



The Mainframe Software Partner
For The Next 50 Years

Enterprise Common Components Installation and Customization Guide MVS Version

Release 3.3

Global technical support contact information is available from our Customer Support Web Site:

<http://go.compuware.com>

This document and the product referenced in it are subject to the following legends:

| Copyright 2015 Compuware Corporation. All rights reserved. Unpublished rights reserved under the Copyright Laws of the United States.

U.S. GOVERNMENT RIGHTS-Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in Compuware Corporation license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable. Compuware Corporation.

This product contains confidential information and trade secrets of Compuware Corporation. Use, disclosure, or reproduction is prohibited without the prior express written permission of Compuware Corporation. Access is limited to authorized users. Use of this product is subject to the terms and conditions of the user's License Agreement with Compuware Corporation.

Abend-AID, Abend-AID Fault Manager, Abend-AID for WebSphere MQ, Abend-AID for CICS, CICS Abend-AID/FX, XPEDITER, XPEDITER/DevEnterprise, File-AID for DB2, STROBE, UNIFACE, Compuware Shared Services, CSS, Compuware Viewing Facility, Compuware/VF, Enterprise Common Components, and ECC are trademarks or registered trademarks of Compuware Corporation.

IBM, CICS, DB2, and MVS/ESA are trademarks or registered trademarks of International Business Machines Corp.

Adobe® Reader® is a trademark of Adobe Systems Incorporated in the United States and/or other countries.

All other company and product names are trademarks or registered trademarks of their respective owners.

Contents

Figures	vii
Tables	ix
Introduction	xi
Manual Structure	xi
Intended Audience	xi
Related Publications	xii
FrontLine Support Web Site	xii
Online Documentation	xii
World Wide Web	xii
Terminology	xii
Getting Help	xiii
Chapter 1. ECC Overview	1-1
What is ECC?	1-1
CSS Overview	1-2
System Environment	1-2
CSS Components	1-2
Common Files and Utilities	1-2
DDIO Files	1-2
CSS Utilities and Batch File Utilities	1-3
CSS Language Processors (LP)	1-3
Language Processor Types	1-3
Security Exit Program	1-4
Executing CSS	1-4
LMS Overview	1-5
LMS Advantages	1-5
Operating Environment	1-5
The License Management Process	1-5
Information Gathering	1-5
License Certificate	1-5
Installation	1-6
HCI Overview	1-6
Chapter 2. Preparing for ECC Installation	2-1
Required Components and Procedures	2-1
Component Selection	2-1
Required Procedures	2-2
ECC Packaging	2-2
ECC Component Prefixes and FMIDs	2-2
ECC Installation Considerations	2-3
Link Lists	2-3
PDS/E Support	2-3
CSS Installation Considerations	2-3
APF Authorization	2-3
CICS Load Library Definition	2-3
TSO Logon PROCs	2-4
Sharing CSS DDIO Files With Multiple CPUs	2-4
CSS Language Processor (LP)	2-4
Pre-processor	2-4

Post-processor	2-6
Language Processor DD Statements	2-6
Writing a Listing to a DDIO File	2-7
Testing and Debugging Programs	2-7
Debugging Production Programs	2-7
CSS Sample Library Members	2-7
LMS Installation Considerations	2-12
APF Authorization	2-12
Execution Sequence	2-12
The License Management System Functions	2-12
License Management Administration	2-13
Establishing the Runtime LMS Environment	2-13
Validating Product Access Requests During Product Use	2-13
HCI Installation Considerations	2-14
APF Authorization	2-14
Security Settings	2-14
Chapter 3. Receive From Network (RFN) Installation	3-1
Installation Methodology	3-1
Chapter 4. Applying Maintenance	4-1
Chapter 5. CSS Customization	5-1
Step 1. Implement the Security Exit Program (Optional)	5-2
Step 2. Install Customized Translation Tables (Optional)	5-3
Step 3. Make New CSS Load Modules Accessible to Compuware Products (Required)	5-4
Step 4. Create the Compile Information Table (Required)	5-4
Step 5. Activate Compuware ISPF Dialogs (Required)	5-15
Option 1 — Dynamically Allocate Libraries when CSS Utilities are Executed 5-15	
Option 2 — Allocate the Libraries at TSO Startup	5-16
Step 6. Link CSS Utilities Tutorials to Your Main Tutorial Panel (Optional)	5-18
Step 7. Prepare the DDIO Files	5-19
Types of DDIO Files	5-19
File Access Methods	5-20
Step 7a. Allocate and Format Abend-AID Report Shared Directories and Databases for Abend-AID Release 9.4 and Later Users	5-21
Create the Report Shared Directory	5-21
Create the Report Databases	5-21
Step 7b. Allocate and Format Source Listing Shared Directories and Databases	5-21
Create the Source Listing Shared Directory	5-21
Create the Source Listing Databases	5-22
Step 7c. Allocate and Format Source Listing Files	5-22
Allocate the Source Listing Files	5-22
Format the Source Listing Files	5-22
Step 7d. Allocate and Format Abend-AID for CICS Shared Directories and Transaction Databases	5-22
Create Abend-AID for CICS Shared Directory (Allocate & Format)	5-22
Create Abend-AID for CICS Transaction Database (Allocate & Format). 5-22	
Step 8. Implement the Language Processor JCL (Required)	5-24
Step 8a. Modify the JCL to Run the COBOL Language Processor	5-24
Preprocessor	5-24
Post-processor	5-25
Step 8b. Modify the JCL to Run the PL/I Language Processor	5-26

Preprocessor	5-26
Post-processor	5-27
Step 8c. Modify the JCL to Run the Assembler Language Processor	5-28
Preprocessor	5-28
Post-processor	5-29
Step 8d. Modify the JCL to Run the C Language Processor	5-29
Preprocessor	5-29
Post-processor	5-30
Step 9. Initialize the zIIP Enablement Service (optional)	5-31
Overview	5-31
Enabling the Service	5-31
Disabling the Service	5-31
Job disablement	5-31
System-wide disablement	5-32
Step 10. Continue with ECC Customization	5-32
Chapter 6. LMS Customization	6-1
Preliminary Considerations	6-1
Installing and Enabling LMS	6-4
Step 1. Transfer License Certificate to Host	6-4
Step 2. Set up License Administration Utility (LAU)	6-5
Step 3. Create License File	6-7
Step 4. Verify License File	6-9
Step 5. Define Nodes.	6-10
Step 6. Import License Certificate	6-12
Step 7. Prepare an APF Authorized LMSINIT	6-14
Step 8. Run LMSINIT	6-15
Step 9. Verify Proper LMS Function	6-16
Step 10. Continue ECC Customization	6-17
Resolving Problems.	6-17
Appendix A. Compuware Maintenance Utility	7-1
JCL Required to Execute the CMU	7-1
Required Datasets	7-1
Control File Format	7-2
Report Output	7-4
Glossary	G-1
Index	I-1

Figures

1-1.	TP Subsystem, HCI and Application	1-6
5-1.	Compile Information Table - Install Menu	5-5
5-2.	Compile Information Table - Select Compilers (1 of 2)	5-6
5-3.	Compile Information Table - Select Compilers (2 of 2)	5-6
5-4.	Compile Information Table - Enter Compiler Libraries (1 of 4)	5-7
5-5.	Compile Information Table - Enter Compiler Libraries (2 of 4)	5-7
5-6.	Compile Information Table - Enter Compiler Libraries (3 of 4)	5-8
5-7.	Compile Information Table - Enter Compiler Libraries (4 of 4)	5-8
5-8.	Compile Information Table - Enter DB2 IDs and Precompile Libraries (1 of 2)	5-9
5-9.	Compile Information Table - Enter DB2 IDs and Precompile Libraries (2 of 2)	5-9
5-10.	Compile Information Table - Enter CICS IDs and Translator Libraries (1 of 2)	5-10
5-11.	Compile Information Table - Enter CICS IDs and Translator Libraries (2 of 2)	5-11
5-12.	Compile Information Table - Enter Miscellaneous Information	5-12
5-13.	Compile Facility - Compile Menu	5-13
5-14.	Compile Facility - Compile Menu	5-14
6-1.	Prepare License Administration Utility Screen	6-5
6-2.	Prepare License Administration Utility - second screen	6-6
6-3.	License File Selection Screen	6-7
6-4.	Delete/Define and Initialize License File Screen	6-7
6-5.	Edit Delete/Define JCL Screen	6-8
6-6.	Parameter Option Screen	6-9
6-7.	License File Selection Screen	6-9
6-8.	Main Compuware License Management Screen	6-10
6-9.	Parameter Option Screen	6-10
6-10.	Maintain Nodes Table Screen	6-11
6-11.	First Import License Certificate Screen	6-12
6-12.	Second Import License Certificate Screen	6-12
6-13.	Edit JCL Screen	6-13
6-14.	Sample LMSINIT Proc	6-15
6-15.	Sample LMSINIT JCL	6-15
6-16.	Reports Screen	6-16
A-1.	General JCL Structure	7-1
A-2.	Example of the Control File input	7-3
A-3.	Summary report without a PRODUCTS DD control file	7-4
A-4.	Full report without a PRODUCTS DD control file	7-5
A-5.	Summary report with a Product DD control File	7-5
A-6.	Full report with a PRODUCTS DD control file	7-6

Tables

- 2-1. Components Utilized by Each Compuware Product2-1
- 2-2. Files Dynamically Allocated by the Preprocessor.....2-5
- 2-3. Language Processor DD Statements.2-6
- 5-1. CSS Procedure Requirements.....5-1
- 5-2. REXX EXECs or CLISTs Used to Invoke CSS utilities5-16
- 5-3. Allocating CSS Libraries.....5-17
- 5-4. DDIO File Formats5-20
- 5-5. SLCXCNTL Sample Library Members for COBOL Language Processor5-26
- 5-6. SLCXCNTL Sample Library Members for PL/I Language Processor5-27
- 6-1. LMS Steps Required6-2

Introduction

This manual contains instructions for installing, customizing, and maintaining Enterprise Common Components (ECC).

Enterprise Common Components is essentially a packaging method for the following Compuware facilities:

- License Management System (LMS)
- Compuware Shared Services (CSS)
- Host Communications Interface (HCI)

Manual Structure

This *Enterprise Common Components Installation and Customization Guide* is composed of the following chapters:

Chapter 1, “ECC Overview”: A description of the individual components that make up Enterprise Common Components with overview information on each of the parts. These include: ECC, CSS, LMS, and HCI.

Chapter 2, “Preparing for ECC Installation”: Critical information for planning your installation of ECC including the LMS, CSS, and HCI components. This chapter also lists which components are required based on your Compuware product, and also lists the libraries that are created during installation with SMP/E.

