UNISYS | Securing Your Tomorrow®

ClearPath OS 2200

Software Products Installation Guide

ClearPath OS 2200 Release 19.0

Contents

Appendix A. Setting Up the Security Environment for Object Module Subsystems

Appendix B. Guidelines for Reloading Subsystems Provided with Unisys Software

Tables

Section 1
Overview

This guide describes the procedure for installing the Software Library Administrator for ClearPath OS 2200 (SOLAR) onto a Dorado Series server. It also describes the procedure for installing OS 2200 software products distributed on Unisys conditioned software package DVD media.

The primary audience for this guide is the person, typically the site or system administrator, responsible for installing and maintaining OS 2200 software products on a system. That person must be familiar with the Exec and the system library structure, and must also refer to the *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604).

This guide describes how to complete the following tasks:

- Install SOLAR, register key tapes, and install FURPUR.

- Register a Unisys conditioned software package tape.

- Install OS 2200 software products from a registered conditioned software package tape.

- Correctly install an object module subsystem.

Product-specific information for SOLAR-installed products supported by this release also is included in this guide. Refer to the *ClearPath OS 2200 Systems Software Planning and Migration Overview* (7831 0349) for specific release and product level information, and for considerations that you must be aware of before installing the products in this release.

1.1. Documentation Updates

This guide contains all the information that was available at the time of publication. Changes identified after release of this document are included in problem list entry (PLE) 19267784. To obtain a copy of the PLE, contact your Unisys representative or access the current PLE from the Unisys Product Support website:

http://www.support.unisys.com/all/ple/19267784

Note: *If you are not logged into the Product Support site, you will be asked to do so.*

1.2. Notation Conventions

Notation conventions used are as follows.

Convention	Example
Uppercase letters indicate a keyword or a reserved word. LEVEL in the example is a reserved word. Unless otherwise noted, uppercase letters can be coded in uppercase or lowercase.	LEVEL,*product_level*
Lowercase italic text indicates variable information that the user must supply. In the example, *product_date* and *product_time* are user-supplied values.	TIME,*product_date,product_time*

Brackets surround an optional field, expression, or value. In the example, the PRINTER and FILE fields are optional.	REPORT;
	  NAME,*report_name* ;
	  [PRINTER,[*printer*][*number*];]
	  [FILE,*filename*]

1.3. SOLAR Processor

SOLAR is the software tool used to install OS 2200 software products that reside on software package tapes. SOLAR replaced COMUS as the principal product installation tool. COMUS is still used to generate symbolic products.

SOLAR consists of the processor and utilities. The SOLAR processor is a full-screen, menu-driven processor that creates runstreams by displaying a series of screens to collect information, depending on the task the user selects from the main menu. The created runstream contains stream generation statements (SGS) and a call to the appropriate SOLAR utility. When the runstream is started, the utilities process the SGSs to complete the specified task. This guide describes the procedure for installing OS 2200 software products. Refer to the *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604) for a complete description of the processor and utilities, and the other software library maintenance tasks you can complete with SOLAR.

1.4. Software Products Distribution

Unisys distributes most OS 2200 software products on conditioned software packages that are installed using SOLAR. You cannot copy and install a product from the software package unless your site is authorized to use the product. Unisys provides each site with a customized key to "unlock" the conditioned packages. This key tape identifies the products that site is authorized to use. After registering the key tape with SOLAR, you can copy and install the authorized products.

A few SOLAR-installable products are delivered without keys (see Appendix C). Appendix C also lists products that are installed with COMUS or SOLAR and those released as a CD-ROM image.

1.5. Separately Packaged Exec Features

Separately packaged Exec features you order are on tapes in SOLAR-installable format. The feature tapes contain the files for all Exec features, but you cannot copy and install a feature from a tape unless your site is authorized for that feature. The key tape you receive with the feature tape identifies those features.

The Exec symbolic features, though SOLAR installable, are not product loadable. You must still perform a system generation to install the features. You might have to alter the Exec generation runstreams to point to the SOLAR-installable files.

You do not have to use SOLAR to install the Exec features. If you wish, you can copy feature files via @COPY,G and install features from those files. You must use SOLAR to register the key tape before you can install features with SOLAR or copy them via @COPY,G.

Installing Exec features using SOLAR has the advantage of allowing you to load all features with one SOLAR command rather than loading each feature separately. See Section 3 for SOLAR installation procedures.

If you decide not to use SOLAR to install the Exec features, File 1 of the SOLAR feature tape has a tape file description (TFD) element that documents the files on the tape (refer to the following example). You can move the tape to the required position and copy it to a file by using @COPY.

SOLAR Feature Tape TFD Element Example

```
LOGICAL PACKAGE
     POS [BLOCK-ID]  NAME,LEVEL[,MODE]     FILE DESCRIPTION * SIZE + ENV
FORMAT   NAME,LEVEL
.   --- ---------  --------------------  ---------------- - ---- - ---   ----
--   -----------
TFD  1,001000000004  #                    U UTIL$,         40
COPYG
TFD  2,001000000024  CARTIS,46R1          M UTIL$,         29
COPYG   CP-SPF, ''8.0 ''


.      .                     .                            .      .
.      .                     .                            .      .
TFD 41,002000005222  SPAIR,5R5            M ABS$.          36, *, 2200
COPYG   CP-SPF, ''8.0 ''
.
. * U = SOLAR UTILITY FILE   M = COMUS MASTER FILE
.   S = SOLAR SPECIAL FILE   I = COMUS INFORMATION FILE
.   P = SOLAR PRODUCT FILE   F = COMUS SPF FILE
.   E = SOLAR EXTRA FILE
.
. + AN ASTERISK (*) IS PRESENT FOR SOLAR INSTALLABLE PRODUCT AND EXTRA FILES.
```

Section 2
Preparing to Install Products with SOLAR

This section describes the steps you must take before you can install products with SOLAR. It is assumed that your system is an existing system and is not being started for the first time. If your system is being started for the first time, refer to the installation guide appropriate to your system. The following step by step product installation sequence can be easily completed by using the CP-FLD fast load tape and the ST INSTALLFLD runstream.

Before you can install products, you must complete the following steps:

1. Install the correct SOLAR level for the ClearPath OS 2200 release level indicated on the cover of this guide.

2. Register the key tape with SOLAR.

3. Install the FURPUR level received with this release.

Notes:

- *The ClearPath OS 2200 products are delivered on logical package DVD media. For additional information on the logical package structure, refer to the* ClearPath OS 2200 DVD Media Quick Reference Card (7862 2032) *and the* ClearPath OS 2200 Systems Software Planning and Migration Overview (7831 0349) documents.

- *When installing a new system or initializing an existing system, follow the standard procedures in the* ClearPath OS 2200 Systems Software Planning and Migration Overview.

- *To be in software license compliance, the System Control Number (SCN) on a product's installation or feature key must match the SCN on the registered Exec image-enabler key. Before registering the key, make sure that the key tape is intended for the system/workload. If necessary, you must take any actions required to ensure that you remain in compliance with your software license agreement. Refer to the* ClearPath OS 2200 Systems Software Planning and Migration Overview (7831 0349) *and the* ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual (7831 0604) *for additional information.*

If there are no installation dependencies or conflicts, you can select products in different logical packages on the same physical package for installation from the same runstream. Refer to the *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604) for detailed information about using the F12 function key from the Product selection screen.

2.1. Installing SOLAR

You install SOLAR from the CP01 package tape using the INSTALLSOLAR runstream. This runstream installs only the first copy of SOLAR contained on a physical package tape, regardless of which logical package it might reside in or how many copies exist on the tape.

Execute the INSTALLSOLAR runstream by entering the following command from your system console:

```
ST INSTALLSOLAR[,1]
```

Note: *To execute this runstream, you must have the appropriate user privileges, as listed in Appendix A.*

The INSTALLSOLAR runstream asks for the reel number of the tape that contains SOLAR by displaying the following prompt:

```
INSTALLSOLAR:  Enter reel number   ▶
```

Specify the reel number as follows:

```
reel-num[/reel-num..][,[options][,[device-type][,[expiration]
[/mmspec][,[compression][,fta]]]]]
```

where:

reel-num

> is the tape reel that contains the SOLAR product.

options

> are the assign options. The default is TJ.

device-type

> is the tape device type. The default is HICL.

mmspec

> is the tape media manager specification.

expiration

> is the tape expiration period.

compression

> is the data compression value. The default is the system default. The compression field applies only to half-inch cartridge tapes.

fta

> is the Fast Tape Access specification. The values allowed are ON and OFF. The default value is OFF if you do not provide a specification. Fast Tape Access applies only to half-inch cartridge tapes.

If the SOLAR installation file contains a SOLAR$CONFIG element, the INSTALLSOLAR runstream uses tape assignment parameter values it contains to override the default values identified above.

If you enter 1 as the second parameter on the ST keyin, the INSTALLSOLAR runstream asks for configuration information pertaining to the cataloging of the SOLAR installation files. The format of these prompts is:

```
Disk equipment type? <FIXED>/REMOVABLE ▶
[- >Enter device type and pack-id: ▶ ]
 Additional product installation file catalog options? < > ▶
 Use file size for initial reserve? Y/<N> ▶
 Load file security? <NONE>/KEYS/PRIVATE  ▶
 Additional configuration SGSs, blank transmit when complete:
 - ▶
```

The INSTALLSOLAR runstream always asks if you want to register the software package with SOLAR. The format of this prompt is:

```
 Register package also? Y/<N> ▶
```

For additional information on these prompts, refer to the INSTALLPKG description in the *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604).

If you prefer, you can execute the INSTALLSOLAR runstream from an alternate file. This requires that you use the appropriate Executive Control Language (ECL) statements to copy the INSTALLSOLAR runstream from the boot tape into the alternate file you create on the system.

To execute the INSTALLSOLAR runstream after copying it into an alternate file, enter

```
 ST alternate-file.INSTALLSOLAR[,1]
```

Specify the reel number and any other information to the prompts as indicated above.

2.2. Registering the Key Tape

After SOLAR has been installed on your system, you must register the key tape. You cannot install or copy product files until the key tape is registered.

There are three ways to register a key tape. You can choose to:

- Use the SOLAR full-screen interface

- Execute the REGISTERPKG runstream

- Call PKGREG directly

  To register a key tape using PKGREG, use the PACKAGE_TAPE_INFO SGS. For information on this SGS, refer to the *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604).

To register a key tape using the SOLAR full-screen interface, follow this procedure:

1. Call the SOLAR processor.

2. Select Register from the Main Menu and transmit.

3. On the Package Information screen, enter the tape reel number information for the key tape and commit (F5).

4. On the Runstream Information screen, specify the file or element where you want SOLAR to write the runstream it generates, select if you want SOLAR to start the runstream now or if you want to start it later yourself, and commit (F5).

To register a key tape using the REGISTERPKG runstream, follow this procedure:

1. Execute the REGISTERPKG runstream by entering the following command from your system console:

   ```
   ST REGISTERPKG
   ```

   Note: *To execute this runstream, you must have the appropriate user privileges, as listed in Appendix A.*

2. When the REGISTERPKG runstream asks for the type of registration being performed and the reel number, specify the reel number as follows:

   ```
   reel-num[/reel-num..][,[options][,[device-type][,[expiration][/mmspec]
   [,compression]]]]
   ```

When the key tape is registered, product keys are added to the system key table.

Registered key tapes are not viewable for selection through the SOLAR interface. There are no products on key tapes, so SOLAR performs no action, other than initial registration. For reference purposes, the key tape has a symbolic KEYTABLE$ element that contains a list of the keys on the tape.

PKGREG produces a summary report that lists the new keys that are added to the system key table.

There are three ways to view all the currently registered keys, sorted by product:

- The SOLAR full-screen Registered Keys Online Report.

- The symbolic KEYTABLE$ element generated by PKGREG in the system library file SYS$*DATA$.

- The PRODRT Key Report.

You can also determine if a product is authorized by displaying the Registered Packages Online Report using the full-screen interface. The <F10> – View Product function key displays information on a selected product, including its authorization status.

Authorizations are on a feature-level basis. A new key is required for a product that was not previously authorized or for a new feature release of a product.

2.3. Removing Registered Keys

Remove the registered keys in one of two ways:

- Use the SOLAR full-screen interface.

- Call PKGREG directly.

To remove registered keys using the SOLAR full-screen interface, follow this procedure:

1. Call the SOLAR processor.

2. Select Remove Keys from the Main Menu and transmit.

3. On the Key Selection screen, select the keys you want to remove by placing a nonblank character in the tab position in front of the key.

4. On the Runstream Information screen, specify the file or element where you want SOLAR to write the runstream it generates, select if you want SOLAR to start the runstream now or if you want to start it later yourself, and commit (F5).

To remove registered keys using PKGREG, use the REMOVE_KEY SGS. For information on this SGS, refer to the *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604).

2.4. Installing FURPUR

You install the File Utility Routines/Program File Utility Routines (FURPUR) from the CP01 package tape using the INSTFURPUR runstream. This runstream installs only the first copy of FURPUR contained on a physical package tape, regardless of which logical package it might reside in or how many copies exist on the tape.

Note: *This step is not required if FURPUR level 32R5A or higher is already installed.*

Execute the INSTFURPUR runstream by entering the following command from your system console:

```
ST INSTFURPUR[,1]
```

The INSTFURPUR runstream issues the same prompts as the INSTALLSOLAR runstream.

Section 3
Installing Software Products with SOLAR

This section describes how to use SOLAR to install OS 2200 software products from a Unisys conditioned software package tape.

3.1. Before You Begin

Before reading any further in this section, make sure you understand the following information:

- System Status

  This guide assumes the following:

  - The correct Exec level for this release has been installed on the system. Refer to the *ClearPath OS 2200 Systems Software Planning and Migration Overview* (7831 0349) for a basic installation sequence.

  - The current version of SOLAR has been installed on the system (see Section 2).

  - The key tape(s) have been registered with SOLAR (see Section 2).

  - The current version of FURPUR has been installed on the system (see Section 2).

- User Privileges

  To start a runstream generated by the SOLAR processor, you must have a specific set of privileges. SOLAR strictly enforces privileges to make sure that only the appropriate personnel can install products. Refer to the *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604) for information on the privileges needed.

- Product-Specific Information

  Before you install products with the SOLAR processor, see Section 4 of this installation guide to familiarize yourself with any product-specific information that might affect the installation of a product.

- Product Interdependencies

  Installing a product on the system does not mean that it is functional. Check the product interdependencies listed in the *ClearPath OS 2200 Systems Software Planning and Migration Overview* (7831 0349) for more information on product dependencies.

- Generation Information

  Generation information for products installed with SOLAR is contained in Section 5, "Software Build Instructions."

Some products use the ELMS subsystem to retrieve and deliver information and error messages. During installation of these products, SOLAR briefly deactivates ELMS in order to rebuild an internal table with information about the newly installed product. If another product using ELMS is running at that time, there is a risk that the product might request a message from ELMS and then terminate abnormally when ELMS fails to respond.

To prevent this, make sure that all products that use ELMS remain idle until installation of the following products is complete:

APEX	NTSI
CIFS	OLTP-TM2200
cpFTP	OSI-TP
DDP-FJT	PCFP
DDP-PPC	QLP
ELMS	SAUTILITIES
EXPIPE	SLIB
EXT-KIT-2200	UC
FLEX	UCOB
I18NLIB	UFTN
LSS	UREP

3.2. Installation Procedure

The following steps describe the procedure needed to install OS 2200 software products from the conditioned software package tapes distributed by Unisys. Refer to the *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604) if necessary.

1. Call the SOLAR processor.

   If you want to change any of the default configuration values, specify a configuration element. The default configuration values are as follows:

   DVD Assign Option—TJ

   DVD Mnemonic Type—DVDTP

Diagnostic File Name—SOLAR*DIAGNOSTIC

Default Runstream File Name—SOLAR*RUNSTREAMS

Load File Security—NONE

Refer to the *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604) for more information on the configuration SGSs you can specify in the configuration element.

After SOLAR reads in the system information, it displays the main menu.

2. (Optional) Check the status of the system before you register any tapes or install any products.

    Tab to the Registered Packages (Online Report) menu selection to display the software packages that have already been registered.

    Tab to the Installed Products (Online Report) menu selection to display the products that are currently installed on the system.

    Refer to the *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604) for more information on using these main menu selections.

3. Tab to Register on the main menu and transmit.

    Note: *You must have completed the steps in Section 2 and registered the key tape(s) before you can register and install any products for which your site is authorized.*

4. Complete the information requested on the screens displayed by SOLAR. When you transmit from the Register menu selection, SOLAR displays the following screens:

    •   Package information

    •   Runstream information

    SOLAR uses these screens to gather the information it needs to create the runstream that registers the conditioned software package tape. Refer to the *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604) for more information about these screens and the information you need to provide.

    You cannot install the products from a conditioned software package tape until that package tape is registered.

    Once SOLAR creates the registration runstream, it redisplays the main menu.

5. Check Section 4 to see if there is any product-specific information for the products you want to install. The product-specific information can include any installation restrictions or a description of the product's modes if the product has multiple modes.

6. Tab to Registered Packages on the main menu and transmit.

7. Complete the information requested on the screens displayed by SOLAR. When you transmit from the Registered Packages menu selection, SOLAR displays the following screens:

- Registered Package

- Logical Package (optional)

- Product Selection

- Product Information (optional)

- Extra File Selection

- Runstream Information

- Additional Installation Information (optional)

SOLAR uses these screens to gather the information it needs to create the runstream that installs the products. Refer to the *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604) for more information about these screens and the information you need to provide.

If there are no installation dependencies or conflicts, you can select products in different logical packages on the same physical package for installation from the same runstream. Refer to the *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604) for detailed information about using the F12 function key from the Product selection screen.

On the runstream information screen, start the runstream SOLAR generates to install the products you specified on the product selection screen.

Once SOLAR creates the runstream, it redisplays the main menu. The products you specified are installed when the runstream completes its execution.

8. Tab to another selection on the main menu, or press the F2 function key to quit.

Section 4
Product-Specific Information

This section contains product-specific information you might need when installing products with SOLAR.

Table 4–1 lists all products installed with SOLAR and identifies those which have product-specific information described in this section. The product-specific information might include the following:

- Installation restrictions

- Initialization procedures, runstreams, and routines

- Installation modes

- Information on whether a product has subsystems

- Location of configuration information

The Subsystem column identifies products that have an object module subsystem or are part of a subsystem. Please read Appendix A before installing these products. This appendix defines how to set up the security environment, which must be done correctly or the product and any products that use it do not operate properly. This appendix also lists the name of the default subsystem definition file. For additional information, refer to the *ClearPath OS 2200 Apex Help* (8207 4154).

The Product Specific Information column indicates with a checkmark (√) when product-specific information is provided later in this section. Refer to the *ClearPath OS 2200 TeamQuest Products Installation Guide* (TQ-01013) for more information on TeamQuest products.

Products not listed in Table 4–1 are released as CD-ROM images (see Appendix C).

Note: *Table 4–1 lists products in order by their full product name. For your convenience, the product support names are listed in the Installation and Support Name column.*

Table 4–1. Products with Product-Specific Information

Product Name	Installation and Support Name	Subsystem	Product-Specific Information
Apex	APEX		√
ASCII COBOL Compiler	ACOB		√
ASCII FORTRAN Compiler	FTN		√
Business Information Server	MAPPER		√
C Compiler	UC	Yes (the subsystem defined by LSS)	√
CIFS	CIFS	Yes	√
CIPHER API	CIPHER-API	Yes	√
ClearPath Extension Kit	EXT-KIT-2200		√
COBOL Compiler	UCOB	Yes (the subsystem defined by LSS)	√
Collector	MAP		√
Common Mathematical Library	CML		√
Communications Application Program Interface	COMAPI	Yes	√
Communications Interface for Transaction Applications	CITA		√
Communications Platform	CPComm	Yes	√
Communications Platform for Open Systems	CPCommOS	Yes	√
COMUS	COMUS		√
Connectivity Services	CS2200	Yes	√
Cryptographic Library	CryptoLib		√
Database Agent	DBAGENT		√
Define File Processor	DFP		√
Display Processing System	DPS	Yes	√
Distributed Data Processing File and Job Transfer	DDP-FJT		√
Distributed Data Processing Program-to-Program Communications	DDP-PPC	Yes	√
Distributed Transaction Integration	DTI		√

Table 4–1. Products with Product-Specific Information

Product Name	Installation and Support Name	Subsystem	Product-Specific Information
ELT Processor	ELT		
Enterprise Application Runtime Test Environment for ClearPath OS 2200		Yes	√
Enterprise Network Database Server	DMS		√
Enterprise Output Manager (EOM)	DEPCON-SERVER		√
Enterprise Relational Database Server	RDMS	Yes	√
Enterprise Relational Database Server Event Analyzer	DB-EVENT-ANALYZER	Yes	
ePortal for ClearPath OS 2200	EPORTAL-2200		√
Extended Language Message System	ELMS	Yes	√
File Administration System	FAS		
FORTRAN Compiler	UFTN	Yes (the subsystem defined by LSS)	√
FTP Services for ClearPath OS 2200	cpFTP	Yes	√
General Syntax Analyzer	GSA		√
I18N Service Library	I18NLIB	Yes	√
Integrated Recovery Utility	IRU	Yes (FSAH component)	√
Interactive Processing Facility	IPF		√
Interconnect	IC2200	Yes	√
OS 2200 Transaction Resource Adapter for the Java™ Platform	J2EE-CON-OS2200		√
Language Support System	LSS	Yes	√
Linking System	LINK	Yes	√
LIST Processor	LST		
Media Manager	MMGR		√
Message Control Bank	MCB	Yes	√
Messaging Integration Services	NTSI	Yes	√

Table 4–1. Products with Product-Specific Information

Product Name	Installation and Support Name	Subsystem	Product-Specific Information
Meta-Assembler	MASM		√
Microsoft Message Queuing Interface	MSMQI		√
Monitor Services Control Program	MSCP		√
Multiple Batch Run Optimizer	EXPIPE	Yes	√
Network Database Server Resource Adapter for the Java Platform	DMS-RA		√
ODBC Access–Corporate Edition for ClearPath OS 2200	INFOACCESS	Yes	√
Open Distributed Transaction Processing	OLTP-TM2200	Yes	√
Open Systems Interconnection Transaction Processing	OSI-TP	Yes	
OS 2200 High Performance Transaction Processing Interconnect	HTPIC-2200	Yes	√
PLUS	PLUS		√
Procedure Definition Processor	PDP		√
Processor Common Input/Output System	PCIOS		√
Programmers Advanced Debugging System	PADS		√
Program-Callable FURPUR	PCFP	Yes	√
Query Language Processor	QLP		√
Relational Database Fast Load	XRLOAD		
Relational JDBC Driver	RDMS-JDBC		√
Remote System Support	RSS		
Repository for ClearPath OS 2200	UREP		√
ROLRUNS	ROLRUNS		
Runtime System for Basic Mode Compilers	UCSRTS		

Table 4–1. Products with Product-Specific Information

Product Name	Installation and Support Name	Subsystem	Product-Specific Information
Runtime System for Extended Mode Compilers	URTS		√
Service Library	SLIB	Yes	√
Shared File System	SFS		√
SOLAR	SOLAR	Yes	√
SOLAR/E	SOLAR/E	Yes	√
Sort/Merge	SORT		√
Symbolic Stream Generator	SSG		√
System Interface for Legacy Application Systems	SILAS		√
System Service Routines Library	SYSLIB		
Tape Labeling Utility	TUTIL		√
TCP/IP Application Services	TAS		√
TeamQuest Baseline®	TQ-BASELINE	Yes	
TeamQuest® CULL	CULL		
TeamQuest® D-Fragger	TQD-FRAGGER		
TeamQuest® IACULL	IACULL		
TeamQuest® Log Analyzer	LA		
TeamQuest® Mass Storage Analysis And Relocation	MSAR		
TeamQuest® Mass Storage Manager	MSMANAGER		
TeamQuest Online®	TQ-ONLINE	Yes	
TeamQuest® Online System Activity Monitor	OSAM	Yes	
TeamQuest® Performance Analysis Routines	PAR		
TeamQuest® PMLog	TQ-PMLOG		
TeamQuest® Probes	TQ-PROBES		
TeamQuest® RemD-Fragger	TQRDFRAGGER		
TeamQuest® Site Administration Utilities	SAUTILITIES		
TeamQuest® Site Management Complex	SIMAN		

Table 4–1. Products with Product-Specific Information

Product Name	Installation and Support Name	Subsystem	Product-Specific Information
TeamQuest® TIP Log Analyzer	TIP-LA		
Transaction Processing utility	TIPUTIL		√
UniAccess ODBC	UNIACCESS-ODBC	Yes	√
Utilization Report Utility for OS 2200	URU-OS2200		
Universal Database Control	UDSC	Yes	√
UPLS	UPLS	Yes (the subsystem defined by LSS)	√
User Authentication	FLEX	Yes	√
WebSphere MQ Version 9 for ClearPath OS 2200	WMQ2200	Yes	√
Web Transaction Server	WEBTS	Yes	√

4.1. Apex Administration and Reporting

The Apex agent installs with SOLAR and has only one installation mode (NORMAL). The SECMGR command line processor is also installed with the Apex agent. The Apex Windows web application is a separate component supplied as a CD-ROM image.

Before installing the Apex agent, choose a user-id and account to be used by the Apex agent background run. The user-id and account are specified during installation by the SOLAR DEFAULT_RUNSTREAM_USERID and DEFAULT_RUNSTREAM_ACCOUNT SGSs.

The Apex agent user-id should be defined as follows:

- For systems configured with Fundamental Security, the Apex agent is required to run under the security officer's user-id.

- For systems configured with Security Level 1 and higher, the Apex agent is required to run under a user-id with the security attributes described in the following table:

Attribute	Value
Run Mode	Batch
Processor Privilege	Read Executive GRS
Privileges	SSAUTHNTICAT, SSCONSOLE, SSSWITCHUSER

Note: Apex authenticates all users and uses impersonation to enforce security. Therefore, there is neither need, nor is it recommended, that the user-id be configured with additional privileges for systems configured with Security Level 1 and higher.

The Apex installation performs the following actions:

- Installs the Apex agent into file SYSLIB*APEX.

- Copies element APEX into SYSLIB*RUN$, which is the runstream for starting the Apex agent. It also inserts an @START image into SYSLIB*RUN$.AUTO$START to start that runstream each time the system is booted.

- Installs and registers the SECMGR command line processor, ELMS messages, and samples into file SYSLIB*APEX. SECMGR was previously a part of the Security Administration product which should be uninstalled before Apex is installed.

- Creates configuration file SYS$CONFIG$*APEX which contains the default configuration. This configuration can later be changed through a console keyin or from the Apex web pages. After installation, access to this file should be changed based upon your site's security policy. Read and write access must be granted to the user-id of the Apex agent and any other administrative user-id that is allowed to update the Apex agent configuration.

Refer to *ClearPath OS 2200 Apex Getting Started Guide* (8207 4055) for complete installation details.

4.2. ASCII COBOL Compiler (ACOB)

ASCII COBOL Compiler can be installed with either SOLAR or COMUS.

Table 4–2 indicates whether the ASCII COBOL common banked library contains Shared File System (SFS) interfaces. The ASCII COBOL library always contains Processor Common Input/Output System (PCIOS) interfaces.

Notes:

- *SOLAR installation modes are not equivalent to COMUS installation modes. For example, none of the SOLAR installation modes installs a flagging compiler.*

- *COMUS installation is currently allowed to accommodate users who require options that are not available in the five SOLAR installation modes.*

- *Users are encouraged to use one of the SOLAR-installable modes, since future ASCII COBOL Compiler installation enhancements are made only to these modes.*

Table 4–2. SFS Interfaces in Common Banked Library

ASCII COBOL Build Performed	SFS Installed Before Build	Library Contains SFS Interfaces
No (released version)	—	Yes
Yes	Yes	Yes
Yes	No	No

All modes listed except TEST install the following:

- Common banked nonflagging compiler (SYSLIB*ACOB)

- Common banked library (SYSLIB*ACOB)

- C$MCS common bank with Transaction Processing (TIP) interfaces

- Relocatable support elements

- Relocatable TIP library

Individual modes also install

- D (the default mode)

  – Relocatable DML library and relocatable support elements (SYSLIB*ACOB)

  – Full relocatable library including SFS interfaces (SYSLIB*ACOB-CB). The relocatable library is not registered with the Collector.

  – CBEP$$ACOB/PART-LIB-CB included in the file SYSLIB*ACOB

  – CBEP$$ACOB/FULL-LIB-CB included in the file SYSLIB*ACOB-DML

- E

  – Full relocatable library including SFS interfaces (SYSLIB*ACOB-CB). The relocatable library is not registered with the Collector.

  – CBEP$$ACOB/FULL-LIB-CB included in the file SYSLIB*ACOB

- F

  – Relocatable DML library and relocatable support elements (SYSLIB*ACOB)

  – CBEP$$ACOB/PART-LIB-CB included in the file SYSLIB*ACOB

  – CBEP$$ACOB/FULL-LIB-CB included in the file SYSLIB*ACOB-DML

- G (no additional items installed)

- TEST

  – Program banked nonflagging compiler (SYSLIB*ACOB-ALT). The compiler is not registered with the Exec.

  – Relocatable library including Shared File System (SFS) interfaces (SYSLIB*ACOB-ALT). The library is not registered with the Collector.

  – Relocatable TIP library (SYSLIB*ACOB-ALTTIP)

4.3. ASCII FORTRAN Compiler (FTN)

ASCII FORTRAN Compiler has the following installation modes:

- AFCB is the default mode
  - Common banked nonflagging ASCII FORTRAN Compiler
  - ASCII FORTRAN library with common banked I/O (type 2)
  - Processors FTN and FTNR registered with the Exec
  - Library file SYSLIB*FTN by default
- FLAGGER
  - Common banked flagging ASCII FORTRAN Compiler
  - ASCII FORTRAN library with common banked I/O (type 2)
  - Processors FTN and FTNR registered with the Exec
  - Library file SYSLIB*FTN by default
- TEST
  - Noncommon banked nonflagging ASCII FORTRAN Compiler
  - Relocatable format ASCII FORTRAN library (type 1)
  - No processors registered with the Exec
  - No library file registered with the Collector
  - No procedure elements installed in SYSLIB*PROC$
  - Installed in file SYSLIB*TESTFTN by default

4.4. Business Information Server (BIS)

Refer to the *Business Information Server for ClearPath OS 2200 Installation* and *Systems Analysis Guide* (7831 9332) for specific information on installing BIS software with SOLAR.

4.5. C Compiler (UC)

C Compiler has the following installation modes:

- A

  This mode requires that Language Support System mode A or mode S also be installed.

  This mode installs C Compiler as part of the object module subsystem defined by Language Support System.

  To call the compiler installed by this mode, enter @UC.

- TEST

  This nonreentrant version of C Compiler can be installed in combination with any other mode. It is not dependent on the version of Language Support System installed, although some version must be present.

  This mode is intended for preproduction testing. You must ensure that the proper INCLUDE files and library files are appropriately referenced at compile and link times.

  To call the compiler installed by this mode, enter @UCNR.

- X

  This installation mode is for Unisys use only.

Refer to Appendix A of the *Application Development Solutions Application Development Programming Guide* (7831 4077) for more details on installing the C Compiler and other compilers and ensuring that the interface products (UDSC, TIP, DPS, and MCB) can interact with the compiler programs.

4.6. CIFS

CIFS for ClearPath OS 2200 has only one installation mode:

NORMAL

- Installs CIFS for ClearPath OS 2200 into SYSLIB*CIFS$SS., and installs the following additional library file:

  ```
  SYS$LIB$*CIFS$LIB
  ```

- Installs SYSLIB*CIFS$DEBUG, which contains problem analysis tools.

- Copies three elements (CIFS-BACK, CIFS-POP, and CIFS-DEACT) into SYSLIB*RUN$., but only puts @START CIFS-BACK into AUTO$START.

4.7. Cipher API

Cipher API has two installation modes. They differ from one another only in the default configuration file that they load.

Mode	Description
A	For systems with CIP1001-PCX connected to an IOP
B	For systems with CIP2001-PCE connected to an IO Manager

The configuration file differences are significant only if you use a cryptographic hardware accelerator. If you do not use a hardware accelerator, then the installation modes are equivalent.

Each of the installation modes does the following:

- Installs the Cipher API subsystem into SYSLIB*CIPHER-API.

- Installs SYSLIB*CIPHER-ELMS, containing status messages, and CIPHER-JAVA, containing the UnisysJCE and associated JNI library.

- Installs the utility file, SYSLIB*CIPHER-UTIL., which contains the following elements:

    - The elements CIPHER-USER/H, CIPHER-UCOB, CIPHER-ACOB, CIPHER-FTN, CIPHER-MASM, and CIPHER-PLUS provide definitions of entry points, variables, and constants that can be included in user programs.

    - The element CIPHER/USERID is a runstream that can be used to establish the security attributes for Cipher API.

    - The element CIPHER is a runstream that initializes the CIPHER subsystem.

    - The elements CMOS-IOP/CFG and CMOS-IOM/CFG are default configuration files.

See the *ClearPath OS 2200 Cipher Application Programming Interface (API) Programming Reference Manual* (3826 6110) for more information about these elements.

Note: *The proper security attributes must be established for Cipher API in order for proper subsystem transitions to occur.*

4.8. ClearPath Extension Kit (EXT-KIT-2200)

CIFS 9R2 or higher must be installed on the system before installing ClearPath Extension Kit.

ClearPath Extension Kit has only one installation mode:

NORMAL

- Installs ClearPath Extension Kit into SYSLIB*NATIVEX.

- Installs installation scripts into SYS$NATIVEX*INSTALL.

- Copies nx-packages.zip and nx-rpms.zip to SYS$NATIVEX*PKGS and SYS$NATIVEX*RPMS respectively. These zip files are used to create, populate, and update /bin /etc /lib64 /run /sbin /srv /usr /var directories in the system root directory.

- Copies nativex-load.rpm to SYS$NATIVEX*NPCLD. Refer to the SAIL Command Center documentation for instructions on how to install this package.

- Registers the processors BASH, GIT, GREP, JAVA, JAVAC, NX, NXLINK, PYADD, PYTHON, PYTHON2, PYTHON3, and SVN with the Exec.

- Copies STNPCBG into SYSLIB*RUN$ and adds an @START STNPCBG to AUTO$START.

4.9. COBOL Compiler (UCOB)

COBOL Compiler has the following installation modes:

- A

  This mode requires that Language Support System mode A or mode S also be installed.

  This mode installs COBOL Compiler as part of the object module subsystem defined by Language Support System.

  To call the compiler installed by this mode, enter @UCOB.

- TEST

  This nonreentrant version of COBOL Compiler might be installed in combination with any other mode. It is not dependent on the version of Language Support System installed, although some version must be present.

  This mode is intended for preproduction testing. You must ensure that the proper COPY PROC files and library files are appropriately referenced at compile and link times.

  To call the compiler installed by this mode, enter @UCOBNR.

- X

  This installation mode is for Unisys use only.

Refer to Appendix A of the *Application Development Solutions Application Development Programming Guide* (7831 4077) for more details on installing COBOL Compiler and other compilers and ensuring that the interface products (UDSC, TIP, DPS, and MCB) can interact with the compiler programs.

4.10. Collector (MAP)

The MAP processor has the following installation modes:

- AFCB-FREE

  Common banked version that frees files that it assigns during a collection. This is the default installation mode.

- AFCB-ASG

  Common banked version that maintains file assigns after the collection.

- NON-AFCB

  A program banked version.

4.11. Common Mathematical Library (CML)

CML has the following installation modes:

- A

  Common banked version. It does not include the general intercept routines. This is the default mode.

- B

  Common banked version. It does include the general intercept routines.

- C

  Noncommon-banked version. It does not include the general intercept routines.

- D

  Noncommon-banked version. It does include the general intercept routines.

4.12. Communications Application Program Interface (COMAPI)

Communications Application Program Interface has four installation modes, mode A, B, C, and D. During the installation process, SOLAR creates the following files on your system.

Mode A	Mode B
SYSLIB*COMAPI	SYSLIB*COMAPI$B
SYSLIB*COMAPI$LIB	SYSLIB*COMAPILIBB
SYSLIB*RUN$.STCOMAPI	SYSLIB*RUN$.STCOMAPI-B
SYSLIB*COMAPI$TOOLS	SYSLIB*COMAPI$TOOL$B

Mode C	Mode D
SYSLIB*COMAPI$C	SYSLIB*COMAPI$D
SYSLIB*COMAPILIBC	SYSLIB*COMAPILIBD
SYSLIB*RUN$.STCOMAPI-C	SYSLIB*RUN$.STCOMAPI-D
SYSLIB*COMAPITOOL$C	SYSLIB*COMAPITOOL$D

Additional Installation Modes

In addition to the four installation modes (A, B, C, and D), Communications Application Program Interface enables you to install modes C through Z by using a utility called INS-MODES. INS-MODES streamlines the build and installation process, allowing you to make a copy of another Communications Application Program Interface mode and install it as mode *n*, where *n* is a single letter C through Z. There is overlap for modes C and D to support users that have their own BDIs for modes C and D from previous levels of COMAPI.

Format

```
SYS$LIB$*COMAPI.INS-MODES
```

When you execute this command, INS-MODES queries for

1. The name of the Communications Application Program Interface file to be used as the source for the SOLAR installation. The file must be the installed file SYSLIB*COMAPI or SYSLIB*COMAPI$B, or the first file from the Communications Application Program Interface product tape.

2. The letter of the mode (C through Z) you want to install. Each installation mode must be unique. For example, if mode G has already been installed, and you specify mode G, the installed mode G is replaced.

3. The user BDI for the Communications Application Program Interface subsystem for the mode being installed.

4. The mode letters and associated BDIs for all COMAPIs installed on your system. The format of the input is described when the prompt appears. The information is used to generate a customized COMAPICALLER call router library module.

INS-MODES makes a copy of the input file and modifies some elements, compiles and links the elements, and creates new installation files. INS-MODES then calls SOLAR and installs the new installation files. When the SOLAR installation has finished successfully, the following message appears:

```
The SOLAR install of COMAPI was successful.
```

If an installation error occurs, the following message appears:

```
The SOLAR install of COMAPI has failed. Please check the file,
DIAG.PL-DIAGINFO/datetime for a description of the error(s).
```

INS-MODES creates the following files on your system:

```
SYS$LIB$*COMAPI$n.
SYS$LIB$*COMAPI$LIB$n.
SYS$LIB$*RUN$.STCOMAPI-n
```

where *n* is the letter of the mode installed (C through Z).

4.13. Communications Interface for Transaction Applications (CITA)

Communications Interface for Transaction Applications provides an interface for remote devices such as PCs, OS 2200 systems, or other systems to access OS 2200 TIP applications using TCP/IP protocols. Other transport or network protocols are not supported.

For communication services, Communications Application Program Interface is used, which in turn attaches to CPComm or CPCommOS. If you wish to use two copies of CPComm or CPCommOS, you must use two copies of Communications Interface for Transaction Applications and two copies of Communications Application Program Interface.

For detailed information on configuring and starting this product after it is installed, see the *Communications Interface for Transaction Applications Configuration and Operations Guide* (7862 6470).

Use SOLAR to register the release tape and install Communications Interface for Transaction Applications. Installation modes are as follows:

- A

  Used to install a normal production environment. Mode A is linked to Communications Application Program Interface mode A. The installed file name is SYSLIB*CITA. Two start runstreams are copied to SYSLIB*RUN$, one to start mode A, and the other to capture diagnostics through Remote System Support.

- B

  Used to install Communications Interface for Transaction Applications in an alternate file. Mode B is linked to Communications Application Program Interface mode B. The installed file name is SYSLIB*CITA$B. Mode B allows for testing a new level or a fix, or to attach to another CPComm or CPCommOS. Two start runstreams are copied to SYSLIB*RUN$, one to start mode B, and the other to capture diagnostics through Remote System Support.

- C through Z

  Modes C through Z are provided through a separate utility called INS-MODES.

INS-MODES

INS-MODES is a utility that streamlines the build and installation process and allows the user to make a copy of another mode and install it as mode C through Z. INS-MODES queries the user for the following:

- The name of the file to be used as the source for the SOLAR installation, which must be the installed file SYSLIB*CITA or SYSLIB*CITA$B, or the second file from the product tape.

- The letter of the mode you want to install (C through Z). Each installation mode must be unique. If a mode G is already installed, and G is specified, the current mode G is replaced.

- The mode of Communications Application Program Interface you want this mode linked to (A through Z). The mode specified must be installed on this system. In particular, the file included in the link must be SYSLIB*COMAPI$LIB.CBEP$COMAPI (mode A) or SYSLIB*COMAPI$LIB$$x$.CBEP$COMAPI (where x is B through Z).

Installation Files

Mode A

SYSLIB*CITA.

SYSLIB*RUN$.STCITA-A

SYSLIB*RUN$.RSSCITA-A

Mode B

SYSLIB*CITA$B.

SYSLIB*RUN$.STCITA-B

SYSLIB*RUN$.RSSCITA-B

Mode x (where x can be C through Z)

SYSLIB*CITA$$x$.

SYSLIB*RUN$.STCITA-$x$

SYSLIB*RUN$.RSSCITA-$x$

4.14. Communications Platform (CPComm)

CPComm has the following installation modes:

- Mode A is the default mode of installation.

- Modes B through H are additional modes of installation. With modes A-H, you can run up to eight copies of Communications Platform concurrently.

- Mode AFCB is an optional mode of installation used in addition to modes A through H. Mode AFCB is required if you use SILAS.

- Mode BACKUP is an optional mode of installation used for cross-platform disaster recovery.

