv*]END.		FUNCTION (UNLOAD)
S-E:*mmmm*[...*mmmm*]END.		FILE (ALL) or multiple FILE (name)
/*	Obsolete	
//PARM DD *	Obsolete	
*vvvv*LINKPATH=*mmmm*LK*xx*		LINKPATH (*pppp*LK*xx*)
,PRESERVE = [NO / YES]		,PRESERVE = [NO / YES]
		CLEAR-LINKS (LK*xx*[LK*xx*]) RRN-RANGE (low to high) CRITERIA (argument)
,RC=*xx*		**RECORD(**[**xx** / **ALL**]**)**
ffff {**ALL.END.** / **elem**[**...elem**]**END.**}		**ELEMENT(**{**ALL** / **elem[...elem,*FILL =]**}**)**
*mmmm*BLANK-LINKS=LK*xx*[... LK*xx*]END.		See CLEAR-LINKS above
/*	Obsolete	

LOAD

Load utility	CONTROL UCL	FUNCTION UCL
DBNAME=dbmod,CORD=*nn*		ENV-DESC (*ed-name*)
	SCHEMA (new/same-schema)	
SORTNAME=name	SORT (name)	
SORTCORE=*nnnn*	Obsolete	
MAXKEY=*nnn*	Obsolete	
V-E:*vvvv*[...*vvvv*]END.		FUNCTION (LOAD)
S-E:*mmmm*[..*mmmm*]END.		FILE (ALL) or multiple FILE (name) LINKPATH (*pppp*LK*xx*) CLEAR-LINKS (LK*xx*[LK*xx*]) SEQUENCE ( )
		RECORD({**xx** / **ALL**} **)**
ffff {**ALL.END.** / **elem[…elem]END.**}		**ELEMENT(** {**ALL** / **elem[,…,elem,*FILL =]**} **)**
/*	Obsolete	

PRINT, MODIFY

Print/modify	CONTROL UCL	FUNCTION UCL
For Print:		
SUMMARY-DATA (*xxxx*)	SUMMARY-DATA (*xxxx*)	
*HDR=option LIST( )	LIST( )	
HEADER (option)	HEADER (option	
DATA-FORMAT (*xxxx*)	DATA-FORMAT (*xxxx*)	
SUPPRESS ([REFER]	SUPPRESS ([REFER])	
[SPACE] [ELEMENT])	[SPACE] [ELEMENT]	
*MAX=*nn*		MAXIMUM (*nn*) under FILE
*START=*nn*		RRN-RANGE (low to high)
⊬PRINT⊬DBMOD=dbmod	ENV-DESC (*ed-name*)	under FILE
SCHEMA (schema)	SCHEMA (schema)	FUNCTION (PRINT)
,FILE=one-file		FILE (ALL) or multiple FILE (name)
,QUAL=(value)		**QUALIFIER** { **DIRECT** / **SERIAL** / **SEQUENTIAL** }
		Sub-parms for KEY, rrn, linkpath
,ARG=(value.)		CRITERIA (argument)
,DATALIST=(value.)		**RECORD(** { **xx** / **ALL** } **)**
.END.		**ELEMENT(** { **ALL** / **elem[…elem]** } **)**

Print/modify	CONTROL UCL	FUNCTION UCL
For Modify:		
	SUMMARY-DATA (*xxxx*)	
*HDR=option		
	LIST({ **ALL** / **NONE** / **BEFORE AFTER** } **)**	
	HEADER (option)	
	DATA-FORMAT (*xxxx*)	
	SUPPRESS ([REFER]	
	[SPACE] [ELEMENT])	
*MAX=*nn*		MAXIMUM (*nn*) under FILE
*START=*nn*		Not supported
⁄MODIFY⁄DBMOD=dbmod	ENV-DESC (*ed-name*)	
	SCHEMA (schema)	
		FUNCTION (MODIFY)
,FILE=one-file		FILE (name)
,QUAL=(value)		**QUALIFIER(** { **DIRECT** / **SERIAL** / **SEQUENTIAL** } **)**
,ARG=(value.)		Sub-parms for KEY, rrn, linkpath CRITERIA (argument)
,DATALIST=(value.)		**RECORD(** { **XX** / **ALL** } **)**
		ELEMENT (elem[,...elem])
,DATAAREA=(value.)		DATA (.data-stringEND.)
,END.		