Chapter 3, “Receive From Network (RFN) Installation”: Step-by-step instructions for SMP/E installing Compuware Shared Services, License Management System, and Host Communications Services using Receive From Network (RFN) procedures.

Chapter 4, “Applying Maintenance”: Mandatory step-by-step instructions for applying maintenance to your installation of ECC.

Chapter 5, “CSS Customization”: Step-by-step instructions for customizing your installation of CSS.

Chapter 6, “LMS Customization”: Step-by-step instructions for setting up and customizing LMS for use with the Compuware products licensed by your organization.

Appendix A, “Compuware Maintenance Utility”: Explains the purpose and function of the Maintenance Utility for reporting all currently applied PTFs to a system, for all specified Compuware mainframe products.

Intended Audience

This manual is intended for installers and application programmers. It is only to be used for installing the MVS/ESA, OS/390, and z/OS versions of Compuware products.

Related Publications

For more information about ECC products refer to the Compuware documentation on FrontLine (<https://go.compuware.com>). For more information about Distributed Viewing Support, refer to the *Abend-AID Installation and Customization Guide* or *Abend-AID for CICS Installation and Customization Guide*. Instructions for accessing this documentation are given in the following paragraphs. For more information on compiling COBOL, PL/I, Assembler, or C/C++ programs, refer to the appropriate IBM manuals.

FrontLine Support Web Site

Access online technical support for Compuware products through our FrontLine support Web site. View or download documentation, frequently asked questions, and product fixes, or e-mail Compuware with questions or comments. To access FrontLine, you must first register and obtain a password at <http://go.compuware.com>.

Online Documentation

The following electronic formats are available on <http://go.compuware.com>:

- View PDF files with the free Adobe Reader, available at <http://www.adobe.com>.
- View HTML files with any standard Web browser.

World Wide Web

Compuware's site on the World Wide Web provides information about Compuware and its products. The address is <http://www.compuware.com>.

Terminology

The following terms are defined as they are used in this manual:

- **DDIO (Dump Dataset Input Output) file** is used generically to mean any of the following types of files created by the CSS utilities:
 - Report File
 - Source Listing File
 - Transaction Databases and Shared Directories
 - Report Shared Directory and Database
 - Source Listing Shared Directory
 - Source Listing Database.

For a list of Compuware products that use/require these types of DDIO files, see Chapter 2, "Allocating and Formatting DDIO Files".

For a list of Compuware products that use/require these types of DDIO files, see Chapter 5, "CSS Customization".

For a list of the types of DDIO files created by the CSS Utilities and which products use/require them, refer to the *Compuware Shared Services User/Reference Guide*.

For a list of the types of DDIO files created by the CSS Utilities and which products use/require them, refer to the *Compuware Shared Services User/Reference Guide*.

- **Member** is used to identify an individual diagnostic report in an Abend-AID report file, a source listing in a source listing file, or a profile in a profile file.
- **Entry** is used to identify an individual transaction diagnostic report in a transaction database for Abend-AID for CICS.

Getting Help

Compuware continually strives to improve our software products and documentation. Feedback from our customers helps us to maintain our quality standards.

While the software should execute according to documented specifications, there are times when problems do occur. If problems arise, check the Compuware manuals for assistance. You may also need to consult your site's technical representative for Compuware assistance.

If you have attempted to solve the problem using the resources listed above and the difficulty persists, contact the Compuware Customer Support Hotline for assistance. Customer support is also available via our FrontLine Support web site as described in "Related Publications".

Please have the following information available before calling. This information will assist the Customer Support personnel in determining the exact cause of your problem in a timely manner.

Note: If you encounter a CSS problem when compiling your program, Compuware Customer Support staff may need additional information. You can obtain this information by using the CSS Problem Documentation Utility. Refer to the "Using the Problem Documentation Utility" appendix in the *Compuware Shared Services User/Reference Guide* for complete instructions. Refer to the "Using the Problem Documentation Utility" appendix in the *Compuware Shared Services User/Reference Guide* for complete instructions. See the "Using the Problem Documentation Utility" appendix for instructions on using this utility.

1. Use the following sample JCL members to identify the PTFs and/or APARs that have been applied to the respective ECC components' libraries:
 - CXLPTFS or CXLZAPS in the CSS SLCXCNTL library
 - \$LMLPTFS in the LMS SLMSCNTL library
 - \$HCLPTFS in the HCI SLHCCNTL library.

These jobs (with the exception of CXLZAPS) will produce a detailed SMP/E report of PTFs applied to the above components. The CXLZAPS job will produce a more compact/summarized version of the CSS PTF listing.

2. Determine the release number of the ECC-enabled product.
3. For CSS problems, identify the compiler language and the release number being processed.
4. Identify the CICS release in use if the problem involves a CICS product.
5. Identify the operating system release to help determine operating system dependencies.
6. If an abend occurs, note the displacement and the module in which the interrupt occurs.
7. Note the sequence of issued commands that resulted in the abend, the data type, and the programming language being used.
8. Locate your Compuware customer number.

Compuware Customer Support
<http://go.compuware.com>

Chapter 1.

ECC Overview

This chapter explains the composition of Compuware's Enterprise Common Components (ECC) and provides overview information on each of its component parts.

What is ECC?

Enterprise Common Components is the packaging method for the following Compuware facilities:

- Compuware Shared Services (CSS)
- License Management System (LMS)
- Host Communications Interface (HCI)
- Compuware Installer

Which parts of ECC you need to install depends on which Compuware products are licensed by your organization and which parts of ECC have already been installed. All Compuware mainframe products require the License Management System. Many Compuware products use Compuware Shared Services to provide crucial core functionality. Abend-AID, Abend-AID for CICS, Strobe, Compuware Workbench, and Compuware Topaz use the HCI component. A table is provided in **Chapter 2, "Preparing for ECC Installation"** that makes it easy to determine exactly which parts of ECC you need to install.

Each of the following components are installed using SMP/E and can be installed individually, depending on your site's requirements:

- **Compuware Shared Services (CSS):** An enterprise-wide tool for sharing compiled program information, CSS is utilized by many Compuware products spanning the Fault Management and Enterprise Testing Solutions product lines. For more information, see "CSS Overview" on page 1-2.
- **License Management System (LMS):** Using LMS enables you to manage access to all the Compuware products used by your organization. LMS consists of several small components which allow you to establish, maintain, diagnose, and upgrade your access to the Compuware products licensed by your enterprise:
 - An ISPF License Administration Utility application
 - A runtime environment
 - A program interface to the runtime environment employed by Compuware products.

LMS License Certificates are English-like text files that are sent to you electronically. You will receive a License Certificate for each new product release under LMS. You can centrally manage all of your organization's License Certificates in a single License File that can then be distributed to various sites.

- **Host Communications Interface (HCI):** The Host Communications Interface is a facility that provides connectivity between mainframe-based programmer productivity software and peer-node software running on other platforms in a network. The HCI also serves as a server activation facility that is responsible for starting and monitoring application programs that are using the communications facilities.

- **Compuware Installer:** use the Compuware Installer to facilitate the SMP/E installation of Compuware mainframe products. See the *Compuware Installer—Mainframe Product SMP/E Installation Guide* for more information.

CSS Overview

Compuware Shared Services (CSS) is an integral component of many Compuware products spanning the Fault Diagnosis, Interactive Analysis and Debugging, File and Data Management, and Application Performance Measurement product lines. CSS provides storage, retrieval, and maintenance for Abend-AID reports, source listings, and transaction dump information in datasets called DDIO files. Shared directories and their attached databases are also types of DDIO files being utilized for more efficiency in maintaining information and activity within these files. CSS also provides language processing support for COBOL, PL/I, Assembler, and C.

The following products currently use CSS:

- Abend-AID
- Abend-AID for CICS
- Compuware Workbench/Topaz
- Strobe
- Xpediter/CICS
- Xpediter/Code Coverage
- Xpediter/TSO and Xpediter/IMS

System Environment

For System Requirement information, see the *Enterprise Common Components Release Notes* available from GO/Frontline (<http://go.compuware.com>).

CSS Components

CSS consists of the following components:

- Compuware common files and utilities
 - DDIO files: Source Listing Files, Report Files, Shared Directories and Associated Report, Transaction, and Source Listing Databases
 - Batch file utilities: CWDDSUTL, CWFXSDUT, CWAASDUT, CWDDLPUT, and CWDDALLU
- Compuware Language Processors (COBOL, Assembler, PL/I, and C)
- Security Exit Program

Common Files and Utilities

CSS components include common files and utilities that provide storage, retrieval, and maintenance functions.

DDIO Files

A generic term used to describe the proprietary file access method that stores Abend-AID reports, transaction dump information, and source listings for several Compuware products. DDIO files can be allocated as either VSAM or sequential datasets. Different products may require different types of DDIO files.

CSS Utilities and Batch File Utilities

CSS Utilities walks you step-by-step through CSS language processor functions and DDIO file manipulation using easy-to-use panels.

The batch file utilities (CWDDSUTL, CWAASDUT, CWDDLPUT, CWFxsdut and CWDDALLU) provide the same functions as CSS Utilities in a batch environment.

For detailed information on these utilities, see Chapter 3, “CSS Utilities”

For information on the available online utility functions refer to the “CSS Utilities” chapter in the *Compuware Shared Services User/Reference Guide*. For information on the available batch utility commands and parameters for CWDDALLU or CWDDSUTL, CWAASDUT, CWFxsdut and CWDDLPUT refer to the “Batch File Utility CWDDALLU” chapter in the *Compuware Shared Services User/Reference Guide*.

CSS Language Processors (LP)

The language processors capture information about a compiler listing and store it in a source listing DDIO file. The majority of this information is gathered from the compiler listing, but optionally, SYSIN and SYSLIB are also examined. The source listing DDIO file may be in standard DDIO, shared directory, or database format.

A language processor can be run using two methods:

- preprocessor
- post-processor

You may use the preprocessor or post-processor (whichever is best for your site as needed.) For information about the pre- and post-processors, refer to the appropriate language processor chapter in the *Compuware Shared Services User/Reference Guide*. See the applicable language processor chapter for when to use the pre- or post-processor.

Language Processor Types

CSS provides language processing support for the following languages:

- COBOL
- PL/I
- Assembler
- C

The COBOL language processor reads the compiler listing produced by the COBOL compiler and writes either a compiler listing or an enhanced listing to the DDIO file. The enhanced listing merges the information normally found in the data division map (DMAP) and the condensed listing (CLIST) or procedure map (PMAP) into the source statement lines. The COBOL compiler options in effect are sorted alphabetically and appear at the beginning of the enhanced listing. The enhanced listing provides you with a condensed hardcopy of the compiler information.