- The following modes are optional installation modes you can use if your site has not completed the migration of all network connections from TSAM to CPComm:

CMSA-CPCA	CMSTEST-CPCA
CMSA-CPCB	CMSTEST-CPCB
CMSA-CPCC	CMSTEST-CPCC
CMSA-CPCD	CMSTEST-CPCD
CMSA-CPCE	CMSTEST-CPCE
CMSA-CPCF	CMSTEST-CPCF
CMSA-CPCG	CMSTEST-CPCG
CMSA-CPCH	CMSTEST-CPCH

Installation Mode A

This is the default mode of installation and is normally used for the CPComm production environment. The first CPComm subsystem is installed using gate bank BDI 05222 and the following installation files:

- SYSLIB*CPCOMM

- CPCOMM*LIB$A

This mode defines the Library Code Name (LCN) CPCOMM$LCN under UCS$EMUSER in SYS$*DATA$.SSDEF$, which specifies that library file CPCOMM*LIB$A is automatically searched when users link their communications programs.

The entry points are defined in the elements CPCOMM*LIB$A.CBEP$CPCOMM and CBEP$$CPCOMM. User programs can also link to this subsystem by explicitly including the entry point element in their link or map source, respectively.

The CPCOMM*LIB$*x* file also contains

- QTSU—A UC module suitable for manipulating queue banks in your TSU application program. This module enables you to create an available queue, acquire queue banks from the queue, and release queue banks to the System Return queue. The queue banks are used with certain interfaces described in the *ClearPath Enterprise Servers Communications Platform Programming Reference Manual* (7844 8446).

- QTSU/H—The UC header file used by the QTSU module.

The elements STCPCOMMA and RSSCPCOMMA are installed in the alternate RUN$ file (SYSLIB*RUN$). STCPCOMMA is a runstream used to start mode A. RSSCPCOMMA is a runstream used to capture CPComm diagnostics through RSS.

Installation Modes B through H

Installation modes B through H can also be used in production but are primarily for

- Testing CPComm configurations
- Testing new levels of CPComm
- Debugging user communications programs

Additional subsystems are installed using the following installation files:

- CPCOMM*CPCOMM$B and CPCOMM*LIB$B

- CPCOMM*CPCOMM$C and CPCOMM*LIB$C

- CPCOMM*CPCOMM$D and CPCOMM*LIB$D

- CPCOMM*CPCOMM$E and CPCOMM*LIB$E

- CPCOMM*CPCOMM$F and CPCOMM*LIB$F

- CPCOMM*CPCOMM$G and CPCOMM*LIB$G

- CPCOMM*CPCOMM$H and CPCOMM*LIB$H

These modes use gate bank BDIs 05223, 05224, 05225, 04773, 04774, 04775, and 04776.

Entry points are defined in the CBEP$CPCOMM and CBEP$$CPCOMM elements of library files CPCOMM*LIB$B through CPCOMM*LIB$H, but no LCNs are inserted into SYS$*DATA$.SSDEF$. User communications programs link to these subsystems by explicitly including the entry point element in their link or map source.

The elements STCPCOMM*x* and RSSCPCOMM*x* (where *x* is mode B through H) are installed in the alternate RUN$ file (SYSLIB*RUN$).

Installation Mode AFCB

Optional installation mode AFCB is used in addition to modes A through H. This mode must be installed to enable the basic mode interface to CPComm. It installs the file CPCOMM*OM$LIB, which contains these elements:

- CPCOMMBMAFCB – An AFCB used by basic mode callers of all eight CPComm subsystems. The BDI 05274 is used by the AFCB.

- CACALLROUTER – An object module that allows your application program to call any installed CPComm without relinking. Refer to the *ClearPath Enterprise Servers Communications Platform Programming Reference Manual* (7844 8446) for details.

- BMCALLROUTER - A relocatable binary element that allows your application program to call any installed CPComm without remapping. Refer to the *Communications Platform Programming Reference Manual* for details.

Mode AFCB is required if you use SILAS.

Installation Mode BACKUP

Optional installation mode BACKUP is used for cross-platform disaster recovery. Refer to the *ClearPath Enterprise Servers Communications Platform Configuration and Operations Guide* (7844 8438) for more information.

Installation Modes CMSA-CPCA through CMSA-CPCH and CMSTEST-CPCA through CMSTEST-CPCH

These optional installation modes are used in addition to modes A through H. You can use these installation modes if your site has not completed the migration of all network connections from TSAM to CPComm. The AFCBs installed with these modes unconditionally redirect all calls from TSAM applications to CPComm so that the applications themselves do not need to be changed.

Each CPComm install mode installs three AFCBs. These AFCBs use BDIs previously used by the TSAM gate bank and TSAM AFCBs.

The Enterprise Output Manager product requires installation mode CMSA-CPCA.

Table 4–3. Communications Platform (CPComm) Install Modes

Communications Platform Install Mode	Redirects TSAM Calls to Communications Platform Mode	Reuses TSAM BDIs
CMSA-CPCA	A	404520 – 404522
CMSA-CPCB	B	404520 – 404522
CMSA-CPCC	C	404520 – 404522
CMSA-CPCD	D	404520 – 404522
CMSA-CPCE	E	404520 – 404522
CMSA-CPCF	F	404520 – 404522
CMSA-CPCG	G	404520 – 404522
CMSA-CPCH	H	404520 – 404522
CMSTEST-CPCA	A	404550 – 404552
CMSTEST-CPCB	B	404550 – 404552
CMSTEST-CPCC	C	404550 – 404552
CMSTEST-CPCD	D	404550 – 404552
CMSTEST-CPCE	E	404550 – 404552
CMSTEST-CPCF	F	404550 – 404552
CMSTEST-CPCG	G	404550 – 404552
CMSTEST-CPCH	H	404550 – 404552

Notes:

- *Only one of modes CMSA-CPCx can be installed. The installation file created by these install modes is CMS1100*CMS$LIB.*

- *Only one of modes CMSTEST-CPCx can be installed. The installation file created by these install modes is CMS1100*TEST$LIB.*

4.15. Communications Platform for Open Systems (CPCommOS)

CPCommOS has the following installation modes:

- Mode A is the default mode of installation.

- Modes B through H are additional modes of installation. With modes A-H, you can run up to eight copies of Communications Platform for Open Systems concurrently.

- Mode AFCB is an optional mode of installation used in addition to modes A through H. Mode AFCB is required if you use SILAS.

- Mode BACKUP is an optional mode of installation used for cross-platform disaster recovery.

- The following modes are optional installation modes you can use if your site has not completed the migration of all network connections from TSAM to CPCommOS:

CMSA-CPCA	CMSTEST-CPCA
CMSA-CPCB	CMSTEST-CPCB
CMSA-CPCC	CMSTEST-CPCC
CMSA-CPCD	CMSTEST-CPCD
CMSA-CPCE	CMSTEST-CPCE
CMSA-CPCF	CMSTEST-CPCF
CMSA-CPCG	CMSTEST-CPCG
CMSA-CPCH	CMSTEST-CPCH

CPCommOS uses the same application-level BDIs that CPComm uses.

You need at least two installation modes if you want to debug a CPCommOS configuration element by executing CPCommOS with the C option while another copy of CPCommOS is running.

Installation Mode A

This is the default mode of installation and is normally used for the CPCommOS production environment. The first CPCommOS subsystem is installed using gate bank BDI 05222 and the following installation files:

- SYSLIB*CPCOMM

- CPCOMM*LIB$A

This mode defines the Library Code Name (LCN) CPCOMM$LCN under UCS$EMUSER in SYS$*DATA$.SSDEF$, which specifies that library file CPCOMM*LIB$A is automatically searched when users link their communications programs.

The entry points are defined in the elements CPCOMM*LIB$A.CBEP$CPCOMM and CBEP$$CPCOMM. User programs can also link to this subsystem by explicitly including the entry point element in their link or map source, respectively.

The CPCOMM*LIB$$x$ file also contains

- QTSU – A UC module suitable for manipulating queue banks in your TSU application program. This module enables you to create an available queue, acquire queue banks from the queue, and release queue banks to the System Return queue. The queue banks are used with certain interfaces described in the *ClearPath Enterprise Servers Communications Platform for Open Systems Programming Reference Manual* (3839 3971).

- QTSU/H – The UC header file used by the QTSU module.

The elements STCPCOMMA and RSSCPCOMMA are installed in the alternate RUN$ file (SYSLIB*RUN$). STCPCOMMA is a runstream used to start mode A. RSSCPCOMMA is a runstream used to capture CPCommOS diagnostics through RSS.

Installation Modes B through H

Installation modes B through H can be used in production, but are primarily for

- Testing CPCommOS configurations

- Testing new levels of CPCommOS

- Debugging user communications programs

Additional subsystems are installed using the following installation files:

- CPCOMM*CPCOMM$B and CPCOMM*LIB$B

- CPCOMM*CPCOMM$C and CPCOMM*LIB$C

- CPCOMM*CPCOMM$D and CPCOMM*LIB$D

- CPCOMM*CPCOMM$E and CPCOMM*LIB$E

- CPCOMM*CPCOMM$F and CPCOMM*LIB$F

- CPCOMM*CPCOMM$G and CPCOMM*LIB$G

- CPCOMM*CPCOMM$H and CPCOMM*LIB$H

These modes use gate bank BDIs 05223, 05224, 05225, 04773, 04774, 04775, and 04776.

Entry points are defined in the CBEP$CPCOMM and CBEP$$CPCOMM elements of library files CPCOMM*LIB$B through CPCOMM*LIB$H, but no LCNs are inserted into SYS$*DATA$.SSDEF$. User communications programs link to these subsystems by explicitly including the entry point element in their link or map source.

The elements STCPCOMM*x* and RSSCPCOMM*x* (where *x* is mode B through H) are installed in the alternate RUN$ file (SYSLIB*RUN$).

Installation Mode AFCB

Optional installation mode AFCB is used in addition to modes A through H. This mode must be installed to enable the basic mode interface to CPCommOS. It installs the file CPCOMM*OM$LIB, which contains the following elements:

- CPCOMMBMAFCB – An AFCB used by basic mode callers of all eight CPCommOS subsystems. BDI 05274 is used by the AFCB.

- CACALLROUTER – An object module that allows your application program to call any installed CPCommOS without relinking. Refer to the *ClearPath Enterprise Servers Communications Platform for Open Systems Programming Reference Manual* (3839 3971) for details.

- BMCALLROUTER - A relocatable binary element that allows your application program to call any installed CPCommOS without remapping. Refer to the *Communications Platform for Open Systems Programming Reference Manual* for details.

Mode AFCB is required if you use SILAS.

Installation Mode BACKUP

Optional installation mode BACKUP is used for cross-platform disaster recovery. Refer to the *ClearPath Enterprise Servers Communications Platform for Open Systems Configuration and Operations Guide* (3850 8032) for more information.

Installation Modes CMSA-CPCA through CMSA-CPCH and CMSTEST-CPCA through CMSTEST-CPCH

These optional installation modes are used in addition to modes A through H. You can use these installation modes if your site has not completed the migration of all network connections from TSAM to CPCommOS. The AFCBs installed with these modes unconditionally redirect all calls from TSAM applications to CPCommOS so that the applications themselves do not need to be changed.

Each CPCommOS install mode installs three AFCBs. These AFCBs use BDIs previously used by the TSAM gate bank and TSAM AFCBs.

The Enterprise Output Manager product requires installation mode CMSA-CPCA.

Table 4–4. Communications Platform for Open Systems (CPCommOS) Install Modes

Communications Platform for Open Systems Install Mode	Redirects TSAM Calls to Communications Platform for Open Systems Mode	Reuses TSAM BDIs
CMSA-CPCA	A	404520 – 404522
CMSA-CPCB	B	404520 – 404522
CMSA-CPCC	C	404520 – 404522
CMSA-CPCD	D	404520 – 404522
CMSA-CPCE	E	404520 – 404522
CMSA-CPCF	F	404520 – 404522
CMSA-CPCG	G	404520 – 404522
CMSA-CPCH	H	404520 – 404522
CMSTEST-CPCA	A	404550 – 404552
CMSTEST-CPCB	B	404550 – 404552
CMSTEST-CPCC	C	404550 – 404552
CMSTEST-CPCD	D	404550 – 404552
CMSTEST-CPCE	E	404550 – 404552
CMSTEST-CPCF	F	404550 – 404552
CMSTEST-CPCG	G	404550 – 404552
CMSTEST-CPCH	H	404550 – 404552

Notes:

- *Only one of modes CMSA-CPCx can be installed. The installation file created by these install modes is CMS1100*CMS$LIB.*

- *Only one of modes CMSTEST-CPCx can be installed. The installation file created by these install modes is CMS1100*TEST$LIB.*

4.16. COMUS

COMUS level 6R1 or higher must be installed with SOLAR. Products that require installation with COMUS must be installed with COMUS level 5R1 or higher.

COMUS has the following installation modes:

- A

  The COMUS, COSMIC, and software generation print (SGP) processors are registered as processors with the Exec.

This mode provides the stand-alone version of COMUS. Only the software products on the system boot tape are required for execution.

This is the default installation mode.

- B

  With this installation mode, COMUS and its associated utilities are installed. The COMUS, COSMIC, and SGP processors are registered as processors with the Exec. If mass storage usage is critical, use this mode because it requires less mass storage and reduces main storage requirements.

This version of COMUS requires that the following software products be installed on the system:

- ASCII COBOL Compiler level 7R1 or higher

- SORT level 15R3D or higher

- PCIOS level 6R1 or higher common-bank libraries.

In addition, the absolute elements in the ABSCB$ file require that the ASCII COBOL Compiler common banks be generated for dual-processor state register (PSR) mode. This means using the C$DML and C$DML8 banks.

4.17. Connectivity Services (CS2200)

Connectivity Services has four installation modes: A, B, C, and D. During the installation process, SOLAR creates the following files on your system.

Mode A

SYSLIB*CS2200
SYSLIB*CS2200$LIB
SYSLIB*RUN$.STCS2200
SYSLIB*CS$CLIENT$A

Mode B

SYSLIB*CS2200$B
SYSLIB*CS2200LIBB
SYSLIB*RUN$.STCS2200-B
SYSLIB*CS$CLIENT$B

Mode C

SYSLIB*CS2200$C
SYSLIB*CS2200LIBC
SYSLIB*RUN$.STCS2200-C
SYSLIB*CS$CLIENT$C

Mode D

SYSLIB*CS2200$D
SYSLIB*CS2200LIBD
SYSLIB*RUN$.STCS2200-D
SYSLIB*CS$CLIENT$D

4.18. Cryptographic Library (CryptoLib)

Cryptographic Library (CryptoLib) has the following installation modes:

- Mode FIPS

  This is the default installation mode. This mode installs the current FIPS-certified CryptoLib into SYSLIB*CRYPTOLIB. See note 1.

- Mode FIPSTEST

This is a test installation mode. This mode installs the current FIPS-certified CryptoLib into SYSLIB*CRYPTOLIBTST.

- Mode NOTFIPS

    This mode installs the latest non-FIPS-certified CryptoLib into SYSLIB*CRYPTOLIB. This mode can contain new features or fixes that have not yet been FIPS certified. See notes 1 and 2.

- Mode NOTFIPSTEST

    This is a test installation mode. This mode installs the latest non-FIPS-certified CryptoLib into SYSLIB*CRYPTOLIBTST. This mode can contain new features or fixes that have not yet been FIPS certified.

Notes:

1. *CryptoLib mode FIPS or NOTFIPS is a required product for the CPComm and Cipher API products.*

2. *Only the NOTFIPS mode is installed when using the CP-FLD tape. If FIPS certification is required, you must install mode FIPS on your system using the CryptoLib release tape.*

4.19. Database Agent (DBAGENT)

DBAGENT provides information to the Apex Agent regarding the Unisys Universal Data System (UDS). For each UDS application group accessed from Apex, there is one DBAGENT installation.

DBAGENT is a SOLAR-installable product. See the *ClearPath OS 2200 Database Agent Administration Guide* for installation instructions.

4.20. Define File Processor (DFP)

The Define File Processor build information and a tape file description are in the last file on the Define File Processor release tape.

4.21. Display Processing System (DPS)

Before you install the Display Processing System, be aware of the following restriction:

When built with the ESECURITY Build parameter set to ON, Display Processing System has a subsystem.

The release tape contains three installable modes: A, B, and C. All modes produce the Display Processing System library file (default = SYSLIB*DPS) containing the following:

- Display Processing System utility processors
- Configuration element

- Run-time handler

- Copy procedures

- Omnibus elements of the internal forms for the FORMGEN, DPSIF, and DPSPW processors

- CONFIGURE debug and protocol elements (mode A and C environments only)

Modes A and C produce the Display Processing System library file SYSLIB*DPSALTPROC, which contains copy procedures in ASCII COBOL Compiler or COBOL Compiler format.

Mode A copies ASCII COBOL Compiler procedures into SYSLIB*DPS and SYSLIB*PROC$ files. Then, mode A copies COBOL Compiler procedures into SYSLIB*DPSALTPROC file. Mode C copies COBOL Compiler procedures into SYSLIB*DPS and SYSLIB*PROC$ files. Then, mode C copies ASCII COBOL Compiler procedures into the SYSLIB*DPSALTPROC file.

The following files also are created at installation time and are not part of the Display Processing System library file:

- Nonconfigured form library [QUAL]*FISENG containing the internal forms for FORMGEN.

- Nonconfigured form library [QUAL]*DISENG containing the internal forms for the Display Processing System File Initialization (DPSIF) and Display Processing System Password (DPSPW) processors.

  Refer to the *Display Processing System (DPS 2200) Administration Guide* (7831 2295) for a description of the [QUAL] configuration directive.

Display Processing System has the following three modes:

- A

  Mode A is the default installation mode. Mode A provides automatic installation of alternate file common banks (AFCB) for the following DPS banks:

  – Configuration bank

    ```
    DPS$CFBK
    ```

  – Handler banks

    ```
    DPS$BK      DPS$BK4
    DPS$BK2     DPS$BK4A
    DPS$BK2A    DPS$BK5
    DPS$BK3     DPS$BK5A
    DPS$BK3A    DPS$BK5B
    DPS$BK3B    DPS$BK5C
    DPS$BK3C    DPS$BK6
    DPS$BK3D
    ```

Mode A places these alternate file common banks in the Display Processing System library file and creates a CBEP$$DPS element in that file. The BDIs that mode A specifies in CBEP$$DPS are:

```
CONFIG$DPS*  EQU   0401070001000
HANDLER$*    EQU   0400372001000
HANDLER2$*   EQU   0401472001000
HANDLER2A$*  EQU   0401471001000
HANDLER3$*   EQU   0401473001000
HANDLER3A$*  EQU   0401475001000
HANDLER3B$*  EQU   0401476001000
HANDLER3C$*  EQU   0403025001000
HANDLER3D$*  EQU   0402415001000
HANDLER4$*   EQU   0401474001000
HANDLER4A$*  EQU   0402416001000
HANDLER5$*   EQU   0402412001000
HANDLER5A$*  EQU   0402413001000
HANDLER5B$*  EQU   0402414001000
HANDLER5C$*  EQU   0403020001000
HANDLER6$*   EQU   0403017001000
```

Mode A automatically registers the Display Processing System library file with the Collector and the Linking System, and the Display Processing System utility processors with the Exec. The processors registered with the Exec include: DPSIF, DPSPW, FLDP, FLMU, FORMGN, LISTER, LOGON, LOGOFF, STATS, TMODE, and TTYPE.

For more information about alternate file common banks, refer to the *ClearPath OS 2200 Exec System Software Executive Requests Programming Reference Manual* (7830 7899).

Mode A copies ASCII COBOL procedures into SYSLIB*PROC$.

- B

  Mode B, the program banked mode, provides relocatable versions of the mode A handlers and configuration for a single-thread Display Processing System. This mode does not provide processor registration with the Exec and does not provide library file registration with the Collector. Mode B is provided primarily as a test of the Display Processing System. Install it in as many separate library files as required.

  You can use mode B as your production DPS system. However, it is better to use mode A or C for the following reasons:

  – Mode B increases the storage requirements if you have numerous DPS application programs.

  – You cannot use the COMUS CONFIGURE command to configure mode B.

  – With mode B, you must recollect your application programs to use higher levels of DPS.

- C

  Mode C is similar to mode A except that COBOL Compiler compatible copy procedures are placed in the system procedure file SYSLIB*PROC$.

4.22. Distributed Data Processing File and Job Transfer (DDP-FJT)

Distributed Data Processing File and Job Transfer (DDP-FJT) has the following installation modes:

- Mode A

  This is program callable and has the Interactive Processing Facility (IPF) but without a Display Processing System (DPS) interface. Mode A is the default mode.

- Mode B (Program Callable and DPS, no IPF)

  This is program callable mode and has DPS, but not IPF.

- Mode C (All Program Callable, IPF and DPS)

  This is all program callable and has both DPS and IPF.

- Mode D

  This does not have DPS or IPF.

- Mode E

  This is program callable and has DPS, but not IPF. Additionally, this mode supports TXFR and PSERVER.

- Mode F

  This is program callable and has both DPS and IPF. Additionally, this mode supports TXFR and PSERVER.

4.23. Distributed Data Processing Program-to-Program Communications (DDP-PPC)

DDP-PPC determines the security level of the system during the SOLAR installation process and loads the appropriate version of the SSD element except when it is installed from a fast-load tape. If you are running Fundamental Security and you are installing DDP-PPC from a fast-load tape, you must enter the following command after the installation and prior to running DDP-PPC:

```
@COPY,A SYS$LIB$*PPC$$GATES.SSD/TRUSTED.SYS$LIB$*PPC$$GATES.SSD
```

If you change security levels after installing the DDP-PPC, you can either reinstall DDP-PPC or copy the appropriate SSD element with the following commands:

- To go from Fundamental Security to Security Level 1 or higher, enter

  ```
  @COPY,A SYS$LIB$*PPC$$GATES.SSD/SHELL,SYS$LIB$*PPC$$GATES.SSD
  ```

- To go from Security Level 1 or higher to Fundamental Security, enter

  ```
  @COPY,A SYS$LIB$*PPC$$GATES.SSD/TRUSTED,SYS$LIB$*PPC$$GATES.SSD
  ```

To perform the copy commands, you must have read and write privileges for the SYSLIB*PPC$$GATES file.

DDP-PPC has an object module subsystem.

For more information, refer to the *Distributed Data Processing Program-to-Program Communications (DDP-PPC) Implementation and Administration Guide* (3787 3270).

4.24. Distributed Transaction Integration (DTI)

Distributed Transaction Integration is a multi-purpose middleware product from Unisys. It provides access from Microsoft Windows environments to applications/transactions running on ClearPath servers as well as providing access to other transaction environments including Oracle Tuxedo and Fujitsu, formerly ICL, TPMS.

With ClearPath release 19.0, the capabilities previously provided by Transaction Integrator (formerly WebTx) is provided by DTI 11.3 installer and Open Transaction Integrator (OpenTI) is provided by OpenTI 12.1 installer. OpenTI 12.1 installer includes installation setup for 32-bit and 64-bit platforms.

Distributed Transaction Integration allows:

- Internet and Intranet access to applications and transactions running on the mainframe side of a ClearPath system.

- An environment where JAVA and CORBA components from other vendors can access applications and transactions on the mainframe side of the ClearPath.

OpenTI allows:

- An environment where Microsoft COM/DCOM/.NET can access applications/transactions on both the ClearPath systems and others mentioned above. Access is fully bi-directional and supports global transactions.

Distributed Transaction Integration provides both application runtime and development environments that integrate the various middleware environments mentioned above. As a result, Distributed Transaction Integration enables developers to integrate existing X/Open-compliant DTP services (Unisys OpenDTP, Oracle Tuxedo, Fujitsu, formerly ICL, TPMS) and Unisys TIP/HVTIP transactions with other popular application environments such as Microsoft COM/DCOM/.NET and Oracle WebLogic.

Note: *Support for Transaction Integrator Component and OS 2200 outbound has been discontinued from ClearPath release 18.0 onwards.*

4.25. ePortal for ClearPath OS 2200 (EPORTAL-2200)

ePortal for ClearPath OS 2200 has two installation modes that represent the two different forms of release media, a SOLAR-installable tape (DVD) and a SOLAR-installable ZIP file that can be downloaded from the Unisys Product Support website.

The installation modes are

- Mode A

  This is the default mode and is used to install ePortal from a tape or DVD.

- Mode B

  This mode is used to install ePortal from a ZIP file.

For both modes, a successful installation of the EPORTAL-2200 product catalogs and creates the following files in the OS 2200 environment:

- SYSLIB*EPORTAL, the ePortal system product file.
- EPORTAL*2200SHARE, the ePortal CIFS share file.

For additional installation and configuration information, refer to the *ClearPath ePortal Business Implementation and Operations Guide* (8215 9641).

4.26. Enterprise Application Runtime Test Environment for ClearPath OS 2200

Enterprise Application Runtime Test Environment for ClearPath OS 2200 (previously LINCLITE 2200) is installed on your system as a chameleon fixed-gate shared subsystem, with the following two programs that use this subsystem:

- LITE – Tests LINC Ispecs
- LITER – Tests LINC reports

For more information, refer to the *Unisys e-@ction Enterprise Application Runtime Test Environment Guide* (7862 1703).

4.27. Enterprise Network Database Server (DMS)

DMS is released on a package tape containing the standard product as part of the UDSC kernel. This tape also includes the product key (DMSKEY). The package tape must be registered and installed using SOLAR as follows:

- Use the product name DMSKEY, mode A, for the product key. Without the key, you cannot use the Data Definition Language (DDL) processor to define a database (schema).

- Use the product name DMS for the standard Enterprise Network Database Server product.

This product has four installation modes:

- NOREGISTER

  This is the default mode. This mode bypasses Exec processor registration and collector registration, and is typically used on all alternate application group installations. This is equivalent to the "test" keyword on the COMUS INSTALL command.

- REGISTER

  When this mode is installed, the DDL, SDDL, ADMLP, FDMLP, PDMLP, and DMU processors are registered as processors with the Exec, and the product file (D$MR) is registered with the collector.

  This mode can be used only once, since processor and collector registration can be used for only one application group. If this mode is used a second time, the product already installed with the REGISTER mode is removed.

- NOPROCESSOR

  This mode bypasses Exec processor registration. SOLAR still performs collector registration. This is equivalent to the "test,exec" keyword on the COMUS INSTALL command.

- NOCOLLECTOR

  This mode bypasses collector registration. SOLAR still performs Exec processor registration. This is equivalent to the "test,map" keyword on the COMUS INSTALL command.

Table 4–4 summarizes the effect of each installation mode on other application groups.

Table 4–5. Effect of Mode on Other Application Groups

Mode	Processor Registration?	Collector Registration?	Effect on Other Application Groups
NOREGISTER (default mode)	No	No	None
REGISTER	Yes	Yes	Removes DMS from any other application with processor or collector registration.
NOPROCESSOR	No	Yes	Removes DMS from any other application with Collector registration.
NOCOLLECTOR	Yes	No	Removes DMS from any other application with processor registration.

You must install all UDSC kernel product release tapes into each application group in which you intend to use this product. The DMS key needs to be installed only once, however, to make all product versions across all application groups fully functional.

In addition to installing the appropriate mode, use SOLAR to install and remove Database procedures (DBP) and user procedures written in Meta-Assembler (MASM) or ASCII COBOL Compiler. For more specific information, refer to the *ClearPath Enterprise Servers Enterprise Network Database Server for ClearPath OS 2200 Administration and Support Guide* (7830 7568).

Use SOLAR to install and remove Database procedures and user procedures written using the COBOL Compiler (UCOB). These DBPs are packaged as object module subsystems. For more specific information, refer to the *ClearPath Enterprise Servers Enterprise Network Database Server for ClearPath OS 2200 Administration and Support Guide* (7830 7568).

4.28. Enterprise Output Manager

Enterprise Output Manager for ClearPath OS 2200 installs with SOLAR and has only one installation mode. Enterprise Output Manager also includes a Windows server component supplied as a CD-ROM image.

4.29. Enterprise Relational Database Server (RDMS)

RDMS is released on a package tape containing the standard product as part of the UDSC kernel. This tape also includes the product key (RDMSKEY). The package tape must be registered and installed using SOLAR.

The standard product is installed into a chameleon subsystem.

Use the product name RDMSKEY, mode A, for the product key. Use the product name RDMS for the standard product.

RDMS has only one installation mode.

You must install all UDSC kernel product release tapes into each application group in which you intend to use RDMS. The RDMS key needs to be installed only once, however, to make all product versions of the product across all application groups fully functional.

4.30. Extended Language Message System (ELMS)

ELMS has an object module subsystem.

If the Linking System is reinstalled to a different level, the ELMS subsystem must be deactivated, as with all other Linker-dependent subsystems. Use SOLAR to deactivate the ELMS subsystem. On its next use, this subsystem is automatically reactivated.

4.31. FORTRAN Compiler (UFTN)

FORTRAN Compiler has the following installation modes:

- A

  This mode requires that Language Support System mode A or mode S also be installed.

  This mode installs FORTRAN Compiler as part of the object module subsystem defined by Language Support System

  To call the compiler installed by this mode, enter @UFTN.

- TEST

  This nonreentrant version of UFTN can be installed in combination with any other mode. It is not dependent on the version of Language Support System installed, although some version must be present.

  This mode is intended for preproduction testing. You must ensure that the proper INCLUDE files and library files are appropriately referenced at compile and link times.

  To call the compiler installed by this mode, enter @UFTNNR.

- X

  This installation mode is for Unisys use only.

4.32. FTP Services for ClearPath OS 2200 (cpFTP)

FTP Services for ClearPath OS 2200 has the following installation modes:

- A

  Mode A is the default installation mode, which is normally used with the Communications Platform (CPComm) or Communications Platform for Open Systems (CPCommOS) production environments.

- B, C, and D

  Mode B, C, and D are normally used for testing Communications Platform (CPComm) or Communications Platform for Open Systems (CPCommOS).

FTP Services for ClearPath OS 2200 has an object module subsystem.

If you want to use CPComm or CPCommOS mode B, C, or D instead of CPComm or CPCommOS mode A, you must specify the install mode as a parameter in the runstream used to start the FTP Services for ClearPath OS 2200 master run. Refer to the *ClearPath Enterprise Servers FTP Services for ClearPath OS 2200 User's Guide* (7847 5753) for more information.

4.33. General Syntax Analyzer (GSA)

Mode A is the default installation mode for GSA. When mode A is installed, it registers the TCON, MCON, and DCON processors with the Exec

4.34. I18N Service Library (I18NLIB)

I18N Service Library has a chameleon subsystem. It has the following installation modes:

- A

  The default locale used in Mode A is POSIX. The POSIX locale has the ISO8859-1 CCS associated to it. The CCS has a character set number of 35 (043) that is used in all file headers. This is the default mode.

- B

  The default locale used in Mode B is en_US.ASCII. This locale has the ASCII CCS associated with it and its CCS number is 1. This enables a code value of 1 to be used in all file headers.

4.35. Integrated Recovery Utility (IRU)

IRU must be registered and installed using SOLAR. IRU has the following installation modes:

- IRU

  This mode (default) installs the IRU processor into file SYSLIB*IRU. FSAH object modules, subsystems, and copy elements that might be installed in SYSLIB*FSAH and/or SYSLIB*ALTFSAH are unaltered.

- FSAH

  This mode installs the FSAH object module, subsystem, and copy elements into file SYSLIB*FSAH. The contents of SYSLIB*IRU and SYS$LIB*ALTFSAH are not changed.

- ALT-FSAH

  This mode installs an alternate FSAH object module, subsystem, and copy elements into SYSLIB*ALTFSAH. The contents of SYSLIB*IRU and SYSLIB*FSAH are not changed. This mode can be used to test a new level of FSAH, to test a version of FSAH containing a fix, or to run user FSAH applications and system software that uses FSAH on different FSAH levels.

Note: *If you want both the IRU processor and the FSAH subsystem installed, install both IRU and FSAH modes.*

For more information on installation and configuration, refer to the *ClearPath Enterprise Servers Integrated Recovery Utility Administration Guide* (7833 1576).

4.36. Interactive Processing Facility (IPF)

Install IPF with the SOLAR PRODLD utility or the SOLAR full-screen interface.

Note: *If IPF level 5R1 or lower release products are installed on your system (including stability update levels such as 5R1A, 5R1B, and so on), you must remove them before using SOLAR to install level 6R1 or higher.*

Use the SOLAR PRODRV utility to remove the following installed products: IPF 1100 Control, IPF 1100 Command Language, IPF 1100 Edit 1100, IPF 1100 Procedures, IPF 1100 User Assistance, or IPF SQL.

IPF can be configured after installation. Configuration instructions are in the *Interactive Processing Facility (IPF 1100) Administration Guide* (7833 3754).

4.37. Interconnect (IC)

Interconnect has different installation modes to support multiple levels of the software. The selected installation mode (A through H) must correspond to the installation mode of the COMAPI library that Interconnect will use. The selected number (1, 2, or 3) denotes the number of copies of Interconnect that are connected to the same mode of COMAPI.

The default installation mode is A1.

Installing the product

Before you install Interconnect, you must register the IC2200 release tape with SOLAR. Refer to the *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604) for detailed instructions.

Perform the following steps to install Interconnect:

1. Determine the installation mode to be used.

2. Use SOLAR to register and install the IC2200 product from the IC2200 release tape.

Verifying the installation

Perform the following steps to verify that the IC2200 installation is successful:

1. From an OS 2200 demand session, type an @PRT,F executive control language statement to verify that the following Interconnect files exist:

    - QUAL*ICLIB

    - QUAL*ICSTATLIB

    - QUAL*ICUTILITY

    - QUAL*ICADMIN

    - QUAL*ICINCLUDE

    - QUAL*ICSS

    where, QUAL is IC2200 by default and IC2200mode when an alternate mode is used.

2. Review the PL-SUMMARY$ element in the SOLAR*DIAGNOSTIC file.

    If the Interconnect software is successfully installed, the following information appears in the following report available in the SOLAR*DIAGNOSTIC.PL-SUMMARY$ file:

    - Product name

    - Release level

    - Installation mode

    - Time and date of installation

3. Verify the PL-DIAGINFO element.

   If the Interconnect software installation is not successful, the SOLAR processor creates a PL-DIAGINFO element in the SOLAR*DIAGNOSTIC file. The SOLAR*DIAGNOSTIC file contains warnings and error messages.

Perform the following actions if the Interconnect installation is not successful:

- Check the contents of the element to determine the cause of the error.

- Contact your Unisys Support Representative for assistance, if required.

For more information, refer to the *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604).

4.38. OS 2200 Transaction Resource Adapter for the Java™ Platform (J2EE-CON-OS2200)

J2EE-CON-OS2200 provides access from application components executing under Oracle WebLogic Server (WLS) and IBM WebSphere Application Server (WAS) to ClearPath OS 2200 TIP, HVTIP, and batch programs. Clients and technologies include Enterprise JavaBeans (EJB), Java Server Pages (JSP), Java servlets, and other Java EE technologies. J2EE-CON-OS2200 enables you to make new or existing applications running in the OS 2200 operating environments available to clients in Java networks outside the legacy systems.

J2EE-CON-OS2200 is installed from its own media. Installation instructions are included on the CD-ROM image in the ReadMe.txt file.

4.39. Language Support System (LSS)

Language Support System has the following installation modes.

- A

  Mode A is the default installation mode. This mode installs a Fast-Load Self-Contained subsystem (FLSS). This mode matches mode A of C Compiler, COBOL Compiler, FORTRAN Compiler, and UPLS.

- S

  This mode also matches mode A of C Compiler, COBOL Compiler, FORTRAN Compiler, and UPLS; however, it installs a Self-Contained subsystem (SCSS).

- X

  This installation mode is for Unisys use only.

All installation modes define object module subsystems.

When Language Support System is installed, the subsystem defined by Language Support System is deactivated using the SOLAR DEACT processor. When mode A of a compiler is installed, the corresponding Language Support System subsystem is also deactivated. Any runs using C Compiler, C++ Compiler, COBOL Compiler, and FORTRAN Compiler might terminate with errors if they are active in a compiler at the time of the deactivation.

Refer to the *Application Development Solutions Application Development Programming Guide* (7831 4077) for a description of special considerations and the steps you need to take when setting up an extended mode environment on a system.

4.40. Linking System (LINK)

The Linking System has the following installation modes:

- A

  This is the default installation mode.

  Both AFCBs and an object module subsystem are installed by this mode.

  The default application group number is 3. You can dynamically change this number by specifying a DEFAULT_APPLICATION_GROUP SGS to SOLAR when installing the Linking System.

  Note: *When a new level of the Linking System is installed, all active subsystems that require ongoing Linking System support must be deactivated so that they are reloaded using the new level of the Linking System. For more information, including a list of the affected products, see Appendix A.*

- ALT

  This mode allows for testing of a different level of the Linking System or of a Linking System with an added fix while leaving the current version of the Linking System unaltered. After installing the Linking System in ALT mode, the ALT versions of LINK, SSDP, OMCR, SSD-List, SSINFO, and ZM2ABS is contained in the SYSLIB*ALTLINK file.

  The ALT versions of these processors are not registered so the file name must be specified when using them. For example

  ```
  @SYS$LIB$*ALTLINK.LINK,s file.source,file2.output
  ```

  Besides the processors associated with the Linking System, the Linking System also consists of a component called the dynamic linker. The dynamic linker is the part of the Linking System that loads an object module. All subsystems and all types of object modules except ZOOMs (loaded by the Exec) must be loaded by the dynamic linker. Besides loading an object module, the dynamic linker is also used to resolve all entry points as they are referenced by the executing program if they were not resolved before execution by using the static linker. The dynamic linker is also used when a call to any of the Linking System program-callable interfaces is used (for example, LS$RES_NAME, or using PADS to debug a subsystem).

To use the ALT mode dynamic linker, do one of the following:

– Create a file named MYALTLINK. The qualifier name for this file must be your project ID. After creating the file, assign it to your run. For example

```
@CAT,P MY$ALT$LINK.
@ASG MY$ALT$LINK.
```

   or

```
@ASG,T MY$ALT$LINK.
```

As long as MYALTLINK is assigned to your run, the alternate mode dynamic linker is used. To revert back to using the regular installed dynamic linker, do an @FREE MYALTLINK. This can be used as an easy way for one run to initially test a new or modified version of the Linking System without affecting anyone else on the system.

– Ask your system administrator to install the USEALT mode of the Linking System. When USEALT is installed, the alternate dynamic linker is used system-wide.

• USEALT

You can install the USEALT mode of the Linking System after the ALT mode is installed. Installation of the USEALT mode creates the SYSLIB*USEALTLINK file containing AFCB element LINK$DEBUG. If AFCB LINK$DEBUG exists, the alternate mode dynamic linker is invoked for all users on the system. If it is no longer desirable to have the alternate mode dynamic linker invoked, the USEALT mode must be uninstalled using the Remove Products selection in SOLAR.

After either MYALTLINK is assigned or the USEALT mode has been installed, the alternate dynamic linker is used to load all object modules and subsystems that are not already loaded. For subsystems that are already loaded, if you wish to use a different copy of the dynamic linker, you must first deactivate the subsystem. That is, if you initially execute or load a subsystem when one level of the Linking System is installed and you later wish to test that subsystem with a different level of the Linking System, you must first deactivate the subsystem.

If you do not deactivate your subsystem, that subsystem continues to use the particular copy of the Linking System that it was originally loaded with. Only after the new level of the Linking System has been installed and the subsystem has been deactivated, will the subsystem use the new copy of the dynamic linker to load and thus have a new copy of the Linking System associated with that subsystem.

You can use the SSINFO processor to determine which subsystems have been loaded by the main mode A and the alternate ALT linkers:

```
@SSINFO,SP ALL/LINKERS
```

- OLD-LINK

  This mode installs a previous release level of the LINK processor while still installing the current release level of the dynamic linker. This install mode can be used to temporarily avoid any compatibility or migration issues you might experience in using the current release level of the LINK processor. Refer to the *ClearPath OS 2200 Systems Planning and Migration Overview* (7831 0349) for information about the compatibility and migration issues for the current release level of LINK.

  Note: *This previous release level of the static linker is also available in SYSLIB*LINKOMLIB.LINK if you have another install mode installed.*

- RSVL-TEST

  This mode is used for Roseville development and testing of the Linking System. Please do not install this mode at your site.

  Refer to the *Application Development Solutions Application Development Programming Guide* (7831 4077) for a description of special considerations and the steps you need to take when setting up an extended mode environment on a system.

4.41. Media Manager (MMGR)

MMGR is a separately installed Independently Linked Exec Subsystem (ILES) product.

Caution

Before installing MMGR, you must first establish the databases for the vendor's tape management system. If you do not establish these databases, all tape assignments fail. Refer to the vendor's documentation for detailed instructions.

MMGR has a single installation mode, mode A. When you install mode A, the following command is executed:

```
@MILES INSTALL,SYS$LIB$*MMGR.MMGR
```

and the following elements are created in SYSLIB*MMGR:

- MMGR (absolute element), the primary MMGR

- LOILES (absolute element), the backup MMGR, which is the log-only ILES

- LOILES (symbolic element), the runstream the Exec uses to start the backup MMGR

During installation, the MILES INSTALL function is executed automatically to install MMGR as an ILES and to have the Exec load it. In addition, the primary ILES user-id --MEDIAMGR– is automatically created with the attributes shown in Table 4–5.

Note: *You can preinstall the –MEDIAMGR– user-id, in which case, it is not created during installation.*

Table 4–6. Attributes Installed by MILES for User-Id
–MEDIAMGR–

Parameter	Value
Clearance Level	Minimum = 0 Maximum = 0
Quantum Timer	Disallow
Defer Interrupt	Disallow
Processor Privilege	Ordinary user
Access Privilege	Ordinary
Sharing Level	System
File	Owned
List Privilege	SSLOGER, SSCONSOLE

1. If you install MMGR without first establishing the databases for the tape management system, the first tape assignment request causes MMGR to abort, and the following message is displayed:

    ```
    MEDIA MANAGER INSTALLED, BUT NOT LOADED – ASK, CONTIN, ERRALL, RETRY
    ```

2. To correct this problem, establish the tape management system; then answer RETRY.

3. If you decide not to use a tape management system after MMGR is installed, read the following note, then proceed to step 4.