FILE STATISTICS

File Stats utility	CONTROL UCL	FUNCTION UCL
DBNAME=dbmod	ENV-DESC (*ed-name*)	
	SCHEMA (schema)	
SORTNAME=name	SORT (name)	
		FUNCTION (UNLOAD)
FILE=one-file		FILE (ALL) or multiple FILE (name)
,LINKS=*mmmm*LK*xx*		LINKPATH (*pppp*LK*xx*[,...,*pppp*LK*xx*])
[,...,*mmmm*LK*xx*]		
	SUMMARY-DATA	
	(**ALL** / **NONE** / **BASE FUNTION** / **CUMULATIVE**)	
STATS = { **ALL.** / **xxxx[,...,xxxx]** }		STATISTICS
		(**ALL** / **BASE** **SIZE** **LINK** **CHAIN** **CODE**)

PERFORMANCE STATISTICS

Perf Stats utility	Not UCL	Not UCL
No control card input; execute program CSIRPORT		No control card input; execute TXJPSTAT JCL Sample for Execution Statistics program

Unlock/Review utility	CONTROL UCL	FUNCTION UCL
⁄UNLOCK⁄dbmod	ENV-DESC (*ed-name*)	
REVIEW	SCHEMA (schema)	
		FUNCTION (UNLOCK) or (REVIEW)
FILE = {**ALL.** / ***ffff*[,...,*ffff*]**}		FILE= ALL. or multiple FILE (name) or FILE (*ffff*[,...,*ffff*])

RECOVER, RESTORE, LOG PRINT

Recover utility	CONTROL UCL	FUNCTION UCL
⊮⊮LOGDATA⊮		
UNIT=TAPE	**LOG – FILE(** { **LOGFILE** / **name** } **)**	Use this CONTROL with choice of FUNCTION on next page
,NAME = { **TLOG** / **name** }	**DEVICE(** { **TAPE** / **DISK** / **VSAM** } **)**	
	ACCESS – METHOD({ **BSAM** / **BDAM** / **ENDS** } **)**	
,STATISTICS=NO or YES		See under FUNCTION ( )
,OPEN = { **ALL** / **DYNAMIC** / **LISTffff[,...ffff]** }		Not supported See under FUNCTION ( ) See under FUNCTION ( )
,SYSTEM=*xxxxxxx* ,STARTVOL=n ,DATE=*xxxxxxxx* ,TIME-CHECK=NO/YES ,SERIAL-CHECK=NO/YES	All not needed	
⊮LOGPRINT⊮	LIST with choice(s)	Use this CONTROL with choice of FUNCTION on next page
ALLRECS=NO/YES ,IMAGES=NO/YES ,FUNCTIONS=NO/YES ,SYSFUNCS=NO/YES COMITS=NO/YES	(ALL or NONE) ([BEFORE] [AFTER] [FUNCTION] [SYSTEM] , [SYSTEM]	ALL includes new BLOCK & DESCRIPTION log records. These choices are printed during the log file scan phase.
	[APPLIED-IMAGES])	For application phase if ALL not used.
,TYPE=HEX/CHAR	DATA-FORMAT	
	([CHAR] [HEX])	
,BEGIN=n		See Key or Rrn-Range in FUNCTION

Recover utility	CONTROL UCL	FUNCTION UCL
,END=n		See Key or Rrn-Range in FUNCTION
⁄⁄RUNDATA⁄⁄		
DUMP=		
,PHASE=		
	LOG-DATA information above.	
		FUNCTION({**RECOVER** / **RESTORE** / **LOGPRINT**} **)**
TO = {**LOGBEGIN / END** / **LASTCOMIT** / **LASTQUIET** / **QUIETnnnn**}		**STATE(** {**LOG – BEGIN / –END** / **LAST – COMMIT** / **LAST – COMMIT**} **)**
		Not supported
		OPEN – FILE({**INITIAL** / **DYNAMIC**} **)**
		STATISTICS({**ALL** / **BASE** / **NONE**} **)**
		FILE (ALL) or multiple FILE (name)
		[KEY-RANGE (low to high)] [RRN-RANGE (low to high)]
,REPORTING = {**YES** / **ALL** / **READ** / **NO**}	There is no separate application phase reporting: except use of LIST on previous page having ALL or having APPLIED-IMAGES as a choice.	

Index

D

E

F

I

L

M

N

O

P

R

S

T

U

V

Y