The COBOL and PL/I listings may be used to provide source support using any of the following products:

- Abend-AID
- Abend-AID for CICS
- Strobe and iStrobe
- Xpediter/CICS
- Xpediter/TSO
- Xpediter/Eclipse

The Assembler listings may be used to provide source support using any of the following products:

- Abend-AID
- Strobe and iStrobe
- Xpediter/CICS
- Xpediter/TSO
- Xpediter/Eclipse

The C language processor reads the compiler listing produced by the C compiler and writes the compiler listing to the DDIO file and SYSCPRT. The source listing in DDIO can then be used to provide source support when using:

- Xpediter/TSO
- Xpediter/CICS
- Strobe and iStrobe

Security Exit Program

CSS also includes the Compuware Security Exit Program. This is an optional user-written exit for your Compuware products. It can be used in conjunction with your existing security package to secure sensitive data used by the following Compuware products:

- Abend-AID
- Abend-AID for CICS
- Xpediter/CICS
- Xpediter/TSO and Xpediter/IMS
- Xpediter/Code Coverage
- Strobe

The Compuware Security Exit enables you to control access to DDIO file members or restrict access to various commands through a user-written exit program. For the Abend-AID for CICS product, you can restrict access to some of the CWFSDUT and CWDDALLU functions for transaction database and source listing files. For transaction databases, you can use the Security Exit Program to restrict access to individual dumps within a transaction database.

If you install a Security Exit program, it will be called by all Compuware products installed at your site. For more information on the Security Exit Program, see the “CSS Security Exit” chapter in the *Compuware Shared Services User/Reference Guide*.

Executing CSS

CSS is primarily executed at the following times:

- Abend-AID and Abend-AID for CICS
 - at compile time (when source support is used)
 - at time of abend or application program failure
 - at view time when using the Abend-AID Viewer
 - at installation and maintenance time when formatting a DDIO file.
 - when a compile listing that has been stored elsewhere is needed for Abend-AID or Abend-AID for CICS viewing with source support.
- Xpediter/TSO and Xpediter/CICS
 - at compile time (when source support is used)
 - at execution time for interactive debugging
 - at installation and maintenance time when formatting a DDIO file.
- Strobe
 - when indexing is used to analyze measurement data
 - to support Web services and SQLAF processing thru CSS TP of HCI

- Compuware Workbench
 - to support Web Services needs through the CSS TP and other HCI related facilities (SSAS and STASK)

LMS Overview

LMS is located on the ECC product shipping package. LMS enables you to centrally administer Compuware's product License Certificates and manage access to Compuware products at your site. LMS includes several components that together enable you to establish, maintain, diagnose, and upgrade access to the Compuware products licensed by your enterprise.

LMS Advantages

LMS provides the following advantages:

- Electronically-delivered License Certificates that contain product access parameters in a human-readable text format
- Programs and reports to verify the contents of the License File and LMS runtime environment
- Capability to employ a single centralized Enterprise License File and centralized administration of License Files
- Easy updates to License Certificates and the License File with no disruption to your Compuware products
- No production impact from the testing of LMS runtime systems, installation and validation of new License Certificates, or installation of new or updated LMS software
- Optional E-mail alerts and SMF logging when licensing events occur.

Operating Environment

LMS operates under IBM MVS/ESA Release 4.3 and more current, and all OS/390 and z/OS releases.

The License Management Process

The license management process begins when you acquire a Compuware product through a license agreement, trial agreement, or beta agreement.

Information Gathering

Compuware obtains the relevant information for license management from you at the time of the agreement. Your organization supplies information such as name and the sites and CPUs for which the product is being licensed. Information about the Compuware product licensed, such as product name, release number, and options licensed, is obtained from your Compuware representative.

License Certificate

Upon the completion of an agreement, Compuware creates a License Certificate representing a portion of the information in that agreement. The License Certificate is used by LMS to provide access to Compuware products. The License Certificate is **not** the same thing as a license agreement. You are still responsible for abiding by your license agreement, and although not its primary role, you will find that the LMS can help in that effort.

The License Certificate, LMS, and the Compuware product are all delivered to you by Compuware. Typically, you receive the License Certificate via e-mail, but other methods can be used, if necessary.

Installation

The first time your enterprise receives LMS, you must install the LMS software. Then you set up your LMS environment, import your License Certificate into a **License File**, activate the **Runtime License Management System**, and install your Compuware product. From then on, access to your Compuware mainframe products occurs transparently.

After you accomplish the initial LMS installation, you would typically only need to import a new product License Certificate and re-initialize the LMS runtime environment in conjunction with events such as the following:

- obtaining a new release of a product under the terms of your software maintenance agreement
- adding new products through additional license agreements
- amending your agreement to include new options
- changing the CPUs licensed in your original agreement.

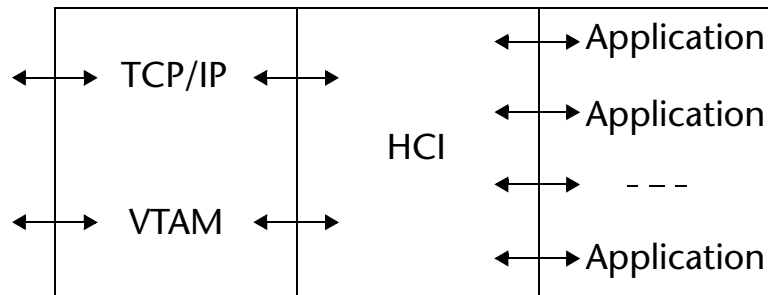
HCI Overview

The Host Communications Interface (HCI) is a facility that provides connectivity between mainframe-based programmer productivity software and peer-node software running on other platforms in a network. These communications facilities provide a simple application programming interface so that Compuware software can be written to communicate with peer programs running on different platforms. Additionally, the communications system is flexible, simple to operate, and reliable. It has been designed to support open-ended communications protocols, and it can operate on older releases of the MVS operating system. It allows application programs to communicate over any one of several protocols without knowing which protocol is in use at any given time.

The HCI also serves as a server activation facility that is responsible for starting and monitoring application programs that are using the communications facilities.

And finally, the HCI has been designed with no architected upper limit on the number of concurrent sessions supported, nor on throughput rates. The following figure shows how the HCI fits into the scheme of the TP application and the MVS subsystem components.

Figure 1-1. TP Subsystem, HCI and Application. HCI Position as middleware



Chapter 2.

Preparing for ECC Installation

This chapter contains important information that can help you prepare to install the Enterprise Common Components (ECC) in the most efficient manner. Review it before beginning the installation process.

Required Components and Procedures

This section explains how to determine which parts of ECC need to be installed at your site and which chapters in this manual contain the applicable customization procedures.

Component Selection

Different Compuware products utilize different portions of ECC. Use Table 2-1 to determine which parts of ECC are needed at your site.

Notice that Compuware's License Management System (LMS) is required for every mainframe product, while HCI and CSS are components for some products.

Table 2-1. Components Utilized by Each Compuware Product

Compuware Product	LMS	CSS	HCI*
Abend-AID	x	x	x
Abend-AID Fault Analytics	x	x	
Abend-AID for CICS	x	x	x
File-AID/Data Solutions	x		
File-AID for DB2	x		
File-AID for IMS	x		
File-AID/MVS	x		
File-AID/RDX	x		
Hiperstation	x		x**
Strobe	x	x	x
Xpediter/CICS	x	x	
Xpediter/Code Coverage	x	x	
DevEnterprise	x	x	x
Compuware Workbench/Topaz		x	x
Xpediter/TSO and /IMS	x	x	
Xpediter/Xchange	x		
* HCI can be optionally installed from the ECC distributed media. Please refer to individual product's documentation for configuration and customization instructions.			
** HCI is only required if using the Hiperstation viewer.			

Required Procedures

Determining which chapters to use in this manual is actually very simple:

- Every mainframe Compuware product uses LMS. Therefore, every installation of a Compuware product requires, at a minimum, that the latest maintenance be applied to LMS. So regardless of which product you are installing, you must perform the applicable procedures in the following chapters:
 - Chapter 3, “Installing Enterprise Common Components” (if current release has not been previously installed)
 - Chapter 4, “Applying Maintenance”
 - Chapter 6, “LMS Customization”
- If the Compuware product you are installing requires CSS, you must also perform the procedures in Chapter 5, “CSS Customization”.

The information within each chapter will direct you as to which steps apply in your situation.

Note: If you are installing the Host Communications Interface (HCI) components, refer to each product manual that uses HCI for specific customization/configuration instructions. **The *Host Communications Interface User/Reference* manual does not contain configuration or customization information for specific products and should not be used to configure HCI.**

ECC Packaging

The ECC product media or as part of a Receive from Network (RFN) order, includes LMS, CSS, and HCI, is packaged for installation using System Modification Program Extended (SMP/E). SMP/E is a method developed by IBM to install software products in the MVS environment.

SMP/E provides enhanced installation management and maintenance tracking capabilities. To facilitate SMP/E, the LMS, CSS, and HCI components are shipped separately from the products, via RFN or via the ECC EP physical media (upon request). The ECC software is distributed with both the base code and all cumulative, approved maintenance to-date. To install the ECC, refer to Chapter 3, “Installing Enterprise Common Components”.

The installation is facilitated by the Compuware Installer consisting of a set of REXX EXECs, JCL skeletons, and panels. This facility prompts you for the installation information and then builds the necessary jobs required to perform the SMP/E installation. You can also view reporting information from the installation facility, such as parameters and execution information.

Cumulative maintenance for ECC components is available for download from the Compuware GO/Frontline web site (<http://go.compuware.com>). It is recommended that you periodically download and apply cumulative maintenance to keep your product version current.

ECC Component Prefixes and FMIDs

Compuware has registered the following element prefixes with IBM for the ECC components:

- LCX for CSS
- LMS for LMS
- LHC for HCI

The FMIDs for each component are listed below.

- MLCXnnn — CSS base code
- NLCXnnn — CSS Japanese support
- MLMSnnn — LMS base code
- MLHCnnn — HCI base code

ECC Installation Considerations

This section describes considerations for installing and customizing ECC that applies to LMS, CSS, and/or HCI. Check your mainframe product installation documentation to verify if these any of the ECC component applications have already been installed. If they are installed, you may consider only applying maintenance. See **Chapter 4, “Applying Maintenance”** for more information.

Link Lists

Refer to your Compuware product documentation to determine whether the load library or libraries for your components should be placed in the link list.