    Note: *You must be running under a security officer's user-id (for Fundamental Security) or under a user-id that has read and write privileges for SYSLIB*MMGR (for Security Level 1 or higher systems).*

4. Assign the SYSLIB*MMGR file.

5. Use the MILES processor to deinstall MMGR:

    ```
    @MILES DEINSTALL,SYS$LIB$*MMGR.MMGR
    ```

6. If the console message described in step 1 is outstanding, answer RETRY.

4.42. Message Control Bank (MCB)

Initially you must install the MCB 10R1 or later release tape by first registering the MCB release tape with SOLAR and then installing MCB with SOLAR. Refer to the *Software Library Administrator (SOLAR) User Reference Manual* (7831 0604). You also use SOLAR to install MCB from the MCB build output tape.

The MCB release tape has the following four installation modes:

UTILA

    The mode UTILA installation creates the file (SYSLIB*MCBUTIL$A) containing various tools that are used to process the MCB configuration and installation of MCB. Installing mode UTILA will not impact your existing application groups or cause any MCB run that might still be active to terminate. SYSLIB*MCBUTIL$A cannot be used to execute MCB. It is used to execute the MCB Configuration and Installation Utility (CIU) to configure and install additional MCBs with site-specific configuration requirements.

    Mode UTILA is the default mode.

UTILB

    The mode UTILB installation creates the file (SYSLIB*MCBUTIL$B) containing various tools that are used to process the MCB configuration and installation of MCB. Similar to mode UTILA, mode UTILB is made available to support another version of an MCB for testing (for example, with a fix applied). Installing mode UTILB will not impact your existing application groups or cause any MCB run that might still be active to terminate. SYSLIB*MCBUTIL$B cannot be used to execute MCB. It is used to execute the MCB Configuration and Installation Utility (CIU) to configure and install additional MCBs with site-specific configuration requirements.

MCB3

    Mode MCB3 installs a pre-configured MCB that can be used with TIP Application Group 3. This mode is not recommended unless the supplied configuration meets your needs. This pre-configured MCB can be installed and executed without configuring MCB if the supplied configuration meets your needs.

    MCB files for mode MCB3:

        MCB3*MCB$.
        MCB3*MCB$LIB.
        SYSLIB*RUN$.STMCB3

I18N

    Mode I18N installs the file I18N$*UTILITY. This file can be used to prepare a transaction for MCB transliteration. Refer to the *OS 2200 Internationalization Library (I18NLIB) User Manual* (7850 5393) for information about the I18N mode.

Once mode UTILA, UTILB, or MCB3 is installed, you can execute the MCB Configuration and Installation Utility (CIU) to configure and install additional MCBs with site-specific configuration requirements. You can also replace MCB3 with your own site-specific configuration.

When MCB is installed using the MCB Configuration and Installation Utility (CIU), the default installation file names for MCB are MCB*n**MCB$ and MCB*n**MCB$LIB (where *n* is the application number). If you want to replace the names of these files, you must specify the MCBFILE configuration parameter when calling the MCB Configuration and Installation Utility (CIU).

```
MCBFILE   qualifier*abs-file.,qualifier*lib-file.
```

This causes MCB to be installed in the files specified on the MCBFILE configuration statement, instead of MCB*n**MCB$ and MCB*n**MCB$LIB.

Note: *When multiple MCBs are installed, they are kept unique by attaching the application number to the installation file names and to the element names of the elements placed in SYSLIB*RUN$. This is accomplished automatically by the MCB Configuration and Installation Utility (CIU), based on your configuration input.*

If you are installing MCB to replace an MCB that is already installed (same application group), you must abort the application group and delete the MCB TIP/DMS files before installing.

If you have multiple MCBs installed (separate application groups), be sure to avoid configuration conflicts between them, particularly with respect to BDIs and file numbers. If two MCBs have a conflict over BDIs (including defaults), installation of the second MCB causes automatic removal of the first one.

Refer to the *MCB Configuration and Operations Guide* (7833 1550) for more details on the MCB Configuration and Installation Utility (CIU).

4.43. Messaging Integration Services (NTSI)

NTSI is a low-level, message-based interface between a program on an OS 2200 node and a program operating on the Microsoft Windows node.

NTSI has an OS 2200 component and a Windows component, each of which is installed separately. The OS 2200 component is on release tapes and the Windows component is on a CD-ROM image issued with each ClearPath software release.

Use SOLAR to install mode A of the NTSI OS 2200 component on an OS 2200 node. Use the INS-MODES processor supplied by NTSI to install alternate modes, if needed.

Communications Application Program Interface (COMAPI) must also be installed in order to use NTSI.

NTSI installs two software subsystems: its own and one for the Mail Integration feature. See Appendix A for more information.

For more information on installing, configuring, and operating NTSI on both ClearPath nodes, refer to the *ClearPath Enterprise Servers Messaging Integration Services for ClearPath OS 2200 Help* (7851 5558).

4.44. Meta-Assembler (MASM)

MASM level 6R3 (and higher) provides only one (default) installation mode, A. There is no longer a common bank version.

4.45. Microsoft Message Queuing Interface (MSMQI)

MSMQI provides an interface between OS 2200 applications and message queues in your network. The message queues enable your OS 2200 applications to interact asynchronously with other applications via your enterprise network. The other applications can reside on Windows systems, other OS 2200 nodes, UNIX systems, and IBM MVS systems.

MSMQI provides a "COBOL friendly" interface to message queues and read and write access to MSMQ queues.

MSMQI also provides queue administrative functions, create and delete. However, MSMQI assumes that queue administration (creating, deleting, moving, and so forth) is primarily via the MSMQ "snap in" in the Computer Manager, MSMQ functions for the Windows Active Directory, or Windows based programs.

MSMQI uses the MSMQ COM object interface as a model, not the older MSMQ C Language SDK APIs. Therefore, MSMQI APIs are not compatible with the FalconMQ Client for OS 2200 APIs.

MSMQI assumes that the ClearPath Windows node is a Windows node executing Message Queuing (MSMQ 2.0 or later).

MSMQI has an OS 2200 component and a Windows based component. The OS 2200 component is installed with NTSI when that product is installed. See the product installation information for NTSI later in this section.

The Windows component is installed from separate media. For more information, refer to *ClearPath OS 2200 Interface for Microsoft Message Queuing Help* (6885 2722).

4.46. Monitor Services Control Program (MSCP)

MSCP is released as both a separately installable (SOLAR-installable) product and a preinstalled product in SYS$*LIB$.

If you have Fundamental Security or Security Levels 1 or 2, you can use either the SOLAR-installable or preinstalled versions of MSCP. If you have Security Level 3, you can use only the SOLAR-installable version of MSCP. If you install MSCP with SOLAR, you can install new versions without rebooting your system.

If you install MSCP with SOLAR, the MSCP background run must be terminated using the E keyin before MSCP is installed. If the MSCP run is not terminated, it is automatically terminated when the new MSCP is installed.

For more detailed information on MSCP, refer to the *ClearPath OS 2200 Exec System Software Administration Reference Manual* (7831 0323).

REPROG Images (Preinstalled MSCP)

If you use the preinstalled version of MSCP, the following REPROG images must be included in your site configuration:

```
REPROG BANKS,D+G+Q+C       ARE  MSB , MSB$
REPROG BANKS,D+G+Q         ARE  MSBDB , 0405253
REPROG BANKS,D+G+Q         ARE  MSBDB , 0405254
REPROG BANKS,D+G+Q         ARE  MSBDB , 0405255
REPROG BANKS,D+G+Q         ARE  MSBDB , 0405256
REPROG BANKS,D+G+Q         ARE  MSBDB , 0405257
REPROG BANKS,D+G+Q         ARE  MSBDB , 0405260
REPROG BANKS,D+G+Q         ARE  MSBDB , 0405261
REPROG BANKS,D+G+Q         ARE  MSBDB , 0405262
```

Note: *The C option must be added to the MSB REPROG image in your site configuration.*

Installing MSCP

When you install MSCP, the following runstreams are added:

- MSCP   (Run-id is MSCP)

- MSCPRT   (Run-id is MSCPRT)

- MSCP/RT   (Run-id is MSCPRT)

If you choose to SOLAR-install the separate MSCP product, the MSCP program must first be terminated using the E keyin. The MSCP product is always installed in the file SYSLIB*MSCP. The default install mode for MSCP is mode A, which inserts an @START,A/X of the real-time MSCPRT runstream into the AUTO$START element in SYS$LIB$*RUN$. You can choose to SOLAR-install the non-real-time version of MSCP by specifying mode B, which inserts an @START,A/X of the non-real-time MSCP runstream into the AUTO$START element in SYS$LIB$*RUN$.

If you do not SOLAR-install the separate MSCP product, the standard SYS$*RUN$.AUTO$START element delivered with the EXEC master tape will automatically @START the SYS$*RUN$.MSCP/RT runstream during every system boot. If you choose to SOLAR-install the MSCP product, the runstreams placed into the alternate RUN$ file (SYSLIB*RUN$) are intended to supersede the default runstreams delivered in SYS$*RUN$. When the SOLAR installation of MSCP adds the @START statement for the MSCP background run into SYSLIB*RUN$.AUTO$START as described above, it will also remove the corresponding @START statement for the MSCP/RT runstream from SYS$*RUN$.AUTO$START. This is done to avoid starting two copies of the MSCP background run during every system boot. During booting, the SYS run will @ADD the contents of both SYS$*RUN$.AUTO$START and SYS$LIB$*RUN$.AUTO$START, so the @START statement for the MSCP background run should always be in one, but not both, of the AUTO$START elements.

Any changes made to elements in file SYS$*RUN$, however, cannot be considered permanent, since the SYS$*RUN$ file will be re-cataloged whenever any Jump Key 4 tape boot is performed. Thus, after SOLAR-installing the MSCP product, you should copy the updated version of SYS$*RUN$.AUTO$START to your system boot tape(s) and also copy the updated SYS$*RUN$.AUTO$START element into the RUN$ file that is specified within the "LIB FILES" SGS used by your EXEC generations.

Security Level 3 customers must do the following before installing and using MSCP:

1. Remove all MSB REPROG statements from the site configuration.

2. Remove the MSCP and MSB absolutes from SYS$*LIB$ (optional). Remove the MSBDB absolute, if it exists.

3. Generate a new boot tape and boot the system.

4. Install MSCP using SOLAR.

5. Manually start up the MSCP background run.

4.47. Multiple Batch Run Optimizer (EXPIPE)

Multiple Batch Run Optimizer has one installation mode, mode A, which installs the following:

- One fixed-gate shared subsystem (FGSS), which is the kernel of Multiple Batch Run Optimizer

- One basic mode common bank, which supports switching between the basic mode PCIOS and the extended mode Multiple Batch Run Optimizer FGSS

- The utility processors @EXPDEF, @EXPASG, and @EXPSTA, which are packaged as zero overhead object modules (ZOOM)

- The system configuration file SYSLIB*PIPE$CONFIG

4.48. Network Database Server Resource Adapter for the Java Platform (DMS-RA)

DMS-RA software product provides a standards-based Java interface to Network Database Server (DMS). It opens DMS data to stand-alone Java applications and Java EE components such as EJBs and servlets.

DMS-RA has two pieces of media, a SOLAR-installable piece and a CD-ROM image that is installed on your Java platform. The SOLAR-installable component has four install modes, A through D, with mode A being the default. More than one mode can be installed at a time. See the *Network Database Server Resource Adapter for the Java™ Platform User Guide* (3850 8362) for installation instructions.

4.49. ODBC Access – Corporate Edition for ClearPath OS 2200 (INFOACCESS)

ODBC Access – Corporate Edition for ClearPath OS 2200 provides a full relational engine for a relational view of Enterprise Network Database Server (DMS) databases. It supports access from Open Database Connectivity (ODBC) tools and tools that support SQL standards (such as ISQL and CLI). Updates are supported.

The ODBC Access client provides the client side for ODBC access to DMS databases.

ODBC Access is SOLAR-installable using Mode A and includes a protected subsystem. See Appendix A for user-id configuration requirements.

For installation and configuration information, refer to the *Distributed Processing Middleware Data Access Installation and Operations Guide* (7862 5753*).

4.50. Open Distributed Transaction Processing (OLTP-TM2200)

Open Distributed Transaction Processing provides the Open Group standard interfaces that allow OS 2200 applications to participate in mission-critical distributed transaction processing (DTP) environments.

The default mode of Open Distributed Transaction Processing installs the English-language version for messages. To translate system messages to another language, you must follow this procedure:

1. Produce a new Extended Language Message System (ELMS) message data bank. For more information, refer to the *OS 2200 Extended Language Message System (ELMS) Programming Guide* (7830 8210).

2. Generate and install Open Distributed Transaction Processing using COMUS and SOLAR.

For additional installation and configuration information, refer to the *Distributed Processing Middleware Open Distributed Transaction Processing Administration Guide Volume 1: Installation, Configuration, and Ongoing Administration* (7833 5072) and the *OSI-TP High-Performance Transaction Processing for XATMI (HTP/x) Implementation and Administration Guide* (2750 2558).

This product has an object module subsystem.

4.51. OS 2200 High Performance Transaction Processing Interconnect (HTPIC-2200)

HTPIC-2200 is an absolute-only version of OS 2200 OSI-TP (HTP/x) that supports communications with DTI.

HTPIC-2200 can only send messages to and receive messages from DTI. If communication with Oracle eLink OSI TP running on UnixWare or Windows is required, then OS 2200 OSI-TP (HTP/x) must be used.

OS 2200 OSI-TP supports interoperability with all platforms supporting OSI-TP, whereas HTPIC-2200 supports only the Windows platform running DTI. If you need to support access to OS 2200 Open Distributed Transaction Processing services from Tuxedo clients, you must install OS 2200 OSI-TP (HTP/x).

HTPIC-2200 and OS 2200 OSI-TP (HTP/x) use the same BDI. Keep in mind that if you have OS 2200 OSI-TP installed, there is no need to also install HTPIC-2200. However, if your environment dictates that both products are added on the same system, you must build one with mode TEST. This creates an alternate with a different BDI. Also, you must build OLTP/TM2200 in TEST mode for the second installation.

4.52. Programming Language for Unisys Subsystems (PLUS)

Programming Language Unisys Subsystems (PLUS) has the following installation modes:

- AFCB

  Common banked version. This is the default mode.

- NON-AFCB

  Program banked version.

Both installation modes use file SYSLIB*PLS.

4.53. Procedure Definition Processor (PDP)

PDP has the following installation modes:

- AFCB (alternate file common bank)

  Common banked version. This is the default mode.

- NON-AFCB

  Program banked version.

4.54. Programmers Advanced Debugging System (PADS)

The Programmers Advanced Debugging System has one installation mode, mode AFCB, which performs the following operations:

- Installs PADS as alternate file common banks into the file SYSLIB*PADS.

- Installs the PADS Standard Procedures in the system library SYSLIB*PADS$PROC$.

- Registers the PADS and NPEER processors with the Exec.

- Assigns the library code name (LCN) PADS$EMLIB to the file SYS$LIB$*PADS and includes it in the Linking System search chain, enabling easy link access to the PADS program callable interfaces for extended mode products.

For additional information, *refer to the OS 2200 Programmers Advanced Debugging System Programming Guide* (7831 0562).

4.55. Processor Common Input/Output System (PCIOS)

PCIOS has the following installation modes:

- A

  This is the default mode. It installs PCIOS as alternate file common banks into the file SYSLIB*PCIOS and registers the file with the collector.

- B

  Installs PCIOS as a relocatable file into the file SYSLIB*PCIOS6R2.

- C

  Installs PCIOS as alternate file common banks into the file SYSLIB*PCIOS. With this installation, an error code of "42" is returned when a MSAM file that has not been closed by a previous user is opened. Install PCIOS with this mode if you are using the Information Management System (IMS).

- D

  Installs PCIOS as a relocatable file into the file SYSLIB*PCIOS6R2. With this installation, an error code of "42" is returned when a MSAM file that has not been closed by a previous user is opened. Install PCIOS with this mode if you are using the IMS with PCIOS relocatable elements.

The PCIOS build information and a PCIOS tape description are in the last file on the PCIOS release tape.

4.56. Program-Callable FURPUR (PCFP)

PCFP has an object module subsystem.

4.57. Query Language Processor (QLP)

QLP has the following installation modes:

- AFCB-DMS-P

  This is the default mode. It installs an AFCB QLP/DMS/PCIOS/RDMS system collected to access DMS and an AFCB PCIOS (an AFCB is a multithread system in which all users access the same copy of the absolute processor). This mode requires separate installation of the File Definition Processor (FDP) and the Application Definition Processor (ADP) before you can access PCIOS files with QLP.

- AFCB-DMS

  This mode installs an AFCB QLP/DMS/RDMS system collected to access DMS.

- AFCB-PCIOS

  This mode installs an AFCB QLP/PCIOS/RDMS system collected to access an AFCB PCIOS. This mode requires separate installation of FDP and ADP before you can access PCIOS files with QLP.

To install the default mode (AFCB-DMS-P), you do not need to identify the mode. To install either of the other modes, you must specify the mode name.

Unlike QLP, the FDP and ADP have only a single-thread installation mode, and you cannot specify a mode for them.

4.58. Relational JDBC Driver (RDMS-JDBC)

RDMS-JDBC software allows Java programs of all types to communicate with databases managed on the ClearPath OS 2200 systems under the Relational Database Server for ClearPath OS 2200.

RDMS-JDBC is a SOLAR-Installable product. See the *Relational JDBC Driver for ClearPath OS 2200 User Guide* for installation instructions.

4.59. Repository for ClearPath OS 2200 (UREP)

UREP is released on a package tape containing the standard product as part of the UDSC kernel. The package tape must be registered and installed using SOLAR.

UREP has three installation modes.

- A

   This is the default mode and the one that must normally be used. If the files do not exist, mode A initializes the application group including the Repository database, RDMS system files, and DMS metadatabase. If the files already exist, mode A migrates the application group to the level of UREP being installed.

- B

   This mode reinitializes the application group. Using this mode destroys all user information in the Repository database, RDMS system files, and DMS metadatabase. Before installing UREP in mode B, you must enter a SUDS II command and then perform Integrated Recovery Utility short recovery. In the case of concurrent application groups, perform these tasks on the application group on each host. Refer to the *ClearPath Enterprise Servers Universal Data System Configuration Guide* (7844 8362) for more information.

- C

   This mode creates new cycles of the UREP product files and loads the UREP common banks, but does not migrate or initialize the application group. You can use this mode to install an updated version of the current UREP level on an existing application group. Do not use this mode for the initial installation on an application group or to migrate from one UDS level to another.

You must install all UDSC kernel product release tapes into each application group in which you intend to use UREP.

Integrated Recovery Utility, Universal Database Control, and Enterprise Relational Database Server for ClearPath OS 2200 must be installed and the application group (3 is the default) must be up and recovered before UREP is installed. Refer to the *ClearPath Enterprise Servers Universal Data System Planning and Installation Overview* (7844 8370) and the *ClearPath Enterprise Servers Universal Data System Configuration Guide* (7844 8362) for more information.

A migration installation of UREP catalogs and initializes new Relational Database Server system files as follows.

Level	Relational Database Server System Files Cataloged and Initialized
To level 11R1 from level 6R2 or lower	RDM$DROPFILE
To level 11R1 from level 5R3 or lower	FRDT$FILE, LOB$SA$FILE, and PART$FILE
To level 11R1 from level 5R1A or lower	DEP$FILE, ECM$FILE, EEAM$FILE, NULL$FILE, PARAMS$FILE, ROUTINE$FILE, SCH$FILE, TRIG$FILE, and TRIGCOL$FILE

Be sure to back up and save these new files along with the other database files. The file qualifier is the standard application qualifier (such as UDS$$SRC if the application is the default application).

If you perform a mode A migration installation of UREP, you must use a user-id that has UREP security officer privileges. If the installation catalogs the Relational Database Server stored procedure files, the user-id must also have access privileges for the following tables: RDMS.AUTHTABLE, UREP.ENTITIES, UREP.ENTITY_ATTRIBUTES, and UREP.ENTITY_NAMES. In most cases, the user-id for the initial installation has those privileges. Refer to the *ClearPath Enterprise Servers Repository for ClearPath OS 2200 Administration Guide* (7830 8087) for more information.

4.60. Runtime System for Extended Mode Compilers (URTS)

Runtime System for Extended Mode Compilers has the following installation modes:

- AFCB

  Common banked version. This is the default mode.

- TEST

  An alternate version of the run-time system common banks and the run-time system service library is installed.

New levels of Runtime System for Extended Mode Compilers must be installed at a time when background runs for other products are not active. If any background runs are active and are written in high-level languages, installation of a new level might cause them to abort. Background runs that might be affected by an installation must be terminated using normal procedures before installation begins and then restarted after installation has completed.

Installation of a new Runtime System for Extended Mode Compilers level might cause CPComm or CPCommOS and any program that calls either CPComm or CPCommOS or the TSAM interface to abort. Therefore, you must install the new level when those programs are not running. The following is a list of products that are known to call either TSAM or CPComm or CPCommOS:

- ClearPath OS 2200 BIS

- Communications Application Program Interface

- Distributed Data Processing Program-to-Program Communications (DDP-PPC)

- Enterprise Output Manager for ClearPath OS 2200

- FTP Services for ClearPath OS 2200

- ODBC Access for ClearPath OS 2200

- Open Distributed Transaction Processing

- System Interface for Legacy Application Systems (SILAS)

- Transmission Control Protocol/Internet Protocol (TCP/IP) Application Services (TAS)

- UniAccess ODBC

- Web Transaction Server

- Network Database OLE DB Provider

Note: *This list increases as products migrate or new products are developed in high-level languages using the extended mode compilers.*

TEST Environment Limitations

The TEST installation environment is designed to test specific levels of compilers, run-time systems, or static linkers and must be used only for that purpose. It is not designed to be used as a production system. The following limitations apply to applications executed within the test installation environment:

- Only one test installation environment can exist on any system. Because each test installation uses the same library code name (LCN), the previous test version must be deinstalled. SOLAR automatically deinstalls previous test installation versions during the installation of a new test product.

- All object modules must be relinked with the static linker to change library code names UCS$EMLIB and UCS$EMUSER to ALT$EMLIB and ALT$EMUSER. All links targeting the Runtime System for Extended Mode Compilers test installation environment must contain the following CHANGE commands:

  ```
  CHANGE LCN (UCS$EMUSER) TO ALT$EMUSER
  CHANGE LCN (UCS$EMLIB) TO ALT$EMLIB
  ```

- Object modules executed in the test installation environment cannot be used on other systems and cannot be used in combination with object modules that have not been relinked to change the library code names.

- The TEST installation environment does not support HVTIP executions.

Refer to the *Application Development Solutions Application Development Programming Guide* (7831 4077) for a description of special considerations and the steps you need to take when setting up an extended mode environment on a system.

4.61. Service Library (SLIB)

Mode A is the default installation mode for SLIB. SLIB has an object module subsystem.

4.62. Shared File System (SFS)

Shared File System has the following installation modes:

- COLLECTOR

  This is the default mode, which installs the Shared File System files (SYSLIB*SFS and SYSLIB*SFSLIB) and registers the SFSLIB file with the Collector. Installing the COLLECTOR mode removes Shared File System from any other application group with Collector registration.

- NOCOLLECTOR

  This mode is similar to the COLLECTOR mode except the SFSLIB file is not registered with the Collector. This mode is equivalent to the TEST=MAP keyword on the COMUS INSTALL command. Installing the NOCOLLECTOR mode has no effect on other application groups.

4.63. Software Library Administrator (SOLAR)

SOLAR has an object module subsystem. It is important to read Appendix A before installing it. SOLAR can execute without the subsystem to register packages, install/remove/save products, or generate reports. If you do not set up the subsystem correctly, the Feature Key Verification (FKEY) utility provided by SOLAR does not operate properly, causing products that call FKEY to fail.

You must not restrict general read access to SYSLIB*SOLAR and SYSLIB*SOLAR-GDEF through ACRs or any other means. The file SYSLIB*SOLAR is included in the Collector (MAP) library search chain to allow basic mode products to easily map in the FKEY utility entry points. The file SYSLIB*SOLAR-GDEF is included in the Linking System subsystem search chain to allow extended mode products to easily link in the FKEY utility entry points. A user-id must have read access to these files to perform a MAP or LINK.

4.64. Sort/Merge (SORT)

SORT can only be installed with SOLAR. SORT CONFIGURE is still supported through COMUS, however, and requires a COMUS registration of SORT.

The following modes are available for SOLAR installation. Each mode installs the common banked errors (SM$ERR) file.

- A

  This is the default mode. It installs the following:
  - Common banked processor (SORT)
  - Nonreentrant processor (SORTNR)
  - Relocatable subroutines (REL$)
  - Common banked subroutine (SM$SUB)
  - Extended Mode Linkages

- B

  Installs the following:
  - Common banked processor (SORT)
  - Relocatable subroutines (REL$)
  - Common banked subroutine (SM$SUB)
  - Extended Mode Linkages

- C

  Installs the following:
  - Nonreentrant processor (SORTNR)
  - Relocatable subroutines (REL$)

- D

  Installs the relocatable subroutines (REL$)

- E

  Installs the following:
  - Relocatable subroutines (REL$)
  - Common banked subroutine (SM$SUB)
  - Extended Mode Linkages

- F

  Installs the following:
  - Common banked subroutine (SM$SUB)
  - Extended Mode Linkages

4.65. Symbolic Stream Generator (SSG)

SSG has the following installation modes:

- A

  This mode registers the SSG processor with the Exec. Mode A is the default mode.

- B

  This mode does not register the SSG processor with the Exec. Thus, the processor call must include the name of the file that contains the processor.

4.66. System Interface for Legacy Application Systems (SILAS)

SILAS has the following installation modes:

- A

  Use this mode to install SILAS in a normal production environment. This is the default installation mode. Installation of SILAS mode A requires CPComm or CPCommOS modes A and AFCB. This mode installs SILAS into SYSLIB*SILAS$A.

  Installing SILAS mode A also installs the elements STSILAS-A and SILASDUMP-A into the alternate RUN$ file (SYSLIB*RUN$). Use runstream STSILAS-A to start mode A of SILAS. Use runstream SILASDUMP-A to capture SILAS diagnostics through RSS.

- B through H

  Use modes B through H to install SILAS in a normal production environment. Installation of SILAS mode x (where x denotes the installation mode) requires CPComm or CPCommOS modes x and AFCB. This mode installs SILAS into SYSLIB*SILAS$$x$.

  Installing SILAS mode x also installs the elements STSILAS-x and SILASDUMP-x into the alternate RUN$ file (SYSLIB*RUN$). Use runstream STSILAS-x to start mode x of SILAS. Use runstream SILASDUMP-x to capture SILAS diagnostics through RSS.

- UOC

  This mode installs the file SILAS*UOC$AFCB which is used for the TIP user-own-code feature. In addition, the Exec loads a user-own-code element into an alternate file common bank (AFCB) called SILAS$UOC.

Starting SILAS with Default Runstreams

Eight runstreams are available for starting SILAS, corresponding to SILAS installation modes A through H. Runstream STSILAS-*x* (where *x* denotes the installation mode) starts mode *x* of SILAS that pairs with CPComm or CPCommOS mode *x*.

Using any of these runstreams has the following prerequisites:

- You must install the corresponding CPComm or CPCommOS (modes A through H).

- You must install CPComm or CPCommOS mode AFCB

- You must use SOLAR to install SILAS.

- You must catalog a program file called SILAS$*CONFIG$. This file must be assignable and readable by the SILAS start runstreams.

- You must have the Remote System Support processor installed on your system.

4.67. Tape Labeling Utility for ClearPath OS 2200 (TUTIL)

TUTIL has two installation modes: A and Test

- A

  Mode A is the default. The normal runstreams are installed into the standard alternate RUN$ file (SYSLIB*RUN$). The TUTIL program is installed as a registered processor in the file SYSLIB*TUTIL. The following runstreams are installed in the alternate RUN$ file:

  – INLBL

  – EINLBL

  – BKLBL

  – EBKLBL

  – RELBL

  – ERELBL

  – UNLBL

  – PRLBL

  – PRLBLS

  Note: *If TeamQuest SAUTILITIES is installed or a previous version of TUTIL is installed, you must deinstall it before installing TUTIL.*

- Test

  This mode allows TUTIL to be installed and coexist with a previous version of TUTIL or with the TeamQuest SAUTILITIES (TLABEL or TLBL). The runstreams all have a version name matching the release level, and are installed in the file SYSLIB*TUTILTST (not the normal alternate RUN$ file). The following runstreams are installed:

- INLBL/1R1B

- EINLBL/1R1B

- BKLBL/1R1B

- EBKLBL/1R1B

- RELBL/1R1B

- ERELBL/1R1B

- UNLBL/1R1B

- PRLBL/1R1B

- PRLBLS/1R1B

The TUTIL program is also installed in the file SYSLIB*TUTILTST. When it is installed in this mode, TUTIL is not registered as a standard processor and the full filename must be specified when it is called.

Restriction

The existing release does not support Tape Volume Security (TVSL > 0).

4.68. TCP/IP Application Services (TAS)

TCP/IP Application Services (TAS) has the following installation modes:

- Mode A

  Provides FTP and Mail support under menu (DPS) or IPF

- Mode B

  Provides FTP and Mail support under menu (DPS) only

- Mode C

  Provides FTP and Mail support under security menu (DPS) or IPF

- Mode D

  Provides FTP and Mail support under security menu (DPS) only

- Mode E

  Provides FTP and Mail support under IPF

- Mode F

  Provides FTP and Mail support without IPF or menu (DPS)

- Mode G

  Provides FTP with menu (DPS) and IPF

- Mode H

  Provides FTP with menu (DPS)

- Mode J

  Provides FTP with IPF

4.69. Transaction Processing Utility (TIPUTIL)

The Transaction Processing utility (TIPUTIL) has the following installation modes:

- UCS mode

  TIP primitives (for file access only) are installed for use in extended mode TIP programs. The communications interface must be the Message Control Bank (MCB).

- TPE mode

  TIP primitives (for file access and message handling) are installed for use in basic mode TIP programs. This is the default mode. The communications interface can be either MCB or COMPOOL.

4.70. UniAccess for OS 2200 Product Suite

The UniAccess for OS 2200 Product Suite is comprised of the following products:

- UniAccess ODBC Server for Relational Database Server for ClearPath OS 2200

  The UniAccess ODBC Server for Relational Database Server provides ODBC capability to the Relational Database Server. The ODBC capability enables you to access Relational Database Server data using a variety of Windows-based client tools and applications.

- UniAccess Transaction Server for ClearPath OS 2200

  The UniAccess Transaction Server for ClearPath OS 2200 allows ODBC-compliant and Open Client applications to invoke Unisys TIP transactions on the OS 2200 system using UniAccess Server-Library (UASL). Using UASL, OS 2200 programmers can create transactions to access a variety of OS 2200 data.

- UniAccess Transaction Client for ClearPath OS 2200

  The UniAccess Transaction Client for ClearPath provides client capabilities on the ClearPath OS 2200. It allows Unisys host-based applications remote access to Microsoft and Sybase SQL Servers. In addition, transaction client applications can access ClearPath OS 2200 TIP transactions through the UniAccess ODBC Server for Relational Database Server and UniAccess Transaction Server.

  UniAccess requires the use of product keys. Refer to the *UniAccess for OS 2200 Systems Read.Me* (7850 5021) file for information on obtaining the product key and other release information.

The UniAccess ODBC Server media has the following installation modes:

- A

  Use this mode for installing the base feature release. This is the default installation mode. It is also the base production mode for the feature release.

  The UniAccess software is loaded into the production files. The UniAccess fixed gate subsystem is installed using the Unisys allocated BDI (0405220).

- B

  Use this mode for installing a stability update version. This is the maintenance update production mode. This mode is used to install a stability release, if one is made available. Until a stability release is available, mode A and mode B are equivalent.

  The UniAccess software is loaded into the production files. The UniAccess fixed gate subsystem is installed using the Unisys allocated BDI (0405220).

- C

  Use this mode for installing a test version of the base feature release. This is the base test mode. This mode is used to install an alternate or test version of UniAccess.

  The UniAccess software is loaded into test files. The UniAccess fixed-gate subsystem is installed using a site selected BDI.

- D

  Use this mode for installing a test version of a stability update release. This is the maintenance update test mode. This mode is used to install a stability release, if one is made available. Until a stability release is available, mode C and mode D are equivalent.

  The UniAccess software is loaded into the test files. The UniAccess fixed-gate subsystem is installed using a site selected BDI.

Refer to the *UniAccess for OS 2200 System Administration Guide*, *UniAccess for OS 2200 Open Client Guide*, *UniAccess for OS 2200 Server-Library Programming Reference*, and the *UniAccess for OS 2200 Client-Library Programming Reference* for information on the UniAccess product suite. Those documents are provided in electronic form on the UniAccess Client CD-ROM image.

UniAccess includes a subsystem.

4.71. Universal Database Control (UDSC)

Universal Database Control is installed for every application group requiring UDSC services. Universal Database Control is composed of two types of subsystems: one protected and one chameleon. The owner of these files must possess certain security attributes.

The installation mode you select identifies the application group installation you are registering with SOLAR. The UDSC name contains the application group number. For example, the mode name for application 3 is AP3, the mode name for application 16 is AP16, and so on. Since there are a maximum of 16 application groups, there are 16 possible mode names for UDSC (AP1 through AP16).

Installing the default application (typically application 3) might require you to supply additional information to the installation in the form of an SGS (to control initialization of the ADT-FILE). Refer to the following documents for more information concerning UDSC installation and configuration:

- *ClearPath Enterprise Servers Universal Data System Configuration Guide* (7844 8362)

- *ClearPath Enterprise Servers Universal Data System Planning and Installation Overview* (7844 8370)

4.72. Universal Compiling System (UPLS)

Universal Compiling System (UPLS) is intended for internal Unisys use. No documentation is available for this product

UPLS has the following installation modes:

- DEFAULT

  This is the default installation mode recommended for customer sites. The UPLS compiler installed by this mode is a nonreentrant, program-banked compiler.

  This mode automatically deinstalls mode A and automatically installs mode BUILD.

  To call the compiler installed by this mode, enter @UPLS.

- A

  This mode requires that Language Support System mode A or mode S also be installed.

  This mode installs UPLS as part of the object module subsystem defined by Language Support System.

  This mode automatically deinstalls mode DEFAULT and automatically installs mode BUILD.

  To call the compiler installed by this mode, enter @UPLS.

- BUILD

  This is a special compiler mode that installs the compilers used to build C Compiler, COBOL Compiler, and FORTRAN Compiler. These BUILD compilers are not registered and are installed automatically by modes A and DEFAULT.

  This mode automatically deinstalls mode A and mode DEFAULT.

- X

  This installation mode is for Unisys use only.

4.73. User Authentication (FLEX)

User Authentication allows your site to implement types of authentication for demand and TIP system access other than the traditional memorized passwords, without modifying your local operating system. Authentication Modules (AM) implementing any form of authentication can be plugged into the Authentication and Session Initiation Subsystem (ASIS). ASIS provides the infrastructure, linking to the AMs that implement the decision-making policies.

Authentication Modules can be supplied by Unisys, third parties, or sites can provide Site Specific Authentication Modules (SSAM) to implement their own authentication methods.

User Authentication has two installation modes.

- NORMAL

  This mode creates an entry in SYSLIB*RUN$.AUTO$START for the ASIS background run that cause the ASIS background run to automatically be started on each system boot.

- NOAUTO

  This is the default installation mode. This mode does not create the AUTO$START entry and requires that the ASIS background run be started manually. User Authentication includes a protected subsystem.

The *User Authentication Administration Guide* (7850 4586) provides details on the features of the User Authentication product, how to set up the required operating environment, and how to write authentication modules.

You must be careful when you reinstall User Authentication. You must terminate the ASIS background run during the SOLAR installation process to prevent users from logging on to the system. The preferred method, entered via the @ASISUT processor or the ASIS console command keyin, is:

```
SHUTDOWN PURGE
```

This shuts down ASIS in a controlled manner and allows users to log on to the system using traditional Exec authentication while ASIS is unavailable. If the ASIS background run is not terminated by the SHUTDOWN command, it terminates abnormally when the SOLAR install does a subsystem deactivation of ASIS. If ASIS terminates abnormally or the PURGE option is not used on the SHUTDOWN command, users attempting to log on receive the following message:

```
The Authentication Subsystem is not available.  This sign-on is
rejected.  Contact your site administrator.
```

If you receive this message, enter an @@SOLI keyin. This keyin solicits a user-id and password with the following message:

```
Enter your user-id/password and clearance level:
```

If "Traditional authentication allowed" is not selected in the system security environment (or TRADITIONAL_AUTHENTICATION = DISABLED is set in the system security environment), then only the security officer is allowed to sign on using their traditional Exec user-id and password. If the security officer user-id is not being used to install User Authentication, take care to ensure that a demand session is open and that it remains open to allow accomplishing the remaining installation steps.

4.74. Utilization Report Utility for OS 2200 (URU-OS2200)

The Utilization Report Utility for OS 2200 (URU) consists of two portions:

- URU Windows Component

  The URU Windows Component is installed on a Windows platform. This Windows Component consists of a client Utilization Report Utility program along with a Windows service. The URU Windows Component is installed from the URU installation CD-ROM image. The installation instructions are included on the CD-ROM image in the ReadMe file.

- URU Background Run

  The URU background run is installed on the OS 2200. This is a SOLAR-installable product. The installation and configuration instructions for the background run are included on the URU installation CD-ROM image in the ReadMe.doc file.

Refer to the *ClearPath OS 2200 Utilization Report Utility Help* (3826 6441) for more information on URU.

4.75. WebSphere® MQ Version 9 for ClearPath OS 2200 (WMQ2200)

A complete installation of the WMQ2200 product installs the product on both the OS 2200 system and the OS 2200 QProcessor. WMQ2200 has two OS 2200 installation modes, and sixteen OS 2200 QProcessor installation modes.

Mode WMQ$A is the default OS 2200 installation mode and installs a set of eight OS 2200 files.

Mode WMQ$B installs the same set of eight files plus two additional files that provide the basic mode interface for the Business Information Server for ClearPath OS 2200 product.

For both modes WMQ$A and WMQ$B, a successful installation of the WMQ2200 product catalogs and populates the following files in the OS 2200 environment:

- WMQ$*MQS$COMMON.
- WMQ$*MQS$LIB, the extended mode library file.
- WMQ$*MQS$EXE, the executables file.

- WMQ$*MQS$RUN, the batch runstreams file.

- WMQ$*MQS$SAMPLES, a samples programs file.

- WMQ$*MQSTOOLS, the tools and utilities file.

- WMQ$*MQS$OM, the compiled or assembled object modules file.

- WMQ$*MQS$CONFIG, the configuration file

A successful mode WMQ$B installation also catalogs and populates these two files:

- WMQ$*MQS$BMCOMMON, the shared object module subsystem definition, and intercepts file for the basic mode interface system.

- WMQ$*MQS$BMLIB, the fixed-gate file for the basic mode interface system.

16 installation modes exist for installing WMQ2200 on the OS 2200 QProcessor. Installation modes Q$1 through Q$16 correspond to the 16 possible Specialty Partitions as configured in the IC$*Processors file and managed by the ICADMIN utility. See the *ClearPath Specialty Engine for OS 2200 Configuration Guide* or the *ClearPath Dorado 6300 Server Installation, Administration, and Operations Guide Volume 2: Enterprise Partition Platforms* for more details on ICADMIN.

For each of the OS 2200 QProcessor installation modes (Q$1-Q$16), the following two files are installed to the OS 2200:

- WMQ$*MQS$RPM, a set of RPM files that are installed on the OS 2200 QProcessor. This file is removed after the installation is complete.

- WMQ$*MQS$QP, a set of elements that control the installation and removal of the WebSphere MQ RPM files on the OS 2200 QProcessor.

WebSphere MQ enables you to generate multiple alternate WMQ2200 installations at your site. Each alternate installation requires a unique file qualifier for its product file set, as well as unique BDIs. The product is an absolute-only product; no PRIMUS changes or local code can be applied through this build capability. You can install both test and production versions of WMQ2200 on the same OS 2200 host.

For additional configuration information, see the *WebSphere® MQ Version 9 for ClearPath OS 2200 Installation, Administration, and Programming Guide*.

4.76. Web Transaction Server (WEBTS)

Refer to the *ClearPath Web Enablement Solutions Web Transaction Server for ClearPath OS 2200 Administration Guide* (7850 4073) for configuration information.

Web Transaction Server includes a subsystem.

Section 5
Software Build Instructions

Software generation is the process used to apply corrections to the source code or to modify the configuration of a software product. The COMUS BUILD command generates OS 2200 software products. This document describes only the major steps needed to generate OS 2200 products.

You can generate software only if your site has symbolic maintenance capability. In the following situations, you must generate the product before you install it:

- Unisys sends you a code change to resolve a problem. You must apply the code change as a temporary correction, so that you can use the product until you receive a maintenance update that incorporates the change.

- You need to apply local code changes to the product to ensure it meets the requirement which is unique to your site.

To apply temporary corrections or local code changes to a product, use COMUS as follows:

1. Use the REGISTER command to register the product release tape or the master tape output from a previous generation in the COMUS database, if you have not completed this task.

2. Use the INSERT command to insert the changes in the COMUS database.

3. Use the BUILD command to generate a new product master tape containing new product files and elements that incorporate the changes.

You should always use the level of COMUS associated with the ClearPath 2200 release level on your system.

5.1. Associated Documents

See the following documents for additional information:

- *ClearPath OS 2200 COMUS End Use Reference Manual* (7830 7758)

  This manual is a comprehensive reference manual for COMUS. COMUS is the software tool used to build and configure OS 2200 software products.

- *ClearPath OS 2200 Executive Control Language (ECL) and FURPUR Reference Manual* (7830 7949).

  This manual provides a complete reference to all Executive Control Language (ECL) and File Utility Routines/Program File Utility Routines (FURPUR) statements and describes the purpose, format, options, and individual fields of each statement. Examples appear throughout the manual to help you understand the various uses of these statements.

- *ClearPath OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604)

  This manual describes the SOLAR processor and utilities that a site can use to install and maintain their software products.

- *ClearPath OS 2200 Symbolic Stream Generator (SSG) Programming Reference Manual* (7831 1784)

  This manual describes the use of SSG to generate, maintain, and update the proper system configuration for OS 2200 Exec, processors, and libraries.