PDS/E Support

ECC components support Partitioned Data Set/Extended (PDS/E) libraries. LMS, CSS, and/or HCI can be installed in PDS/E object libraries. Input source members for the CSS preprocessor or compiled listings for the CSS post-processor may reside in a data PDS/E library. The following are supported for PDS/E libraries:

- PDS/E object libraries are supported with DFSMS 1.1 and above.
- PDS/E data libraries are supported with DFSMS 1.0 and above.

CSS Installation Considerations

This section describes considerations for installing Compuware Shared Services.

APF Authorization

Following IBM system integrity guidelines, Compuware recommends that your CSS load library (SLCXLOAD) **NOT** be APF-authorized. Depending on your MVS system installation options, libraries placed in the link list may be APF-authorized automatically.

Note: If you are using Abend-AID and Abend-AID is APF-authorized, then CSS should also be APF-authorized. If Abend-AID is not APF-authorized, then CSS should not be APF-authorized. The **ONLY** time the CSS load library should be APF-authorized is when you are using the Abend-AID load library as APF-authorized.

If you will be installing and configuring the Compuware Workbench Mainframe Interface TP, this component does require an APF-authorized library. However, this component will install into a separate and very small load library (SLCXAUTH) and the main CSS load library need not be APF-authorized subject to the note above.

CICS Load Library Definition

The CSS library must be included with Compuware product libraries in the CICS startup JCL or RDO definitions for the following product:

- Xpediter/CICS (all supported releases)

TSO Logon PROCs

If the CSS load library is not in the link list and dynamic library allocation is not used, the CSS load library must be placed before all Compuware product load libraries in the STEPLIB or ISPLLIB concatenations in the TSO logon PROC. The CSS message and panel libraries must be placed before all Compuware product message and panel libraries in the ISPMLIB and ISPLLIB concatenations in the TSO logon PROC.

Sharing CSS DDIO Files With Multiple CPUs

If you are sharing CSS DDIO files among multiple CPUs, the following qnames must be changed from LOCAL to GLOBAL enqueues.

- ABENDAID
- ABENDDAM
- ABENDSMF
- CWCATALG

Note: These qnames are needed for **all** products. Please refer to the individual Compuware product documents with which CSS was distributed for other product requirements.

For IBM's GRS (Global Resource Serialization), these qnames must be added to the inclusion RNL. This RNL is defined in the GRSRNLxx member of SYS1.PARMLIB for MVS/XA and later. For pre-MVS/XA systems, these qnames are maintained in the ISGSIRNL entry point of ISGGRNLO. Please consult the IBM product documentation for further details.

The CSS sample library contains an example of how qnames should be coded in the GRSRNLxx member of SYS1.PARMLIB. See "CSS Sample Library Members" on page 2-7 for more information.

For resource serialization packages other than GRS (for example MIM), consult the product documentation for changing LOCAL to GLOBAL enqueues.

CSS Language Processor (LP)

The language processor captures information about a compiler listing and stores it in a source listing DDIO file. The majority of this information is gathered from the compiler listing, but optionally, SYSIN and SYSLIB are also examined. The source listing DDIO file may be in standard DDIO, shared directory, or database format. Releases prior to Xpediter/TSO 7.3, Xpediter/CICS 7.6, and Abend-AID for CICS 4.5 do not support source listing shared directories. Instead, individual source databases may be specified for lower levels.

The language processor can be run using two methods:

- pre-processor
- post-processor

You may use the pre-processor or post-processor (whichever is best for your site as needed). This section discusses the benefits of the preprocessor and the post-processor and describes when to use them.

Pre-processor

Some information about a program module is not always available from the compiler listing. The pre-processor, in addition to gathering information from the compiler listing, gathers information from SYSIN and SYSLIB. The preprocessor provides additional functionality for Xpediter/CICS and Xpediter/TSO, and is required for PL/I programs that use the %NOPRINT statement.

When setting up the language processors, Compuware recommends that you use the pre-processor instead of the post-processor. The post-processor should only be used in situations where compiled source listings have been stored.

Pre-processor Steps

1. Determines the proper compiler options required to process the listing file.
2. Automatically invokes the compiler or assembler to compile or assemble your source program.
3. Writes the listing to the DDIO file. An enhanced listing for COBOL programs can be produced, if desired. PL/I and Assembler listings are written to SYSPRINT. C listings are written to SYSCPRT.

Pre-processor Benefits

- **Better handling of compiler errors.** By using the preprocessor, you can avoid the potential problem of processing a listing with compiler errors. The preprocessor, in conjunction with the CONDDIO parameter, internally checks the return code from the compiler and doesn't write a DDIO listing that contains errors. For example, if you set the CONDDIO parameter to 8, the preprocessor will write the compiler listing to the DDIO file unless the compiler return code exceeds 8. The preprocessor can process the compiler errors more effectively than the post-processor. Compuware recommends that you use the preprocessor rather than the post-processor.
- **Capturing suppressed source code.** When any of the following parameters are used, sections of source code can be suppressed from the compiler listing:

For COBOL COPY SUPPRESS
 For PL/I %NOPRINT
 For Assembler PRINT OFF or PRINT NOGEN
 For C NOSHOWINC and NOEXPMAC

The preprocessor captures this information from SYSIN and SYSLIB, or by forcing on more revealing compiler options, then altering the listing to emulate the options you requested.

- **Automated compiler options.** The post-processor requires that certain compiler options be specified in order to process all needed sections of the compiler listing. The preprocessor can automatically pass the required options to the compiler.
- **Simplified JCL.** While the post-processor requires that the user add a step after the compile step, the preprocessor requires only that minor modifications be made to your existing compile step.

Files Dynamically Allocated

The following files are dynamically allocated by the Compuware preprocessor. The attributes can be overridden by specifying a DD in the JCL.

Table 2-2. Files Dynamically Allocated by the Preprocessor

DDname	Attributes
CWPWRKn (0-6)	BLKSIZE=19000 SPACE(TRK,(100,80)) UNIT=SYSDA
SORTWK01	BLKSIZE=19000 SPACE(TRK,(100,80)) UNIT=SYSDA
TEMPLIN	BLKSIZE=3200 SPACE(TRK,(200,100)) UNIT=SYSDA
CWPERRM	SYSOUT=*
SYSPRINT	SYSOUT=*
SYSOUT	SYSOUT=*
CWPPRTI (PL/I)	BLKSIZE=19000 SPACE(TRK,(100,200)) UNIT=SYSDA

Table 2-2. Files Dynamically Allocated by the Preprocessor

DDname	Attributes
CWPPRTI (all other languages)	BLKSIZE=16093 SPACE(TRK,(100,200)) UNIT=SYSDA DISP=MOD
CWPPDS2	SPACE (TRK,(100,80,80)) UNIT=SYSDA

CWPPRTO is dynamically allocated as a work file (for the preprocessor). SYSPRINT is where the listing goes.

Post-processor

The postprocessor is executed as a single step and it can be a separate job. It reads in the listing created from the compiler. Information is gathered from the source listing, XREF, data maps, and object code sections of the listing. This information is stored in a DDIO file member and is used by various Compuware products.

While the preprocessor is the preferred method of loading the compiled listing to the DDIO, there are situations that necessitate the use of the postprocessor as a viable alternative. This is commonly found in production environments where the listings may be archived. In order for the listings to populate the DDIO, they must be compiled using the required options specified in the appropriate language processor chapters in the *Compuware Shared Services User/Reference Guide*. In order for the listings to populate the DDIO, they must be compiled using the required options specified in the appropriate language processor chapter. Note that these are usually the default options at many shops.

If the programs contain suppressed source code, certain additional control statements must be added to the language processor. Please view the Language Processor chapters in the *Compuware Shared Services User/Reference Guide* for the programming language desired for more details. Please view the Language Processor chapters for the programming language desired for more details.

If the programs are compiled with OFFSET as well as OPTimize), certain additional control statements must be added to the language processor. Please view the Language Processor chapters in the *Compuware Shared Services User/Reference Guide* for the programming language desired for more details. Please view the Language Processor chapters for the programming language desired for more details.

Postprocessor Benefits

- Exact match of compiled output listings with the executable load module. This is especially important in production environments where time is critical.
- Eliminates the risk of copybook or source code changes prior to recompiling the code since the listing reflects the code that is being executed.
- Significantly less time is needed to process source listings as opposed to recompiling the source code.

Language Processor DD Statements

Table 2-6 lists the DD statements for the Assembler, COBOL, PL/I, and C language processors. Not every DD Name applies to every language processor.

Table 2-3. Language Processor DD Statements.

DDname	Purpose
CWPWRK (0-6)	LP work files
CWPPRMO	LP parameter input dataset
CWPDDIO	Target source listing DDIO file
CWPPRTI	Compiler listing input to postprocessor

Table 2-3. Language Processor DD Statements.

DDname	Purpose
CWPPRTO	Compiler listing output from postprocessor
CWPLOAD	Option LOAD or OBJECT module input to postprocessor
CWPDECK	Option DECK module input to postprocessor
CWPERRM	LP Error Messages
CWPWBNV	Workbench navigation information

Note: CWPLOAD and CWPDECK correspond to the SYSLIN and SYSPUNCH compiler outputs. If you specify compiler option OBJECT, you should use CWPLOAD in your postprocessor JCL. If you specify compiler option DECK, you should use CWPDECK in your postprocessor JCL. Do not feed the SYSLIN output into CWPDECK or the SYSPUNCH output into CWPLOAD. The updated OBJECT contained in CWPLOAD and CWPDECK should be used in the program's linkedit step. The preprocessor does not use CWPLOAD and CWPDECK.

Writing a Listing to a DDIO File

Several options exist to place listings in a DDIO file. Each option requires that some changes be made to your JCL. The type of changes needed depend on whether you are using the preprocessor or the post-processor, and the type of debugging session. Refer to "CSS Sample Library Members" for a list of the sample JCL that can be used for the various options.

Testing and Debugging Programs

The actual method used for handling testing and debugging programs depends on whether you are preprocessing or postprocessing:

- **Preprocessing** — Run the preprocessor and link your program.
- **Postprocessing** — Compile your program, run the post-processor, and then link your program.

With either of these methods, the language processor places a listing in the DDIO file whenever a program is compiled.

Debugging Production Programs

If the original compiler listing is available in machine-readable format, you can retrieve the listing and use it as input to the post-processor in order to reprocess the listing and recreate it in the DDIO file. The post-processor can process listings stored by most products.

Note: If the listing you use as input to the post-processor contains any other information (such as translator or link edit output), in addition to the compiler information, you may not obtain the desired results.