- *ClearPath OS 2200 Software Planning and Migration Overview for Release 18.0* (7831 0349)

  This overview contains a product interdependency matrix that you can use to determine product dependencies.

5.2. Software Generation Preparation

Software generation requires the following preparation steps. After these steps are completed, COMUS is used to create the updated product. When the software generation is completed, the updated product is ready for installation.

5.2.1. Installation of Required Products

Install the software products required to generate the product. Table 4-1 lists the products you need to install before you can build any product. See the *OS 2200 Software Library Administrator (SOLAR) User Reference Manual* (7831 0604) for more information on installing products.

5.2.2. COMUS Database Initialization

Use the **COMUS COINIT** processor to initialize the COMUS database if it has not already been initialized. See the *ClearPath OS 2200 COMUS End Use Reference Manual* for more information about the COINIT processor, a discussion of when you need to initialize the database, and a complete example.

5.2.3. COMUS Database Product Registration

Use the **COMUS REGISTER** command to register the product in the COMUS database. The SOLAR PKGREG utility does not replace the COMUS REGISTER command. Before you can generate a product you must register it with the COMUS database. The **COMUS REGISTER** command copies the control information necessary to execute the BUILD and other COMUS commands copies into the COMUS database. See the *ClearPath OS 2200 COMUS End Use Reference Manual* for more information.

5.2.4. Application of Source Code Change

If you are applying a source code change, create a change (CHG) document for the product by using the **COMUS INSERT** command. See the *ClearPath OS 2200 COMUS End Use Reference Manual* for more information.

5.3. Product Generation

Use COMUS to generate a product to create a new product master tape that you can use to install or load the product on your system.

Following is the input to the software generation process:

- A software package tape (also called the product master tape) or a set of mass storage files from a previous product generation.

- Any changes for the product that you have already inserted in the COMUS database and now want to apply to the product.

- A newly created or updated configuration element (if applicable) in a program file on mass storage.

- Tapes containing optional product features (if applicable).

Following is the output from the software generation process:

- A tape (called the new master tape) or a set of mass storage files that contains the updated product generation files, including a new product absolute.

- Printed listings or listings sent to mass storage or tape that document the generation.

5.3.1. First Generation and COMUS Database Build Default Values

The first time COMUS is used to generate a product level, COMUS requests default values for all subsequent product generation runstreams. While executing the BUILD command, COMUS queries for the information needed to build the generation runstream. Each question lists its possible responses.

If you do not understand a question, or require more information on possible responses, respond with the COMUS help character, which is a question mark "**?**".

If you enter no response and transmit (called a "null transmit"), COMUS automatically uses the default response shown between less than, greater than signs (<>).

If you want to view or redefine the product defaults in subsequent builds, use the **Q** option on the **BUILD** command to get the following prompt:

```
Redefine the product defaults (LIST, Y, or <N>) ? ▶
```

L or **LIST**   The current default values will be listed and the preceding prompt repeated.

Y or **YES**   Questions will be asked to redefine the defaults.

N or **NO**   The current default values will be used and processing will continue

In the following COMUS Build Session examples:

- ▶ represents user input is expected.
- **ABC** is a variable representing the COMUS database qualifier.
- **PRODUCT** is a variable representing the product being generated.
- **xRy** is a variable representing the level of the product being generated.

The following prompts for default values are applicable to all products. Additional prompts unique to each product are listed in that product's "COMUS Default Values" section.

```
▶@QUAL ABC
```

This statement specifies *ABC* as the COMUS database qualifier, which is used to access the COMUS database in this session. All database file names have a specified qualifier.

```
▶@COMUS
>COMMAND ?   ▶BUILD PRODUCT,xRy
```

This BUILD command directs COMUS to apply changes to PRODUCT level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level. For example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

Define *PRODUCT* default values

Your responses to the COMUS prompts for default values become the product defaults and are enclosed in angle brackets < > during the subsequent build sessions. If you respond with **Q** for query to any prompt, COMUS does not create a default and resolicits the value during subsequent build sessions.

For each default question you can enter one of the following responses:

1. An appropriate value.

2. A null string or spaces will maintain the current default value.

3. QUERY - This keyword causes COMUS to ask for the default on every BUILD of the product.

4. BLANK - This keyword sets the value of the default to null.

```
Default generation recovery mode (ON or <OFF>) ? >
```

ON	The generated runstream establishes restart points to aid in generation restart in case of a recoverable system stop.
OFF	No restart processing takes place.
QUERY	COMUS asks for the recovery mode during all subsequent generations of the product.

```
Default project id - <Q$Q$Q$> ▶
```

The Default project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Default runid - <> ▶
```

The Default run ID is used in the RUN-ID field of the @RUN control statement.

```
Default run options - <> ▶
```

The Default run options are the options used on the @RUN control statement.

```
Default run priority - <> ▶
```

The Default run priority is used in the SCHEDULING-PRIORITY field of the @RUN control statement.

```
Default tape equipment type - <T>   ▶
```

Enter the default tape equipment type that is to be used in tape assignment statements for all generations of **PRODUCT xRy**. If **QUERY** is entered, the tape equipment type will be solicited on every generation.

```
Default tape assign options - <TF>  ▶
```

Enter the default tape assign options that is to be used in tape assignment statements for all generations of **PRODUCT xRy**. If **QUERY** is entered, the tape equipment type will be solicited on every generation.

```
Default generation type (DISK/DISK or <TAPE/TAPE>) ? ▶
```

Enter the type of input and output media to be used. COMUS will use the default generation type in subsequent generations.

DISK/DISK
COMUS will not query for generation type in subsequent build generations. All generations will be DISK/DISK.

TAPE/TAPE (recommended)
COMUS will not query for generation type in subsequent build generations. All generations will be TAPE/TAPE.

QUERY
COMUS will query for the generation type in every subsequent generation.

You may then enter a generation type other than DISK/DISK or TAPE/TAPE. Generation types are as follows:

D/D	Disk input, disk output
D/DI	Disk input, disk output. Initial - new file names will be solicited for the output files.
D/T	Disk input, tape output
T/T	Tape input and tape output. This is most common and is used for this example.
T/D	Tape input and disk output. Disk file names will solicited.
	Refer to the *ClearPath OS 2200 COMUS End Use Reference Manual* for more information on generation types.

> **Note:** *The build examples in this manual assume the default generation type as TAPE/TAPE.*

```
Permanent SGSs <END> ? ▶
```

As part of the product defaults, COMUS allows you to specify permanent stream generation statements (SGS). COMUS automatically includes these SGSs in all subsequent generation runs of and retains them with the other product defaults. If you enter an SGS, COMUS repeats the question until you enter END or a null

transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* for more information on SGSs.

5.3.2. Requesting a Generation Print Output

During the selection of COMUS default values, most products (but not all) query for the following.

```
PRODUCT printout: tape, printer or query - <T>/P/Q ? ►
```

Define the normal handling of generation printout.

T	tape	Save generation printout on tape
P	printer	Queue generation printout to printer
Q	query	Determine handling at generation time

Your selection will result in different queries in the subsequent generation sessions. If you select "Q", this query will be repeated during each subsequent COMUS build session.

Requesting a Generation Print Output Tape with Print Consolidation

Queries that occur during the BUILD command dialog provide a choice of tape or printer for the generation output device and the option of using print consolidation, which places all generation output on one tape reel. The default responses T then Y for the following BUILD command dialog default queries request print consolidation for tape output:

```
PRODUCT printout: tape, printer, or query - <T>/P/Q ? ►T
Print consolidation required - <Y>/N ? ►Y
Consolidate print - <Y>/N ? ►Y
```

This reconfirms the request for print consolidation.

```
OLDCONSP - reel/file./NONE/<> ? ►
```

Requests the output tape reel number or file from a previous **PRODUCT** generation to be merged with the new output from this generation.

Following are the possible responses:

- **Tape reel** specifications: **reel1[/.../reeln][,options,*type*,expiration-period]**

   For *type*, specify a high-density tape equipment type to fit all output on one physical tape reel.

- **Cataloged tape file** specifications:
  filename.[,options,*type*,reel1[/.../reeln],expiration-period]

  For *type*, specify a high-density tape equipment type to fit all output on one physical tape reel.

- **NONE or null** (transmit without typing anything) response for no print consolidation tape reel or file from a previous generation

```
NEWCONSP - reel/file./NONE/<> ? ▶
```

Requests the output tape reel number or file for this **PRODUCT** generation. Possible responses:

- **Tape reel number** specifications:
  reel1[/.../reeln][,options,*type*,expiration-period]

  For *type*, specify a high-density tape equipment type to fit all output on one physical tape reel.

- **Cataloged tape file** specifications:
  filename.[,options,*type*,reel1[/.../reeln],expiration-period]

  This file must already be cataloged before the generation runstream starts. For *type*, specify a high-density tape equipment type to fit all output on one physical tape reel.

- **NONE or null** (transmit without typing anything) response. This places the CO$PRINT file on the new master tape created by the generation. This is the tape reel or file specified in response to the NEWMAST query during the BUILD command dialog.

For tape reel number and tape file specifications, see the tape assignment (@ASG) executive control language statement description in the *OS 2200 Executive Control Language (ECL) End Use and Programming Reference Manual*.

Requesting a Generation Print Output Tape without Print Consolidation

Queries that occur during the BUILD command dialog provide a choice of tape or printer for the generation output device and the option of using print consolidation, which places all generation output on one tape reel. The responses **T** then **N** for the following BUILD command dialog default queries request tape output without print consolidation:

```
product printout: tape, printer, or query - <T>/P/Q ? ▶
```

The **T** response during the initial build default session will result in all generation print output being saved to the NEWMASTER tape.

If Q was chosen during the initial build default session, this query will appear during each subsequent COMUS build session:

```
Printout to tape or printer - <T>/P ? ▶
```

This specifies the destination of the generation printout.

Note: *This query will not appear if a default type was specified during COMUS default selection.*

P	It is queued to the default printer specified in the COMUS database.
TAPE or **T**	It is saved on the output NEWMASTER tape (or disk if the output NEWMASTER is a disk). The following query will appear:

```
Print will be saved - queue for output too - Y/<N> ?  ▶
```

Y: It is saved on the output NEWMASTER and you are queried for an output device:

```
Output device - <    > ?
```

N: It is saved on the output NEWMASTER and you are not queried for an output device.

```
Print consolidation required - <Y>/N ?      ▶N
```

The preceding responses produce the following additional queries in every BUILD command dialog:

```
Consolidate print - <Y>/N ? ▶
```

This reconfirms the request for print consolidation.

```
Assign a new tape for the generation output ? <Y>/N ▶
```

- The default answer **Y** produces the next prompt.
- Answer **N** to send generation output to a printer instead of tape, in which case the next prompt does not occur.

```
GENOUT - reel/file./NONE/<> ? ▶
```

Requests the output tape reel number or file for this *PRODUCT* generation. Possible responses are as follows:

- **Tape reel number** specifications:
  reel1[/.../reeln][,options,*type*,expiration-period]

  For *type*, specify a high-density tape equipment type to fit all output on one physical tape reel.

- **Cataloged tape file** specifications:
  filename.[,options,*type*,reel1[/.../reeln],expiration-period]

  This file must already be cataloged when the generation runstream starts. For *type*, specify a high-density tape equipment type to fit all output on one physical tape reel.

- **NONE or null** (transmit without typing anything) response. This places the PRNTF$ file on the new master tape output by the generation. This is the tape reel or file specified in response to the NEWMAST query during the BUILD command dialog.

For tape reel number and tape file specifications, see the tape assignment (@ASG) executive control language statement description in the *OS 2200 Executive Control Language (ECL) End Use and Programming Reference Manual*.

Requesting a Generation Print Queued to a Printer

Queries that occur during the BUILD command dialog provide a choice of tape or printer for the generation output device. The response **P** requests the print be sent to a print queue.

```
product printout: tape, printer, or query - <T>/P/Q ? ▶
```

The **P** response during the initial build default session will result in all generation print output being queued to the default printer specified in the COMUS database.

If Q was chosen during the initial build default session, this query will appear during each subsequent COMUS build session.

```
Printout to tape or printer - <T>/P ? ▶
```

This specifies the destination of the generation printout.

Note: *This query will not appear if a default type is specified during COMUS default selection.*

P It is queued to the default printer specified in the COMUS database.

TAPE or **T** It is saved on the output NEWMASTER tape (or disk if the output NEWMASTER is a disk). The following query will appear:

```
Print will be saved - queue for output too - Y/<N> ? ▶
```

Y: It is saved on the output NEWMASTER and you are queried for an output device:

```
Output device - <     > ?
```

N: It is saved on the output NEWMASTER and you are not queried for an output device.

5.3.3.   Console Messages

The following message displays at the system operator's console to indicate the start of the generation runstream. **Pro** is a variable that is a 3 character version of the product name. For example, SRT indicates SORT, TAS indicates TAS, and so on.

►*pro*GEN START

A LOAD message displays to tell the operator to load the current product release master tape on a particular tape drive. In this example, it is drive TAPEG0.

►LOAD R12345 TAPEG0 TAPE – 1 *pro*GEN

A LOAD message displays for a blank, unlabeled tape that becomes the new product master tape.

►LOAD UNLABELED BLANK M12345 TAPEG0 RING TAPE – 1 *pro*GEN

A FIN message displays at the system operator's console when the product generation runstream is finished executing.

►*pro*GEN FIN

5.4. Software Product COMUS Build Sessions

This section provides examples of COMUS Build Sessions unique to each product. These examples assume the product has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, refer to 5.3.1.

If you do not understand a question, or require more information on possible responses, respond with the COMUS help character, which is a question mark "**?**".

If you enter no response and transmit (called a "null transmit"), COMUS automatically uses the default response shown between less than, greater than signs (< >).

In the following COMUS Build Session examples:

- Text following a ► represents user input.
- **ABC** is a variable representing the COMUS database qualifier.
- **xRy** is a variable representing the level of the product being generated.

For more information on the BUILD command, see the *ClearPath OS 2200 COMUS End Use Reference Manual*.

5.4.1. ACOB

ACOB COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the ACOB default session:

```
Number of elements per print file – <10> ? ►
```

This question requests the number of elements to be assembled within each file on the print tape or in each part of a break pointed print file. A 10 response

assembles 10 elements within one file on the print tape or in one part of a break pointed print file.

```
Do you have a user common bank matrix (USERSGS) element – Q/Y/<N> ? ▶
```

This question asks whether you have a USERSGS element to define user common banks. The USERSGS element contains SGSs which are needed to define the contents of user subprogram common banks.

Q or **QUERY**   COMUS queries during each ASCII COBOL build whether you have a USERSGS element to define user common banks.

Y or **YES**   You have USERSGS element. The name of this element is requested during each ASCII COBOL build.

N or **NO**   You do not have a USERSGS element.

```
Do you want to build an ASCII or Fieldata C$DML common bank
–  Q/F/<A> ? ▶
```

This question asks whether an ASCII or Fieldata C$DML common bank should be built. The released version of the C$DML common bank contains only the ASCII C$DML routines. The Fieldata version of C$DML common bank contains both the ASCII and the Fieldata C$DML routines. Unless your programs contain extensive use of USAGE COMP-4 or USAGE DISP-1 data items, an ASCII C$DML common bank should be built.

Q   Requires you to decide at each ASCII COBOL build whether an ASCII or Fieldata C$DML common bank should be built.

F   Always build a Fieldata C$DML common bank.

A   Always build an ASCII C$DML common bank.

```
Enter the UCSRTS library filename –filename/QUERY/<SYS$LIB$*UCSRTS> ? ▶
```

This question asks which file should be used for the UCSRTS library. UCSRTS is a product that contains subroutines that the ASCII COBOL run-time system uses for memory management. The ASCII COBOL common bank C$INITIAL contains the UCSRTS relocatable elements. A **QUERY** response means COMUS requests the file name for the UCSRTS library during each ASCII COBOL build.

```
Do you want common bank or relocatable SORT –  Q/R/<C> ? ▶
```

This question asks whether the ASCII COBOL common bank C$SORT interfaces with the SORT common bank or contains the SORT relocatable elements.

Q   COMUS asks how the ASCII COBOL common bank C$SORT interfaces with SORT at each ASCII COBOL build.

C   Causes the ASCII COBOL common bank C$SORT to interface with the SORT common bank.

R   Causes the ASCII COBOL common bank C$SORT to contain the SORT relocatable elements.

```
Enter the alternate SORT CBEP (CBEP$$XSORT) filename -
filename/QUERY/<SYS$*SRT$PAR> ? ▶
```

This question requests the file name that contains the alternate SORT CBEP (CBEP$$XSORT) element. For the ASCII COBOL common bank C$SORT to reference the SORT common bank, ASCII COBOL requires the alternate SORT CBEP. Refer to the *SORT Installation Guide* for additional information. If answered with **Q** or **QUERY**, COMUS will request the file name for the alternate SORT CBEP (CBEP$$XSORT) during each ASCII COBOL build.

ACOB COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates ACOB. This example assumes ACOB has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, refer to 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?  ▶ BUILD ACOB,xRy
```

This BUILD command directs COMUS to apply changes to ACOB level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<  >) ?  ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (< >) ?  ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ? ▶
```

Specify the reel number of the product master tape used as an input to the generation. The MASTER reel number is the number of the tape containing ASCII COBOL level *xRy* release tape files. A reel number can contain up to six letters or digits. It is used only to identify the product master tape in a LOAD message when the generation runstream executes.

```
Generation id ? ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Enter your user common bank matrix (USERSGS) element - >(Q*F.E/V) ? ▶
```

This question asks whether you have a USERSGS element to define user common banks. The USERSGS element contains SGSs which are needed to define the contents of user subprogram common banks.

```
Generation type - component/ALL/<UPDATE> ? ▶
```

Specify the following type of generation to be performed:

UPDATE   Only the updated elements will be assembled.

ALL   All elements will be assembled.

component   In addition to the updated elements, specify one or more components. Enter ? to see a list of components.

```
RELTAPE - reel/file./<> ? ▶
```

Specify the reel number of the product master tape used as output to the generation.

```
PRINT - reel/file./<> ? ▶
```

Specify the reel number to be used for the new product master tape to be produced by this generation.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and

OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/ACOBxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started.

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/ACOBxRy'**. After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/ACOB*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED  **************************
COMMAND ?  ▶exit
```

When the preceding runstream has completed its execution without any error, the ACOB level *xRy* will be generated.

5.4.2. CKRS

CKRS COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following default values will also be queried for by COMUS during the CKRS default session:

```
Enter location of PLUS compiler ▶
```

An appropriate level of the PLUS compiler must be available in a cataloged file for use by the generation process. The name of the file (and element, if different from PLS) should be given in response to the location query, using standard QUALIFIER*FILENAME.ELEMENT notation. There is no current default.

```
Enter location of PLUS library routines ▶
```

An appropriate level of the PLUS libraries must be available in a cataloged file for use by the generation process. The name of the file (and element, if different from PLS) should be given in response to the location query, using standard QUALIFIER*FILENAME.ELEMENT notation. There is no current default.

Short or long listings on PLUS compiles ? (S/<L>) ▶

L   A full gen with long (object code) listings uses approximately 3000 pages.

S   A full gen with short (source and cross reference) listings uses approximately 1000 pages.

CKRS COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates CKRS. This example assumes CKRS has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, refer to 5.3.1.

▶@QUAL *ABC*
▶@COMUS
COMMAND ? ▶BUILD CKRS,*xRy*

This BUILD command directs COMUS to apply changes to CML level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

Project id for this generation (<   >) ? ▶

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

Run id for this generation (<   >) ? ▶

The RUNID to be used on the @RUN control statement.

MASTER – reel/file./<> ? ▶

The MASTER reel number is the number of the tape containing the product source code.

Generation id ? ▶

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

Generation heading (<>) ? ▶

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of CKRS.

```
Type of generation (FULL,<UPDATE>) ?
```

FULL   Updates are applied to elements with changes, and all elements are recompiled.

UPDATE   Updates are applied to elements with changes, and only those elements that have been changed are recompiled.

```
Type of CULL (ALL,MSM,PLS,<NONE>)?
```

ALL   A cull of the entire output file.

MSM   A cull of the MASM elements and procedures.

PLS   A cull of the PLUS elements and procedures.

NONE   No cull listing is produced.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/CKRSxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in '*ABC**COMRUN(1).1/CKRS*xRy*'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/CKRS*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED *************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the CKRS level *xRy* will be generated.

5.4.3.   CML

CML COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates CML. This example assumes CML has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, refer to 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?  ▶BUILD CML,xRy
```

This BUILD command directs COMUS to apply changes to CML level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<   >) ?  ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation  (<   >) ?  ▶
```

The RUNID to be used on the @RUN control statement

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
FULL or UPDATE generation - F/<U> ? ▶
```

F or **FULL**	All product components are compiled and the selected changes are applied.
U or **UPDATE**	Only those elements with changes are compiled.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of CML.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/CMLxRy'
View the runstream (Y or <N>) ? ►
Print a copy of the runstream (Y or <N>) ? ►
Start the runstream (Y or <N>) ? ►
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in '*ABC**COMRUN(1).1/CML*xRy*'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement

@START *ABC**COMRUN.1/CML*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED ************************
COMMAND ? ►exit
```

When the preceding runstream has completed its execution without any error, the CML level *xRy* will be generated.

5.4.4. CMR

CMR COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following default values will also be queried for by COMUS during the CMR default session:

```
Enter the file that contains the PLUS compiler <SYS$LIB$*PLS> ►
```

Enter the filename that contains the PLUS compiler. Transmit to use the default of SYSLIB*PLS.

```
Enter file to be used for PLUS RLIB <SYS$LIB$*PLS>  ►
```

Enter filename that will be used for RLIB.PLUS RLIB. Transmit to use the default of SYSLIB*PLS.

```
Enter PLUS compiler options <GM>
```

Enter the options to be used when compiling PLUS elements. The value must be alphabetic. Transmit to use the default options of GM.

```
Enter file that contains the MASM compiler <SYS$LIB$*MASM>
```

Enter the filename that contains the PLUS compiler. Transmit to use the default of SYSLIB*MASM.

```
Enter file to be used for MASM  RLIB <SYS$LIB$*MASM>  ▶
```

Enter filename that will be used for RLIB.MASM RLIB. Transmit to use the default of SYSLIB*PLS.

```
Enter MASM compiler options < > ▶
```

Enter the options to be used when compiling MASM elements. The value must be alphabetic.

CMR COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates CMR. This example assumes CMR has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, refer to 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?  ▶BUILD CMR,xRy
```

This BUILD command directs COMUS to apply changes to CMR level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<   >) ?  ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<   >) ?  ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Do you want the generation output printed (Y,<N>)?  ▶
```

Y   Save the print. If answered yes, you will be queried for a print queue.

```
        Printer for @SYM <PR>? ▶
```

N   Do not save the print

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of CKRS.

```
Type of update (FULL,<UPDATE>)?
```

FULL   Updates are applied, and all elements are recompiled.

UPDATE   Updates are applied to elements with changes, and only those elements that have been changed are recompiled.

```
Type of CULL (ALL,MSM,ASM,FOR,PLS,<NONE>)? ▶
```

Enter the type of elements in the symbolic output file should be CULL'ed. The choices are as follows:

ALL　A cull of the entire output file.

MSM　A cull of the MASM elements and procedures.

ASM　A cull of the ASM elements and procedures.

FOR　A cull of the FORTRAN elements and procedures.

PLS　A cull of the PLUS elements and procedures.

NONE　No cull listing is produced.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/CMRxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y　Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

　　The runstream has been started.

N　You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/CMRxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

　　@START *ABC*COMRUN.1/CMRxRy

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED *************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the CMR level *xRy* will be generated.

5.4.5. COMUS

Required Installation Modes

ACOB

If ACOB is installed using COMUS, then ACOB mode B installation is required to generate COMUS. COMUS requires the NONFLAGGER ACOB compiler, both the Common-banked and relocatable ACOB run-time libraries, and a common-banked data manipulation language (DML).

If you do not have the nonflagger ACOB compiler installed on your system but you have the common-banked and relocatable ACOB run-time libraries and a common-banked DML, you must copy the nonflagger ACOB compiler from the second logical file on the ACOB release tape into a mass storage file. When creating the generation runstream for COMUS, include the following additional generation statement (SGS):

```
CO$ACOBPROCESSOR CALL NAME IS ''QUAL*FILE.ACOB/BANKED''
```

Where QUAL*FILE is the qualifier and file name of the alternate file containing the nonflagger ACOB compiler.

ACOB mode B installation allows file names to be changed. COMUS expects the relocatable run-time library to be in the file SYSLIB*ACOB. COMUS expects the ACOB DML library to be in the file SYSLIB*ACOB-DML. If these files are not used, you must provide an alternate library file SGS as described in this section. If you include the Shared File Services (SFS 2200) in your ACOB library system, the stand-alone version of COMUS includes a reference to an SFS common bank. This common bank is not acessed by the COMUS absolute.

If ACOB is installed using SOLAR, then ACOB mode D installation is required to generate COMUS.

When creating the generation runstream for COMUS, include the following SGS:

```
CO$ACOBLIBRARY FILE IS ''SYS$LIB$*ACOB-CB''
```

The SOLAR installable version of ACOB places the relocatable ACOB run-time libraries into this file.

PCIOS

PCIOS mode B relocatables are required to generate COMUS. COMUS expects the PCIOS relocatables to be in the file SYSLIB*PCIOSREL, the file into which PCIOS mode B is installed. PCIOS mode B does not conflict with either PCIOS modes A or C, therefore modes B and A or modes B and C can both be installed.

If PCIOS mode B is not installed, you must copy the fifth logical file belonging to PCIOS containing the PCIOS relocatables (REL$B) from the software package tape or PCIOS release tape to a mass storage file. In addition, if the file into which the PCIOS relocatables are copied is not named SYSLIB*PCIOSREL, an additional CO$PCIOSLIBRARY SGS with the following format must be specified during the BUILD command:

```
CO$PCIOSLIBRARY FILE IS ''QUAL*FILE''
```

Where QUAL*FILE is the name of the alternate file into which the PCIOS relocatables have been copied.

SORT

SORT mode A relocatable installation is required to generate COMUS.

SPECIFYING LIBRARY FILE LOCATION

If the preceding product installation modes are not installed on your system, you must provide an alternate library file to be used during the generation. The CO$XYZLIBRARY SGS is used to specify the location of the file containing the product library file. When creating the generation runstream for COMUS, include the following SGS:

```
CO$XYZLIBRARY FILE IS ''QUAL*FILE''
```

Where *XYZ* is the name of the product and *QUAL*FILE* is the qualifier and file name of the alternate product library file.

COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2 in the *ClearPath OS 2200 COMUS End Use Reference Manual*, the following values will also be queried for by COMUS during the COMUS default session:

```
Default Maximum Run Time (<9999>): ▶
```

Enter the maximum run time for the @RUN control statement. This is the estimated standard units of processing (SUP) usage in minutes. The maximum value allowed is 9999.

```
Default Maximum Pages (<99999>): ▶
```

Enter the maximum pages for the @RUN control statement. The maximum value allowed is 99999.

```
Send consolidated print output to NEWMASTER (Y or <N>) ? ►
```

Consolidated print is done automatically and sent to a tape. You can choose which tape to send it to. If you choose to send it to an alternate reel other than the NEWMASTER, you will be queried for the reel. The default is to query for the alternate reel. For more information on print consolidation, see 5.3.2 and 5.3.3 in the *ClearPath OS 2200 COMUS End Use Reference Manual*.

```
Default name of the PLUS File (SYS$LIB$*PLS>) ? ►
```

Enter the PLUS Library file to be used for compilations and collections. If a PLUS file is not needed, enter the word BLANK.

COMUS Build Session for COMUS

The following COMUS session example shows the prompts for the BUILD command that generates COMUS. This example assumes COMUS has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
►@QUAL ABC
►@COMUS
COMMAND ?  > BUILD COMUS,xRy
```

This BUILD command directs COMUS to apply changes to COMUS level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<  >) ? ►
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation  (<   >) ? ►
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ? ►
```

The MASTER reel number is the number of the tape containing the product source code.

```
NEWMASTER - reel/file./<> ? ►
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of COMUS.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<  >) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Type of update (FULL,<UPDATE>)?
```

FULL Changes are applied, and all elements are recompiled.

UPDATE Changes are applied, and only those elements that have changed are recompiled.

```
OLDCONSP - reel/file./NONE/<> ? ▶
```

Enter a reel number or file that contains the consolidated print file from a previous generation. This query is used only for UPDATE generations.

```
NEWCONSP - reel/file./NONE/<> ? ▶
```

Enter an alternate reel or file for storing the consolidated print file from this generation. The current generation output will be saved in the file PRNTF$. This file will not be printed during this generation.

```
Type of CULL (ALL or <NONE>) ? ▶
```

CULL generates a printed cross-reference listing for all tags or variables in the source code. You can enter the following type of CULL to be performed:

ALL   Listings are produced for all source code.

NONE   No CULL is performed.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
Generate a Stand Alone COMUS or one that utilizes Common Banked
libraries (SA, CB, or <BOTH>) ? ▶
```

You can generate the following two versions of COMUS:

SA   The first is the standalone version. To generate this version, you must have the ACOB and PCIOS relocatables available on your system.

CB   The second is the common-bank version. To generate this version, you must have the ACOB and PCIOS common-banks installed on your system.

```
The runstream has been saved in 'ABC*COMRUN(1).1/COMUSxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with: **The runstream has been started**

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/COMUSxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START ABC*COMRUN.1/COMUSxRy

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED ************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the COMUS level *xRy* will be generated.

5.4.6.   CPFTP

CPFTP COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following default values will also be queried for by COMUS during the CPFTP default session:

```
Enter the RTS object module library name <SYS$LIB$*EMOMRTS> :    ▶
```

Supply the name of the file that contains the URTS$TABLE object module element.

```
Enter the ELMS library file name <SYS$LIB$*ELMS> :     ▶
```

Supply the name of the file that contains the Extended Message System (ELMS) library.

CPFTP COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates CPFTP. This example assumes CPFTP has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶BUILD CPFTP,xRy
```

This BUILD command directs COMUS to apply changes to CPFTP level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<  >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (< >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ?  ▶
```

Specify the reel number of the product master tape used as an input to the generation. A reel number may contain up to six letters or digits. It is used only to identify the product master tape in a LOAD message when the generation runstream executes.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
NEWMAST - reel/file./<> ? ▶
```

Specify the reel number to be used for the new product master tape to be produced by this generation. A reel number may contain up to six letters or digits. It is used only to identify the tape in a LOAD message when the generation runstream executes.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
Enter type of BUILD: <SYM>/REL/NONE▶
```

SYM Recompiles all cpFTP source elements affected by changes applied during this generation and relinks all executable object module elements. Specify **SYM** if you supplied change numbers for this cpFTP generation. A SYM generation includes a REL generation.

REL Links only executable object module elements. Specify a REL generation when it is necessary to relink cpFTP executable object module elements.

NONE No symbolic updates or linking of executable object modules will be performed.

```
Do you want all elements recompiled (Y/<N>) ? ▶
```

Y Recompile all product elements, including those affected by changes.

N Recompile only elements affected by changes.

```
Enter cpFTP site level <> ▶
```

Specify the site original level. This ID is allowed 3 characters or less. The cpFTP level is combined with cpFTP system level and site original level. The cpFTP level that will be shown on console at the cpFTP master run starting as follows:

xRy00xxx where *xRy00* is the cpFTP system level and xxx is the site original level.

```
Assign a new tape for the generation output ? <Y>/N ▶
```

Note: *This query will not appear if a default type was specified during COMUS default selection.*

N Send generation output to default printer specified in the COMUS. database

Y Send print to a tape. Prompts the following query:

```
GENOUT - reel/file./NONE/<> ? ▶
```

Requests the output tape reel number or file for this *PRODUCT* generation. Possible responses are:

- **Tape reel number** specifications:
  reel1[/.../reeln][,options,*type*,expiration-period]

  For *type*, specify a high density tape equipment type to fit all output on one physical tape reel.

- **Cataloged tape file** specifications:
  filename.[,options,*type*,reel1[/.../reeln],expiration-period]

  This file must already be cataloged when the generation runstream starts. For *type*, specify a high density tape equipment type to fit all output on one physical tape reel.

- **NONE or null** (transmit without typing anything) response. This places the PRNTF$ file on the new master tape output by the generation. This is the tape reel or file specified in response to the NEWMAST query during the BUILD command dialog.

For tape reel number and tape file specifications, see the tape assignment (@ASG) executive control language statement description in the *OS 2200 Executive Control Language (ECL) End Use and Programming Reference Manual*.

```
The runstream has been saved in 'ABC*COMRUN(1).1/CPFTPxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:
The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in '*ABC**COMRUN(1).1/COMUS*xRy*'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/CPFTP*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED   ************************
COMMAND ? ▶EXIT
```

When the preceding runstream has completed its execution without any error, the CPFTP level *xRy* will be generated.

5.4.7. CULL

CULL COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the IACULL default session:

```
Default maximum run time (<9999>) ? >▶
```

Enter the maximum run time for the @RUN control statement. This is the estimated standard units of processing (SUP) usage in minutes. The maximum value allowed is 9999.

```
Default maximum pages (<99999> ) ? ▶
```

Enter the maximum pages for the @RUN control statement. The maximum value allowed is 99999.

```
Send consolidated print output to NEWMASTER (Y or <N>) ? ▶
```

Consolidated print is done automatically and sent to a tape. You can choose which tape to send it to. If you choose to send it to an alternate reel other than the NEWMASTER, you will be queried for the reel. The default is to query for an alternate reel. For more information on print consolidation, see 5.3.2 and 5.3.3.

```
Default name of the PLUS file (<SYS$LIB$*PLS>) ? ▶
```

Enter the PLUS library file to be used for compilations and collections. If a PLUS file is not needed, enter the word BLANK.

CULL COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates CULL. This example assumes CULL has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?  ▶build cull,xRy
```

This BUILD command directs COMUS to apply changes to CULL level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<    >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product

generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<     >) ? ▶
```

The RUNID to be used on the @RUN control statement

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of CULL.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ?  ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Type of generation (FULL or <UPDATE>) ?
```

FULL   Changes are applied, and all elements are recompiled.

UPDATE   Changes are applied, and only those elements that have changed are recompiled.

```
OLDCONSP - reel/file./NONE/<> ? ▶
```

Enter a reel number or file that contains the consolidated print file from a previous generation. This query is used only for UPDATE generations.

```
NEWCONSP - reel/file./NONE/<> ? ▶
```

Enter an alternate reel or file for storing the consolidated print file from this generation. The current generation output will be saved in the file PRNTF$. This file will not be printed during this generation.

```
Type of CULL (ALL or <NONE>) ? ▶
```

CULL generates a printed cross-reference listing for all tags or variables in the source code. You can enter the type of CULL to be performed:

ALL   Listings are produced for all source code.

NONE   No CULL is performed.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/CULLxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

             The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/CULLxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

             @START ABC*COMRUN.1/CULLxRy

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED  ************************
COMMAND ? > exit
```

When the preceding runstream has completed its execution without any error, the CULL level *xRy* will be generated.

5.4.8.   DDP-FJT

DDP-FJT COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following default values will also be queried for by COMUS during the DDP-FJT default session.

```
DDP-FJT generation output: Tape/<Printer>/Query ▶Q
```

Define the normal handling of generation printout:

- Tape: Output saved to TAPE.

- Printer: Output saved to print Queue or sent to printer.

- Query: Determine handling at generation time.

```
Perform Print consolidation: <Yes>/No ▶
```

All generation printout can be consolidated onto a single tape. This tape will contain the current version of all elements in the product. Print consolidation can be done in the product generation or afterwards by using SGP. This requires that print consolidation be performed for every generation.

```
System relocatable library: Query/<SYS$LIB$*SYSLIB> ▶Q
```

This file will be used as the system relocatable library during the collection of the DDP-FJT absolutes.

```
Are you using DPS with DDP-FJT: <Yes>/No ▶
```

Is DPS installed on the same system for use with DDP-FJT ?

- If DDP-FJT is to use the DPS 2200 full screen interface, use the default answer **Y**. In this case, the next query for the DPS relocatable library occurs.

- You must install DDP-FJT in mode B, C, E, or F for the DPS 2200 full screen interface.

- If DDP-FJT is to use the program-callable interface only, answer **N**. In this case, the query for the DPS relocatable library does not occur.

```
Enter DPS library:  Query/<SYS$LIB$*DPS> ▶Q
```

This file will be used as the library for DPS PLUS copy elements during the compilation of FJT relocatables.

```
Are you using IPF with DDP-FJT: <Yes>/No ▶
```

Is IPF installed on the same system for use with DDP-FJT ?

You must install DDP-FJT in mode A, C or F for use with IPF 2200.

```
Enter IPF relocatable library: Query/<SYS$LIB$*IPF-2> ▶Q
```

This file will be used as the IPF relocatable library during the collection of the DDP-FJT absolutes.

This file must have been created from a mixed mode (not absolute only) master tape for the required level of IPF 2200 Control software.

```
Enter PLUS library: Query/<SYS$LIB$*PLS> ▶Q
```

This file will be used as the library for PLUS and INLINE absolutes and PLUS relocatables during compilation and collection.

```
Enter DDP-PPC relocatable library: Query/<SYS$LIB$*DDP-PPC-1> ▶Q
```

This file needed for collection of DDP-FJT absolutes.

```
Enter DDP-PPC copy library: Query/<SYS$LIB$*DDP-PPC-3> ▶Q
```

This file will be used as the DDP-PPC copy library during the generation of DDP-FJT.

```
Enter ELMS library: Query/<SYS$LIB$*ELMS> ▶Q
```

This file will be used as the library for ELMS PLUS copy and relocatables during compilation and collection.

```
Enter UCS run time library: Query/<SYS$LIB$*URTS> ▶Q
```

The fully qualified file name of the library that contains the UCS run time library elements.

```
Enter Free Standing Audit Handler file: QUERY/<SYS$LIB$*FSAH> ▶Q
```

The fully qualified file name of the file that contains the IRU free standing audit handler (FSAH) run time and copy library elements.

DDP-FJT defaults completed.

DDP-FJT COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates DDP-FJT. This example assumes DDP-FJT has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ? ▶BUILD DDP-FJT,xRy
```

This BUILD command directs COMUS to apply changes to DDP-FJT level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (< >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (< >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ? ▶
```

Specify the reel number of the product master tape used as an input to the generation. A reel number may contain up to six letters or digits. It is used only to identify the product master tape in a LOAD message when the generation runstream executes.

> Reel(s): reel1/../reeln,asgopt,equip,exp/mmgr,ring,comp,buff,pool
> File: filename.,asgopt,equip,reel1/../reeln,exp/mmgr,ring,comp,buff,pool

Input can be continued on the next line at any point by specifying a semicolon (;) at the end of the current line without intervening spaces. If parameters are omitted, commas must be retained as place holders, however trailing commas should not be specified.

```
Generation id ? ▶
```

Enter a unique generation identifier which will be recorded in the system generation log (SGL). For some products this will also be used as part of the product sign-on line.

```
Generation heading (<>) ? ▶
```

Enter the heading to be used on generation printouts. Only 30 characters are allowed.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
NEWMAST - reel/file./<> ? ▶
```

Specify the reel number to be used for the new product master tape to be produced by this generation. A reel number may contain up to six letters or digits. It is used only to identify the tape in a LOAD message when the generation runstream executes.

> Reel(s): reel1/../reeln,asgopt,equip,exp/mmgr,ring,comp,buff,pool
> File: filename.,asgopt,equip,reel1/../reeln,exp/mmgr,ring,comp,buff,pool

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and the *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
Re-Compile ALL elements: Yes,<No> ▶
```

Yes ALL elements will be re-compiled.

No Only elements with changes will be re-compiled.

```
Enter system relocatable library: <SYS$LIB$*SYSLIB> ▶
```

This file will be used as the system relocatable library during the collection of the DDP-FJT absolutes.

```
Are you using DPS with DDP-FJT: <Yes>/No ▶
```

Is DPS installed on the same system for use with DDP-FJT ?

```
Enter DPS library: <SYS$LIB$*DPS> ▶
```

This file will be used as the library for DPS PLUS copy elements during the compilation of FJT relocatables.

```
Enter PLUS library: <SYS$LIB$*PLS> ▶
```

This file will be used as the library for PLUS and INLINE absolutes and PLUS relocatables during compilation and collection.

```
Enter ELMS library: <SYS$LIB$*ELMS> ▶
```

This file will be used as the library for ELMS PLUS copy and relocatables during compilation and collection.

```
Enter UCS run time library: <SYS$LIB$*URTS> ▶
```

The fully qualified file name of the library that contains the UCS run time library elements.

```
Are you using IPF with DDP-FJT: <Yes>/No ▶
```

Is IPF installed on the same system for use with DDP-FJT ?

```
Enter IPF relocatable library: <SYS$LIB$*IPF-2> ▶
```

This file will be used as the IPF relocatable library during the collection of the DDP-FJT absolutes.

```
Enter DDP-PPC library: <SYS$LIB$*DDP-PPC-1> ▶
```

This file will be used as the DDP-PPC relocatable library during the collection of DDP-FJT absolutes.

```
Enter DDP-PPC Copy library: <SYS$LIB$*DDP-PPC-3> ▶
```

This file will be used as the DDP-PPC copy library during the generation of DDP-FJT.

```
Enter Free Standing Audit Handler file: <SYS$LIB$*FSAH> ▶
```

The fully qualified file name of the file that contains the IRU free standing audit handler (FSAH) run time and copy library elements.

```
Send GEN output to: <Tape>/Printer ▶
```

This specifies the destination of the generated printout.

WARNING

Print consolidation is only possible if the generated printout is kept on tape.

```
Consolidate print - <Y>/N ▶
OLDCONSP - reel/file./NONE/<> ▶
NEWCONSP - reel/file./NONE/<> ▶

The runstream has been saved in 'ABC*COMRUN(1).1/DDP-FJTxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶

The runstream has been saved in 'ABC*COMRUN(1).1/DDP-FJTxRy'
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED ************************
COMMAND ? ▶EXIT
END COMUS
```

When the preceding runstream has completed its execution without any error, the DDP-FJT level *xRy* will be generated.