CSS Sample Library Members

The following list describes the members of the CSS sample JCL library that you may work with during installation (depending on your installation options). Use these members only as needed if directed to do so by a specific Compuware product installation guide. These members are distributed and updated via SMP/E. The members will be located in the following files after the SMP/E Apply and Accept steps are complete.

- COMPWARE.MLCX mmm .SLCXCNTL (target library)
- COMPWARE.MLCX mmm .ALCXCNTL (distribution library)

where mmm indicates the CSS release number.

To print the CSS sample library members, submit the JCL contained in library member CXPRINT after the distributed media has been unloaded.

CWASSECD	Security Exit program DSECT.
CWASSECU	Sample Security Exit program.
CWCMTRHE	Sample customized horizontal translation table for mixed-case English.
CWCMTRHT	Sample customized horizontal translation table for uppercase English.
CWCMTRHU	Sample customized horizontal translation table for mixed-case English with Euro character.
CWCMTRVE	Sample customized vertical translation table for mixed-case English.
CWCMTRVT	Sample customized vertical translation table for uppercase English.
CWCMTRVU	Sample customized vertical translation table for mixed-case English with Euro character.

Note: The following five members are specific for Abend-AID. They allow browse access to diagnostic reports and source listings directly through ROSCOE or print access via a job submitted from ROSCOE to run in batch. Refer to the members for additional information.

CWROSBAT	ROSCOE sample members.
CWROSBRW	ROSCOE sample members.
CWROSCOE	ROSCOE sample members.
CWROSDIR	ROSCOE sample members.
CWROSPDF	ROSCOE sample members.
CWUTCLSE	Sample CLIST to invoke CSS online utilities with English messages and panels.
CWUTCLSJ	Sample CLIST to invoke CSS online utilities with Japanese messages and panels.
CWUTREXE	Sample REXX EXEC to invoke CSS online utilities with English messages and panels.
CWUTREXJ	Sample REXX EXEC to invoke CSS online utilities with Japanese messages and panels.
CWVFCLPT	CLIST to invoke the language processor debugging aid.

Note: The debugging aid is a diagnostic tool to be used only under the supervision of Compuware Customer Support in the event of a problem related to the language processor.

CWVFRXPT	Sample REXX EXEC to invoke the Compuware language processor debugging aid.
-----------------	--

Note: The debugging aid is a diagnostic tool to be used only under the supervision of Compuware Customer Support.

CXAADIRX	Abend-AID CWAASDUT DIRX sample JCL.
CXAAEXPO	Abend-AID CWAASDUT EXPORT command sample JCL.
CXAAIMPO	Abend-AID CWAASDUT IMPORT command sample JCL.
CXAAMOVE	Abend-AID CWAASDUT MOVE command sample JCL.
CXALLBAA	JCL for creating sequential Abend-AID database files.
CXALLBLP	JCL for creating sequential source listing database files.
CXALLBSD	JCL for creating Abend-AID for CICS sequential transaction databases.
CXALLDAA	JCL for creating Abend-AID shared directories.
CXALLDLP	JCL for creating Source Listing Shared Directories.
CXALLDS	JCL to allocate a sequential DDIO file.
CXALLMC	JCL for creating Abend-AID for CICS shared directories.
CXALLVAA	JCL for creating VSAM Abend-AID database files.
CXALLVLP	JCL for creating VSAM Source Listing Database files.
CXALLVS	JCL to allocate a standard VSAM Abend-AID report or source listing DDIO file.
CXALLVSD	JCL for creating Abend-AID for CICS VSAM transaction databases.
CXASM	JCL for running the Assembler language post-processor.
CXASMPRE	JCL for running the Assembler language preprocessor.
CXC	Sample JCL for running the C post-processor.
CXCAPTUR	Sample JCL for creating a tape from the output of the CSS Problem Documentation Utility.
CXCFGEXT	Sample JCL to extract a source file from the configuration module in a CSS load library.
CXCFGINI	Sample source configuration file.
CXCFGSET	Sample JCL to build a configuration module in a CSS load library, from a source configuration file you create.
CXCIREG	Sample JCL to register the Contact Information dataset.
CXCIVSAM	Sample JCL to create the Contact Information dataset.
CXCOBPRE	JCL for running the COBOL language preprocessor.
CXCOB1	JCL for the step to be added after the compile step in your current COBOL compile and link edit JCL. This is used in order to process the compiler listing through the Compuware language post-processor.
CXCOB2	JCL to process COBOL compiler listings stored in machine-readable format.

CXCOB99	Sample JCL for running the COBOL post-processor.
CXCPRE	Sample JCL for running the C preprocessor.
CXC2	Sample JCL to postprocess C compiler listings stored in machine-readable format.
CXDCRADB	JCL to allocate and CREATE a multivolume Abend-AID report database.
CXDCRASD	JCL to allocate and CREATE an Abend-AID shared directory and a multi-volume report database.
CXDCLDDB	JCL to allocate and CREATE a multi-volume source listing database.
CXDCLSD	JCL to allocate and CREATE a source shared directory and a multi-volume Source Listing database.
CXDVRTDB	JCL to allocate and CREATE a multi-volume transaction database.
CXDVRTSD	JCL to allocate and CREATE an Abend-AID for CICS shared directory and a multi-volume transaction database.
CXDDSUTL	Sample JCL for running the CWDDSUTL batch file utility.
CXDDUNLP	Sample JCL to execute the CWDDUNLP source extraction utility.
CXDFMAA	JCL to allocate and FORMAT a standard multi-volume Abend-AID report DDIO file.
CXDFMSL	JCL to allocate and FORMAT a standard multi-volume source listing DDIO file.
CXEXPORT	Sample JCL for running the CWDDSUTL EXPORT command.
CXFLAG	Sample JCL for the CWDDSUTL utility FLAG command.
CXFMTAA	Sample JCL and CWDDSUTL control statement for formatting Abend-AID report files.
CXFMTSL	Sample JCL and CWDDSUTL control statement for formatting a source listing file.
CXCFDIRX	Abend-AID for CICS CWFXSDUT DIRX sample JCL.
CXCFEXPO	Abend-AID for CICS CWFXSDUT EXPORT command sample JCL.
CXCFEXTL	Sample JCL to export a Abend-AID for CICS transaction and a source listing to a single tape.
CXCFIMPO	Abend-AID for CICS CWFXSDUT IMPORT command sample JCL.
CXCFMOVE	Abend-AID for CICS CWFXSDUT MOVE command sample JCL.
CXIMPORT	Sample JCL for running the CWDDSUTL IMPORT command.
CXJCLSEC	JCL to assemble and link edit the Security Exit program.
CXJCLTRT	JCL to assemble and link edit the custom translation tables.
CXLPASM	Sample Assembler language processor options.
CXLPC	Sample C language processor options.

CXLPCOBB	Sample COBOL language processor options (batch programs).
CXLPCOBC	Sample COBOL language processor options (CICS programs).
CXLDIRX	LP source CWDDLPUT DIRX sample JCL.
CXLPEXPO	LP source CWDDLPUT EXPORT command sample JCL.
CXLPIMPO	LP source CWDDLPUT IMPORT command sample JCL.
CXLPMOVE	LP source CWDDLPUT MOVE command sample JCL.
CXLPPLI	Sample PL/I language processor options.
CXLPTFS	Sample JCL to run SMPE LIST on CSS Target/Distribution zones.
CXLZAPS	Sample JCL to list the PTFs and APARs on a CSS load library
CXMODMAP	Sample JCL member to map the CSECTs in a load module.
CXOTFPRE	Sample CLIST stub for preprocessing exit for OTF
CXOTFPST	Sample CLIST stub for postprocessing exit for OTF
CXPDDSU	Sample JCL to print a listing or report from a DDIO file.
CXPLI	JCL for running the PL/I language post-processor.
CXPLIPRE	JCL for running the PL/I language preprocessor.
CXPLI2	JCL to postprocess PL/I compiler listings stored in machine-readable format.
CXPRCTL	IEBTPCH control statements for job CXPRINT.
CXPRINT	JCL to print the entire CSS installation library.
CXRELS	PTF/APAR tracking module. Do not modify this member.
CXR00003	Sample Tutorial main menu.
CXSPRE01	Sample Skeleton for on-the-fly preprocessing.
CXSPST01	Sample Skeleton for on-the-fly postprocessing.
CXTUTOR	Alternate Sample Tutorial main panel.
CXUECEXT	Sample CLIST for the Compile Facility JOB card exit
CXUECIN2	REXX EXEC used to invoke the Compile Information installation screens
DDIODAYS	REXX utility to delete DDIO file members by age.
DDIODAYJ	Sample JCL for running the DDIODAYS REXX utility. This utility deletes DDIO file members by age.
FDXTRN01	File-AID DB2 Interface CLIST.
GRSRNLXX	Sample GRS qname inclusion RNL.
P@CPRE01	Sample panel for preprocessing exit for on the fly.

P@CPST01	Sample panel for postprocessing exit for on the fly.
P@HPRE01	Sample Help panel for preprocessing exit for on-the-fly processing.
P@HPST01	Sample Help panel for postprocessing exit for on-the-fly processing.
ATSOECLP	Sample TSO logon procedure for Xpediter/TSO via Compuware Workbench.
CSPFINIT	Sample JCL to create and initialize the Compuware Shared Profile dataset.
CSSTPCFG	Sample Compuware Workbench TP configuration file member.
CXCFGBLD	Sample JCL to create the Compuware Workbench configuration file dataset.
CXTPDEF	Sample HCI configuration parameters to be inserted into an existing HCI configuration.
CXTP01	Sample HCI PARMLIB member for the Compuware Workbench TP.
HCICFG	Sample JCL and macros that define a total HCI configuration for the Compuware Workbench TP (use with a newly installed or copied HCI).
HCIJCL	Sample HCI start-up JCL.
HCIJOURN	Sample JCL to create the HCI journals.
HCILOAD	Sample JCL to create the HCI configuration load library.
HCIPARM	Sample JCL to create the HCI PARMLIB library and member.

LMS Installation Considerations

This section describes considerations for installing and customizing LMS.

APF Authorization

Following IBM system integrity guidelines, Compuware requires that your LMS load library be APF-authorized. You must provide an APF-authorized load library into which you place the modules for LMSINIT since all load modules for LMSINIT must reside in an APF-authorized load library.

Execution Sequence

In order to ensure availability of Compuware products with LMS-controlled access, the LMS runtime environment must be established before any of those products are initialized. For this reason, Compuware strongly recommends that you institute an MVS start-up procedure based on the sample proc provided. It should run automatically as part of IPL and IML processing prior to any procs utilized by other Compuware products. It may be necessary to consult your site's MVS system programmer.

The License Management System Functions

The License Management System has three basic functions.

License Management Administration

You will create and maintain your License File using the License Administration Utility (LAU) installed with your License Management Software. Your License File is used as the source from which Compuware products will validate access for your site. When you receive License Certificates for Compuware products, your organization's License Administrator must use the LAU to **import** the License Certificates into the License File. Additional features of the LAU assist in the maintenance of your License Files. In addition to browsing the License Certificates in your License File, you may run a number of administrative reports from the LAU. These reports will reveal the contents of your License File, from a number of perspectives, allowing you to determine the state of the License Certificate for any particular Compuware product release.

Depending on the requirements of your organization, you may have more than one License File. Each License File can be centrally administered from the License Administration Utility.

In addition to using the LAU to administer licenses in individual license files, an optional feature called the Centralized Licensing Facility (CLF) can be used to manage your licensing environment from a single z/OS system regardless of where your production or test LPARs reside. Please see the *License Management System User/Reference Guide—Centralized Licensing Facility* for more information about CLF.

Establishing the Runtime LMS Environment

You will use the License Files created and maintained by the License Administration Utility as input to a program that establishes your Compuware LMS runtime environment. This program is named LMSINIT. LMSINIT is the License Management System program that reads the License File¹ and constructs the License Cache and License Management System subsystem against which Compuware product runtime license access requests are later made.

Important: You must re-establish your runtime License Management System environment at each IPL, and whenever you have an update to your License File, the update must be made available to your z/OS systems. At IPL startup, LMSINIT must be completed before starting up any Compuware product.

Validating Product Access Requests During Product Use

Your access to your Compuware products will be validated when you use your Compuware products. The Compuware product will make a request of the License Management System to determine if your site has a valid License Certificate for the product release. The requests are made by the product at various times during its execution depending on the product's needs. The Compuware product will access the LMS subsystem established, request License File information and act upon the information. If the License Certificate information is valid, users will proceed with their use of the product without disruption. Any abnormal License Management event detected will be reported to the product's user and may optionally be reported by e-mail to your organization's License Administrator. The disposition and frequency of the reporting of these events can be controlled by the customer using the "Message Suppression and Distribution" feature of LMS. See the *License Management System User/Reference Guide—Centralized Licensing Facility* for more information about this feature. Optionally, these events may also be recorded in SMF Logging.

Note: LMSINIT can read more than one License File if you are an organization that administers Compuware product access for more than one Compuware customer.

1. LMSINIT can read more than one License File if you are an organization that administers Compuware product access for more than one Compuware customer.

HCI Installation Considerations

This section describes considerations for installing the Host Communications Interface which is a requirement for several Compuware products.

APF Authorization

The HCI component library (COMPWARE.MLHCnnn.SLHCAUTH) must be APF-authorized. This library is used in the HCI regions that serve various Compuware products. See Table 2-1 on page 2-1 for the specific products.

Security Settings

HCI can switch ACID (UserID) while processing certain requests. Some security package settings may then prevent the HCI from writing to JES sysout after this is done. Installation with such strict security settings need to allow the HCI task access to the JESSPOOL class.

Installations that have such settings that limit any or all user access to output from other users can do so by setting the appropriate profiles for the security class JESSPOOL.

Profiles for JESSPOOL are in the format:

```
Local-nodename.userid.jobname.jobid.dsnumber.name
```

Using combinations of userIDs and wildcard characters, output viewing can be limited on a userid or groupid basis as the installation chooses.

Please consult your installation's appropriate security software administration documentation for additional information. For RACF installations, controlling access to output via the JESSPOOL class is documented in IBM's *z/OS Security Server RACF Security Administrator's Guide*, section entitled, "Controlling Access to Spool Data".

Chapter 3.

Receive From Network (RFN) Installation

Installation Methodology

Compuware's primary product delivery and installation method is Receive From Network (RFN). The RFN process enables you to receive the product directly to your z/OS environment and install it using System Modification Program Extended (SMP/E).

After you select a licensed product from Compuware's online ordering system, you will receive an e-mail message containing sample JCL for creating a Compuware Installer dialog PDS.

Documentation for using the Compuware Installer to install one or more ECC components (CSS, LMS, HCI) is contained in the *Compuware Installer Mainframe Product SMP/E Installation Guide* document attached to the order e-mail.

If you ordered the ECC EP Physical media, you can obtain a copy of this document from Compuware GO (<http://go.compuware.com>) in the ECC documentation section. The document also contains information for using the EP physical media to SMP/E install the ECC components.

Chapter 4.

Applying Maintenance

All maintenance for each supported Compuware product release is provided on Compuware GO (<http://go.compuware.com>). Select **Enterprise Common Components** from the **Product Support: My Product** selection menu, the Enterprise Common Components web page appears. Locate and select the **Fixes/Downloads** link and select the desired product and release for which to apply maintenance.

Follow the instructions on the web site for acquiring and applying cumulative maintenance.

Optionally, you may apply maintenance by getting a current copy of the ECC EP physical media and using it to generate the JCL for applying maintenance through SMP/E. Refer to the *Compuware Installer — Compuware Products SMP/E Installation Reference* for additional details on applying maintenance from a Compuware EP.

CAUTION:

Products using address spaces or server regions where CSS and/or HCI libraries are allocated must be shut down while maintenance is being applied to their CSS libraries and restarted after the maintenance process is complete. This applies especially to Abend-AID for CICS; HCI regions supporting the Compuware Workbench, Strobe and DevEnterprise; and Xpediter/CICS users.

Chapter 5.

CSS Customization

The procedures in this chapter are used to customize Compuware Shared Services (CSS). **If you are installing Abend-AID 11.02.00 or higher or Abend-AID for CICS, you must install HCI prior to customizing CSS.** For information about configuring the HCI server for Abend-AID, refer to the *Abend-AID User/Reference Guide* or the *Abend-AID for CICS Installation and Customization Guide*.

As shown in Table 5-1, some of the steps in this chapter are required, while others are optional.

Table 5-1. CSS Procedure Requirements

Step Number	Required	Optional
"Step 1. Implement the Security Exit Program (Optional)"		x
"Step 2. Install Customized Translation Tables (Optional)"		x
"Step 3. Make New CSS Load Modules Accessible to Compuware Products (Required)"	x	
"Step 4. Create the Compile Information Table (Required)"	x	
"Step 5. Activate Compuware ISPF Dialogs (Required)"	x	
"Step 6. Link CSS Utilities Tutorials to Your Main Tutorial Panel (Optional)"		x
"Step 7. Prepare the DDIO Files"	x	
"Step 8. Implement the Language Processor JCL (Required)"	x	
"Step 9. Initialize the zIIP Enablement Service (optional)"		
"Step 10. Continue with ECC Customization"		

Step 1. Implement the Security Exit Program (Optional)

The Security Exit program is an optional user-coded module for establishing security at each site. If you want to ensure full security access to file resources, refer to the “CSS Security Exit” chapter in the *Compuware Shared Services User/Reference Guide* for detailed information on using the program. A sample Security Exit program is located in SLCXCNTL sample library member CWASSECU.

If the Security Exit program is present, it is called for each member selected. The Security Exit program determines if the user command is permitted, and passes the appropriate return code. If the command is unacceptable for one of the selected members, it will not be executed at all.

Because CSS is used by more than one Compuware product, the Security Exit program is executed for each Compuware product that uses CSS. If you are using multiple Compuware products, you must update your Security Exit program to accommodate those products.

Notes:

1. The sample Security Exit program is for illustration purposes only and **does not** provide any useful functions. You must update it to serve the needs of your site.
2. This Security Exit program is invoked in the MVS environment and does not provide for CICS services.
3. If Abend-AID for CICS is installed at your site, or if you are installing Abend-AID for CICS, the Security Exit program must be reentrant.

To Implement the Security Exit Program:

Use the following procedure to implement the CWASSECU Security Exit program:

1. Create the Security Exit program. Use the sample Security Exit program as the basis for your exit.

You must use the program name CWASSECU because other programs reference it and search specifically for this program name.
2. Assemble and linkedit the Security Exit program as a separate load module named CWASSECU. Linkedit CWASSECU into the same load library as CSS. The SLCXCNTL library member CXJCLSEC contains sample JCL that can be used to assemble and linkedit CWASSECU to the CSS load library.
3. If CICS Abend-AID/FX or Abend-AID for CICS is installed at your site, the Security Exit program must be reentrant.

Step 2. Install Customized Translation Tables (Optional)

Note: If you omit this step, the default translation tables will be used.

You must perform this step if you are using Japanese language support with Abend-AID.

You may optionally perform this step if you have Abend-AID. When necessary, Abend-AID products dump storage in both hexadecimal and character representation. These storage areas will appear in one of two formats:

- Horizontal dump
- Vertical dump

The character representation for either format is controlled by a default translation table. The default character set is mixed-case English. You can override either, or both, of these default translation tables with customized translation tables. Different customized translation tables can be specified for either dump format. These customized tables may also allow display of certain otherwise non-displayable fields.

If you install vertical and/or horizontal customized translation tables, CWCMTVRT must be used as the load module name for the vertical translation table, and CWCMTRHT must be used as the load module name for the horizontal translation table. A different internal name can be defined to identify a table. The internal table name must be eight characters long. The table length, excluding the table name, must be 256 bytes. Therefore, the total table length is 264 bytes. If an error is made installing a customized translation table, the Compuware default table is used.

The following SLCXCNTL sample library members contain sample translation tables:

CWCMTVRT	Vertical translation table for uppercase English.
CWCMTRHT	Horizontal translation table for uppercase English.
CWCMTRVE	Vertical translation table for mixed-case English. This is the default vertical translation table that will be used if you choose not to install a customized vertical translation table.
CWCMTRHE	Horizontal translation table for mixed-case English. This is the default horizontal translation table that will be used if you choose not to install a customized horizontal translation table.
CWCMTRVU	Vertical translation table for mixed-case English with the Euro Character X'9F' included.
CWCMTRHU	Horizontal translation table for mixed-case English with the Euro character X'9F' included.

CSS SLCXCNTL sample library member CXJCLTRT contains the sample JCL to assemble and link-edit customized translation tables.

Step 3. Make New CSS Load Modules Accessible to Compuware Products (Required)

You must perform this step.

Ensure that the new CSS load modules can be accessed by Compuware products by using one or more of the following methods. Regardless of the method chosen, the CSS load library (SLCXLOAD or a copy of it) should be concatenated **in front** of any Compuware product library(ies). The method you choose will depend on the Compuware products you have at your site.