5.4.9. DDP-PPC

DDP-PPC COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the DDP-PPC default session.

```
Send Generation output to: Tape/<Printer>/Query ▶Q
```

Define the normal handling of generation printout:

- Tape: Save generation printout on tape.

- Printer: Queue generation printout to printer.

- Query: Determine handling at generation time.

```
Perform Print consolidation: YES/<NO> ▶Y
```

All generation printout can be consolidated onto a single tape. This tape will contain the current version of all elements of the product. Print consolidation can be done within the product generation or afterwards by using SGP. This requires that print consolidation be performed for every generation.

```
System relocatable LIB file: Query,<SYS$LIB$*SYSLIB> ▶Q
```

>This file will be used as the system relocatable library during the collection of the DDP-PPC absolutes.

```
Are you using DPS with DDP-PPC: <Yes>/No ▶
```

>Is DPS installed on the same system that you will be using DDP-PPC ?

```
DPS  library  file: Query,<SYS$LIB$*DPS> ▶Q
```

>This file will be used as the library for DPS PLUS copy elements during the compilation of PPC relocatables.

```
PLUS library file:  Query,<SYS$LIB$*PLS> ▶Q
```

>This file will be used as the library for PLUS and INLINE absolutes and PLUS relocatables during compilation and collection.

```
ELMS library file: Query,<SYS$LIB$*ELMS> ▶Q
```

>This file will be used as the library that contains the ELMS copy elements during the compilation of PPC elements.

```
UCS Run Time LIB file: Query,<SYS$LIB$*URTS> ▶Q
```

>The fully qualified file name of the library that contains the UCS run time library elements.

DDP-PPC defaults complete.

DDP-PPC COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates DDP-PPC. This example assumes DDP-PPC has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
@QUAL ABC
@COMUS
COMMAND ? ▶build DDP-PPC,xRy
```

>This BUILD command directs COMUS to apply changes to DDP-PPC level xRy. For stability releases, do not enter a stability level. Enter only the base product level: for example, enter xRy for $xRyA$, $xRyB$, or any other stability level.

```
Project id for this generation (< >) ? ▶
```

>The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product

generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (< >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ? ▶
```

Specify the reel number of the product master tape used as an input to the generation. A reel number may contain up to six letters or numerals. It is used only to identify the product master tape in a LOAD message when the generation runstream executes.

Reel(s): reel1/../reeln,asgopt,equip,exp/mmgr,ring,comp,buff,pool
File: filename.,asgopt,equip,reel1/../reeln,exp/mmgr,ring,comp,buff,pool

Input can be continued on the next line at any point by specifying a semicolon (;) at the end of the current line without intervening spaces. If parameters are omitted, commas must be retained as place holders, however trailing commas should not be specified.

```
Generation id ? ▶
```

Enter a unique generation identifier which will be recorded in the system generation log (SGL). For some products this will also be used as part of the product sign-on line.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit. This information is saved in the SGL document in the COMUS database.

```
New change number (<END>) ? ▶
```

Enter the COMUS database CHG document numbers to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
NEWMAST - reel/file ? ▶
```

Specify the reel number to be used for the new product master tape to be produced by this generation. A reel number may contain up to six letters or digits. It is used only to identify the tape in a LOAD message when the generation runstream executes.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and the *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
Type of GEN to perform: Full/<Inc> ▶F
```

INC AN INCREMENTAL UPDATE IS PERFORMED. RECOMMENDED FOR DEVELOPMENT CENTER USERS TO UPDATE THE PCF.

FULL A INCREMENTAL UPDATE IS DONE AND ALL ELEMENTS ARE COMPILED.

```
Enter system relocatable library: <SYS$LIB$*SYSLIB> ▶
```

This file will be used as the system relocatable LIBRARY during the collection of DDP-PPC absolutes.

```
Are you using DPS with DDP-PPC: <Y>/N ▶
```

Is DPS installed on the same system that you will be using DDP-PPC ?

```
Enter DPS library file: <SYS$LIB$*DPS> ▶
```

This file will be used as the library for DPS PLUS copy elements during the compilation of PPC relocatables.

```
Enter PLUS library file: <SYS$LIB$*PLS> ▶
```

This file will be used as the library for PLUS and INLINE absolutes and relocatables during compilation and collection.

```
Enter UCS Run Time library: <SYS$LIB$*URTS> ▶
```

The fully qualified file name of the library that contains the UCS Run Time library elements.

```
Enter ELMS library file: <SYS$LIB$*ELMS> ▶
```

This file will be used as the library that contains the ELMS copy elements during the compilation of PPC elements.

```
Send Gen output to: Tape/<Printer> ▶
```

This specifies the destination of the generated printout.

> **WARNING**
>
> Print consolidation will only be possible if the generated printout is kept on tape.

```
Consolidate GEN output: <Yes>/No ▶
```

This specifies whether or not print consolidation will be performed as part of the generation. For more information on print consolidation, see 5.3.2 and 5.3.3.

```
OLDCONSP - reel/file./NONE/<> ▶
NEWCONSP - reel/file./NONE/<> ▶

The runstream has been saved in 'ABC*COMRUN(1).2/DDP-PPCxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶

The runstream has been saved in 'ABC*COMRUN(1).2/DDP-PPCxRy'
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED ********************
COMMAND ? ▶exit
END COMUS
```

When the preceding runstream has completed its execution without any error, the DDP-PPC level *xRy* will be generated.

SNAPPC Sample Programs

The DDP-PPC release includes the source for sample SNAPPC ACOB and C programs which may be used as the basis for development of SNAPPC application programs. These programs are depicted in the SNAPPC manuals. The SAMPLES file contains these elements.

5.4.10. DFP

DFP COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates DFP. This example assumes DFP has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
▶COMMAND ? ▶BUILD DFP,xRy
```

This BUILD command directs COMUS to apply changes to DFP level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<   >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<   >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER – reel/file./<> ?  ▶
```

Specify the reel number of the product tape or the name of the utility file to be used as input to the generation. The MASTER reel number is the number of the tape containing DFP release tape files. A reel number may contain up to six letters or digits.

```
Generation id ? ▶
```

Specify the generation ID to appear in the system generation log (SGL) document in the COMUS database. The generation ID is your identifier for this new version of DFP.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify a reason for the generation or enter END. This information is saved in the SGL document in the COMUS database.

```
New change number (<END>) ▶
```

Enter the change numbers for CHG documents to be applied to DFP. These changes must have already been inserted in the COMUS. See the INSERT command description in the *OS 2200 COMUS End* database *Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
FULL or UPDATE generation - F/<U> ? ▶
```

F or **FULL**   All product components are compiled and the selected changes are applied.

U or **UPDATE**   Only those elements with changes are compiled.

```
NEWMASTER - reel/file./<> ? ▶
```

Specify the reel number for the new product tape to be produced by this generation. Print will be saved - queue for output too - Y/<N> ?

```
Where is the PLUS LIBRARY located -  <SYS$LIB$*PLS.>/Q*F. ?
```

Enter the PLUS library file to be used for compilations and collections.

```
Additional SGSs (<END>) ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/DFPxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/DFPxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC*COMRUN.1/DFPxRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED   *************************
COMMAND ?  ▶EXIT
```

When the preceding runstream has completed its execution without any error, the DFP level *xRy* will be generated.

5.4.11. ELMS

ELMS COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the FAS default session:

```
Name for system relocatable file: <SYS$LIB$*SYSLIB.>/QUERY ? ▶
```

Please enter the default to be used for the name of the system relocatable file.

```
Processor listings: none, short, long or query - N/<S>/L/Q ? ▶
```

Several system processors which are called during the BUILD process print source listings. The length of these listings is controlled by options set on the processor call cards, and may be one of N (none), S (short) or L (long). To control the size of and amount of information contained in the listings from the BUILD command which you will receive, enter a listing length option to be used on the processor calls required for the BUILD of this product.

ELMS COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates ELMS. This example assumes ELMS has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶ build elms,xRy
```

This BUILD command directs COMUS to apply changes to ELMS level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<>) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<>) ?   ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ?
```

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ?  ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>)  ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
FULL or UPDATE generation - F/<U> ?  ▶
```

F or **FULL**	All product components are compiled and the selected changes are applied.
U or **UPDATE**	Only those elements with changes are compiled.

```
                        NEWMASTER - reel/file./<> ?
```

Print will be saved on tape - queue for output too - Y/<N> ?

```
Additional SGSs (<END>) ?  ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ELMS*COMRUN(1).1/ELMSxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

                           The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in '*ABC**COMRUN(1).1/ELMS*xRy*'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

                           @START *ABC**COMRUN.1/ELMS*xRy*

```
UPDATING ACCESS FILES ...
CCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED  ************************
COMMAND ? > exit
```

When the preceding runstream has completed its execution without any error, the ELMS level *xRy* will be generated.

ELMS Product Specific SGS

The following SGSs (Stream Generation Statements) can be used with ELMS:

```
CO$XYZPROCESSOR CALL NAME IS 'QUAL*FILE.PROCESSOR'
```

This SGS is allowed for any processor called during an ELMS generation. For example, SSG, MASM, ELT, ED, MCON, PLS (requires LIBRARY FILE IS "Q*F"suffix), INLINE, UC, LINK, MAP, or UPAS.

```
CO$LINKLIBRARY FILE IS 'QUAL*FILE.'
```

In general, this SGS would not be specified. The ELMS build uses this file in an @USE LINK$PF,fn. statement to point the LINK processor to an alternate file for resolution of library code names.

```
CO$RTSLIBRARY FILE IS 'QUAL*FILE.'
```

In general, this SGS would not be specified. The ELMS build uses this file in an @USE RTS$LIB,fn. statement to use an alternate file for resolution of UCS Runtime System BDI references.

```
CO$EMOMLIBRARY FILE IS 'QUAL*FILE.'
```

In general, this SGS would not be specified. The ELMS build uses this file in an @USE EMOM$LIB,fn. statement to locate the runtime system's extended mode object modules.

```
CO$OMORLIBRARY FILE IS 'QUAL*FILE.'
```

This SGS must be specified. The ELMS build uses this file to resolve references to the Linking System's OMOR (Object Module Output Routines) package. This package is located on the Linking System tape and must be copied to disk prior to the ELMS build.

COMUS standard SGSs CO$MARGIN and CO$HDG are not supported by the ELMS build skeleton.

5.4.12. ELT

ELT COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates ELT. This example assumes ELT has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶BUILD ELT,xRy
```

This BUILD command directs COMUS to apply changes to ELT level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<   >) ?  ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation  (<   >) ?  ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ?  ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
FULL or UPDATE generation - F/<U> ? ▶
```

F or **FULL**	All product components are compiled and the selected changes are applied.
U or **UPDATE**	Only those elements with changes are compiled.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of ELT. Print will be saved on tape - queue for output too - Y/<N>.

```
Consolidate print - <Y>/N ?

OLDCONSP - reel/file./NONE/<> ?
NEWCONSP - reel/file./NONE/<> ?

Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/CMLxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/ELTxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START ABC*COMRUN.1/ELTxRy

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED ************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the ELT level xRy will be generated.

5.4.13. FAS

FAS COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following value will also be queried for by COMUS during the default session for FAS.

```
FAS printout: COMUS print file, printer or query - <T>/P/Q ? ▶
```

T   COMUS print file   The print will go to the print file as specified by COMUS. If the output media is a tape, the print file will go the print file on the tape. If the output media is mass storage, the print will go to the print file on mass storage.

P   printer   Queue generation printout to printer.

Q   query   Determine handling at generation time.

```
PLUS library file - <SYS$LIB$*PLS> ? ▶
```

Enter the PLUS library file to be used for compilations and collections.

FAS COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates FAS. This example assumes FAS has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?  ▶build fas,xRy
```

This BUILD command directs COMUS to apply changes to FAS. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<Q$Q$Q$>) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<FAS>) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ? ▶
```

The generation ID is your site identifier for this new version of FAS. This command directs COMUS to apply changes to FAS. For stability releases, enter the stability level. For example, enter xRyA, xRyB, etc.

```
Generation heading (<>) ? ▶
Generation reason (<END>) ? ▶
```

You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Enter the COMUS database CHG document numbers to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
FULL or UPDATE generation (F/<U>) ? ▶
```

F or **FULL**   All product components are compiled and the selected changes are applied.

U or **UPDATE**   Only those elements with changes are compiled.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of FAS.

```
Print will be saved in the COMUS print file.
Do you want it queued to a printer also - Y/<N> ?      ▶
```

Y   It is saved on the output NEWMASTER and you are queried for an output device:

```
Output device - < > ?
```

N   It is saved on the output NEWMASTER and you are not queried for an output device.

The FAS printout will be saved in the COMUS output file. It can also be queued for printing by answering 'Y'.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and the *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'FAS*COMRUN(1).1/FASxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in '*ABC**COMRUN(1).1/FAS*xRy*'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement

@START *ABC**COMRUN.1/FAS*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED  ************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the
FAS level *xRy* will be generated.

5.4.14. FLIT

FLIT COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following value will
also be queried for by COMUS during the IPF default session.

```
Name PLUS library/compiler file (Q*FN.). <SYS$LIB$*PLS>   ▶
```

Define the name of the file which contains the PLUS relocatable library, PLUS
compiler and INLINE processor needed to generate FLIT xRy.

FLIT COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command
that generates FLIT. This example assumes FLIT has been registered with the COMUS
database and the default values have already been selected. For information on how
to select the default values, see "First Generation and COMUS Database Build Default
Values."

```
▶@QUAL ABC
▶@COMUS
▶COMMAND ? ▶BUILD FLIT,xRy
```

This BUILD command directs COMUS to apply changes to FLIT level *xRy*. For
stability releases, do not enter a stability level. Enter only the base product level;
for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<    >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The
project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is
not required, however, to avoid conflicts in file names used for product
generations, a unique project identifier should be used for each generation when
performing concurrent generations.

```
Run id for this generation (<    >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ?  ▶
```

Specify the reel number of the product tape or the name of the utility file to be used as input to the generation. The MASTER reel number is the number of the tape containing FLIT release tape files. A reel number may contain up to six letters or digits.

```
Generation id ? ▶
```

Specify the generation ID to appear in the system generation log (SGL) document in the COMUS database. The generation ID is your identifier for this new version of FLIT.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify a reason for the generation or enter END. This information is saved in the SGL document in the COMUS database.

```
New change number (<END>)  ▶
```

Enter the change numbers for CHG documents to be applied to FLIT. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
NEWMASTER - reel/file./<> ?  ▶
```

Specify the reel number for the new product tape to be produced by this generation.

```
Type of generation (FULL,<UPDATE>) ?
```

FULL	Updates are applied to elements with changes, and all elements are recompiled.
UPDATE	Only those elements with changes are recompiled.

```
Name PLUS library/compiler file (Q*FN.). <SYS$LIB$*PLS.>
```

Define the name of the file which contains the PLUS relocatable library, PLUS compiler and INLINE processor needed to generate FLIT xRy.

```
Additional SGSs (<END>)
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and the *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/FLITxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/FLITxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC*COMRUN.1/FLITxRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED|
BUILD TASK COMPLETED   **************************
COMMAND ?  ▶EXIT
```

When the preceding runstream has completed its execution without any error, the FLIT level *xRy* will be generated.

5.4.15. FTN (ASCII FORTRAN)

FTN COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the FTN default session.

```
Number of elements per print file - <500> ? ▶
```

Define the number of elements contained within each file on print tape or in each part of a breakpointed print file.

```
Listing options, brief, none or full - B/<N>/F ? ▶
```

Define the listing options to be used in the generation. The shortest available listing is specified by 'NONE'. BRIEF generally means an S option type of listing giving source symbolics.

```
UCSRTS relocatables filename - <SYS$LIB$*UCSRTS> ? ▶
```

Define what file contains the UCSRTS relocatables. These relocatables need to be available for the FTN library build. The standard system file from a UCSRTS build is SYSLIB*UCSRTS and the current default is SYSLIB*UCSRTS. They will also be accessed correctly if they happen to be in SYS$*RLIB$ or installed by COMUS.

```
PLUS compiler and library filename - <SYS$LIB$*PLS> ? ▶
```

Define what file contains the PLUS compiler and library. The file specified should contain the level of PLUS released with the level of FTN you are building. The default is currently SYSLIB*PLS.

FTN COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates FTN. This example assumes FTN has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ? ▶build ftn,xRy
```

This BUILD command directs COMUS to apply changes to ASCII FORTRAN level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<PROJECT>) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation  (<RUN>) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of FTN.

```
Generation type, full or update - F/<U> ? ▶
```

    F or **FULL**   All product components are compiled and the selected changes are applied.

    U or **UPDATE**   Only those elements with changes are compiled.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. See Section "FTN SSGs" for a listing of SGSs you can apply only to FTN library and compiler builds. In addition, you can include any SGSs listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit.

```
The runstream has been saved in 'ABC*COMRUN(1).1/FTNxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y	Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:
	The runstream has been started
N	You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/FTNxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:
	@START *ABC**COMRUN.1/FTN*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED *************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the FTN level *xRy* will be generated.

FTN SSGs

This is a listing of SSGs (Stream generation statements) that you can apply only to FTN library and compiler builds.

- INCLUDE *name*

  This SGS causes assembly or compilation of the specified element name. Use it to assemble or compile an element that has no temporary corrections.

- NEW *type name*
  - Where *type* is FOR or FTN, MASM, PDP, PLS, or DCL.

FOR or FTN	FORTRAN elements
MASM	Assembler elements
PDP	Assembler procedure elements
PLS	PLS elements
DCL	Declare elements

  - Where *name* is the name of a complete new element.

This SGS indicates that corrections for the specified element name are a complete new element and not corrections for an element in the base symbolic.

Note: *You cannot introduce new MASM or FORTRAN procedure elements.*

- NOT *name*

  This SGS indicates that although there are temporary corrections for the specified element name or a full generation is to be done, the named element is not to be assembled or compiled.

- TYPECHECKING IS OFF

  ASCII FORTRAN automatically checks the number and types of arguments passed to functions and subroutines. When there is no match with the expected arguments, diagnostics are printed and execution resumes. This SGS shuts off typechecking for all absolutes created with this library by causing the typechecking run time routine to do an immediate return.

  Compiled code calls the typechecking run time routine when optimization is not used or there is a COMPILER(ARGCHK=ON) statement in the source. A COMPILER(ARGCHK=OFF) statement or a COMPILER(STD=66) statement in the source also shuts off typechecking for that element. The default is to do typechecking.

- TYPECHECKING IS ON

  If you turn argument typechecking off in the library routine with the TYPECHECKING IS OFF SGS, it remains off during future builds. Use the TYPECHECKING IS ON SGS to turn on typechecking in a library where it had been previously turned off.

- LINK USING IBJ$

  When you use this SGS, the resulting compiler acts as if each program compiled has the COMPILER(LINK=IBJ$) statement in it. This can be valuable when your site has projects that always link between programs by using the IBJ$ collector linkage. This SGS is off by default, and linking is done by the LMJ instruction on register X11 (after doing an LXI,U X11,0).

- DICTIONARY PAGE *count/size* IS *n*

  count

  > refers to the original number of dictionary pages for which a main storage area is reserved. Executive requests (ERs) to MCORE$ are done as more area is needed. A maximum of 511 pages can be defined, although the compiler stops doing MCORE$ and instead does input/output (I/O) to the spill file when main storage contains about 120,000 words of dictionary. The value of the count parameter times the value of the size parameter must be greater than or equal to 1000 + the value of the size parameter, and the value of the count parameter must be greater than 2.

size

> is the size of the dictionary pages. It must be a multiple of the disk preparation (prep) factor. When you get an ERROR 4 DCT-SIZE error message followed by a SPILL FILE LIMIT EXCEEDED message, you must increase the size. The minimum dictionary page size is 112.

This SGS alters the number or size of dictionary pages (symbol table information) originally in main storage.

Default:

> DICTIONARY PAGE COUNT IS 5
> DICTIONARY PAGE SIZE IS 448

Example:

> DICTIONARY PAGE COUNT IS 3
> DICTIONARY PAGE SIZE IS 672

- TEXT PAGE *count/size* IS *n*

count

> is the original number of in-core text pages for which a main storage area is reserved.

size

> is the size of the in-core text pages.

This SGS alters the number or size of the in-core text pages that the compiler uses. More the text pages in core, lower the I/O SUP charges are because of the spill file I/O. However, the compiler D-bank also gets larger. The minimum number of text pages allowed is four, and the minimum text page size is 448. Increase the text page size when your site has very large programs that do not compile because the spill file limit is exceeded. (This condition is indicated by a SPILL FILE LIMIT EXCEEDED message preceded by an ERROR 4 TXT-SIZE message.) Set the page size to a multiple of the disk prep factor for better efficiency in I/O.

Default:

> TEXT PAGE COUNT IS 6
> TEXT PAGE SIZE IS 896

Example:

> TEXT PAGE COUNT IS 4
> TEXT PAGE SIZE IS 448

> **Note:** *The initial size of the compiler D-bank resulting from the use of the preceding SGSs must not cause its collected last address (LASTD$) to get so large that results in collection errors.*

- PAGESIZE IS *n*

    This SGS alters the number of lines printed between headings and page ejects for the pseudo-assembler and storage allocation portions of the listings the ASCII FORTRAN compiler generates. The source and cross-reference listings are not affected, since they do not repeat headings and, therefore, no page is ejected. This number does not include the lines printed for user specified headings from @HDG cards. It also does not work properly when "n" is greater than the number of lines per page used for the system default.

    Default: PAGESIZE IS 57

    Example: PAGESIZE IS 45

- INHIBIT DIVIDE OPTIMIZATION

    This SGS inhibits optimization from changing certain single-precision division to multiples. Following is an example:

    X = Y/16380.   @ THIS IS USUALLY CONVERTED TO . . .

    X = Y*6.10501E-5   @ . . . THIS

    The preceding conversion loses a small amount of precision; when your site needs that small amount of extra precision, use the INHIBIT SGS. This conversion is done only when you use optimization (V or Z options). The divide is much slower than the multiplication to which it is converted. The default allows the conversion.

- NOZ *name*

    This SGS inhibits the use of optimization when compiling the specified PLUS element name.

- MAPL

    When this SGS is present, one of the several collections of the compiler has the L option on it.

FTN Performance Considerations

Altering Dictionary and Text Page Sizes

Internal buffer sizes and numbers affect compiler performance. Your site can change the default number and size of initial main storage dictionary (symbol table) pages and also the number and size of the main storage text pages. (Text pages hold the internal form of the executable statements of your program.) Both of these page types are accessed in a virtual manner, and their size and number affect input/output (I/O) charges and compilation speed.

For example, when your site generally compiles small programs that do not fully utilize the allocated internal storage or exceed the allocated storage by a small amount, you can generate the compiler to allocate fewer main storage pages. This provides smaller compiler D-bank area with a minimal increase in I/O time. (I/O is caused by the compiler going out to the spill file when its main storage area is full.)

When your site generally compiles large programs that require many more pages than can be resident at one time, increase the number of text pages originally in main storage. Compromising text area D-bank size for a larger number of pages in main storage reduces spill file swapping, resulting in less I/O time.

The ASCII FORTRAN compiler executes ERs to MCORE$ for dictionary space when it runs out of the original main storage area. (It does not do this for text space, however.) Therefore, increasing the amount of initial main storage dictionary area has only a small effect on performance. Dropping the number of original in-core dictionary pages can help system throughput slightly by requiring less initial-load D-bank, but this has to be balanced against the charges for ERs to MCORE$ (ER MCORE$) that then occur on more programs.

The following SGSs change the default sizes:

- DICTIONARY PAGE COUNT IS *n*
- DICTIONARY PAGE SIZE IS *n*
- TEXT PAGE COUNT IS *n*
- TEXT PAGE SIZE IS *n*

Note: *When you increase the size or number of pages, the initial-load last address of the compiler must not go so high that it results in collector truncation errors.*

The dictionary pages hold information (dictionary entries) on named items in your program such as variables, constants, common blocks, equivalence sets, subroutine names, and labels. In addition, the compiler generates internal entries such as internal generated labels. You can roughly approximate the number of 448-word dictionary pages that a program needs by using the following formula:

No-of-dictionary-pages = 3 + No-of-named-items / 32

You cannot estimate the number of text pages needed, since so much depends on the data types and types of operations performed.

When your site has many small jobs, usually four dictionary pages and four text pages are sufficient. This reduces the ASCII FORTRAN D-bank size. Since the I-banks are shared among the various runs, this can significantly increase the system throughput when ASCII FORTRAN is heavily used. However, this type of compiler can hamper large compilations by increasing the spill file I/O charges dramatically. Reducing the number of initial main storage dictionary and text pages to this amount reduces the D-bank of an ASCII FORTRAN compiler by about 2,250 decimal words.

When your site does predominantly large, optimized ASCII FORTRAN compilations, you may want to increase the number or size of in-core text pages. (A default text page is 896 words long.) The optimization phase scans the text several times and

divides the program into blocks that can be less than a page or larger than a few pages. The optimization phase may therefore thrash on text pages, causing drastically larger compilation I/O times. The compiler does not perform ERs to MCORE$ for text pages as it does for dictionary pages, so when the text pages are full, the compiler swaps entries to the spill-file PSF$ instead. The optimization phase can benefit from more or larger text pages in main storage when thrashing occurs on spill-file I/O.

Increasing the text or dictionary page size can help when your site gets a fatal SPILL FILE LIMIT EXCEEDED message on some large programs.

Type1 or Type2 Library

Unisys supplies a type2 library on the release tape as the preferred type of library. This form of library has the I/O elements collected in a common bank absolute, C2F$, that is installed in the system. (The C2F$ absolute links, by an LIJ instruction, to the PCIOS common banks for file I/O.) A type1 library has the I/O elements as local relocatables. When your program is mapped with a type1 library, it has all ASCII FORTRAN I/O and possibly all PCIOS in its absolute. Following two line program:

```
PRINT*,'HELLO'
END
```

has the following approximate sizes and collection times when collected in an absolute element.

Library Type	Program Size (I/D)	Absolute Size (Sectors)	Collection Time (Total SUPs)
Type2	300/2,000 words	73	8.7 seconds
Type1 (No PCIOS)	16,000/2,000 words	649	16.4 seconds
Type1 (With PCIOS)	22,000/2,000 words	878	17.5 seconds

All I/O is collected in the absolute when you use a type1 library because the library is designed to be common banked. It is brought into the collection completely, not element by element as needed by your program.

File TEST of the release tape has a type1 library for testing purposes.

5.4.16. FURPUR

FURPUR COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates FURPUR. This example assumes FURPUR has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

▶@QUAL *ABC*
▶@COMUS
COMMAND ? ▶build furpur,*xRy*

This BUILD command directs COMUS to apply changes to FURPUR level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<>) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<  >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ? ▶
```

The generation ID is your site identifier for this new version of FURPUR.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Enter the COMUS database CHG document numbers to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
FULL or UPDATE generation - F/<U> ?
```

Specify whether this generation is to assemble every FURPUR symbolic element (FULL) or only those elements which have temporary corrections (UPDATE).

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of FURPUR.

```
Print will be saved on tape - queue for output too - Y/<N>  ▶
Consolidate print <Y>/N ? ▶
```

Y   It is saved on the output NEWMASTER and you are queried for an output device:

```
Output device - < > ?
```

N   It is saved on the output NEWMASTER and you are not queried for an output device.

For more information on print consolidation, see 5.3.2 and 5.3.3.

```
OLDCONSP - reel/file./NONE/<> ? ▶
```

Enter a reel number or file that contains the consolidated print file from a previous generation. This query is used only for UPDATE generations.

```
NEWCONSP - reel/file./NONE/<> ? ▶
```

Enter an alternate reel or file for storing the consolidated print file from this generation.

```
Do you want to include new PCFP relocatable and definition elements?
(Y,<N>)
```

You may include new PCFP relocatable and definition (MSD omnibus) elements if you have SOLAR installed an updated PCFP, either from a release tape, from an Interim Correction (IC) tape, or from electronically transferred IC mass storage files.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit.

```
Change the number of 3600-word buffers? (Y, <N>) ? ▶
```

The 3600-word buffers are used internally by FURPUR for many commands. The main external affect is to select the tape block size that is written by the @COPY,G command when tape compression is turned on or when the D option is included.(@COPY,GD). When 'n' 3600-word buffers are selected, the tape block size will be n * 1794 words.

If you answer yes (Y) to this prompt, the following prompt appears:

```
      Enter the number of 3600-word buffers - (1-7,<8>,9-16)  ▶
```

You may select from 1 to 16 3600-word buffers if the default is not appropriate.

```
The runstream has been saved in 'ABC*COMRUN(1).1/FURPURxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

> **Y**   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:
>
> **The runstream has been started**

> **N**   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/FURPURxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:
>
> @START *ABC**COMRUN.1/FURPUR*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED  *************************
COMMAND ? > exit
```

When the preceding runstream has completed its execution without any error, the FURPUR level *xRy* will be generated.

5.4.17. GSA

GSA COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the GSA default session:

```
Default maximum run time (<9999>) ? >▶
```

Enter the maximum run time for the @RUN control statement. This is the estimated standard units of processing (SUP) usage in minutes. The maximum value allowed is 9999.

```
Default maximum pages (<99999> ) ? ▶
```

Enter the maximum pages for the @RUN control statement. The maximum value allowed is 99999.

```
Send consolidated print output to NEWMASTER (Y or <N>) ? ▶
```

Consolidated print is done automatically and sent to a tape. You can choose which tape to send it to. If you choose to send it to an alternate reel other than the NEWMASTER, you will be queried for the reel. The default is to query for an alternate reel. For more information on print consolidation, see 5.3.2 and 5.3.3.

```
Default name of the PLUS file (<SYS$LIB$*PLS>) ? ▶
```

Enter the PLUS library file to be used for compilations and collections. If a PLUS file is not needed, enter the word BLANK.

GSA COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates GSA. This example assumes GSA has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶build gsa,xRy
```

This BUILD command directs COMUS to apply changes to GSA level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<    >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<     >) ? ▶
```

The RUNID to be used on the @RUN control statement

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of IACULL.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ?  ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Type of generation (FULL or <UPDATE>) ?
```

FULL	Changes are applied, and all elements are recompiled.
UPDATE	Changes are applied, and only those elements that have changed are recompiled.

```
OLDCONSP - reel/file./NONE/<> ? ▶
```

Enter a reel number or file that contains the consolidated print file from a previous generation. This query is used only for UPDATE generations.

```
NEWCONSP - reel/file./NONE/<> ? ▶
```

Enter an alternate reel or file for storing the consolidated print file from this generation. The current generation output will be saved in the file PRNTF$. This file will not be printed during this generation.

```
Type of CULL (ALL or <NONE>) ? ▶
```

CULL generates a printed cross reference listing for all tags or variables in the source code. You can enter the type of CULL to be performed:

ALL	Listings are produced for all source code.
NONE	No CULL is performed.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/GSAxRy'

View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

      Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

            The runstream has been started

      N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/GSAxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

            @START *ABC*COMRUN.1/GSAxRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED ************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the GSA level *xRy* will be generated.

5.4.18. IACULL

IACULL COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the IACULL default session:

```
Default maximum run time (<9999>) ? >▶
```

Enter the maximum run time for the @RUN control statement. This is the estimated standard units of processing (SUP) usage in minutes. The maximum value allowed is 9999.

```
Default maximum pages (<99999> ) ? ▶
```

Enter the maximum pages for the @RUN control statement. The maximum value allowed is 99999.

```
Send consolidated print output to NEWMASTER (Y or <N>) ? ▶
```

Consolidated print is done automatically and sent to a tape. You can choose which tape to send it to. If you choose to send it to an alternate reel other than the NEWMASTER, you will be queried for the reel. The default is to query for an alternate reel. For more information on print consolidation, see 5.3.2 and 5.3.3.

```
Default name of the PLUS file (<SYS$LIB$*PLS>) ? ▶
```

Enter the PLUS library file to be used for compilations and collections. If a PLUS file is not needed, enter the word BLANK.

IACULL COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates IACULL. This example assumes IACULL has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?  ▶build iacull,xRy
```

This BUILD command directs COMUS to apply changes to IACULL level xRy. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<    >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from the set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<     >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of IACULL.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ?  ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Type of generation (FULL or <UPDATE>) ?
```

FULL   Changes are applied and all elements are recompiled.

UPDATE Changes are applied, only those elements that changed are recompiled.

```
OLDCONSP - reel/file./NONE/<> ? ▶
```

Enter a reel number or file that contains the consolidated print file from a previous generation. This query is used only for UPDATE generations.

```
NEWCONSP - reel/file./NONE/<> ? ▶
```

Enter an alternate reel or file for storing the consolidated print file from this generation. The current generation output will be saved in the file PRNTF$. This file will not be printed during this generation.

```
Type of CULL (ALL or <NONE>) ? ▶
```

CULL generates a printed cross reference listing for all tags or variables in the source code. You can enter the type of CULL to be performed:

ALL   Listings are produced for all source code.

NONE   No CULL is performed.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
Generate a Stand Alone IACULL or one that utilizes
Common Banked libraries (SA, CB, or <BOTH>) ? ▶
```

Enter how to generate the absolutes.

SA   A Stand Alone COMUS absolute will be generated and placed in file 2.

CB   The PCIOS and ACOB Common Banked libraries will be used. The COMUS absolute is placed in file 3.

BOTH Two COMUS absolutes will be generated. The standalone absolute is placed in file 2, and the one using common banks in file 3.

```
The runstream has been saved in 'ABC*COMRUN(1).1/IACULLxRy'
View the runstream (Y or <N>) ? ►
Print a copy of the runstream (Y or <N>) ? ►
Start the runstream (Y or <N>) ? ►
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/IACULLxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC*COMRUN.1/IACULLxRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED   ************************
COMMAND ? > exit
```

When the preceding runstream has completed its execution without any error, the IACULL level *xRy* will be generated.

5.4.19. IPF

IPF 2200 Sources of Information

To configure, build, and install IPF 2200, you need, at a minimum, the following sources of information:

- IPF 2200 build tape

- *OS 2200 Interactive Processing Facility (IPF 2200) Administration Guide* (7833 3754)

- *OS 2200 Software Products Installation Guide* (7831 0612)

- *ClearPath OS 2200 COMUS End Use Reference Manual* (7830 7758)

- *OS 2200 Symbolic Stream Generator (IPF 2200) Programming Reference Manual* (7831 1784). This manual describes the directives that a programmer uses to direct and manipulate one or more symbolic input streams and create one or more output streams.

- *OS 2200 Interactive Processing Facility (IPF 2200) Administration Guide* (7833 3754). This guide contains information which helps a site manage and maintain IPF 2200. The intended audience for this guide is the person who configures, builds, and installs IPF 2200, usually the site administrator. This document contains the following information:

o   A brief product overview and description of the features and enhancements of IPF 2200 levels 6R1 and above.

o   A description of the configure process, including the configuration menus and directives.

o   A description of the files on the IPF 2200 release tape.

o   Information on sample procedures and how to install them.

o   Information on compatibility issues with previous levels of IPF 2200.

o   A description of operational considerations such as alternate installation of the Display Processing System (DPS 2200) for use with IPF 2200, and restrictions for IPF 2200.

o   Information on the sources of possible errors in IPF 2200, and what to do when IPF 2200 has an internal system error.

o   A description of how a site administrator can convert the message text of message modules from English to another language, a list of languages currently supported by IPF 2200 User Assistance, and a description on how a site administrator can display the IPF 2200 tutorials in languages other than English.

IPF COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the IPF default session:

```
Processor listings: none, short, long or query - N/<S>/L/Q ? ▶
```

Enter the letter corresponding to the listing length to be used during the generation. Your choice affects the options used on the various processor calls performed during the generation and the overall size of the print output from the generation. Several of the system processors are called during the BUILD process print source listings. The length of these listings is controlled by options set on the processor call statements, and can be either one of the following: **N (none)**, **S (short)**, or **L (long)**. To control the size of and amount of information contained in the listings from the BUILD command which you will receive, enter a listing length option to be used on the processor calls required for the BUILD of this product. Enter **Q (for QUERY)** if you want to determine the processor listing length at generation time.

```
IPF printout: tape, printer, file, or query - <T>/P/F/Q ? ▶
```

Define the normal handling of generation printout.

T	**tape**	Save generation printout on tape
P	**printer**	Queue generation printout to printer
F	**file**	Save generation printout to the file IPF-PRINT
Q	**query**	Determine handling at generation time

If you wish to get a consolidated generation printout, you must send it to tape. To obtain a full consolidated generation printout, you must perform at least one full generation. You can then perform update generations and consolidate the new portions with the existing consolidated print tape to get a current full generation printout tape.

```
Workspace size in tracks: <256>/{128 to 262143}/Q ? ▶
```

Enter the size in tracks to be used when IPF 2200 creates each user's workspace file. The file can be from 128 to 262143 tracks in length. It is recommended that you do not make the workspace size greater than 1000 tracks unless you have unusual circumstances that dictate workspaces that large. Enter **Q** (for QUERY) if you want to determine the workspace size at generation time.

```
Workspace deletion time: never, logoff, run termination, or query –
<N>/L/R/Q? ▶
```

Enter the letter corresponding to the time at which IPF 2200 should delete the workspace for each user.

N	**never**	The workspace is never deleted
L	**logoff**	The workspace is deleted at LOGOFF time
R	**run**	The workspace is deleted at run termination time
Q	**query**	Determine deletion time at generation time

```
Do you want to activate the internationalization (I18N) feature?
<N>/Y/Q ▶
```

Make the internationalization (I18N) feature active or inactive for future product generations. Making this feature active enables IPF 1100 to handle characters and text according to the different coded character sets (CCSs).

Enter **Q** (for QUERY) if you want to make this choice at product generation time.

Respond **Y** to activate the I18N feature.

Respond **N** or take the default to leave the feature inactive.

IPF COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates IPF. This example assumes IPF has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

Note: *Your user ID must be allowed to execute the ER MODPS$ in order to execute the COMUS BUILD command for IPF 2200.*

```
►@QUAL ABC
►@COMUS
```
COMMAND ? ►build ipf,*xRy*

This BUILD command directs COMUS to apply changes to IPF level *xRy*. For
stability releases, do not enter a stability level. Enter only the base product level;
for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<   >) ? ►
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The
project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is
not required, however, to avoid conflicts in file names used for product
generations, a unique project identifier should be used for each generation when
performing concurrent generations.

```
Run id for this generation (<run>) ? ►
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ? ►
```

The MASTER reel number is the number of the tape containing the product source
code.

```
Generation id ?  ►
```

Specify the generation ID to appear in the System Generation Log (SGL) document
in the COMUS database.

```
Generation heading (<>) ? ►
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ►
```

Specify the reason for the generation. This information is saved in the SGL in the
COMUS database. You can enter any text describing the reason for this
generation. If you enter text, COMUS repeats the question until you enter END or
a null transmit.

```
New change number (<END>) ? ►
```

Specify the COMUS database CHG document numbers for the changes to be
applied during this generation. These changes must have already been inserted in
the COMUS database. See the INSERT command description in the *ClearPath
OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS
repeats the question until you enter END or a null transmit.

```
FULL or UPDATE generation - F/<U> ? ▶
```

> **F** or **FULL**   All product components are compiled and the selected changes are applied.
>
> **U** or **UPDATE**   Only those elements with changes are compiled.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of IPF 2200.

```
Printout to tape, printer, or file - <T>/P/F ? ▶
```

> **P** or **Printer**   The system printer is to print all generation print output.
>
> **T** or **Tape**   All generation print output is to be stored on tape rather than be printed.
>
> **F** or **File**   COMUS asks what is to be done with generation print output during all subsequent generations of the product.

```
Print will be saved on tape - queue for output too - Y/<N> ? ▶
```

> **Y**   It is saved on the output NEWMASTER and you are queried for an output device:
>
> ```
> Output device - < > ?
> ```
>
> **N**   It is saved on the output NEWMASTER and you are not queried for an output device.

```
OLDCONSP - reel/file./NONE/<> ? ▶
```

Enter a reel number or file that contains the consolidated print file from a previous generation. This query is used only for UPDATE generations.

```
NEWCONSP - reel/file./NONE/<> ? ▶
```

Enter an alternate reel or file for storing the consolidated print file from this generation.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/IPFxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/IPFxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/IPF*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED  ************************
COMMAND ? > exit
```

When the preceding runstream has completed its execution without any error, the IPF level *xRy* will be generated.

5.4.20. LA

LA COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the LA default session:

```
Default maximum run time (<9999>) ? ▶
```

Enter the maximum run time for the @RUN control statement. This is the estimated standard units of processing (SUP) usage in minutes. The maximum value allowed is 9999.

```
Default maximum pages (<99999> ) ? ▶
```

Enter the maximum pages for the @RUN control statement. The maximum value allowed is 99999.

```
Send consolidated print output to NEWMASTER (Y or <N>) ? ▶
```

Consolidated print is done automatically and sent to a tape. You can choose which tape to send it to. If you choose to send it to an alternate reel other than the NEWMASTER, you are queried for the reel. The default is to query for an alternate reel. For more information on print consolidation, see 5.3.2 and 5.3.3.

```
Default name of the PLUS file (<SYS$LIB$*PLS>) ? ▶
```

Enter the PLUS library file to be used for compilations and collections. If a PLUS file is not needed, enter the word BLANK.

LA COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates LA. This example assumes LA has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶build la,xRy
```

This BUILD command directs COMUS to apply changes to LA level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (< >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (< >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER – reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
NEWMASTER – reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of LA.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Type of generation (FULL or <UPDATE>) ?
```

> **FULL** Changes are applied and all elements are recompiled.
>
> **UPDATE** Changes are applied, only those elements that changed are recompiled.

```
OLDCONSP - reel/file./NONE/<> ? ▶
```

Enter a reel number or file that contains the consolidated print file from a previous generation. This query is used only for UPDATE generations.

```
NEWCONSP - reel/file./NONE/<> ? ▶
```

Enter an alternate reel or file for storing the consolidated print file from this generation. The current generation output will be saved in the file PRNTF$. This file will not be printed during this generation.

```
Type of CULL (ALL or <NONE>) ? ▶
```

CULL generates a printed cross reference listing for all tags or variables in the source code. You can enter the type of CULL to be performed:

> **ALL** Listings are produced for all source code.
>
> **NONE** No CULL is performed.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/LAxRy'
View the runstream (Y or <N>) ?  ▶
Print a copy of the runstream (Y or <N>) ?  ▶
Start the runstream (Y or <N>) ?  ▶
```

Y Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/LAxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START ABC*COMRUN.1/LAxRy

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED   ************************
COMMAND ?   ▶exit
```

When the preceding runstream has completed its execution without any error, the LA level xRy will be generated.

5.4.21. LINK

LINK COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the LA default session:

The default processors for a LINK BUILD are defined through CO$xyzPROCESSOR SGSs. To change a processor call, enter a new CO$xyzPROCESSOR SGS when you are queried for permanent SGSs. The CO$xyzPROCESSOR SGS format follows:

CO$xyzPROCESSOR SGS CALL NAME IS ''call name'' ;
OPTIONS ARE ''options'' ;
LIBRARY FILE IS ''library file name''

LINK COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates LINK. This example assumes LINK has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
►@QUAL ABC
►@COMUS
```
COMMAND ?   ►BUILD LINK,*xRy*

This BUILD command directs COMUS to apply changes to LINK level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (< >) ? ►
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation  (< >) ? ►
```

The RUNID to be used on the @RUN control statement

```
MASTER - reel/file./<> ? ►
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?   ►
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ►
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ►
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
FULL or UPDATE generation - F/<U> ? ▶
```

F or **FULL**   All product components are compiled and the selected changes are applied.

U or **UPDATE**   Only those elements with changes are compiled.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of LINK.

```
PTAPE - reel/file./<> ? ▶
```

The PTAPE contains the generation printout.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/LINKxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in '*ABC**COMRUN(1).1/LINK*xRy*'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/LINK*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED *************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the LINK level *xRy* will be generated.

5.4.22. LIST

LIST COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates LIST. This example assumes LIST has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

Note: *You must use the name LST when communicating with COMUS about the LIST processor. COMUS has a command called LIST and will confuse the product LIST with the command LIST.*

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶BUILD LST,xRy
```

This BUILD command directs COMUS to apply changes to LIST level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (< >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<   >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER – reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ?  ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? >
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ?  ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Type of generation (FULL or <UPDATE>) ?  ▶
```

F or **FULL**	All product components are compiled and the selected changes are applied.
U or **UPDATE**	Only those elements with changes are compiled.

```
NEWMASTER - reel/file./<> ?  ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of LIST.

```
Print will be saved on tape - queue for output too - Y/<N>
```

Y	It is saved on the output NEWMASTER and you are queried for an output device:
	```Output device - < > ?```
N	It is saved on the output NEWMASTER and you are not queried for an output device.

```
Consolidate print - <Y>/N ?
```

For more information on print consolidation, see 5.3.2 and 5.3.3.

```
OLDCONSP - reel/file./NONE/<> ?
```

Enter a reel number or file that contains the consolidated print file from a previous generation. This query is used only for UPDATE generations.

```
NEWCONSP - reel/file./NONE/<> ?
```

Enter an alternate reel or file for storing the consolidated print file from this generation. The current generation output will be saved in the file PRNTF$. This file will not be printed during this generation.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/LSTxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with: **The runstream has been started**

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in '*ABC**COMRUN(1).1/LST*xRy*'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/LST*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED  *************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the LIST level *xRy* will be generated.

5.4.23. LSS

LSS COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the LSS default session:

Set default character set keyword option for UCS compilers:

A - ASCII The ASCII character set (EQ keyword NO-SHIFT) -default-
K - KANJI The KANJI character set (EQ keyword SHIFT)

```
Select the default compiler character set keyword option (K/<A>) ?>
```

You may change the default compiler keyword option setting of SHIFT I NO-SHIFT at this point. The default is NO-SHIFT (ASCII), so a null <transmit> or an A <transmit> selects NO-SHIFT. You may answer K <transmit> to select SHIFT (KANJI). In this example, the user has simply entered a null <transmit>.

The LSS build routine will echo your selection with the following message:

```
Will build for ASCII character set (compiler keyword = NO-SHIFT)....
```

```
LSS Default Processors
```

The default processors for an LSS BUILD are defined through CO$xyzPROCESSOR SGSs. To change a processor call, enter a new CO$xyzPROCESSOR SGS when you are queried for permanent SGSs. The CO$xyzPROCESSOR SGS format follows:

CO$xyzPROCESSOR CALL NAME IS ''call name'' ;
OPTIONS ARE ''option'' ;
LIBRARY FILE IS ''library file name''

```
View a table of default processors - (<Y>/N) ? >
View an example of changing a CO$xyzPROCESSOR SGS - (<Y>/N) ? >
```

LSS COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates LSS. This example assumes LSS has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
    ▶@QUAL ABC
    ▶@COMUS
COMMAND ?   ▶BUILD LSS,xRy
```

This BUILD command directs COMUS to apply changes to LSS level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (< >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation  (< >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
FULL or UPDATE generation (F/<U>) ? ▶
```

> **F** or **FULL**   All product components are compiled and the selected
> changes are applied.
>
> **U** or **UPDATE**   Only those elements with changes are compiled.

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the LSS tape containing the product
source code.

```
Generation id ?   ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document
in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the
COMUS database. You can enter any text describing the reason for this
generation. If you enter text, COMUS repeats the question until you enter END or
a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be
applied during this generation. These changes must have already been inserted in
the COMUS database. See the INSERT command description in the *ClearPath
OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS
repeats the question until you enter END or a null transmit.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the
updated version of LSS.

```
PTAPE - reel/file./<> ? ▶
```

Enter a tape reel or file where the build output will be stored.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200
COMUS End Use Reference Manual*, but they apply to this generation only. If you
enter any SGSs, COMUS repeats the question until you enter END or a null
transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and the

OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/LSSxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/LSSxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/LSS*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED *************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the LSS level *xRy* will be generated.

5.4.24. MAP

MAP COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates MAP. This example assumes MAP has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?    ▶BUILD MAP,xRy
```

This BUILD command directs COMUS to apply changes to MAP level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<  >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<>) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
NEWMASTER - reel/file./<> ? >
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of MAP.

```
Type of generation (FULL,<UPDATE>) ?
```

FULL   Updates are applied to elements with changes, and all elements are recompiled.

UPDATE   Only those elements with changes are recompiled.

```
Type of CULL - ALL/MSM/<NONE>   ▶
```

CULL generates a printed cross reference listing for all tags or variables in the source code. Enter the type of CULL to be performed:

ALL   Listings are produced for all source code.

MSM   MASM elements and procedures

NONE   No CULL is performed.

```
Print out to printer or tape - P/<T> ▶
```

The print will be saved on the output NEWMASTER tape. It can also be queued for printing by answering "Y".

```
Queue print out for printing also - Y/<N> ▶
```

Specify the destination of the generated print out.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs

```
The runstream has been saved in 'ABC*COMRUN(1).1/MAPxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

         The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/MAPxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

         @START MAP*COMRUN.1/MAP*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED   *************************
COMMAND ? > exit
```

When the preceding runstream has completed its execution without any error, the MAP level *xRy* will be generated.

5.4.25. MASM

MASM COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates MASM. This example assumes MASM has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

During the generation of MASM, references to Linking System entry points must be satisfied. Sites that have installed the Linking System will automatically resolve those references by using a common bank entry point element in the Linking System library file. Sites without the Linking System will resolve the references with a special definition element in the MASM file REL$. If your site installs the Linking System after a MASM generation, MASM should be rebuilt.

> ►@QUAL *ABC*
> ►@COMUS
> **COMMAND ?**   ►BUILD MASM,*xRy*

This BUILD command directs COMUS to apply changes to MASM level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (< >) ? ►
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation  (< >) ? ►
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ? ►
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Type of generation (FULL or <UPDATE>) ?
```

FULL   Changes are applied and all elements are recompiled.

UPDATE   Changes are applied, only those elements that changed are recompiled.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of MASM.

```
Print will be saved on tape - queue for output too - Y/<N> ? ▶
```

Y   It is saved on the output NEWMASTER and you are queried for an output device:

```
Output device - < > ?
```

N   It is saved on the output NEWMASTER and you are not queried for an output device.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and

OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/MASMxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in '*ABC**COMRUN(1).1/MASM*xRy*'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/MASM*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED *************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the MASM level *xRy* will be generated.

MASM Configuration Variables

The following table describes MASM configuration variables that can be altered to produce a MASM absolute tailored to your site's needs. Create COMUS CHG documents for the appropriate element as described below. Add those CHG documents to your COMUS generation when prompted.

The variable SMMAX determines the maximum DBANK size. SMRES sets the initial storage reserve for an assembly. These variables have been set to produce the maximum system performance over a large variety of assembly sizes. Altering these variables may degrade MASM performance.

The variables LIBMAX, SRCHPROC$, SRCHMASM, SRCHSYSLIB, and SRCHRLIB$ are used to alter the MASM library search chain.

LIBMAX determines the number of MASM$PFx files (in addition to MASM$PF) that MASM searches for procedures and $INCLUDE elements. LIBMAX can be set to search from 1 to 50 additional MASM$PFx files.

The SRCHname variables can be turned off to reduce the number of system files that MASM searches for procedures and $INCLUDE elements.

VARIABLE NAME	DEFAULT VALUE	ELEMENT NAME	DESCRIPTION
SMMAX	0777776	ASMDEF	Maximum D-bank size
SMRES	062500	ASMDEF	Initial storage reserve
LIBMAX	1	RDLIB	Number of MASM$PFx files MASM can search in the library search chain (limit of 50)
SRCHPROC$	1	RDLIB	Turn on library search chain for PROC$
SRCHMASM	1	RDLIB	Turn on library search chain for MASM
SRCHSYSLIB	1	RDLIB	Turn on library search chain for SYSLIB
SRCHRLIB$	1	RDLIB	Turn on library search chain for RLIB$

5.4.26. MSAR

MSAR COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the MSAR default session:

```
Default maximum run time (<9999>) ? >
```

Enter the maximum run time for the @RUN control statement. This is the estimated standard units of processing (SUP) usage in minutes. The maximum value allowed is 9999.

```
Default maximum pages (<99999> ) ? > 9999
```

Enter the maximum pages for the @RUN control statement. The maximum value allowed is 99999.

```
Send consolidated print output to NEWMASTER (Y or <N>) ? >
```

Consolidated print is done automatically and sent to a tape. You can choose which tape to send it to. If you choose to send it to an alternate reel other than the NEWMASTER, you will be queried for the reel. The default is to query for an alternate reel. For more information on the consolidated print, see 5.3.2 and 5.3.3.

```
Default name of the PLUS file (<SYS$LIB$*PLS>) ? > blank
```

Enter the PLUS library file to be used for compilations and collections. If a PLUS file is not needed, enter the word BLANK.

MSAR COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates MSAR. This example assumes MSAR has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
►@QUAL ABC
►@COMUS
COMMAND ?   ►BUILD MSAR,xRy
```

This BUILD command directs COMUS to apply changes to MSAR level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<    >) ? ►
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<    >) ? ►
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ? ►
```

The MASTER reel number is the number of the tape containing the product source code.

```
NEWMASTER - reel/file./<> ? ►
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of MSAR.

```
Generation id ?  ►
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ►
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Type of generation (FULL or <UPDATE>) ?
```

> **FULL**   Changes are applied and all elements are recompiled.

> **UPDATE**   Changes are applied, only those elements that changed are recompiled.

```
OLDCONSP – reel/file./NONE/<> ? ▶
```

Enter a reel number or file that contains the consolidated print file from a previous generation. This query is used only for UPDATE generations.

```
NEWCONSP – reel/file./NONE/<> ? ▶
```

You can enter an alternate reel or file for storing the consolidated print file from this generation. The current generation output will be saved in the file PRNTF$. This file will not be printed during this generation.

```
Type of CULL (ALL or <NONE>) ? ▶
```

CULL generates a printed cross reference listing for all tags or variables in the source code. Enter the type of CULL to be performed:

> **ALL**   Listings are produced for all source code.

> **NONE**   No CULL is performed.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in *the ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
Maximum number of disk packs and cataloged tape file reels (<25>) ?   ▶
```

Enter the maximum number of disk packs allowed in the file record. The default is 25. Bigger the value larger will be the MSAR$DB file.

```
Maximum number of tape reels (<25>) ?   ▶
```

Enter the maximum number of tape reels allowed in the file record. The default is 25. Bigger the value larger will be the MSAR$DB file.

```
The runstream has been saved in 'ABC*COMRUN(1).1/MSARxRy'

View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/MSARxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement: @START *ABC*COMRUN.1/MSARxRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED  ************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the MSAR level *xRy* will be generated.

5.4.27. OSAM

OSAM COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the OSAM default session:

```
Default maximum run time (<9999>) ? ▶
```

Enter the maximum run time for the @RUN control statement. This is the estimated standard units of processing (SUP) usage in minutes. The maximum value allowed is 9999.

```
Default maximum pages (<99999> ) ?  ▶
```

Enter the maximum pages for the @RUN control statement. The maximum value allowed is 99999.

```
Send consolidated print output to NEWMASTER (Y or <N>) ?  ▶
```

Consolidated print is done automatically and sent to a tape. You can choose which tape to send it to. If you choose to send it to an alternate reel other than the NEWMASTER, you are queried for the reel. The default is to query for an alternate reel. For more information on print consolidation, see 5.3.2 and 5.3.3.

```
Default name of the PLUS file (<SYS$LIB$*PLS>) ?  ▶
```

Enter the PLUS library file to be used for compilations and collections. If a PLUS file is not needed, enter the word BLANK.

OSAM COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates OSAM. This example assumes OSAM has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶BUILD OSAM,xRy
```

This BUILD command directs COMUS to apply changes to OSAM level xRy. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<    >) ?  ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<   >) ?  ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER – reel/file./<> ?  ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of OSAM level xRy.

```
Generation id ?   ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Type of generation (FULL or <UPDATE>) ?
```

FULL Changes are applied, and all elements are recompiled.

UPDATE Changes are applied, and only elements that have changed are recompiled.

```
OLDCONSP - reel/file./NONE/<> ? ▶
```

Enter a reel number or file that contains the consolidated print file from a previous generation. This query is used only for UPDATE generations.

```
NEWCONSP - reel/file./NONE/<> ? ▶
```

You can enter an alternate reel or file for storing the consolidated print file from this generation. The current generation output will be saved in the file PRNTF$. This file will not be printed during this generation.

```
Type of CULL (ALL or <NONE>) ? ▶
```

CULL generates a printed cross reference listing for all tags or variables in the source code. Enter the type of CULL to be performed:

ALL   Listings are produced for all source code.

NONE   No CULL is performed.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/OSAMxRy'

View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/OSAMxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC*COMRUN.1/OSAMxRy

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED   ************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the OSAM level *xRy* will be generated.

5.4.28. OSI-TP

OSI-TP COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the OSI-TP default session:

```
Build OSI-TP in TEST Mode: Query/Yes/<No> ▶Q
```

A build in TEST Mode allows you to specify an alternate qualifier for the installed files and an alternate BDI# for the fixed gate subsystem.

```
Is OLTP-TM2200 Alternate install being used: Yes/<No>/Query  ▶Q
```

Have you generated OLTP-TM2200 in mode TEST with a set of alternate user BDI numbers in place of the UNISYS system BDI numbers for the standard installation?

```
Send Generation output to: Tape/<Printer>/Query  ▶Q
```

Define the normal handling of generation output:

- TAPE: Send to tape.
- PRINTER: Send to printer or print queue.
- QUERY: Determine handling at generation time.

```
Print consolidation required: <Yes>/No ▶N
```

All generation printout can be consolidated onto a single tape. This tape will contain the current version of all elements of the product. Print consolidation can be done within the product generation or afterwards by using SGP. This requires that print consolidation be performed for every generation.

```
Enter DDP relocatable LIB: Query/<SYS$LIB$*DDP-PPC-1>  ▶Q
```

This file will be used as the DDP-PPC object module library during the generation of the OSI-TP elements.

```
Enter  DDP  Copy  Library: Query,<SYS$LIB$*DDP-PPC-3>  ▶Q
```

This file will be used as the DDP-PPC copy library during the compilation of the OSI-TP source elements.

```
Enter RTS Object Module LIB: Query,<SYS$LIB$*EMOMRTS>  ▶Q
```

This file must contain the URTS$TABLES object module during the linking of OSI-TP elements.

```
Enter ELMS library file: Query,<SYS$LIB$*ELMS>  ▶Q
```

This will be the Extended Language Message System library during the generation of the OSI-TP elements.

OSI-TP defaults complete.

OSI-TP COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates OSI-TP. This example assumes OSI-TP has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
```
COMMAND ? ▶BUILD OSI-TP,*xRy*

This BUILD command directs COMUS to apply changes to OSI-TP level *xRy*. For
stability releases, do not enter a stability level. Enter only the base product level;
for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (< >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The
project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is
not required, however, to avoid conflicts in file names used for product
generations, a unique project identifier should be used for each generation when
performing concurrent generations.

```
Run id for this generation (< >) ? ▶
```

The RUNID to be used on the @RUN control statement

```
MASTER reel/file./ <> ? ▶
```

Specify the reel number of the product master tape used as an input to the
generation. A reel number may contain up to six letters or digits. It is used only to
identify the product master tape in a LOAD message when the generation
runstream executes.

```
Generation id ? ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document
in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the
COMUS database. You can enter any text describing the reason for this
generation. If you enter text, COMUS repeats the question until you enter END or
a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be
applied during this generation. These changes must have already been inserted in
the COMUS database. See the INSERT command description in the *ClearPath
OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS
repeats the question until you enter END or a null transmit.

```
NEWMAST - reel/file./<> ? ▶
```

Specify the reel number to be used for the new product master tape to be produced by this generation. A reel number may contain up to six letters or digits. It is used only to identify the tape in a LOAD message when the generation runstream executes.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in *the ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and the *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
Re-Compile ALL elements: <Yes>/No ▶
```

Answer **Y** to perform a full generation, which recompiles all product elements, including those affected by changes.

```
Build OSI-TP in TEST Mode: Yes/<No> ▶
```

A build in TEST Mode allows you to specify an alternate qualifier for the installed files and an alternate BDI# for the fixed gate subsystem.

-- IF YES

```
Build OSI-TP in TEST Mode: Yes/<No> ▶Y
```

```
Enter BDI# for OSI-TP TM subsystem, <20nnnn> ▶201234
```

Select a BDI# for the TEST Mode OSI-TP fixed gate subsystem from the pool of BDI numbers that are available for customer use. Enter the BDI# using the format 20nnnn, where nnnn is the rightmost four digits of the BDI number.

```
Is BDI 201234 correct for TM subsystem: Yes/<No> ▶Y
```

```
Enter TM file qualifier: <OSITP$TEST> ▶
```

Supply a valid qualifier to be used for the TEST Mode files. The qualifier must be different than SYSLIB and >2 chars.

```
Is OLTP-TM2200 Alternate install being used: Yes/<No> ▶
```

Have you generated OLTP-TM2200 in mode TEST with a set of alternate user BDI numbers in place of the UNISYS system BDI numbers for the standard installation?

--IF YES

```
Is OLTP-TM2200 Alternate install being used: Yes/<No> ▶Y

Enter BDI# for OLTP-TM2200 fgss: <20nnnn> ▶204321
```

Enter the BDI# for the OLTP-TM2200 fixed gate subsystem that you had selected from the pool of BDI numbers. Enter the BDI number using the format 20nnnn, where nnnn is the rightmost four digits of the BDI#. The default UNISYS BDI# for OLTP-TM2200 fixed gate subsystem is 0204753.

```
Is BDI 204321 correct for OLTP-TM2200 fgss: <Yes>/No ▶


    --------------------------------------------------------------------------------


Enter DDP-PPC Relocatable LIB: <SYS$LIB$*DDP-PPC-1> ▶
```

This file will be used as the DDP-PPC object module Library during generation of OSI-TP elements. Q*F>4chars

```
Enter  DDP-PPC  Copy  Library: <SYS$LIB$*DDP-PPC-3> ▶
```

This will be used as the DDP-PPC copy library during the compilation of OSI-TP elements. Q*F>4chars

```
Enter RTS Object Module Library: <SYS$LIB$*EMOMRTS> ▶
```

This file must contain the URTS$TABLES object module during the linking of OSI-TP elements. Q*F>4chars

```
Enter ELMS library file: <SYS$LIB$*ELMS> ▶
```

This will be the Extended Language Message System Library during generation of OSI-TP elements. Q*F>4chars

```
Send GEN output to: <Tape>/Printer ▶
```

This specifies the destination of the generated printout.

WARNING

Print consolidation will only be possible if the generated printout is kept on tape.

```
Consolidate print: <Yes>/No ▶
```

For more information on print consolidation, see 5.3.2 and 5.3.3.

```
NEWCONSP - reel/file./NONE/<> ? ▶
```

Enter an alternate reel or file for storing the consolidated print file from this generation. The current generation output will be saved in the file PRNTF$. This file will not be printed during this generation.

```
The runstream has been saved in 'ABC*COMRUN(1).1/OSI-TPxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶

The runstream has been saved in 'ABC*COMRUN(1).1/OSI-TPxRy'
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED *******************
COMMAND ? ▶exit
END COMUS
```

When the preceding runstream has completed its execution without any error, OSI-TP level *xRy* will be generated.

5.4.29. PADS

PADS COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates PADS. This example assumes PADS has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶BUILD PADS,xRy
```

This BUILD command directs COMUS to apply changes to PADS level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (< >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation  (< >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ?  ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?   ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ?  ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ?  ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ?  ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
NEWMASTER - reel/file./<> ?  ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of PADS.

```
Type of generation (FULL or <UPDATE>) ?
```

> **FULL**   Changes are applied, and all elements are recompiled.
>
> **UPDATE**   Changes are applied, and only elements that have changed are recompiled.

```
Additional SGSs (<END>) ?  ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/PADSxRy'

View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/PADSxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/PADS*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED ************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the PADS level *xRy* will be generated.

5.4.30. PAR

PAR COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the PAR default session:

```
Default maximum run time (<9999>) ? ▶
```

Enter the maximum run time for the @RUN control statement. This is the estimated standard units of processing (SUP) usage in minutes. The maximum value allowed is 9999.

```
Default maximum pages (<99999> ) ? ▶
```

Enter the maximum pages for the @RUN control statement. The maximum value allowed is 99999.

```
Send consolidated print output to NEWMASTER (Y or <N>) ? ▶
```

Consolidated print is done automatically and sent to a tape. You can choose which tape to send it to. If you choose to send it to an alternate reel other than the NEWMASTER, you will be queried for the reel. The default is to query for an alternate reel. For more information on print consolidation, see 5.3.2 and 5.3.3.

```
Default name of the PLUS file (<SYS$LIB$*PLS>) ? ▶
```

Enter the PLUS library file to be used for compilations and collections. If a PLUS file is not needed, enter the word BLANK.

PAR COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates PAR. This example assumes PAR has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶BUILD PAR,xRy
```

This BUILD command directs COMUS to apply changes to PAR level xRy. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter xRy for $xRyA$, $xRyB$, or any other stability level.

```
Project id for this generation (<     >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<    >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
NEWMASTER - reel/file./<> ? ▶
```

The number of the tape that will hold the updated version of PAR level xRy.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Type of generation (FULL or <UPDATE>) ?
```

FULL   Changes are applied, and all elements are recompiled.

UPDATE   Changes are applied, and only elements that have changed are recompiled.

```
OLDCONSP – reel/file./NONE/<> ? ▶
```

Enter a reel number or file that contains the consolidated print file from a previous generation. This query is used only for UPDATE generations.

```
NEWCONSP – reel/file./NONE/<> ? ▶
```

Enter an alternate reel or file for storing the consolidated print file from this generation. The current generation output will be saved in the file PRNTF$. This file will not be printed during this generation.

```
Type of CULL (ALL or <NONE>) ? ▶
```

CULL generates a printed cross reference listing for all tags or variables in the source code.

ALL   Listings are produced for all source code. You must have CULL installed on your system in order to use this option.

NONE   No CULL is performed.

```
Additional SGSs (<END>) ? >
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/PARxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/PARxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/PAR*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED   ************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the PAR level *xRy* will be generated.

5.4.31. PCIOS

PCIOS COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates PCIOS. This example assumes PCIOS has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

►@QUAL *ABC*
►@COMUS
COMMAND ? ►BUILD PCIOS,*xRy*

This BUILD command directs COMUS to apply changes to PCIOS level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

>Project id for this generation (<   >) ? ►

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

>Run id for this generation (<   >) ? ►

The RUNID to be used on the @RUN control statement.

>MASTER-reel/file./<> ? ►

Specify the reel number of the product master tape used as an input to the generation. A reel number can contain up to six letters or digits. It is used to identify the product master tape in a LOAD message when the generation runstream executes.

> Generation id ? ►

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

Generation heading (<>) ? ►

Specify a heading to appear on the printed output for the generation run.

Generation reason (<END>) ? ►

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

New change number (<END>) ? ►

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
>FULL or UPDATE generation - F/<U> ?  ▶
```

F or **FULL**	All product components are compiled and the selected changes are applied.
U or **UPDATE**	Only those elements with changes are compiled.

```
>NEWMASTER - reel/file./<> ?  ▶
```

Specify the reel number to be used for the new product master tape to be produced by this generation. The NEWMASTER reel number is the number of the tape that will hold the new, generated version of PCIOS.

```
Print will be saved - queue for output too - Y/<N> ?  ▶
```

Y	It is saved on the output NEWMASTER and you are queried for an output device:
	```Output device - < > ?```
N	It is saved on the output NEWMASTER and you are not queried for an output device.

```
>Additional SGSs (<END>) ?  ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
>The runstream has been saved in 'ABC*COMRUN(1).1/PCIOSxRy'
View the runstream (Y or <N>) ?  ▶
Print a copy of the runstream (Y or <N>) ?  ▶
Start the runstream (Y or <N>) ?  ▶
```

Y	Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:
	The runstream has been started
N	You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/PCIOSxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:
	@START *ABC*COMRUN.1/PCIOS*xRy*

```
>UPDATING ACCESS FILES...
>ACCESS FILES HAVE BEEN UPDATED
>BUILD TASK COMPLETED  ************************
>COMMAND?    ▶EXIT
```

When the preceding runstream has completed its execution without any error, the PCIOS level *xRy* will be generated.

5.4.32. PDP

PDP COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates CML. This example assumes CML has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶BUILD PDP,xRy
```

This BUILD command directs COMUS to apply changes to PDP level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<   >) ?  ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation  (<   >) ?  ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ?  ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ?  ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Type of generation (FULL or <UPDATE>) ?
```

> **FULL** Changes are applied, and all elements are recompiled.
>
> **UPDATE** Changes are applied, and only elements that have changed are recompiled.

```
NEWMASTER - reel/file./<> ?  ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version.

```
Print will be saved - queue for output too - Y/<N> ?  ▶
```

> **Y** It is saved on the output NEWMASTER and you are queried for an output device:
>
> ```
> Output device - <    > ?
> ```
>
> **N** It is saved on the output NEWMASTER and you are not queried for an output device.

```
Additional SGSs (<END>) ?  ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and the *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/PDPxRy'
```

The generated output runstream is stored in the default element of the default file.

```
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in '*ABC**COMRUN(1).1/PDP*xRy*'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement

@START *ABC**COMRUN.1/PDP*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED *************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the PDP level *xRy* will be generated.

5.4.33. PMD

PMD COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates PMD. This example assumes PMD has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶BUILD PMD,xRy
```

This BUILD command directs COMUS to apply changes to PMD level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<>) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation  (<>) ?  ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ?  ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ?  ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ?  ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ?  ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
FULL or UPDATE generation - F/<U> ?  ▶
```

F or **FULL** All product components are compiled and the selected changes are applied.

U or **UPDATE** Only those elements with changes are compiled.

```
NEWMASTER - reel/file./<> ?  ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version.

```
Print will be saved on tape - queue for output too - Y/<N> ? ▶
```

Y   The printout is saved on the output NEWMASTER and you will be queried for an output device:

```
Output device - <    > ?
```

N   The printout is saved on the output NEWMASTER and you will not be queried for an output device.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/PMDxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in '*ABC**COMRUN(1).1/PMD*xRy*'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/PMD*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED *************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the PMD level *xRy* will be generated.

5.4.34. RSS

RSS COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the RSS default session:

```
BRKPT print output to file - <Y>/N/QUERY ?  ▶
```

YES	@BRKPT PRINT$,file
NO	Generation print output in PRINT$
QUERY	Determine handling at generation time

```
Cataloged or temporary BRKPT file - <C>/T/QUERY ?  ▶
```

C/CAT	Catalogued file (cycle +1)
T/TEMP	Temporary file
QUERY	Determine handling at generation time

```
BRKPT file name - <PRINT> ?   ▶
```

This is the EXTERNAL name of the BRKPT file. You can enter "qualifier*filename", or just "filename" if you want the project ID to be used as the qualifier.

```
SYM the print output file - Y/<N>/QUERY ?      ▶
```

YES	@SYM PRINT,,DMS
NO	Do not SYM
QUERY	Determine handling at generation time

RSS COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates RSS. This example assumes RSS has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
►@QUAL ABC
►@COMUS
```
COMMAND ?   ►BUILD RSS,*xRy*

This BUILD command directs COMUS to apply changes to RSS level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<    >) ? ►
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from the set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<   >) ? ►
```

The RUNID to be used on the @RUN control statement

```
MASTER - reel/file./<> ? ►
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?   ►
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ►
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ►
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ►
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
NEWMASTER - reel/file./<> ?   ▶
```

Enter the reel number(s) or catalogued tape file name for assigning the NEWMASTER tape. File name is denoted by the presence of a period or an asterisk. If a reel number is not specified, default spaces are used.

```
Type of generation (FULL or <UPDATE>) ?
```

FULL Changes are applied, and all elements are recompiled.

UPDATE Changes are applied, and only elements that have changed are recompiled.

```
Enter types of elements to compile    type/ALL/<END> ▶
```

Specify the following element type for the COMPILE SGS:

ADMLP	ELT	PLSCOPY
ASM	FLT	PLSENV
ASMP	FOR	SDDL
ASMPROC	FORP	SSG
COB	INL	SYNTRAN
COBP	MAP	TCL
CTS	MCON	TCON
DCON	MSD	UDMLP
DDL	MSM	UDMLPCOPY
DOC	PLS	

"AL" compiles all elements

```
Enter names of elements to process element-name/<END> ▶
```

Specify an element for the PROCESS SGS.

```
Enter types of elements to CULL type/ALL/<END> ▶
```

Specify the types of elements in the symbolic output file for which a CULL is to be produced.

ASM	MSM	MSD	PLS	COB	SSG

"ALL" produces a single cull of the entire output file.

```
Additional SGSs (<END>) ? >
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
Enter file which contains PLUS compiler <SYS$LIB$*PLS> ? ▶
```

Enter filename, no period, that contains the PLUS compiler.

```
Enter file to be used for PLUS RLIB <SYS$LIB$*PLS> ? ▶
```

Enter filename, no period, that contains the PLUS library.

```
Enter file to be used for SYSTEM REL library <SYS$LIB$*SYSLIB>  ? ▶
```

Enter filename, no period, that contains the SYSTEM REL library.

```
Enter file that contains the DDP-PPC relocatables <SYS$LIB$*DDP-PPC-1>  ? ▶
```

Enter filename, no period, that contains the DDP-PPC relocatables.

```
The runstream has been saved in 'ABC*COMRUN(1).1/RSSxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/RSSxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/RSS*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED  *************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the RSS level *xRy* will be generated.

5.4.35. SIMAN

SIMAN COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the SIMAN default session:

```
Default maximum run time (<9999>) ? ▶
```

Enter the maximum run time for the @RUN control statement. This is the estimated standard units of processing (SUP) usage in minutes. The maximum value allowed is 9999.

```
Default maximum pages (<99999> ) ? ▶
```

Enter the maximum pages for the @RUN control statement. The maximum value allowed is 99999.

```
Send consolidated print output to NEWMASTER (Y or <N>) ? ▶
```

Consolidated print is done automatically and sent to a tape. You can choose which tape to send it to. If you choose to send it to an alternate reel other than the NEWMASTER, you will be queried for the reel. The default is to query for an alternate reel. For more information on print consolidation, see 5.3.2 and 5.3.3.

```
Default name of the PLUS file (<SYS$LIB$*PLS>) ? ▶
```

Enter the PLUS library file to be used for compilations and collections. If a PLUS file is not needed, enter the word BLANK.

SIMAN COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates SIMAN. This example assumes SIMAN has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶BUILD SIMAN,xRy
```

This BUILD command directs COMUS to apply changes to SIMAN level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<        >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product

generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<      >) ? ▶
```

  The RUNID to be used on the @RUN control statement

```
MASTER - reel/file./<> ? ▶
```

  The MASTER reel number is the number of the tape containing the product source code.

```
NEWMASTER - reel/file./<> ? ▶
```

  The NEWMASTER reel number is the number of the tape that will hold the updated version.

```
Generation id ?  ▶
```

  Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

  Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

  Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

  Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Type of generation (FULL or <UPDATE>) ?
```

    FULL   Changes are applied, and all elements are recompiled.

    UPDATE Changes are applied, and only elements that have changed are recompiled.

```
OLDCONSP - reel/file./NONE/<> ? ▶
```

  Enter a reel number or file that contains the consolidated print file from a previous generation. This query is used only for UPDATE generations.

```
NEWCONSP - reel/file./NONE/<> ? ▶
```

You can enter an alternate reel or file for storing the consolidated print file from this generation. The current generation output will be saved in the file PRNTF$. This file will not be printed during this generation.

```
Type of CULL (ALL or <NONE>) ? ▶
```

CULL generates a printed cross reference listing for all tags or variables in the source code. Enter the type of CULL to be performed:

> **ALL**   Listings are produced for all source code.
>
> **NONE**   No CULL is performed.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/SIMANxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

> **Y**   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:
>
>   **The runstream has been started**
>
> **N**   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in '*ABC**COMRUN(1).1/SIMAN*xRy*'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:
>
>   @START *ABC**COMRUN.1/SIMAN*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED  *************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the SIMAN level *xRy* will be generated.

5.4.36. SORT

SORT COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates SORT. This example assumes SORT has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
►@QUAL ABC
►@COMUS
COMMAND ?   ►BUILD SORT,xRy
```

This BUILD command directs COMUS to apply changes to SORT level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
>Project id for this generation (<    >) ? >►
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
>Run id for this generation (<     >) ? ►
```

The RUNID to be used on the @RUN control statement

```
>MASTER - reel/file./<> ? ►
```

Specify the reel number of the product master tape used as an input to the generation. A reel number can contain up to six letters or digits. It is used only to identify the product master tape in a LOAD message when the generation runstream executes.

```
Generation id ?   ►
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ►
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ►
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
FULL or UPDATE generation - F/<U> ? ▶
```

      F or **FULL**   All product components are compiled and the selected changes are applied.

      U or **UPDATE**   Only those elements with changes are compiled.

```
>NEWMASTER - reel/file./<> ? ▶
```

Specify the reel number to be used for the new product master tape to be produced by this generation. A reel number can contain up to six letters or digits. It is used only to identify the tape in a LOAD message when the generation runstream executes.

```
>Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
>Sym generation printfile - <NO>/YES ? ▶
```

Sort generation listings are saved in a printfile which is copied to the output newmaster. This printfile may be symmed to the default printer during the execution of the build runstream by responding YES.

WARNING

FULL generations create a large listing.

```
>The runstream has been saved in 'ABC*COMRUN(1).1/SORTxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y	Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N	You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in '*ABC**COMRUN(1).1/SORT*xRy*'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/SORT*xRy*

```
>UPDATING ACCESS FILES ...
>ACCESS FILES HAVE BEEN UPDATED
>BUILD TASK COMPLETED  ********************
>COMMAND ? ▶EXIT
```

When the preceding runstream has completed its execution without any error, the SORT level *xRy* will be generated.

Configuring SORT Internal Parameters

See the *OS 2200 SORT/Merge Programming Guide* (7831 0687) for information on configuring SORT internal parameters. Please note that configuring SORT is required only if you want to change the parameters from their default value.

5.4.37. SSG

SSG COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the SSG default session:

```
Default maximum run time (<9999>) ? ▶
```

Enter the maximum run time for the @RUN control statement. This is the estimated standard units of processing (SUP) usage in minutes. The maximum value allowed is 9999.

```
Default maximum pages (<99999> ) ? ▶
```

Enter the maximum pages for the @RUN control statement. The maximum value allowed is 99999.

```
Send consolidated print output to NEWMASTER (Y or <N>) ? ▶
```

Consolidated print is done automatically and sent to a tape. You can choose which tape to send it to. If you choose to send it to an alternate reel other than the NEWMASTER, you will be queried for the reel. The default is to query for an alternate reel. For more information on print consolidation, see 5.3.2 and 5.3.3.

```
Default name of the PLUS file (<SYS$LIB$*PLS>) ? ▶
```

Enter the PLUS library file to be used for compilations and collections. If a PLUS file is not needed, enter the word BLANK.

SSG COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates SIMAN. This example assumes SIMAN has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶BUILD SSG,xRy
```

This BUILD command directs COMUS to apply changes to SSG level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (< >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (< >) ? ▶
```

The RUNID to be used on the @RUN control statement

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version.

```
Generation id ?   ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Type of generation (FULL or <UPDATE>) ?
```

> **FULL**　　Changes are applied, and all elements are recompiled.
>
> **UPDATE**　Changes are applied, and only elements that have changed are recompiled.

```
OLDCONSP - reel/file./NONE/<> ? ▶
```

Enter a reel number or file that contains the consolidated print file from a previous generation. This query is used only for UPDATE generations.

```
NEWCONSP - reel/file./NONE/<> ? ▶
```

Enter an alternate reel or file for storing the consolidated print file from this generation. The current generation output will be saved in the file PRNTF$. This file will not be printed during this generation.

```
Type of CULL (ALL or <NONE>) ? ▶
```

CULL generates a printed cross reference listing for all tags or variables in the source code. Enter the type of CULL to be performed:

> **ALL**　　Listings are produced for all source code.
>
> **NONE**　No CULL is performed.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/SSGxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/SSGxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/SSG*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED   ************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the SSG level *xRy* will be generated.

5.4.38. SYSLIB

SYSLIB COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates SYSLIB. This example assumes SYSLIB has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶BUILD SYSLIB,xRy
```

This BUILD command directs COMUS to apply changes to SYSLIB level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (< >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation  (< >) ?
```
▶

The RUNID to be used on the @RUN control statement.

```
MASTER - reel/file./<> ?
```
▶

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?
```
▶

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ?
```
▶

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ?
```
▶

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ?
```
▶

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
Type of generation (FULL or <UPDATE>) ?
```

FULL Changes are applied, and all elements are recompiled.

UPDATE Changes are applied, and only elements that have changed are recompiled.

```
NEWMASTER - reel/file./<> ?
```
▶>

The NEWMASTER reel number is the number of the tape that will hold the updated version.

```
Print will be saved - queue for output too - Y/<N> ? ▶
```

Y   It is saved on the output NEWMASTER and you are queried for an
          output device:

```
Output device - <    > ?
```

N   It is saved on the output NEWMASTER and you are not queried for an
          output device.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200
COMUS End Use Reference Manual*, but they apply to this generation only. If you
enter any SGSs, COMUS repeats the question until you enter END or a null
transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and
*OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference
Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/SYSLIBxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch
          run using the @START control statement. COMUS responds with:

          The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds
          with: **The runstream has been saved in
          'ABC*COMRUN(1).1/SYSLIBxRy'** After the generation runstream
          is complete, you can start the runstream from demand mode with the
          @START statement:

          @START *ABC*COMRUN.1/SYSLIBxRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED ************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the
SYSLIB level *xRy* will be generated.

Stream Generation Status for SYSLIB

SYSLIB assumes that the PLUS compiler, the INLINE processor, and the PLUS library reside in the file SYSLIB*PLS. If they do not, enter the following as permanent SGSs or additional SGSs during the COMUS Build Session to define new default values. (Note that the quotation marks are two apostrophes (' '), and not double quotation marks (").)

```
CO$PLSPROCESSOR CALL NAME IS ''file.element-of-plus-compiler'';
OPTIONS ARE ''GKLMRZ'';
LIBRARY FILE IS ''file-of-plus-library''
CO$INLINEPROCESSOR CALL NAME IS ''file.element-of-inline-processor'';
OPTIONS ARE ''S''
```

The files that you specify here must exist when you perform the build; otherwise, you will get errors.

Every time you enter a CO$PLSPROCESSOR SGS during a build, you must enter the complete SGS, including the LIBRARY FILE IS field.

5.4.39. TAS (TCP/IP APPLICATION SERVICES)

TAS COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the TAS default session:

```
TAS build output: Tape/<Printer>/Query ▶ Q
```

Define the normal handling of generation printout:

- Tape: Save generation output on tape.
- Printer: Send generation output to a print Queue.
- Query: Determine handling at generation time.

```
Use Print consolidation: Yes/<No> ▶
```

All generation output can be consolidated onto a single tape. This tape will contain the current version of all elements. Print consolidation can be done within the product generation or afterwards using SGP. This requires that print consolidation be used on every generation.

```
Enter PLUS relocatable library name: <SYS$LIB$*PLS> ▶ Q
```

This file will be used as the PLUS relocatable library source during the collection of the TAS absolutes.

```
Enter DDP-PPC library name: <SYS$LIB$*DDP-PPC-1> ▶ Q
```

This file will be used as the PLUS COPY library for those modules that relate to DDP-PPC functions.

```
Enter filename for Free Standing Audit Handler (FSAH)
Library file: <SYS$LIB$*FSAH> ▶ Q
```

Enter fully qualified filename of the file that contains the IRU Free Standing Audit Handler (FSAH) runtime and copy library elements.

```
Enter UCS run-time library file: <SYS$LIB$*URTS> ▶ Q
```

The fully qualified file name of the library that contains the UCS run time library elements.