- Compuware recommends that you place the CSS load library in the link list. This will provide access to the CSS load modules for TSO users and batch jobs that need CSS (for example, language processors, CSS utilities, and/or Abend-AID). This also minimizes any JCL changes that might be needed for a CSS upgrade. (A refresh of the link list is required to activate this change.) Refer to “APF Authorization” on page 2-12 for additional information and cautions about APF authorizations.
- If you have a supported release of Xpediter/CICS installed at your site, you are required to include the CSS load library with the Xpediter/CICS library in the CICS startup JCL or RDO definitions for each CICS region.
- You may also place the CSS load library in the STEPLIB or ISPLLIB for individual TSO logon PROCs, and in the STEPLIB or JOBLIB for batch jobs. These library specifications will override the CSS load library specified in the link list. This is useful for testing a new version of CSS prior to placing it into the link list for production use.

Notes:

1. If you have previously installed CSS and referenced the load library in batch jobs, compile PROCs, CLISTs, or logon PROCs, CSS will be loaded from the STEPLIB, ISPLLIB, or JOBLIB where it is specified rather than from the link list. Compuware recommends that you review these locations and remove or modify the CSS load library DDs as appropriate. If you have any Compuware product that uses CSS from a STEPLIB, ISPLLIB, or JOBLIB concatenation, you should include the CSS load library in front of the Compuware product in that library concatenation.

Beginning with CSS 6.0, you have the ability to dynamically allocate the CSS load library as part of the ISPLLIB concatenation. Please refer to Option 1 listed in the next step for more information.

2. For Abend-AID for CICS, specify the CSS, and HCI load library names in the server JCL. Refer to the *Abend-AID for CICS Installation and Customization Guide* for more information.

3.

Step 4. Create the Compile Information Table (Required)

You should perform this step if you plan to use the CSS Utilities or the Eclipse Compile Facility within Host Explorer (see *Compuware Workbench Installation Guide*).

The purpose of this step is to create the compile information table, which is stored in SLCXTABL. This table contains information about your compilers, compiler libraries, DB2 pre-compile libraries, CICS translator libraries and some miscellaneous items. A series of ISPF dialog panels will take you through the table creation process. When you have finished creating the table, you will have to make it available to the CSS Utilities. See “Step 5. Activate Compuware ISPF Dialogs (Required)” on page 5-15.

For the Eclipse Compile Facility, the compile information table is accessed to populate the Preprocessor Compile Libraries dialog when the Import button is selected (see Host Explorer product Help in the Compuware Workbench).

Create the Compile Information Table:

1. You must edit the CXUECIN2 member of the SLCXCNTL sample library. To avoid having your modifications overlaid by future maintenance, you may want to copy the member to a non-SMP/E controlled library that is in your SYSPROC concatenation. Follow the instructions that are in the member. You will need the name of your CSS panel library, SLCXPENU, and your CSS REXX EXEC/CLIST library, SLCXEXEC.
2. If you placed the member, CXUECIN2, in a library that is in your SYSPROC concatenation, then you may invoke the REXX EXEC as follows: TSO %CXUECIN2. Otherwise, you may invoke it as follows: TSO EXEC 'your.dataset.library(CXUECIN2)'.

The first panel displayed is the "Install Menu". See Figure 5-1below.

Figure 5-1. Compile Information Table - Install Menu

```

Compuware CSS Utilities ----- Install Menu -----
COMMAND ==>

Enter the name of the library, which is used to store the compile information
that you supply from the options below. If the library does not exist, it
will be allocated.
==>

If you have already supplied the compile information in a prior installation
and you want to copy that information to a new library, then enter the
'old' library name below. Use the 'C' option to copy from the library named
below to the library named above.
==>

Enter an option on the command line.
1.) Select compilers
2.) Enter compiler libraries
3.) Enter DB2 IDs and precompile libraries (optional)
4.) Enter CICS IDs and translator libraries (optional)
5.) Enter miscellaneous information
A.) Process options 1 through 5
C.) Copy a prior installation's information to a new library

Press ENTER to process or enter END command to terminate

```

The first field is the name of the Compile Information table. All of the information entered will be saved in this table. If the file does not exist, then it will be allocated for you.

The second field is used during the COPY command. See "C. Copy a Prior Installation's Information to a New Library" on page 5-12.

Options:

1. Select Compilers

This option displays the available compilers that the Compile Facility supports. Place a Y next to each compiler that your site uses and you want your programmers to be able to use with the Compile Facility. You must enter this option first and select at least one compiler. See Figure 5-2 on page 5-6 and Figure 5-3 on page 5-6 below.

Figure 5-2. Compile Information Table - Select Compilers (1 of 2)

```

Compuware CSS Utilities ----- Select Compilers -----
COMMAND ==>
  Use 'Page UP' and 'Page DOWN' to scroll this screen.
                                                    More:    +

COBOL:
OS/VS COBOL           ==> N (Y/N)
VS COBOL II           ==> N (Y/N)
COBOL/370             ==> N (Y/N)
COBOL/MVS             ==> N (Y/N)
COBOL/390             ==> N (Y/N)
ENTERPRISE/COBOL     ==> N (Y/N)
COBOL5                ==> N (Y/N)
CA OPTIMIZER         ==> N (Y/N)
  5.1 DDNames        ==> N (Y/N)

Assembler:
Assembler H           ==> N (Y/N)
High Level Assembler ==> N (Y/N)

PL/I:
PL/I                 ==> N (Y/N)
PLI/370              ==> N (Y/N)

Press ENTER to process or enter END command to terminate
  
```

Figure 5-3. Compile Information Table - Select Compilers (2 of 2)

```

Compuware CSS Utilities ----- Select Compilers -----
COMMAND ==>
  Use 'Page UP' and 'Page DOWN' to scroll this screen.
                                                    More:    -

CA OPTIMIZER         ==> N (Y/N)
  5.1 DDNames        ==> N (Y/N)

Assembler:
Assembler H           ==> N (Y/N)
High Level Assembler ==> N (Y/N)

PL/I:
PL/I                 ==> N (Y/N)
PLI/370              ==> N (Y/N)
VA/PLI               ==> N (Y/N)
ENTERPRISE/PLI       ==> N (Y/N)

C Language:
C/MVS                ==> N (Y/N)
C/390                ==> N (Y/N)
C/ZOS                ==> N (Y/N)

Press ENTER to process or enter END command to terminate
  
```

Note: On the Compile Information Table screen shown in the Figure 5-2 on page 5-6, enter **YES** in the **5.1 DDNames** field if you are using CA OPTIMIZER and the release 5.1 DDNames SYSUT7, SYSUT8, and CAISTATS are to be allocated.

2. Enter Compiler Libraries

This option allows you to supply the library names for the compiler you selected in Option 1. See Figure 5-4 on page 5-7 through Figure 5-7 on page 5-8. The compilers that you selected will have their corresponding arrow indicators highlighted and the field names will be modifiable.

Figure 5-4. Compile Information Table - Enter Compiler Libraries (1 of 4)

```

Compuware CSS Utilities ----- Compiler Libraries -----
COMMAND ==>
  Use 'Page UP' and 'Page DOWN' to scroll this screen.
                                                    More:   +

COBOL Libraries:
OS/VS COBOL           ==>
Linkedit SYSLIB      ==>

VS COBOL II          ==>
Linkedit SYSLIB      ==>

COBOL/370            ==>
Linkedit SYSLIB      ==>

COBOL/MVS             ==>
Linkedit SYSLIB      ==>

COBOL/390            ==>
Linkedit SYSLIB      ==>

ENTERPRISE/COBOL     ==>
Linkedit SYSLIB      ==>

  Press ENTER to process or enter END command to terminate
  
```

Figure 5-5. Compile Information Table - Enter Compiler Libraries (2 of 4)

```

Compuware CSS Utilities ----- Compiler Libraries -----
COMMAND ==>
  Use 'Page UP' and 'Page DOWN' to scroll this screen.
                                                    More:   - +

Linkedit SYSLIB      ==>

CA OPTIMIZER         ==>
Linkedit SYSLIB      ==>

ASSEMBLER Libraries:
Assembler H          ==>

High Level Assembler ==>

PL/I Libraries:
PL/I                 ==>
Linkedit SYSLIB      ==>
Linkedit SYSLIB      ==>

PLI/370              ==>
Linkedit SYSLIB      ==>

  Press ENTER to process or enter END command to terminate
  
```

Figure 5-6. Compile Information Table - Enter Compiler Libraries (3 of 4)

```

Compuware CSS Utilities ----- Compiler Libraries -----
COMMAND ==>
  Use 'Page UP' and 'Page DOWN' to scroll this screen.
                                                    More:  - +

  Linkedit SYSLIB      ==>

  VA/PLI               ==>
  Run-time LIBRARY    ==>
  Linkedit SYSLIB     ==>

  ENTERPRISE/PLI     ==>
  Run-time LIBRARY    ==>
  Linkedit SYSLIB     ==>

  C Language Libraries:
  C/MVS               ==>
  LE LIBs             ==>
  Compiler SYMSGSGS   ==>
  Linkedit SYSLIB     ==>

  C/390               ==>
  LE LIBs             ==>

  Press ENTER to process or enter END command to terminate

```

Figure 5-7. Compile Information Table - Enter Compiler Libraries (4 of 4)

```

Compuware CSS Utilities ----- Compiler Libraries -----
COMMAND ==>
  Use 'Page UP' and 'Page DOWN' to scroll this screen.
                                                    More:  -

  Linkedit SYSLIB      ==>

  C Language Libraries:
  C/MVS               ==>
  LE LIBs             ==>
  Compiler SYMSGSGS   ==>
  Linkedit SYSLIB     ==>

  C/390               ==>
  LE LIBs             ==>
  Compiler SYMSGSGS   ==>
  Linkedit SYSLIB     ==>

  C/ZOS               ==>
  LE LIBs             ==>
  Compiler SYMSGSGS   ==>
  Linkedit SYSLIB     ==>

  Press ENTER to process or enter END command to terminate

```

Libraries entered on these screens depend on the compile processors used at your site. The following types of libraries can be entered on this screen:

- Compiler libraries
- SYSLIB libraries
- Compiler dependent libraries

Compiler libraries describe the datasets that contain the compile processor load modules. These libraries are required only if the processor libraries are not already included in the concatenation lists for LPA or LINKLIST. When these libraries are supplied they will appear in the STEPLIB DD of the compile JCL.

The SYSLIB libraries describe the datasets to be used for the primary resolution of external references by the linkage editor or binder. For example, the L/E library, SCEELKED. When these libraries are supplied, they will appear in the SYSLIB DD of the linkedit step.

Compiler dependent libraries are for those compilers that have additional library requirements. For example, the C language SYSMSGS library or the L/E library, SCEERUN.