```
Are you using DPS menus with TAS: <Yes>/No ▶
Do you install TAS in mode A, B, C, D, G, or H for DPS menus?
```

- If TAS is to use the DPS 2200 full-screen interface, use the default answer **Y**. In this case, the next query for the DPS relocatable library occurs. In response, supply the name of the file that contains the DPS 2200 relocatable, copy, and utility elements on the system where you are doing the TAS generation.

- You must install TAS in mode A, B, C, D, G, or H for DPS menus.

- If TAS is to use the program-callable interface only, answer **N**. In this case, the query for the DPS relocatable library does not occur.

```
Enter DPS library name: <SYS$LIB$*DPS> ▶ Q
```

This file will be used as the library for DPS PLUS COPY elements during the compilation of TAS relocatables.

```
Use IPF with TAS: <No>/Yes ▶Y
```

Did you install TAS in mode A, C, E, G or J for use with IPF?

```
Enter IPF relocatable library: <SYS$LIB$*IPF-2> ▶ Q
```

This file will be used as the IPF relocatable library during the collection of the TAS absolutes.

```
Use the SMTP feature of TAS: <Yes>/No ▶
Do you install TAS in mode A, B, C, D, E, or F for SMTP use?
```

Answer **Y** to provide the TAS mail processor or **N** if you are not using the mail processor. This is the Simple Mail Transfer Protocol (SMTP) feature. You must install TAS in mode A, B, C, D, E, or F for use with SMTP.

```
Did you install the full DMS1100 product or are you using the
DMS subproduct from UDS Control: <FULL>/SUB  ▶
```

Indicate whether your site has the full DMS 2200 product installed, or whether you are using the DMS subproduct that is supplied as part of UDS Control. You must have full DMS 2200 to support DMS applications.

```
Enter the universal DMS qualifier: <UDS$$SRC>  ▶Q
```

The application group qualifier for the application under which you installed TAS.

```
Enter the UDSC file name: <UDS$$SRC*DMRMT$>  ▶Q
```

The fully qualified file name of the library that contains the UDSC relocatable element LINKER.

```
Enter fully qualified filename and element that contains the DMS
entry point definitions: <UDS$$SRC*DMRMT$.CBEP$$DMS>  ▶Q
```

The DMS entry points are required during the collection of those TAS absolutes that interface with DMS.

```
Use the SMTP Menu Security feature: Yes/<No>  ▶
```

Do you install TAS in mode C or D ?

TAS defaults complete.

TAS COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates TAS. This example assumes TAS has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ? ▶BUILD TAS,xRy
```

This BUILD command directs COMUS to apply changes to TAS level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (< >) ?  ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (< >) ? ▶
```

The RUNID to be used on the @RUN control statement. Generation recovery is not available for TAS.

```
MASTER – reel/file./<> ? ▶
```

Specify the reel number of the product master tape used as an input to the generation. A reel number may contain up to six letters or digits. It is used only to identify the product master tape in a LOAD message when the generation runstream executes.

```
Generation id ? ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit. Only 48 characters per line are accepted.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
NEWMAST – reel/file./<> ? ▶
```

Specify the reel number to be used for the new product master tape to be produced by this generation. A reel number may contain up to six letters or digits. It is used only to identify the tape in a LOAD message when the generation runstream executes.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

DOMAP SGS causes mapping of TAS mail routines that interface with DMS 2200. DPSLIB <dps-library> SGS is required if a DOMAP SGS is specified.

```
Compile ALL elements: Yes/<No> ►Y
```

Yes   ALL ELEMENTS ARE RECOMPILED.

No   ONLY ELEMENTS WITH CHANGES ARE RECOMPILED.

```
Which mode are you installing with this BUILD: <A> ►
```

There are 9 install mode for TAS:

- Mode A: FTP, SMTP, Menus, IPF

- Mode B: FTP, SMTP, Menus

- Mode C: FTP, SMTP, Menus Security, IPF

- Mode D: FTP, SMTP, Menus Security

- Mode E: FTP, SMTP, IPF

- Mode F: FTP, SMTP

- Mode G: FTP, Menu, IPF

- Mode H: FTP, Menu

- Mode J: FTP, IPF

The *OS 2200 TCP/IP Application Services (TAS) Implementation and Administration Guide* (3787 3221) describes available installation modes.

Note: *The SOLAR installation mode for the NEWMASTER tape resulting from this BUILD can only be installed in that mode.*

```
Enter DPS library file: <SYS$LIB$*DPS> ►
```

```
Enter PLUS library file: <SYS$LIB$*PLS> ►
```

```
Enter DDP-PPC COPY library file: <SYS$LIB$*DDP-PPC-1>        ►
```

FSAH: Free Standing Audit Handler

```
              Enter IRU FSAH library file: <SYS$LIB$*FSAH> ▶

              Enter UCS run time library file: <SYS$LIB$*URTS> ▶

              Enter IPF relocatable library file: <SYS$LIB$*IPF-2> ▶

              Is the FULL DMS1100 product installed OR are you using the
              DMS subproduct from UDS Control: <FULL>/SUB ▶
              Enter universal DMS qualifier: <UDS$$SRC> ▶
              Enter the UDSC filename: <UDS$$SRC*DMRMT$> ▶

              Enter the fully qualified file name and relocatable
              element that contains your DMS entry point definitions
              <UDS$$SRC*DMRMT$.CBEP$$DMS> ▶

              Gen output to: <Tape>/Printer ▶

              Assign a new tape for generation output: <Yes>/No ▶
              GENOUT - reel/file./NONE/<> ? ▶

              The runstream has been saved in 'ABC*COMRUN(1).1/TASxRy'
              View the runstream (Y or <N>) ? ▶
              Print a copy of the runstream (Y or <N>) ? ▶
              Start the runstream (Y or <N>) ? ▶
              The runstream has been saved in 'ABC*COMRUN(1).1/TASxRy'
              UPDATING ACCESS FILES ...
              ACCESS FILES HAVE BEEN UPDATED
              BUILD TASK COMPLETED ************************
              COMMAND ? ▶exit
              END COMUS
```

When the preceding runstream has completed its execution without any error, TAS level *xRy* will be generated.

5.4.40. UC (USC C)

UC COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the UC default session.

The default processors for a UC BUILD are defined through CO$xyzPROCESSOR SGSs. To change a processor call, enter a new CO$xyzPROCESSOR SGS when you are queried for permanent SGSs. The CO$xyzPROCESSOR SGS format follows:

```
  CO$xyzPROCESSOR  CALL NAME IS "call name" ;
  OPTIONS ARE "options" ;
  LIBRARY FILE IS "library file name"
```

Use double apostrophes in place of the quotes.

```
View a table of default processors (<Y>/N) ? ▶
View an example of changing a CO$xyzPROCESSOR SGS (<Y>/N) ? ▶
```

UC COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates UC. This example assumes UC has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶BUILD UC,xRy
```

This BUILD command directs COMUS to apply changes to UCS C level xRy. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter xRy for $xRyA$, $xRyB$, or any other stability level.

```
Project id for this generation (< >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation  (< >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
LSS-SRC - reel/file./<> ? ▶
```

This is the reel number of the tape containing the LSS product. LSS work files are used to build UC.

```
What is the product release level of LSS on this tape? ▶
```

This is the product release level of LSS on the tape you have provided in the previous query.

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*.

```
FULL or UPDATE generation – F/<U> ? >
```

F or **FULL**	All product components are compiled and the selected changes are applied.
U or **UPDATE**	Only those elements with changes are compiled.

```
NEWMASTER – reel/file./<> ? ▶>
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of UCS.

```
PTAPE – reel/file./<> ? ▶
```

Enter the reel number of a tape to receive the build print. You can also insert a file name by using a period.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/UCxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in '_ABC_*COMRUN(1).1/UC_xRy_'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START _ABC_*COMRUN.1/UC_xRy_

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED ************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the UC level _xRy_ will be generated.

5.4.41. UCOB (UCS COBOL)

UCOB COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the UCOB default session.

The default processors for a UCOB BUILD are defined through CO$xyzPROCESSOR SGSs. To change a processor call, enter a new CO$xyzPROCESSOR SGS when you are queried for permanent SGSs. The CO$xyzPROCESSOR SGS format follows:

CO$xyzPROCESSOR CALL NAME IS "call name" ;
OPTIONS ARE "options" ;
LIBRARY FILE IS "library file name"

Use double apostrophes (' ') in place of the quotes.

```
View a table of default processors (<Y>/N) ? ▶
View an example of changing a CO$xyzPROCESSOR SGS (<Y>/N) ? ▶
```

UCOB COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates UCOB. This example assumes UCOB has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?   ▶BUILD UCOB,xRy
```

This BUILD command directs COMUS to apply changes to UCOB level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<       >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from the set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation  (<     >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
This compiler build requires the matching LSS tape for this release.
LSS-SRC - reel/file./<> ? ▶
```

This is the reel number of the tape containing the LSS product. LSS work files are used to build UCOB.

```
What is the product release level of LSS on this tape? ▶
```

This is the product release level of LSS on the tape you have in the previous query.

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath*

OS 2200 COMUS End Use Reference Manual. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
FULL or UPDATE generation - F/<U> ? ▶
```

F or **FULL**   All product components are compiled and the selected changes are applied.

U or **UPDATE**   Only those elements with changes are compiled.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of UCS COBOL.

```
PTAPE - reel/file./<> ? ▶
```

Enter the reel number of a tape to receive the build print. You can also insert a file name by using a period.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/UCOBxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/UCOBxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/UCOB*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED ************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the UCOB level *xRy* will be generated.

5.4.42. UCSRTS

UCSRTS COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the UCSRTS default session:

```
UCSRTS printout: save, print or query - <S>/P/Q ?      ▶
```

Define the normal handling of generation printout.

S	Save	Save generation printout - may be printed later
P	Prrint	Queue generation printout to printer - will not be saved
Q	Query	Determine handling at generation time

```
MASM processor filename (Q*F) - <SYS$LIB$*MASM> ? ▶
```

The default MASM processor is in the file SYSLIB*MASM. If an alternative MASM processor is required, specify the name of the file containing it.

```
MASM listings: short, long, none, or query - <S>/L/N/Q ?      ▶
```

The default MASM processor is in the file SYSLIB*MASM. If an alternative MASM processor is required, specify the name of the file containing it.

UCSRTS COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates UCSRTS. This example assumes UCSRTS has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?    ▶BUILD UCSRTS,xRy
```

This BUILD command directs COMUS to apply changes to UCSRTS level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<        >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation (<     >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
MASTER-reel/file./<> ? ▶
```

Specify the reel number of the product master tape used as an input to the generation. A reel number can contain up to six letters or digits. It is used to identify the product master tape in a LOAD message when the generation runstream executes.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
FULL, UPDATE, PROC, SORT, or CSMS generation - F/<U>/P/S/C ? ▶
```

F	FULL	Assemble all UCSRTS symbolic elements
U	UPDATE	Assemble elements with updates
P	PROC	Assemble Sort/Merge procedures and interface routines for ASCII FORTRAN and PL/I (PROC)
S	SORT	Assemble Sort/Merge interface routines for ASCII FORTRAN and PL/I (SORT)
C	CSMS	Assemble Common Storage Management System routines (CSMS).

```
NEWMASTER - reel/file./<> ? ▶
```

Specify the reel number to be used for the new product master tape to be produced by this generation. The NEWMASTER reel number is the number of the tape that will hold the new, generated version of UCSRTS.

```
Print will be saved - queue for output too - Y/<N> ? ▶
```

Y   The build printout will be saved and the following query will appear:

```
Output device - <DMS> ? ▶
```

Specify the name of the printer or print group.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/UCSRTSxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUNxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/UCSRTS*xRy*

```
>UPDATING ACCESS FILES...
>ACCESS FILES HAVE BEEN UPDATED
>BUILD TASK COMPLETED  ************************
>COMMAND?    ▶EXIT
```

When the preceding @START runstream has completed without any error, the UCSRTS level *xRy* will be generated.

5.4.43. UFTN

UFTN COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the UCOB default session:

The default processors for a UCOB BUILD are defined through CO$xyzPROCESSOR SGSs. To change a processor call, enter a new CO$xyzPROCESSOR SGS when you are queried for permanent SGSs. The CO$xyzPROCESSOR SGS format follows:

```
CO$xyzPROCESSOR  CALL NAME IS "call name" ;
OPTIONS ARE "options" ;
LIBRARY FILE IS "library file name"
```

Use double apostrophes (' ') in place of the quotes (").

```
View a table of default processors (<Y>/N) ? ▶
View an example of changing a CO$xyzPROCESSOR SGS (<Y>/N) ? ▶
```

UFTN COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates UFTN. This example assumes COMUS has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?  ▶BUILD UFTN,xRy
```

This BUILD command directs COMUS to apply changes to UFTN level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (<        >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from the set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation  (<     >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
This compiler build requires the matching LSS tape for this release.
LSS-SRC - reel/file./<> ? ▶
```

This is the reel number of the tape containing the LSS product. LSS work files are used to build UFTN.

```
What is the product release level of LSS on this tape? ▶
```

This is the product release level of LSS on the tape you have provided in the previous query.

```
MASTER - reel/file./<> ? ▶
```

The MASTER reel number is the number of the tape containing the product source code.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
FULL or UPDATE generation - F/<U> ? ▶
```

  F or **FULL**   All product components are compiled and the selected changes are applied.

  U or **UPDATE** Only those elements with changes are compiled.

```
NEWMASTER - reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of UFTN.

```
PTAPE - reel/file./<> ? ▶
```

Enter the reel number of a tape to receive the build print. You can also insert a file name by using a period.

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. These SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/UFTNxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

  The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/UFTNxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

  @START *ABC*COMRUN.1/UFTNxRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED ************************
COMMAND ? ▶exit
```

When the preceding runstream has completed its execution without any error, the UFTN level *xRy* will be generated.

5.4.44. URTS

URTS COMUS Default Values

In addition to the default values discussed in 5.3.1 and 5.3.2, the following values will also be queried for by COMUS during the URTS default session:

```
Build TEST mode URTS (Y/<N>) ? ▶
```

You must respond **YES** or **NO**; QUERY is not a valid response for this question. COMUS uses your response as the product default. Therefore, this default will

affect ALL subsequent use of the BUILD command for the UCS Runtime System unless you redefine your product defaults.

- Respond **YES** to build a TEST mode UCS Runtime System as well as an alternate file common bank (AFCB) mode UCS Runtime System.

- Respond **NO** to build only an AFCB mode UCS Runtime System.

Note: *If you choose to build a TEST mode UCS Runtime System, it will be built for all subsequent builds as well.*

COMUS displays these default definition queries for the first generation of each product and whenever you specify the **Q** option on the BUILD command. If this is not the first generation of this product, the first line you see is "Project ID for this generation." COMUS uses the defaults for this and all subsequent generations of the UCS Runtime System level xRy (unless you specify the **Q** option on the BUILD command to redefine the defaults).

The default processors for an URTS BUILD are defined through CO$xyzPROCESSOR SGSs. To change a processor call, enter a new CO$xyzPROCESSOR SGS when you are queried for permanent SGSs. The CO$xyzPROCESSOR SGS format follows:

```
CO$xyzPROCESSOR CALL NAME IS ''call name'' ;
OPTIONS ARE ''options'' ;
LIBRARY FILE IS ''library file name''

View a table of default processors - (<Y>/N) ? ▶
View an example of changing a CO$xyzPROCESSOR SGS - (<Y>/N) ? ▶
```

URTS COMUS Build Session

The following COMUS session example shows the prompts for the BUILD command that generates URTS. This example assumes URTS has been registered with the COMUS database and the default values have already been selected. For information on how to select the default values, see 5.3.1.

```
▶@QUAL ABC
▶@COMUS
COMMAND ?  ▶BUILD URTS,xRy
```

This BUILD command directs COMUS to apply changes to the UCS Runtime System level *xRy*. For stability releases, do not enter a stability level. Enter only the base product level; for example, enter *xRy* for *xRyA*, *xRyB*, or any other stability level.

```
Project id for this generation (< >) ? ▶
```

The project ID is used in the PROJECT-ID field of the @RUN control statement. The project ID can be 1 to 12 characters from set A through Z, 0 through 9, -, and $. It is not required, however, to avoid conflicts in file names used for product generations, a unique project identifier should be used for each generation when performing concurrent generations.

```
Run id for this generation  (< >) ? ▶
```

The RUNID to be used on the @RUN control statement.

```
FULL or UPDATE generation – F/<U> ? ▶
```

F or **FULL**   All product components are compiled and the selected changes are applied.

U or **UPDATE**   Only those elements with changes are compiled.

```
MASTER – reel/file./<> ? ▶
```

The MASTER reel number is the number of the UCS Runtime System tape containing the product source code.

```
Generation id ?  ▶
```

Specify the generation ID to appear in the System Generation Log (SGL) document in the COMUS database.

```
Generation heading (<>) ? ▶
```

Specify a heading to appear on the printed output for the generation run.

```
Generation reason (<END>) ? ▶
```

Specify the reason for the generation. This information is saved in the SGL in the COMUS database. You can enter any text describing the reason for this generation. If you enter text, COMUS repeats the question until you enter END or a null transmit.

```
New change number (<END>) ? ▶
```

Specify the COMUS database CHG document numbers for the changes to be applied during this generation. These changes must have already been inserted in the COMUS database. See the INSERT command description in the *ClearPath OS 2200 COMUS End Use Reference Manual*. If you enter any CHGs, COMUS repeats the question until you enter END or a null transmit.

```
NEWMASTER – reel/file./<> ? ▶
```

The NEWMASTER reel number is the number of the tape that will hold the updated version of the UCS Runtime System.

```
PTAPE – reel/file./<> ? ▶
```

```
Additional SGSs (<END>) ? ▶
```

Enter any additional SGSs. See Section "SGSs FOR Changing URTS (UCS Runtime System) I/O Configuration Values" for a description of the SGSs that change the input/output (I/O) configuration values for the UCS Runtime System. Allowable SGSs include any listed in the *ClearPath OS 2200 COMUS End Use Reference Manual*, but they apply to this generation only. If you enter any SGSs, COMUS repeats the question until you enter END or a null transmit. See the *ClearPath OS 2200 COMUS End Use Reference Manual* and *OS 2200 System Processor SSG Skeleton (SYSKEL) Operations Reference Manual* for more information on SGSs.

```
The runstream has been saved in 'ABC*COMRUN(1).1/URTSxRy'
View the runstream (Y or <N>) ? ▶
Print a copy of the runstream (Y or <N>) ? ▶
Start the runstream (Y or <N>) ? ▶
```

Y   Instructs COMUS to schedule the runstream for execution as a batch run using the @START control statement. COMUS responds with:

The runstream has been started

N   You must start (@START) the runstream yourself. COMUS responds with: **The runstream has been saved in 'ABC*COMRUN(1).1/URTSxRy'** After the generation runstream is complete, you can start the runstream from demand mode with the @START statement:

@START *ABC**COMRUN.1/URTS*xRy*

```
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED ************************
COMMAND ? ▶EXIT
```

When the preceding runstream has completed its execution without any error, the URTS level *xRy* will be generated.

SGSs FOR Changing URTS (UCS Runtime System) I/O Configuration Values

The following is a list of optional SGSs that you can use to change the I/O configuration values for the UCS Runtime System.

PREPFACTOR <integer>

PREPFACTOR defines a new value for the system data format (SDF) sequential file prepping factor. The SDF sequential prepping factor specifies the minimum buffer size used in an SDF sequential I/O operation. All SDF sequential buffer sizes are a multiple of the prepping factor.

Use PREPFACTOR (along with BLOCKSIZE) to control the amount of buffer space allocated for SDF sequential files. Large I/O buffers will reduce I/O standard unit of processing (SUP) charges but may cause heap bank space (URTS$TABLES) shortages.

BLOCKSIZE <integer>

BLOCKSIZE defines a new value for the default SDF sequential block size used during I/O operations when no block size is specified for SDF sequential file open requests, or if the Runtime System implicitly opens an SDF sequential file. You must use BLOCKSIZE in conjunction with PREPFACTOR or the default SDF prepping factor, whichever applies. The value specified for BLOCKSIZE must be larger than the SDF prepping factor in use or it will have no effect on the size of the SDF sequential I/O buffers allocated. This occurs because all SDF sequential buffers must be a multiple of the SDF sequential prepping factor.

Use BLOCKSIZE and PREPFACTOR (or the default SDF prepping factor) together to control the amount of buffer space allocated for SDF sequential files. Large I/O buffers will reduce I/O SUP charges but may cause heap bank space (URTS$TABLES) shortages.

Valid values for <integer> are 112, 224, 448, 1792, 3584, 168, and 14336. The initial default value is 1792.

CONVERSION <integer>

Use this SGS to set the value for the default conversion table. Valid values for <integer> are:

0	(default) Conversion of ASCII to/from ISO EBCDIC with the 128 character set.
1	Conversion of ASCII to/from ISO EBCDIC with the lowercase alphabetic character set.
2	Conversion of ASCII to/from IBM EBCDIC with the KATAKANA character set.
3	Conversion of ASCII to/from IBM EBCDIC with the lowercase alphabetic character set.
4	Conversion of ASCII to/from ISO EBCDIC with the KATAKANA character set.

HEAPSIZE <integer>

HEAPSIZE defines the Runtime System heap area size used for record and buffer areas, Executive request (ER) packet areas, I/O tables, General Syntax Analyzer (GSA 2200) messages, and data staging for other basic mode routines. Valid values for <integer> are between 1000 and 170000. The initial default value for HEAPSIZE is 10000.

MAXSIZE <integer>

To change the maximum size of the UFTN Unit Table, enter the MAXSIZE SGS with an integer value. The initial default value for MAXSIZE is 100. Valid values for <integer> are 1 to 512.

UNIT <unit-number>,[*]<unit-type>

To change the value of existing units or to add units to the UFTN Unit Table, enter one UNIT SGS for each new unit you want to define or redefine. The following values are valid for the unit-type parameter:

0	Not Specified
1	SDF Sequential
2	ANSI Tape
3	Card Reader
4	Printer
5	Card Punch
6	Alternate Card Reader
7	Alternate Printer
8	Alternate Card Punch

If the <unit-type> is flagged with an asterisk (*), the unit will also be designated as a REREAD unit.

Appendix A
Setting Up the Security Environment for Object Module Subsystems

Note: *This appendix applies only to object module and chameleon subsystems for sites with Security Level 1 or higher (SENTRY_CONTROL = TRUE). If your system has Fundamental Security (SENTRY_CONTROL = FALSE), a generic security environment is automatically provided, and you have no control over its function. For more information, refer to the* ClearPath OS 2200 Apex Help (8207 4154).

When installing software products that have a subsystem or are part of a subsystem, as identified in Section 4, you must set up the security environment as described in this appendix to have the subsystem function correctly and securely.

The security requirements for each subsystem vary and also depend on your site's security environment. Subsystem entry-point protection is available with Security Level 1 or higher and is always in effect with Security Level 3.

This appendix describes the following:

- Security attributes of subsystems

- Security requirements with Security Level 1 or 2

- Security requirements for customer created subsystems

- Procedure for setting up the security environment

A.1. Security Attributes of Subsystems

The security attributes of a subsystem define

- The security environment that a subsystem executes in

- Who can enter the subsystem when subsystem entry-point protection is in effect

- Who can deactivate the subsystem

Each object module subsystem receives its security attributes from the user-id that owns the file into which the subsystem definition element is installed.

Each user-id that owns a subsystem definition file must have the following security attributes:

- General Access Permission Privilege

- Subsystem Sharing Level – Application

- Processor Privilege – Can Only Read Executive GRS

- Access Privilege – Trusted or Shell

Any additional security attributes required depend on the subsystem type, the security environment, and the individual product requirements as described later in this appendix.

For information about how to set those attributes, see the *ClearPath OS 2200 Apex Help* (8207 4154).

The following figure shows a sample security environment for an object module subsystem.
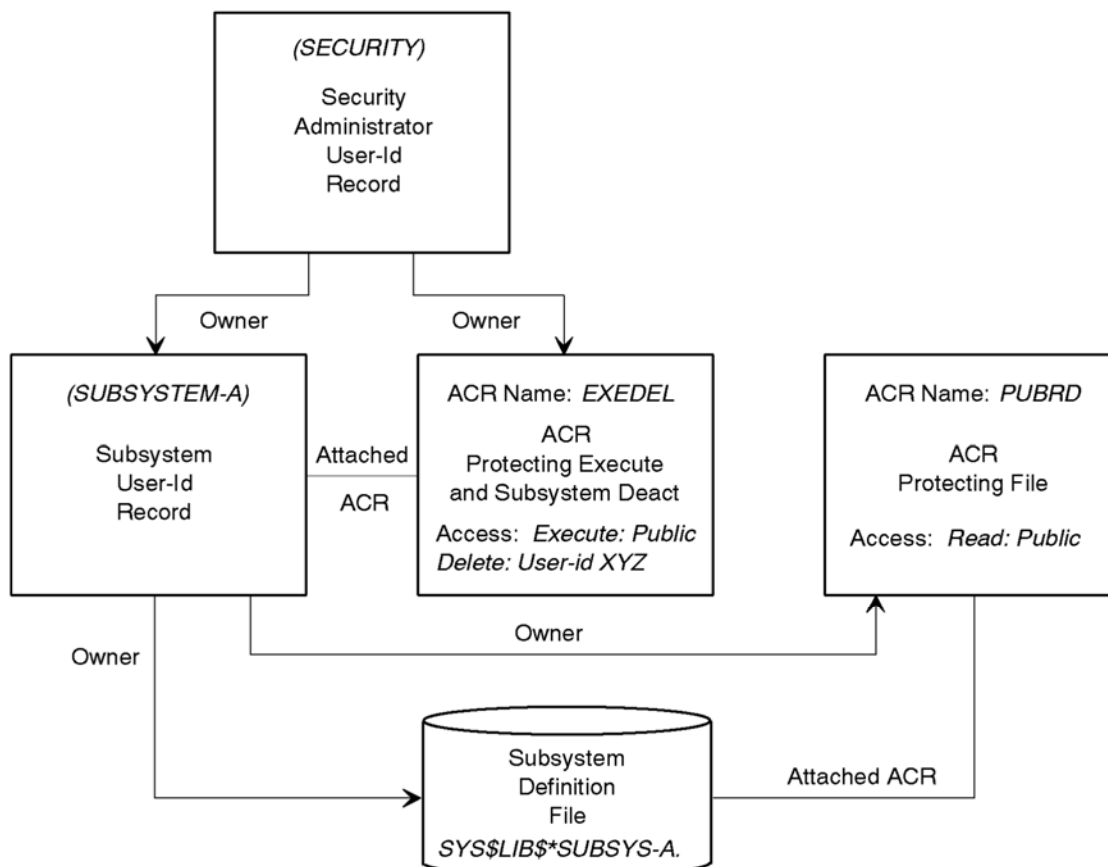
Note: *Examples of user-id names, file names, ACR names, and ACR attributes are in italics.*

A.1.1. Security Access to Enter a Subsystem

There are no constraints on who can access an object module subsystem when subsystem entry-point protection is not in effect.

When entry-point protection is in effect, there are no constraints on who can access *chameleon* subsystems, but all users are subjected to a security check before gaining access to *protected* object module subsystems.

Note: *An SSANYCALLME privilege is provided to automatically satisfy the trusted privilege relationship between subsystems. A subsystem with SSANYCALLME can be entered by any subsystem that passes mandatory and discretionary access validations, regardless of the trusted privileges held by either subsystem. Normally, with object module subsystem entry-point protection in effect, transitions are not allowed unless the trusted privileges held by the calling subsystem are a subset of or equal to the trusted privileges held by the target subsystem. SSANYCALLME is unenforced when delivered; so all subsystems have the privilege. However, sites can later enforce it to regulate its use.*

A.1.2. Security Access to Deactivate a Subsystem

To deactivate a subsystem, you must satisfy the following security constraints:

- The caller (deactivator) must have *enter access* to the subsystem—that is, enter access to the user-id record of the owner of the file containing the subsystem's definition element.

- The caller (deactivator) must have *execute access* to the file containing the subsystem's definition element.

- One of the following must be true:

  − The caller (deactivator) must be the owner of the subsystem.

  − The user-id record of the subsystem owner (that is, the owner of the subsystem's subsystem definition file) must have an access control record (ACR) attached that grants delete access to the caller.

  − The user-id record of the subsystem owner is a public user-id record.

These conditions do not mean the caller needs the ability to actually delete the owner's user-id record. The conditions only mean that the ACR is granting delete access. In fact, other security mechanisms usually prevent the caller from deleting the user-id record itself.

Only a small number of users must be allowed to deactivate subsystems. Normally this would include system administrators, operations personnel, or persons who install software products.

In addition to using DEACT to deactivate subsystems, object module subsystems are deactivated when using library maintenance tools such as INSTALL, LIBLOAD, PRODLD, and PRODRV.

A.1.3. Saving and Loading Library Files That Contain Subsystems

The tools used to save and load the library files of products with object module subsystems have an effect on the security environment of the subsystem and might affect user access to those products.

The following subsections provide information you must consider before selecting a specific tool for loading and saving library files.

LIBSAVE/FAS and LIBLOAD/FAS

When you save and restore library files using the File Administration System (FAS) version of LIBSAVE and LIBLOAD, the file security attributes of object module subsystems are also saved and restored. If you transfer or move these library files from one system to another, the security attributes might not be restored properly unless both systems have the same security environment defined.

LIBSAVE/FAS saves the security attributes of all subsystem files, and LIBLOAD/FAS attempts to restore these saved attributes. LIBSAVE/FAS does not save the user-id records and ACRs that are contained in the security database. LIBLOAD/FAS can only reestablish the saved security environment if the user-ids and ACRs are present when LIBLOAD/FAS is executed. Therefore, be careful to save a copy of the security database whenever you do a LIBSAVE/FAS, and be sure to restore that copy (if needed) when doing a LIBLOAD/FAS.

LIBSAVE and LIBLOAD

When you save and restore library files using LIBSAVE and LIBLOAD, FURPUR is used to save and load the library files. FURPUR does not save or restore security attributes. During the load, the files loaded inherit the security attributes of the user-id executing LIBLOAD. You then need to reestablish the security environment as described in this appendix.

PRODSV and PRODLD

PRODSV and PRODLD use Program-Callable FURPUR (PCFP) to load the library files. During the load, the files loaded inherit the security attributes of the user-id executing PRODLD. You then need to reestablish the security environment as described in this appendix.

A.2. Security Requirements with Security Level 1 or 2

This subsection describes the security requirements for subsystems with and without subsystem entry-point protection configured with Security Level 1 or 2.

Every subsystem must have a user-id attached as the owner of its subsystem definition file. That user-id must have the following security attributes:

- General Access Permission Privilege

- Subsystem Sharing Level—Application

- Processor Privilege—Can Only Read Executive GRS

- Access Privilege—Trusted or Shell

You can assign those attributes using Apex as described in the *ClearPath OS 2200 Apex Help* (8207 4154).

All protected subsystems must have their own unique user-ids. Protected subsystems that do not require additional security privileges could use a single user-id, but Unisys recommends that each protected subsystem have its own unique user-id. Table A–1 shows the user-ids that Unisys recommends for each Unisys product with a subsystem. The table lists subsystem type, recommended user-id, and the default subsystem definition file name. The recommended user-id typically contains a hyphen as the first and last character to eliminate most conflict with existing site-defined user-ids.

Note: *Chameleon subsystems can use the same recommended user-id. See Table A–2 for requirements.*

Table A–1. Subsystem Information

Product Installation Name	Subsystem Type	Recommended User-Id	Default Subsystem Definition File
CIFS	Protected	-CIFS-ADMIN-	SYSLIB*CIFS$SS.
Cipher-API	Protected	-CIPHER-SS-	SYSLIB*CIPHER-API.
COMAPI	Protected	-COMAPI-	SYSLIB*COMAPI.
CPComm	Protected	-CPCOMM-SS-	SYSLIB*CPCOMM.
CPCommOS	Protected	-CPCOMM-SS-	SYSLIB*CPCOMM.
cpFTP	Protected	-FTP-SUBSYS-	SYSLIB*CPFTP$LIB.
CS2200	Protected	-CS2200-SS-	SYSLIB*CS2200.
DDP-PPC	Protected	-DDP-PPC-	SYSLIB*PPC$$GATES.
DPS	Protected	-DPS-	SYSLIB*DPS.
ELMS	Chameleon	-CHAMELEON-	SYSLIB*ELMS.
EXPIPE	Protected	-EXPIPE-	SYSLIB*EXPIPE.
FLEX	Protected	-ASIS-	SYSLIB*ASIS-OMS.
HTPIC-2200	Protected	-HTPIC-2200-	SYSLIB*XTP$$GATES.
IC2200	Protected	-IC2200-	IC2200[mode]*ICSS.
INFOACCESS	Protected	-INFOACCESS-	SYSLIB*IASOCK.
I18NLIB	Chameleon	-CHAMELEON-	SYSLIB*I18NLIB.
IRU	Chameleon	-CHAMELEON-	SYSLIB*FSAH. (for mode FSAH) SYSLIB*ALTFSAH. (for mode ALTFSAH)
LINK	Chameleon	-CHAMELEON-	SYSLIB*LINK.
LSS	Chameleon	-CHAMELEON-	SYSLIB*LSS.
MCB	Protected	-MCB-SS-	MCB*n*MCB$. where *n* is application number 1 – 16.
NTSI	Chameleon	-CHAMELEON-	SYSLIB*NTSI-MSI.
NTSI	Protected	-NTSI-	SYSLIB*NTSI.
OLTP-TM2200	Protected	-OTM-	OTM$*TM$COMMON.
OSI-TP	Protected	-OSI-TP-	SYSLIB*XTP$$GATES.
OSAM	Protected	-OSAM-	SYSLIB*OSAMGATE
PCFP	Chameleon	-CHAMELEON-	SYSLIB*PCFP.
RDMS	Chameleon	-CHAMELEON-	SYSLIB*RDMS$CFGSDEF. or application-qualifier*RDMS$CFGSDEF.
RSA	Chameleon	-CHAMELEON-	SYSLIB*RSA$CFGSDEF. or

Table A–1. Subsystem Information

Product Installation Name	Subsystem Type	Recommended User-Id	Default Subsystem Definition File
			application-qualifier*RSA$CFGSDEF.
SLIB	Chameleon	-CHAMELEON-	SYSLIB*SLIB.
SOLAR	Chameleon	-CHAMELEON-	SYSLIB*SOLAR.
SOLAR/E	Chameleon	-CHAMELEON-	SYSLIB*SOLAR.
TQ-BASELINE	Protected	-TQBASELINE-	SYSLIB*TQ–BSLGATE.
TQ-ONLINE	Protected	-TQBASELINE-	SYSLIB*TQ–BSLGATE.
UDSC	Chameleon Protected	-CHAMELEON- UDS*x*, where *x* is application number	UDS$$SRC*UDSC$CFGSDEF. or application-qualifier*UDSC$CFGSDEF. UDS$$SRC*UDSC$PFGSDEF. or application-qualifier*UDSC$PFGSDEF.
UNIACCESS-ODBC	Protected	-UNIACCESS-	SYSLIB*UAFG. (for mode A and B installations) SYSLIB*UAFGT. (for mode C and D installations)
WMQ2200	Protected	-MQS-	WMQ$*MQS$COMMON.
WMQ2200	Chameleon	-MQS-	WMQ$*MQS$BMCOMMON.
WEBTS	Chameleon	-CHAMELEON-	SYSLIB*WEBTS$SS.

Table A–2 specifies the recommended (without entry-point protection) or required (with entry-point protection) security attributes for each user-id.

As shown, all chameleon subsystems can use the same user-id. That user-id must have a clearance level range of zero to zero and a maximum compartment set of null. In addition, the user-id must not possess any privileges from the trusted privilege set (TPS) described later in this appendix.

Without subsystem entry-point protection,

- You might need to adjust the security attributes for object module subsystems if your site has users running at other mandatory security attributes. See the *ClearPath OS 2200 Apex Help* (8207 4154) for more information about protection of subsystems.

- Each subsystem user-id must have an ACR attached to it that controls who can deactivate the subsystem. Although not absolutely required in that environment, the ACR must also specify who has execute (enter) access to the subsystem. In most cases this is public execute access, which helps your system remain crossboot compatible with prior releases and minimizes future impact if you choose to configure entry-point protection at a later date.

With subsystem entry-point protection,

- The mandatory access control attributes specified in Table A–2 establish all protected subsystems as untrusted subsystems with a clearance level of zero and a compartment set of null. You might need to adjust these attributes if your site has users running at other mandatory security attributes.

- Each subsystem user-id must have an ACR attached to it that controls who can deactivate or execute (enter) the subsystem. In most cases, this ACR is "public execute" and restrict delete access (ability to deactivate the subsystem) to designated system administrators.

Either with or without entry-point protection, the file containing the subsystem definition for each subsystem must be protected by an ACR that prevents unauthorized modification of the subsystem definition. That ACR would normally specify public read access.

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
User Authentication (FLEX)	User-id name	–ASIS–
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public
		Read, write, execute, delete, modify, ACR_Delete: USERID EQ SUBSYS-ADMIN
		Read, write, execute, delete: USERID EQ -ASIS- or USERID EQ ASIS
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Shell (Ex: ring 1)
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 63
	Compartment set	Access to ALL compartments
	Trusted privileges	SSAUTHNTICAT, SSBAFC, SSBYCOMP, SSBYPASSOWNR, SSCCL, SSCONSOLE, SSLOGER, SSREADEXEC,
	Other privileges	SSGAP
	ERs, CALLs	SUBSYS$DEACT, SERVER$CTRL, DUMP$SUBSYS
	User-ID maintenance	Administrator
	File ACR name	PUBRD
	File ACR arguments	Read, execute: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
Business Information Server for ClearPath OS 2200 (BIS)	User-id name	-BISAPI-
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 0
	Compartment set (applies only to Security Level 2 or higher)	Maximum = NULL, Default = NULL (no compartments assigned)
	Trusted privileges	No trusted privileges are required by or must be given to this product.
	Other privileges	SSGAP
	ERs, CALLs	None
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
All products with chameleon subsystems	User-id name	-CHAMELEON-
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 0
	Compartment set (applies only to Security Level 2 or higher)	Maximum = NULL, Default = NULL (no compartments assigned)
	Trusted privileges	No trusted privileges are required by or must be given to these products.
	Other privileges	SSGAP
	ERs, CALLs	None
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
CIFS for ClearPath OS 2200	User-id name	–CIFS–ADMIN– On Fundamental systems, the user-id must be the Security Officer.
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public Delete: USERID EQ SUBSYS-ADMIN
	Run mode	Batch, if this user-id is used to start the CIFS-BACK run.
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 0, or any level where Minimum and Maximum are equal. Multiple level clearance levels are not supported.
	Compartment set (applies only to Security Level 2 or higher)	Maximum = NULL, Default = NULL (compartments are not supported.)
	Trusted privileges	SSAUTHNTICAT, SSBYPASSOWNR
	Other privileges	SSLASH, SSRUNXOPT, SSBPFC, SSGAP, SSMROOC, SSDRG, SSBAFC, SSCONSOLE, SSLOGER
	ERs, CALLS	SERVE$, MODPS$
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
Cipher API	User-id name	–CIPHER–SS– On Fundamental systems, the user-id must be the Security Officer.
	Record access	ACR_Controlled
	Attached ACR name	CIPHER
	ACR arguments	Execute: Public Delete: USERID EQ CIPHERXQT
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 63
	Compartment set (applies only to Security Level 2 or higher)	Access to ALL compartments
	Trusted privileges	SSBYPASSOWNR, SSBYCOMP, SSCCL, SSBAFC
	Other privileges	SSGAP (bank creation General Access Permission) SSCONSOLE (ER KEYIN$ and COM$ privileged functions) SSADID (Perform absolute IO assignment) SSLOGER (Create Exec Log Entries)
	ERs, CALLS	SUBSYS$DEACT, INIT$NETWORK, DUMP$SUBSYS, CONNECT$TIP, PB$CON, MODPS$, SERVE$
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
Communications Application Program Interface (COMAPI)	User-id name	–COMAPI–
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 63
	Compartment set (applies only to Security Level 2 or higher)	Access to ALL compartments, even if no compartments are defined.
	Trusted privileges	SSBAFC, SSBYCOMP, SSBYPASSOWNR, SSCCL
	Other privileges	SSGAP, SSCONSOLE
	ERs, CALLs	None
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
Communications Platform (CPComm)	User-id name	–CPCOMM–SS–
	Record access	ACR_Controlled
	Attached ACR name	CPCOMM
	ACR arguments	Execute: Public
		Delete: USERID EQ CPCOMMXQT or SUBSYS-ADMIN
		(Refer to the *ClearPath Enterprise Servers Communications Platform Configuration and Operations Guide* (7844 8438) for a description of the CPCOMMXQT privileges.)
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 63
	Compartment set (applies only to Security Level 2 or higher)	Access to ALL compartments
	Trusted privileges	SSAUTHNTICAT, SSBAFC, SSBYCOMP, SSBYPASSOWNR, SSCCL
	Other privileges	SSGAP
		SSCONSOLE
		SSADID
		SSLOGER (Create Exec Log Entries)
	ERs, CALLs	SUBSYS$DEACT, INIT$NETWORK, DUMP$SUBSYS
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
Communications Platform for Open Systems (CPCommOS)	User-id name	–CPCOMM–SS–
	Record access	ACR_Controlled
	Attached ACR name	CPCOMM
	ACR arguments	Execute: Public
		Delete: USERID EQ CPCOMMXQT or SUBSYS-ADMIN
		(Refer to the *ClearPath Enterprise Servers Communications Platform for Open Systems Configuration and Operations Guide* (3850 8032) for a description of the CPCOMMXQT privileges.)
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 63
	Compartment set (applies only to Security Level 2 or higher)	Access to ALL compartments
	Trusted privileges	SSAUTHNTICAT, SSBAFC, SSBYCOMP, SSBYPASSOWNR, SSCCL
	Other privileges	SSGAP
		SSCONSOLE
		SSADID
		SSLOGER (Create Exec Log Entries)
	ERs, CALLs	SUBSYS$DEACT, INIT$NETWORK, DUMP$SUBSYS
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
Connectivity Services (CS2200)	User-id name	–CS2200–SS–
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public
		Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 63
	Compartment set (applies only to Security Level 2 or higher)	Access to all compartments, even if no compartments are defined.
	Trusted privileges	SSBAFC, SSBYCOMP, SSBYPASSOWNR, SSCCL, SSAUTHNTICAT
	Other privileges	SSGAP, SSCONSOLE, SSTOKEN
	ERs, CALLs	SUBSYS$DEACT, DUMP$SUBSYS
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
Distributed Data Processing Program-to-Program Communications (DDP-PPC)	User-id name	–DDP–PPC–
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public Delete: USERID EQ SUBSYS-ADMIN
	Run mode	Batch
	Processor privilege	Read Executive GRS
	Access privilege	Shell (i.e., ring 1)
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 0
	Compartment set (applies only to Security Level 2 or higher)	Maximum = NULL, Default = NULL (no compartments assigned)
	Trusted privileges	No trusted privileges are required by or must be given to this product.
	Other privileges	SSGAP, SSCONSOLE, SSRUNXOPT
	ERs, CALLs	CMS$REG, CONNECT$TIP, PB$CON, RT$PID, TIP$SM
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
Display Processing System (DPS)	User-id name	–DPS–
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 0
	Compartment set (applies only to Security Level 2 or higher)	Maximum = NULL, Default = NULL (no compartments assigned)
	Trusted privileges	No trusted privileges are required by or must be given to this product.
	Other privileges	SSGAP
	ERs, CALLs	PB$CON, CONNECT$TIP
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
Multiple Batch Run Optimizer (EXPIPE)	User-id name	–EXPIPE–
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public Delete: USERID EQ SUBSYS–ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 0
	Compartment set (applies only to Security Level 2 or higher)	Maximum = NULL, Default = NULL (no compartments assigned)
	Trusted privileges	No trusted privileges are required by or must be given to this product.
	Other privileges	SSGAP
	ERs, CALLs	None
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
FTP Services for ClearPath OS 2200 (cpFTP)	User-id name	–FTP–SUBSYS–
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 63
	Compartment set (applies only to Security Level 2 or higher)	Maximum = ALL, Default = ALL
	Trusted privileges	SSBYPASSOWNR, SSBYCL, SSBAFC, SSBYCOMP
	Other privileges	SSBPFC, SSBRWK, SSCRCL, SSDRG, SSLASH, SSCSPF, SSDACR, SSFDELE, SSCHDR, SSCONSOLE, SSMROOC, SSANYCALLME, SSGAP, SSSMOQUE
	ERs, CALLs	MODPS$, RSI$
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
HTPIC-2200	User-id name	–HTPIC–2200–
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public
		Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Shell
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 0
	Compartment set	Maximum = NULL, Default = NULL
	Trusted privileges	None
	Other privileges	SSGAP, SSCONSOLE
	ERs, CALLs	None
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
Interconnect	User-id name	–IC2200–
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public
		Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 63 (or both set to the same number that matches the other IC2200 installed files)
	Compartment set	Maximum = NULL, Default = NULL
	Trusted privileges	None
	Other privileges	SSGAP, SSRLODCB, SSWRSUBDAC, SSAUTHNTICAT
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
ODBC Access – Corporate Edition for ClearPath OS 2200, Network Database OLE DB Data Provider	User-id name	–INFOACCESS–
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 0
	Compartment set (applies only to Security Level 2 or higher)	Maximum = NULL, Default = NULL (no compartments assigned)
	Trusted privileges	No trusted privileges are required by or must be given to this product.
	Other privileges	SSGAP, SSWRSUBDAC, SSDRG, SSCONSOLE
	ERs, CALLs	MODPS$, PB$CON, FCREG$, TF$KEY, CONNECT$TIP
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
Message Control Bank (MCB)	User-id name	-MCB-SS-
	Record access	ACR_Controlled
	Attached ACR name	MCB
	ACR arguments	Execute: Public
		Delete: USERID EQ MCBXQT or SUBSYS-ADMIN
		(Refer to the *Message Control Bank (MCB) Configuration and Operations Guide* (7833 1550) for a description of the MCBXQT privileges.)
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 63
	Compartment set (applies only to Security Level 2 or higher)	Access to ALL compartments
	Trusted privileges	SSBAFC, SSBYCOMP, SSBYPASSOWNR, SSCCL
	Other privileges	SSAGNAME, SSCONSOLE, SSGAP
	ERs, CALLs	AUDIT$, CMS$REG, DUMP$SUBSYS, MCABT$, MQF$, QI$CON, TIPQ, TIPXMIT
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
WebSphere MQ Version 9 for ClearPath OS 2200 (WMQ2200)	User-id name	-MQS-
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public Delete: USERID EQ SUBSYS-ADMIN
	Run mode	Batch
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum= 0, Maximum = 0
	Compartment set (applies only to Security Level 2 or higher)	Maximum = NULL, Default = NULL (no compartments assigned)
	Other privileges	SSGAP, SSCONSOLE
	ERs, CALLs	DUMP$SUBSYS
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
Messaging Integration Services (NTSI)	User-id name	– NTSI –
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Trusted
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 0
	Compartment set (applies only to Security Level 2 or higher)	Maximum = NULL, Default = NULL (no compartments assigned)
	Trusted privileges	No trusted privileges are required by or must be given to this product.
	Other privileges	SSGAP, SSAUTHENTICAT
	ERs, CALLs	None
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
TeamQuest OSAM	User-Id name	-OSAM-
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Shell
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 0
	Compartment set (applies only to Security level 2 or higher)	Maximum = NULL, Default = NULL (no compartments assigned)
	Trusted privileges	No trusted privileges are required or should be given to these products.
	Other privileges	SSGAP
	ERs, CALLs	None
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
Open Systems Interconnection Transaction Processing (OSI-TP)	User-id name	–OSI–TP–
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public
		Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Shell
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 0
	Compartment set	Maximum = NULL, Default = NULL
	Trusted privileges	None
	Other privileges	SSGAP, SSCONSOLE
	ERs, CALLs	None
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
Open Distributed Transaction Processing (OLTP-TM2200)	User-id name	–OTM–
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public
		Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Shell
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 0
	Compartment set	Maximum = NULL, Default = NULL
	Trusted privileges	None
	Other privileges	SSGAP,SSCONSOLE
	ERs, CALLs	PB$CON, CONNECT$TIP, DM$IOW, DM$FAC, MQF$, TIP$SM, TIP$TALK, TIP$Q, QI$CON, CMS$REG, RT$PID, SESSION$CTRL
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
TeamQuest Baseline, TeamQuest Online	User-id name	–TQBASELINE–
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Shell
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 0
	Compartment set (applies only to Security Level 2 or higher)	Maximum = NULL, Default = NULL (no compartments assigned)
	Trusted privileges	No trusted privileges are required by or must be given to these products.
	Other privileges	SSGAP
	ERs, CALLs	None
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
Universal Database Control (UDSC)	User-id name	UDS*x*, where *x* is application number
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Shell
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 0
	Compartment set (applies only to Security Level 2 or higher)	Maximum = NULL, Default = NULL (no compartments assigned)
	Trusted privileges	No trusted privileges are required by or must be given to this product.
	Other privileges	SSGAP
	ERs, CALLs	PB$CON, CONNECT$TIP, READY$STEPS, REMOTE$STEP
	File ACR name	PUBRD
	File ACR arguments	Read: Public

Table A–2. Security Attributes for Subsystem User-Ids, ACRs, and Files

Product	Parameter	Value
UniAccess ODBC	User-id name	–UNIACCESS–
	Record access	ACR_Controlled
	Attached ACR name	EXEDEL
	ACR arguments	Execute: Public Delete: USERID EQ SUBSYS-ADMIN
	Run mode	None
	Processor privilege	Read Executive GRS
	Access privilege	Shell
	Sharing level	Application
	Clearance level	Minimum = 0, Maximum = 0
	Compartment set (applies only to Security Level 2 or higher)	Maximum = NULL, Default = NULL (no compartments assigned)
	Trusted privileges	None
	Other privileges	SSGAP, SSCONSOLE
	ERs, CALLs	PB$CON, RT$PID, CMSREG, TIPSM, TIP$TALK
	File ACR name	PUBRD

A.3. Security Requirements for Customer-Created Subsystems

Before installing any subsystems, read this entire appendix to understand the basics of what must be done to set up the security environment for subsystems when installing object module subsystems created at a site. Refer to the *ClearPath OS 2200 Apex Help* (8207 4154) for information concerning protecting subsystems.

A.3.1. Installing Chameleon Subsystems

Chameleon subsystems must have the -CHAMELEON- user-id, or a user-id with similar characteristics, attached as the owner of their subsystem definition file (see Security Requirements with Security Level 1 or 2).

If your site has Security Level 3 or Security Levels 1 or 2 with subsystem entry-point protection configured, the subsystem definition file must also be in a system-low file (see Security Requirements with Security Level 1 or 2).

The subsystem definition file owner user-id must not contain any privileges in the trusted privilege set.

A.3.2. Installing Protected Subsystems

Protected subsystems must have their own unique user-id if any specific attributes are required. (The subsystem designer must be contacted to determine any attributes the subsystem requires.) You can create this user-id using either method described in Procedure for Setting Up the Security Environment.

A.4. Procedure for Setting Up the Security Environment

This subsection tells how to use sample runstreams supplied by Unisys to set up the security environment for all Unisys subsystems. It also describes procedures for manually creating the user-ids and ACRs required for a subsystem.

A.4.1. Using Sample Runstreams

Unisys supplies a set of sample runstreams you can use to set up the security environment for all Unisys subsystems. Since these runstreams are provided as samples, they might not exactly fit your local security environment. You can, however, use any available text editor to tailor the runstreams to fit your local security environment.

The sample runstreams create all of the user-id records and ACRs recommended in this appendix and attach them to the default subsystem definition files as documented in this appendix. If you install a product in files other than its default installation files, you must modify the runstreams or manually attach the proper user-id and ACR to the files.

For the Security Level 1 environment, the runstreams supplied by Unisys are found in the SYS$*RUN$ file on the Exec release (master) tape in the following elements:

```
USERID/INSTALL-S1
ACR/INSTALL-S1
USERID/UPDATE-S1
USERID/ATTACH-S1
```

For the Security Level 2 environment, the runstreams supplied by Unisys are found in the SYS$*RUN$ file on the Exec release (master) tape in the following elements:

```
USERID/INSTALL-S2
ACR/INSTALL-S2
USERID/UPDATE-S2
USERID/ATTACH-S2
```

Unless you modify these runstreams, they create user-id records and ACRs for all products that use subsystems, even if you do not have all of these products installed on your system.

The USERID/INSTALL runstream creates an ACR named EXEDEL to attach to most subsystem user-id records. That ACR specifies a user-id named SUBSYS–ADMIN as the only user having delete access to the user-id, thus giving SUBSYS–ADMIN the capability to deactivate the subsystem. Unless you modify this runstream, a user-id record for SUBSYS–ADMIN is created, but it is given no run modes. This user-id must exist before the EXEDEL ACR, which references it, can be created. You can modify the runstream to remove the creation of this user-id and change the reference to it in the EXEDEL ACR to specify an existing user-id. Otherwise, update the SUBSYS–ADMIN user-id that is created and tailor it to your site requirements.

An ACR named CPCOMM is also created which references SUBSYS-ADMIN. The CPCOMM ACR also gives delete access to a user-ID named CPCOMMXQT, which is assumed to be the user-id used to execute either the CPComm or CPCommOS product.

In the following instructions, it is assumed you are the security officer or a suitably privileged security administrator.

1. Install all desired software products using SOLAR as specified in Sections 3 and 4. At a minimum, you must install Apex or SIMAN. Do not attempt to use any of the products that use subsystems (except SOLAR) until after you complete step 6.

2. Review the sample runstreams for your security environment and modify them as needed to conform to your local security environment.

3. Start the USERID/INSTALL runstream appropriate for your security environment using the security officer's user-id.

4. When the USERID/INSTALL run ends (FIN), start the ACR/INSTALL runstream appropriate for your security environment using the security officer's user-id.

5. When the ACR/INSTALL run ends (FIN), start the USERID/UPDATE runstream appropriate for your security environment using the security officer's user-id.

6. When the USERID/UPDATE run ends (FIN), start the USERID/ATTACH runstream appropriate for your security environment using the security officer's user-id.

7. Save all installed subsystem products using LIBSAVE/FAS.

8. Save a corresponding copy of your security files using the SEC,SAVE keyin as described in the *ClearPath OS 2200 Apex Help* (8207 4154).

A.4.2. Manually Creating the Required Subsystem User-Ids and ACRs

This subsection describes a procedure you can use to manually create the user-ids and ACRs required for a subsystem, attach ACRs to the user-ids, attach the user-ids to the subsystem definition files as their owners, and attach ACRs to the subsystem definition files for file protection. For detailed information about the mechanics of performing each step, see the *ClearPath OS 2200 Apex Help* (8207 4154).

These instructions assume you are the security officer or an appropriately privileged security administrator and that you have already used SOLAR to install the product that uses an object module subsystem.

You can install products before performing this procedure, but you must not (and will not be able to) execute them (except SOLAR itself). Steps 1 through 6 can be performed before a product is installed, but step 7 cannot be done until the SOLAR installation of the product has been completed.

1. Determine the type of subsystem and security attributes required from Table A–2, as appropriate for your security environment.

2. If the subsystem type is chameleon, and you have not previously created the -CHAMELEON- user-id, or if the subsystem type is protected, proceed to step 3. Otherwise, proceed to step 7.

3. Log on to the system in demand mode using the security officer's user-id or another appropriately privileged security administrator user-id.

4. Create an ACR to control who can access or deactivate the subsystem. This ACR is owned by your user-id. Use the name recommended in Table A–2, or one of your choosing. Give that ACR the following attributes:

   a. Specify execute access with the restrictions recommended in Table A–2. In most cases, this is PUBLIC. You could also restrict access to protected subsystems by specifying a list of user-ids, projects, accounts, or times of day.

   b. Specify delete access and specify who is allowed to deactivate the subsystem. You must at least include any user-ids recommended in Table A–2. In most cases, this is the user-id of a subsystem administrator. You do not need to specify your own user-id. As the owner, you are always granted all discretionary accesses. You might want to include user-ids for other system administrators or operators who might need to deactivate the subsystem.

5. Create the user-id security record for the subsystem using the name and security attributes recommended in Table A–2 as follows:

   a. Do not specify any run modes unless a run mode is indicated in Table A–2.

   b. Specify the minimum and maximum clearance level for the subsystem as listed in Table A–2.

   c. Specify that access to the user-id record is controlled by the ACR you created in step 4. Also ensure that the user-id has at least one ACR remaining to be created so that one can be created under this user-id in step 6.

   d. If Table A–2 indicates that the user-id must have access to all compartments, select that option. If Table A–2 says the user-id maximum or default compartment set must be other than NULL, specify the required compartments for the user-id. The compartment set for almost all subsystems is NULL.

   e. Select the following subsystem parameters for the user-id:

      I.   Select Can Only Read Executive GRS for all subsystems.

      II. Specify a subsystem sharing level of Application.

      III. Select Trusted or Shell as the access privilege. Select Shell only if it is shown as required in Table A–2. Never specify Kernel. Kernel is not a legal access privilege for a subsystem; it causes the subsystem load to be terminated.

      IV. Select privileged interfaces (Executive requests or CALL interfaces) as specified in Table A–2. If your site controls any interfaces that are not normally controlled on the released system, you might need to add them for subsystems that use the interfaces.

      V.   Select privileges for the subsystem. The SSGAP (general access permission) privilege is required for all subsystems. Select other privileges (trusted or other) as specified in Table A–2.

6. Create an ACR to control access to subsystem files as follows:

   a. This ACR must be owned by the user-id created in step 5.

   b. Name the ACR as recommended in Table A–2.

   c. Specify the type of access to be controlled as "read," then specify access restrictions as recommended in Table A–2. In most cases, this is PUBLIC.

7. In this step, you change the owner of the product subsystem definition file to the user-id created in step 5 (or one that might have previously been created if this is a chameleon subsystem) and attach the ACR created in step 6 to this file.

    Take the following steps:

    a. Using Cataloged File Security Maintenance, enter the directory, qualifier, and file name of the subsystem definition file for the subsystem. Always use a directory of STD. See Table A–1 for the default qualifier and file name for each Unisys subsystem.

    b. Change the file owner to the user-id created in step 5 (or the previously created chameleon user-id if Table A–1 lists the subsystem as a chameleon subsystem).

    c. Specify that the file is semi-private and identify the ACR created to protect subsystem files (step 6) as the controlling ACR.

    d. Specify the subsystem's user-id as the file owner (File Owned by) for the ACR.

    e. Specify a clearance level of zero for the -CHAMELEON- user-id, and in most cases zero for protected subsystems. Specify a nonzero clearance level only if required in Table A–1.

    f. Do not specify any compartments for the -CHAMELEON- user-id. Specify compartments for protected subsystems only if required in Table A–1. The All Compartments flag must not be set for the -CHAMELEON- user-id and must not be set in most cases for protected subsystems.

Appendix B
Guidelines for Reloading Subsystems Provided with Unisys Software

Reloading a subsystem is similar to reloading a common bank. But, before reloading a subsystem, you must first deactivate it. This allows a new copy of the subsystem to be loaded the next time it is called.

SOLAR performs this process whenever a product containing a subsystem is installed. In some situations, however, it might be necessary to manually reload a subsystem. In those cases, it is important to follow the procedures specified in the following tables.

A common situation in which active subsystems need to be reloaded is when a different level of the Linking System is installed. In this case, any subsystem that requires ongoing Linking System support might not be functional under certain conditions until it is reloaded with the different level of the Linking System. Subsystems can be deactivated manually as described in the following tables or by performing an autorecovery through a $! keyin after installing the new level of Linking System. Using autorecovery might be easier when a large number of subsystems must be reloaded. Table B–1 contains the process for reloading this set of subsystems.

You can use the Linking System SSINFO processor to determine which subsystems are currently active (@SSINFO,STP). You can also use SSINFO to print a list of DEACT command lines for all active subsystems (@SSINFO,D).

If you have an alternate Linking System installed on your system (mode ALT), some subsystems might have been loaded with the main Linking System (mode A) and some with the alternate Linking System. If you install a new main linker, only those subsystems that were previously loaded with a main linker need to be reloaded. Similarly, if you install a new alternate linker, only those subsystems that were previously loaded with the alternate linker need to be reloaded. You can use the SSINFO processor to list the subsystems loaded by the main mode A linker (@SSINFO MAIN/LINKER), the alternate mode ALT linker (@SSINFO ALT/LINKER), or both (@SSINFO ALL/LINKERS).

For more information about the SSINFO processor, specify the H option to get a help screen, or refer to the *ClearPath OS 2200 Linking System Subsystems Programming Guide* (7830 7451).

Notes:

- *There might be certain incompatibilities between subsystems that are already active and new linkers. For detailed information, refer to the the* ClearPath OS 2200 Linking System Subsystems Programming Guide (7830 7451). *The document also includes guidelines for designing subsystems that minimize the impact of installing a new version of the Linking System.*

- *For information on the security access requirements necessary to deactivate a subsystem, see Appendix A.*

Table B–1. Subsystems to Be Reloaded with Linking System Installation

Product	Reload Process
Extended Language Message System (ELMS)	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*ELMS.SSDEFABS 2. The subsystem is reloaded by the next call made to ELMS.
Language Support System (LSS) (Mode S)	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*LSS.UCS$SSDEF 2. The subsystem is reloaded by the next usage of C Compiler, COBOL Compiler, or FORTRAN Compiler. A mode A install of LSS (the default) does not need to be deactivated when a new Linking System is installed.
Messaging Integration Services (NTSI)	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB* NTSI.SSDEF$NTSI 2. The subsystem is reloaded when you run the NTSI configuration.
Open Distributed Transaction Processing (OLTP-TM2200)	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT OTM$*TM$COMMON.SSDEF$ 2. Reinstall the *TMSCONFIG-file*. @OTMTMRUN.TMSCON,L *TMSCONFIG-file* 3. Restart background runs. Refer to the *Distributed Processing Middleware Open Distributed Transaction Processing Administration Guide Volume 1: Installation, Configuration, and Ongoing Administration* (7833 5072) for more information.

Table B–1. Subsystems to Be Reloaded with Linking System Installation

Product	Reload Process
Open Systems Interconnection Transaction Processing (OSI-TP)	1. Terminate the background run with a console keyin. XTP TERM 2. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*XTP$$GATES. XTP-SSD 3. Restart the background run, which reloads the subsystem. @START SYSLIB*RUN$.XTP$RUN Refer to the *OSI-TP High Performance XATMI (HTP/x) Implementation and Administration Guide* for more information.
OS 2200 High Performance Transaction Processing Interconnect (HTPIC-2200)	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*XTP$$GATES.SSD 2. Restart the background run, which reloads the subsystem. @START SYSLIB*RUN$.XTP$RUN Refer to the *OSI-TP High-Performance Transaction Processing for XATMI (HTP/x) Implementation and Administration Guide* (2750 2558) for more information.
WebSphere MQ Version 9 for ClearPath OS 2200 (WMQ2200)	1. Deactivate the MQS subsystem containing the WebSphere MQ engine. @ADD WMQ$*MQSTOOLS.DEACT/MQS 2. Deactivate the UNX subsystem @ADD WMQ$*MQSTOOLS.DEACT/UNX 3. To reload the MQS subsystems you can restart the OS 2200 daemon using the following command: @wmq$*mqs$exe.startup The UNX subsystem is not reloaded unless it is called.

Table B–2 contains the remaining subsystems provided by Unisys software. You do not need to reload this set of subsystems when the Linking System level is changed. But follow these processes if you need to reload any of these subsystems for some other reason.

Table B–2. Other Unisys Subsystems

Product	Reload Process
CIFS	To deactivate the subsystem, use the keyin: CIFS DEACT or start the deactivation run: ST SYSLIB*RUN$.CIFS-DEACT,,,*<accnt/userid>* ***Note:*** *The user-id must have read and write access to the CIFS subsystem file, SYSLIB*CIFS$SS.* After the background run (CIFSB) run has terminated, the CIFS-BACK runstream must be manually restarted. Refer to the *ClearPath Enterprise Servers CIFS for ClearPath OS 2200 User, Programmer, and Administrator Reference Manual* (7859 6137) for more information.
Communications Application Program Interface (COMAPI)	1. Terminate the background run with the console keyin. *keyin-id* TERM 2. Deactivate the COMAPI subsystem. @ADD SYSLIB*COMAPI.DEACT 3. Restart the background run. @START SYSLIB*COMAPI.STCOMAPI Refer to the *ClearPath Enterprise Servers Communications Application Program Interface (COMAPI) User's Guide* (7851 5749) for more information.
Communications Platform (CPComm)	Restart Communications Platform. The subsystem is deactivated and reloaded. For example, the mode A runstream is in SYSLIB*RUN$.STCPCOMMA.
Communications Platform for Open Systems (CPCommOS)	Restart Communications Platform for Open Systems. The subsystem is deactivated and reloaded. For example, the mode A runstream is in SYSLIB*RUN$.STCPCOMMA.
Connectivity Services (CS2200)	1. Terminate the background run with the console keyin. *keyin-id* TERM 2. Deactivate the CS2200 subsystem. @ADD SYSLIB*CS2200.DEACT 3. Restart the background run. @START SYSLIB*RUN$.STCS2200 Refer to the *ClearPath Dorado Servers Connectivity Services User's Guide* (8231 0673) for more information.
Display Processing System (DPS)	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*DPS.SSDEF 2. The subsystem is reloaded on the next reference to it.

Table B–2. Other Unisys Subsystems

Product	Reload Process
Distributed Data Processing Program-to-Program Communications (DDP-PPC)	1. Terminate all users of DDP-PPC. 2. Terminate the background run. 3. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*PPC$$GATES.SSD 4. Restart the background run that reloads the subsystem. 5. Restart the users.
Enterprise Application Runtime Test Environment for ClearPath OS 2200	Restart Enterprise Application Runtime Test Environment for ClearPath OS 2200. Refer to the *Unisys e-@ction Enterprise Application Runtime Test Environment Guide* (7862 1703) for more information.
FTP Services for ClearPath OS 2200 (cpFTP)	1. Terminate the background run with a console keyin. CPFTP TERM 2. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*CPFTP$LIB.SSDEF 3. Restart the background run, which reloads the subsystem. @START SYSLIB*RUN$.CPFTP Refer to the *ClearPath Enterprise Servers FTP Services for ClearPath OS 2200 User's Guide* (7847 5753) for more information.
I18N Service Library (I18NLIB)	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*I18NLIB.SSDEF$ 2. The subsystem is reloaded the next time it is referenced.
Integrated Recovery Utility (IRU) FSAH component	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*FSAH.FSAH-SSDEF or @SYSLIB*SOLAR.DEACT SYSLIB*ALTFSAH. FSAH-SSDEF 2. The FSAH subsystems are reloaded the next time they are referenced.
Language Support System (LSS) Mode A	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*LSS.FLSS$SSDEF 2. The subsystem is reloaded by the next usage of C Compiler, COBOL Compiler, or FORTRAN Compiler.
Linking System (LINK)	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*LINK.LINKING-SYS The subsystem is reloaded on the next reference to it.

Table B–2. Other Unisys Subsystems

Product	Reload Process
Message Control Bank (MCB)	1. Terminate the background run with the console keyin. *keyin-id* TERM 2. Deactivate the MCB subsystem. @ADD MCB*n**MCB$.DEACT where *n* is the application group number for the MCB instance. 3. Restart the background run. @START SYSLIB*RUN$.STMCB*n* **Note:** *Restarting MCB requires its intended Application Group to be UP and also requires execution of the Integrated Recovery Utility (IRU).* Refer to the *Message Control Bank (MCB) Configuration and Operations Guide* (7833 1550) for more information.
Multiple Batch Run Optimizer (EXPIPE)	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*EXPIPE.SS-DEF 2. Delete the pipe definition file if the previous pipe definitions are unnecessary. @DELETE SYS$*PIPE$DEF//EXWKEY. 3. The subsystem is reloaded and the pipe definition file is recataloged on the next reference to it.
ODBC Access–Corporate Edition	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*IASOCK (F-cycle).SSDEF$ 2. The subsystem is reloaded the next time it is referenced.
Program-Callable FURPUR (PCFP)	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*PCFP.FP-SSDEF 2. The subsystem is reloaded on the next reference to it.
Service Library (SLIB)	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*SLIB.SS-DEF 2. The subsystem is reloaded the next time it is referenced.
SOLAR and SOLAR/E	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*SOLAR.SSDEF$ 2. The subsystem is reloaded the next time it is referenced.

Table B–2. Other Unisys Subsystems

Product	Reload Process
TeamQuest Baseline	1. Deactivate the following subsystems: • @SYSLIB*SOLAR.DEACT • SYSLIB*TQ-BSLGATE.TQ-SSD • @SYSLIB*SOLAR.DEACT • SYSLIB*TQ-BSLGATE.TQ-TMSSD 2. The subsystems are reloaded the next time they are referenced.
TeamQuest Online	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*TQ-BSLGATE.TQ-SSD 2. The subsystem is reloaded on the next reference to it.
TeamQuest OSAM	1. Terminate any executing OSAM process using its documented method. 2. Deactivate the following subsystems: @SYSLIB*SOLAR.DEACT SYSLIB*OSAMGATE.OSAM-LCSSD @SYSLIB*SOLAR.DEACT SYSLIB*OSAMGATE.OSAM-MSSSD 3. Each subsystem is reloaded on the next reference.
UniAccess for OS 2200 Systems	1. Terminate the UACS background run. 2. Deactivate the subsystem. @ADD SYSLIB*UAUTIL.DEACTUAFG . (for modes A and B) @ADD SYSLIB*UAUTILT.DEACTUAFG. (for modes C and D) 3. Reload the subsystem with the UACF utility. The actual command used depends on the location of the configuration file, and the installation mode. For additional information, refer to the *UniAccess for OS 2200 System Administration Guide*. The following example assumes that all of the defaults have been taken. @SYSLIB*UAUTIL.UACF,I

Table B–2. Other Unisys Subsystems

Product	Reload Process
Universal Database Control (UDSC)	1. Disable application group(s) by using the Universal Database Control SUDS utility II command. This deactivates the Universal Database Control protected subsystem. If the application group is inactive (no threads active or registered in Universal Database Control), the following can be used; however, this is not recommended standard operating procedure. @SYSLIB*SOLAR.DEACT application-qualifier*UDSC$PFGSDEF.PFGS-SSDEF 2. Deactivate the Universal Database Control chameleon subsystem. @SYSLIB*SOLAR.DEACT application-qualifier*UDSC$CFGSDEF.CFGS-SSDEF 3. Deactivate the Relational Database Server chameleon subsystem. @SYSLIB*SOLAR.DEACT application-qualifier*RDMS$CFGSDEF.RDMS-SSDEF 4. Deactivate the Relational Syntax Analyzer chameleon subsystem. @SYSLIB*SOLAR.DEACT application-qualifier*RSA$CFGSDEF.RSAC-SSDEF 5. The subsystems are reloaded on the next reference to them. 6. Use Integrated Recovery Utility to SHORTRECOVER the application. Refer to the *ClearPath Enterprise Servers Universal Data System Administration and Support Reference* (7831 0737) for more information.
User Authentication (FLEX)	1. Terminate ASIS using either of the following: • console keyin ASIS TERM • @ASISUT processor command SHUTDOWN PURGE QUIET 2. Restart ASIS with the relink option START ASIS,1 Refer to the *ClearPath Enterprise Servers User Authentication Administration Guide* (7850 4586) for more information.
Web Transaction Server	1. Deactivate the subsystem. @SYSLIB*SOLAR.DEACT SYSLIB*WEBTS$SS.SSDEF$ 2. The subsystem is reloaded on the next reference to it.

Appendix C
Exceptions to SOLAR-Keyed Installation

Most OS 2200 products are released on conditioned software package tapes and require a key tape to copy and install them using SOLAR. This appendix lists products that are exceptions to that method. They include:

- SOLAR-installable products that are delivered without SOLAR-keys
- Products that can be installed with COMUS or SOLAR
- Products that are released as a CD-ROM image

C.1. SOLAR-Installable Products Delivered without SOLAR-Keys

The following are the SOLAR-installable products that are delivered without SOLAR installation keys:

- CP-FLD tape
- CP-OE2
- Exec boot tape
- Exec release master/Symbolics
- Enterprise Output Manager (EOM) for ClearPath OS 2200
- ODBC Data Access

    Note: *ODBC Data Access requires a Product Capability Key, which varies depending on the system type and number of licensed users of ODBC Data Access.*

- Software Controlled Performance (SCP) key tape or file
- SOLAR

C.2. Products Installed with Either COMUS or SOLAR

ASCII COBOL Compiler can be installed with either COMUS or SOLAR.

C.3. Products Released as a CD-ROM Image

Table C–1 lists the products that are released as CD-ROM image.

Table C–1. Products Released as CD-ROM Image

Primus Name	Install Name
Apex	Apex
DMS-RA	DMS-RA
DTI Integrator	Distributed Transaction
INFOACCESS	ClearPath OS 2200 Data Access
J2EE-CON-OPENDTP	Open Distributed Transaction Processing Resource Adapter for the Java™ Platform
J2EE-CON-OS2200	OS 2200 Transaction Resource Adapter for the Java™ Platform
MSMQI	ClearPath OS 2200 Interface for Microsoft Message Queuing
NTSI	Messaging Integration Services
OLTP-TM2200	OPENDTP Java and Twrap
OUTPUT-MGR	Enterprise Output Manager
SP-Operation	Operations Sentinel
TQ-BASELINE	TeamQuest View®
TQ-MODEL	TeamQuest Model®
TQ-ONLINE	TeamQuest View®
UNIACCESS-ODBC	UNIACCESS ODBC
URU-OS2200	Utilization Report Utility for OS 2200
WebTS	Unisys Web Transaction Server

Appendix D
Product Build Information

Table D–1 provides the build information for each product.

Table D–1. Product Build Information

Product	Build Information	Build Documentation	Document Number
ACOB	Buildable	Refer to Section 5.4.1, "ACOB"	
APEX	Not Buildable		
CIFS	Not Buildable		
CIPHER-API	Not Buildable		
CITA	Buildable	*ClearPath Enterprise Servers Communications Interface for Transaction Applications Configuration and Operations Guide*	7862 6470
CKRS	Buildable	Refer to Section 5.4.2, "CKRS"	
CML	Buildable	Refer to Section 5.4.3, "CML"	
CMR	Buildable	Refer to Section 5.4.4, "CMR"	
COMAPI	Buildable	*ClearPath Enterprise Servers Communications Application Program Interface (COMAPI) User's Guide*	7851 5749
COMUS	Buildable	Refer to Section 5.4.5, "COMUS". *ClearPath OS 2200 COMUS End Use Reference Manual*	7830 7758
CPCOMM	Buildable	*ClearPath Enterprise Servers Communications Platform Configuration and Operations Guide*	7844 8438

Table D–1. Product Build Information

Product	Build Information	Build Documentation	Document Number
CPCOMMOS	Buildable	*ClearPath Enterprise Servers Communications Platform for Open Systems Configuration and Operations Guide*	3850 8032
CPFTP	Buildable	Refer to Section 5.4.6, "CPFTP"	
CS2200	Not Buildable		
CRYPTOLIB	Not Buildable		
CULL	Buildable	Refer to Section 5.4.7, "CULL"	
DAP	Not Buildable		
DBAGENT	Not Buildable		
DDP-FJT	Buildable	Refer to Section 5.4.8, "DDP-FJT"	
DDP-PPC	Buildable	Refer to Section 5.4.9, "DDP-PPC"	
DEPCON	Not Buildable		
DFP	Buildable	Refer to Section 5.4.10, "DFP"	
DMSKEY	Not Buildable		
DMSSUB	Buildable	*ClearPath Enterprise Servers Universal Data System Configuration Guide*	7844 8362
DMS-RA	Not Buildable		
DPREP1100	Not Buildable		
DPS	Buildable	*Display Processing System (DPS 2200) Administration Guide*	7831 2295
DTI	Not Buildable		
ECLIPSE-2200	Not Buildable		
ELMS	Buildable	Refer to Section 5.4.11, "ELMS"	
ELT	Buildable	Refer to Section 5.4.12, "ELT"	
EPORTAL-2200	Not Buildable		

Table D–1. Product Build Information

Product	Build Information	Build Documentation	Document Number
EXEC **Note:** *The following Exec SPEFS are installable through an Exec System Generation:* MHFS SECOPT1 SECOPT2 SECOPT3 SIPIPM TAVR UDUPLEX VTH XPCEXEC XTCEXEC	Buildable	*ClearPath OS 2200 Exec System Software Installation and Configuration Guide*	7830 7915
ARC	Buildable		
DMPLIB	Buildable	Comus Buildable- use COMUS build defaults.	
MMGR	Not Buildable		
MSCP	Not Buildable		
TIPUTIL	Buildable	*ClearPath OS 2200 Exec System Software Installation and Configuration Guide*	7830 7915
EXPIPE	Not Buildable		
EXT-KIT-2200	Not Buildable		
FAS	Buildable	Refer to Section 5.4.13, "FAS"	
FDP-ADP	Buildable	*Query Language Processor (QLP 2200) Configuration Guide*	7833 4851
FLEX	Not Buildable		
FLIT	Buildable	Refer to Section 5.4.14, "FLIT"	
FTN	Buildable	Refer to Section 5.4.15, "FTN (ASCII FORTRAN)"	

Table D–1. Product Build Information

Product	Build Information	Build Documentation	Document Number
FURPUR	Buildable	Refer to Section 5.4.16, "FURPUR." Build requires that the same CP level for the products PCFP, SOLAR, and SYSLIB be installed or available as input.	
GSA	Buildable	Refer to Section 5.4.17, "GSA"	
HTPIC-2200	Not Buildable		
I18NLIB	Not Buildable		
IACULL	Buildable	Refer to Section 5.4.18, "IACULL"	
INFOACCESS	Not Buildable		
INTERCONNECT	Not Buildable		
IPF	Buildable	Refer to Section 5.4.19, "IPF"	
IRU	Buildable	*ClearPath Enterprise Servers Integrated Recovery Utility Administration Guide*	7833 1576
J2EE-CON-OPENDTP	Not Buildable		
J2EE-CON-OS2200	Not Buildable		
LA	Buildable	Refer to Section 5.4.20, "LA"	
LINK	Buildable	Refer to Section 5.4.21, "LINK"	
LIST	Buildable	Refer to Section 5.4.22, "LIST"	
LSS	Buildable	Refer to Section 5.4.23, "LSS"	
MAP	Buildable	Refer to Section 5.4.24, "MAP"	
MASM	Buildable	Refer to Section 5.4.25, "MASM"	
MCB	Buildable	*Message Control Bank (MCB) Configuration and Operations Guide.*	7833 1550

Table D–1. Product Build Information

Product	Build Information	Build Documentation	Document Number
MSAR	Buildable	Refer to Section 5.4.26, "MSAR"	
MSMANAGER	Not Buildable		
MSMQI	Not Buildable		
NTSI	Not Buildable		
OLTP-TM2200	Buildable	*Distributed Processing Middleware Open Distributed Transaction Processing Administration Guide, Volume 1: Installation, Configuration, and Ongoing Administration*	7833 5072
OSAM	Buildable	Refer to Section 5.4.27, "OSAM"	
OSI-TP	Buildable	Refer to Section 5.4.28, "OSI-TP"	
PADS	Buildable	Refer to Section 5.4.29, "PADS"	
PAR	Buildable	Refer to Section 5.4.30, "PAR"	
PCFP	Not Buildable		
PCIOS	Buildable	Refer to Section 5.4.31, "PCIOS"	
PDP	Buildable	Refer to Section 5.4.32, "PDP"	
PLUS	Not Buildable		
PMD	Buildable	Refer to Section 5.4.33, "PMD"	
QLP	Buildable	*Query Language Processor (QLP 2200) Configuration Guide*	7833 4851
RDMSKEY	Not Buildable		
RDMSSUB	Buildable	*ClearPath Enterprise Servers Universal Data System Configuration Guide*	7844 8362
RDMS-JDBC	Not Buildable		
ROLRUNS	Not Buildable		

Table D–1. Product Build Information

Product	Build Information	Build Documentation	Document Number
RSS	Buildable	Refer to Section 5.4.34, "RSS."	
SAUTILITIES	Not Buildable		
SFS	Buildable	*ClearPath Enterprise Servers Universal Data System Configuration Guide*	7844 8362
SILAS	Buildable	*ClearPath Enterprise Servers System Interface for Legacy Application Systems (SILAS) Configuration and Operations Guide*	7851 5475
SIMAN	Buildable	Refer to Section 5.4.35, "SIMAN"	
SLIB	Not Buildable		
SOLAR	Not Buildable		
SOLAR/E	Not Buildable		
SORT	Buildable	Refer to Section 5.4.36, "SORT"	
SP-OPERATION/ SP-OPERATION CP	Not Buildable		
SSG	Buildable	Refer to Section 5.4.37, "SSG"	
SYSLIB	Buildable	Refer to Section 5.4.38, "SYSLIB"	
TAS	Buildable	Refer to Section 5.4.39, "TAS (TCP/IP APPLICATION SERVICES)"	
TQ-BASELINE	Not Buildable		
TQ-D-FRAGGER	Not Buildable		
TQ-MODEL	Not Buildable		
TQ-ONLINE	Not Buildable		
TQ-PMLOG	Not Buildable		
TQ-PROBES	Not Buildable		
TQ-REMD-FRAGGER	Not Buildable		
TQ-TIP-LA	Not Buildable		

Table D–1. Product Build Information

Product	Build Information	Build Documentation	Document Number
TUTIL	Not Buildable		
UC	Buildable	Refer to Section 5.4.40, "UC (USC C)"	
UCOB	Buildable	Refer to Section 5.4.41, "UCOB (UCS COBOL)"	
UCSRTS	Buildable	Refer to Section 5.4.42, "UCSRTS"	
UDSC	Buildable	*ClearPath Enterprise Servers Universal Data System Configuration Guide*	7844 8362
UFTN	Buildable	Refer to Section 5.4.43, "UFTN"	
UNIACCESS-ODBC	Not Buildable		
UPLS	Not Buildable		
UREPSUB	Buildable	*ClearPath Enterprise Servers Universal Data System Configuration Guide*	7844 8362
URTS	Buildable	Refer to Section 5.4.44, "URTS"	
URU-OS2200	Not Buildable		
WEBTS	Not Buildable		
WMQ2200	Buildable	*WebSphere ® MQ Version 9 for ClearPath OS 2200 Installation, Administration, and Programming Guide*	3843 3744
XRLOAD	Not Buildable		

Index

T

tape file description (TFD) element, 1-3
Tape Labeling Utility (TUTIL) product-specific
    information, 4-58
TeamQuest Baseline
    reload process, B-7
    security attributes, A-31
    user-id and default file name, A-7
TeamQuest Online
    reload process, B-7
    security attributes, A-31
    user-id and default file name, A-7
TIPUTIL product-specific information, 4-60
Transaction Processing utility (TIPUTIL)
    product-specific information, 4-60
TSAM interface abort, 4-54
TUTIL product-specific information, 4-58

U

UDSC product-specific information, 4-61
UniAccess for OS 2200 Product Suite
    product-specific information, 4-60
UniAccess ODBC Server
    reload process, B-7
    security attributes, A-33
    user-id and default file name, A-7

Universal Database Control
    product-specific information, 4-61
    reload process, B-8
    security attributes, A-32
    user-id and default file name, A-7
UPLS product-specific information, 4-62
UREP product-specific information, 4-52
User Authentication
    product-specific information, 4-63
    reload process, B-8
    security attributes, A-9
    user-id and default file name, A-6

W

Web Transaction Server
    product-specific information, 4-65
    reload process, B-8
    user-id and default file name, A-7
WebSphere MQ Version 9 for ClearPath OS
    2200
    security attributes, A-26