3. Enter DB2 IDs and Precompile Libraries (Optional)

If your site has DB2, this option will allow you to define DB2 identifiers and associate the identifiers with DB2 libraries. See Figure 5-8 and Figure 5-9 below. When an identifier is selected, the associated datasets will appear in the STEPLIB DD of the precompile step and in the SYSLIB DD of the linkedit step. The DB2 IDs will appear on the Compile Menu. See Figure 5-13 on page 5-13.

Figure 5-8. Compile Information Table - Enter DB2 IDs and Precompile Libraries (1 of 2)

```

Compuware CSS Utilities --- DB2 Precompile Libraries -----
COMMAND ==>
  Use 'Page UP' and 'Page DOWN' to scroll this screen.
                                     More:      +
  These libraries are used in the DB2 precompile step.  The ID is displayed
  on the 'Compile Menu'.  When the programmer selects one of the IDs from
  the 'Compile Menu', the ID is matched to the corresponding library name(s)
  that are listed here.  You may have up to two libraries with the same ID
  name.  The 'Compile Menu' can display up to 8 unique IDs.

      ID                      DB2 Precompile Libraries
(1) ==>                      ==>
(2) ==>                      ==>
(3) ==>                      ==>
(4) ==>                      ==>
(5) ==>                      ==>
(6) ==>                      ==>
(7) ==>                      ==>
(8) ==>                      ==>
(9) ==>                      ==>
(10) ==>                     ==>
(11) ==>                     ==>

      Press ENTER to process  or  enter END command to terminate
  
```

Figure 5-9. Compile Information Table - Enter DB2 IDs and Precompile Libraries (2 of 2)

```

Compuware CSS Utilities --- DB2 Precompile Libraries -----
COMMAND ==>
  Use 'Page UP' and 'Page DOWN' to scroll this screen.
                                     More:      -
      ID                      DB2 Precompile Libraries
(1) ==>                      ==>
(2) ==>                      ==>
(3) ==>                      ==>
(4) ==>                      ==>
(5) ==>                      ==>
(6) ==>                      ==>
(7) ==>                      ==>
(8) ==>                      ==>
(9) ==>                      ==>
(10) ==>                     ==>
(11) ==>                     ==>
(12) ==>                     ==>
(13) ==>                     ==>
(14) ==>                     ==>
(15) ==>                     ==>
(16) ==>                     ==>

      Press ENTER to process  or  enter END command to terminate
  
```

In the column on the left, type in a DB2 identifier. It can be from one to six characters in length. You may have up to two identifiers with the same name. In addition, you may have up to eight unique DB2 identifiers.

In the column on the right, type in the DB2 dataset names that are required to perform the DB2 precompile step.

For Example:

	ID	DB2 Precompile Libraries
(1) ==>	D610S1	'SYS1.DSN610.SDSNLOAD'
(2) ==>	D610S1	'SYS1.DSN610.SDSNEXIT'
(3) ==>	D710S1	'SYS1.DSN710.SDSNLOAD'
(4) ==>	D710S1	'SYS1.DSN710.SDSNEXIT'
(5) ==>	D810S1	'SYS1.DSN810.SDSNLOAD'
(6) ==>	D810S1	'SYS1.DSN810.SDSNEXIT'

There are two D610S1 identifiers. That is the maximum for a single identifier. Also, there are three unique identifiers D610S1, D710S1, and D810S1. You may have up to eight of these.

4. Enter CICS IDs and Translator Libraries (Optional)

If your site has CICS, this option will allow you to define CICS identifiers and associate the identifiers with CICS libraries. See Figure 5-10 and Figure 5-11 below. When an identifier is selected, the associated datasets will appear in the STEPLIB DD of the translator step and in the SYSLIB DD of the linkedit step. The CICS IDs will appear on the Compile Menu. See Figure 5-13 on page 5-13.

Figure 5-10. Compile Information Table - Enter CICS IDs and Translator Libraries (1 of 2)

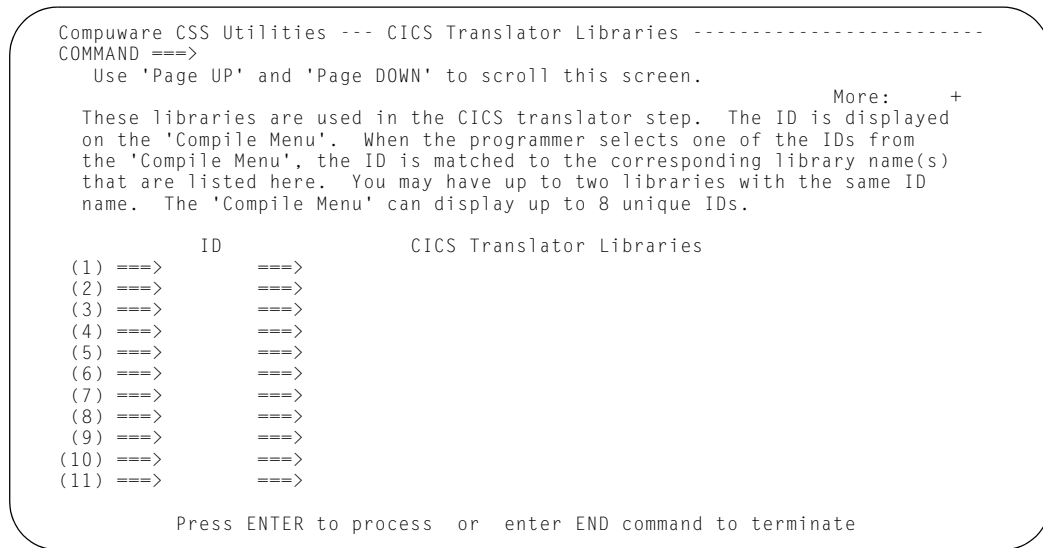


Figure 5-11. Compile Information Table - Enter CICS IDs and Translator Libraries (2 of 2)

```

Compuware CSS Utilities --- CICS Translator Libraries -----
COMMAND ==>
Use 'Page UP' and 'Page DOWN' to scroll this screen.
More: -

          ID                      CICS Translator Libraries
(1) ==>          ==>
(2) ==>          ==>
(3) ==>          ==>
(4) ==>          ==>
(5) ==>          ==>
(6) ==>          ==>
(7) ==>          ==>
(8) ==>          ==>
(9) ==>          ==>
(10) ==>         ==>
(11) ==>         ==>
(12) ==>         ==>
(13) ==>         ==>
(14) ==>         ==>
(15) ==>         ==>
(16) ==>         ==>

Press ENTER to process or enter END command to terminate

```

In the column on the left, type in a CICS identifier. It can be from one to six characters in length. You may have up to two identifiers with the same name. In addition, you may have up to eight unique CICS identifiers.

In the column on the right, type in the CICS dataset names that are required to perform the CICS translator step.

For example:

```

          ID                      CICS Translator Libraries
(1) ==> CTS310 ==> 'SYS1.CICS310.LOADLIB1'
(2) ==> CTS310 ==> 'SYS1.CICS310.LOADLIB2'
(3) ==> CTS220 ==> 'SYS1.CICS220.LOADLIB1'
(4) ==> CTS220 ==> 'SYS1.CICS220.LOADLIB2'
(5) ==> CTS210 ==> 'SYS1.CICS210.LOADLIB1'
(6) ==> CTS210 ==> 'SYS1.CICS210.LOADLIB2'

```

There are two CTS310 identifiers. That is the maximum for a single identifier. Also, there are three unique identifiers CTS310, CTS220, and CTS210. You may have up to eight of these.

5. Enter Miscellaneous Information

There are some options that don't fall under the previously mentioned options. Those items are found on this screen. See Figure 5-12 on page 5-12.

Figure 5-12. Compile Information Table - Enter Miscellaneous Information

```

Compuware CSS Utilities ---- Miscellaneous Settings -----
COMMAND ==>

To prohibit foreground compiles, set the following value to 'N'.
Foreground Compile ==> Y

Job Statement Exit: (Executed before batch job submission)
Exit ==>
The exit syntax uses the ISPF SELECT service syntax.
For example:
Exit ==> CMD(command) or CMD(command) NEWAPPL(applid) PASSLIB

Press ENTER to process or enter END command to terminate

```

Foreground Compile: Some sites do not want their staff doing foreground compiles. If you do not want your users doing foreground compiles, then set this field to N. The Foreground literal will not be displayed as one of the Preparation options. See Figure 5-14 on page 5-14.

Job Statement Exit: This is an exit point in which the JOB card is passed to the stated entity before job submission for the Batch option. The invocation of the exit is also done before the JCL is displayed with the Editjcl option.

The invocation of the exit uses the ISPF SELECT service syntax.

The following are examples:

```
Exit ==> PANEL(panelid) or PANEL(panelid) NEWAPPL(applid) PASSLIB
```

```
Exit ==> CMD(command) or CMD(command) NEWAPPL(applid) PASSLIB
```

```
Exit ==> PGM(program-name) or PGM(program-name) PARM(parameters) or
```

```
PGM(program-name) PARM(parameters) NEWAPPL(applid) PASSLIB
```

Note: *Batch*, *Editjcl* and *Foreground* refer to Preparation types on the Compile Menu. See Figure 5-13 on page 5-13.

A. Process Options 1 through 5

Entering this option on the command line will cause options 1 through 5 to be displayed sequentially.

C. Copy a Prior Installation's Information to a New Library

The Copy process will take an existing compile information table with member AAUTCTBL and copy the member to a new compile information table. If the new table already has member AAUTCTBL, a message window appears asking if you want to overwrite the existing member or bypass the update.

This process may be used to create a back-up of your completed compile information table. Also, it may be used to carry existing compile information forward to a new release. Thus, avoiding having to retype the compile information.

Place a C on the command line. In the first field, enter the dataset name that you want to contain the compile information. In the second field, enter the

dataset name of an existing compile information table. Hit enter to copy the information.

Later on during the CSS installation, you will need the dataset name of the compile information table.

The Compile Menu is displayed below to assist you in your understanding of how the values that you supply during this installation will affect the display of the Compile Menu.

Figure 5-13. Compile Facility - Compile Menu

```

Compuware CSS Utilities ----- Compile Menu ----- CSS V8.6.0
COMMAND ==>

Primary Commands: Listing (Display output) SSetup (Display general settings)

Compile Profile: DEFAULT - DEFAULT COMPILE PROFILE
                  _ Display the Compile Profile list

Source Dsname ==> 'USERID.COB.SOURCE'
Preparation ==> BATCH      (Batch/Editjcl/Foreground)          <-Opt 5
Language ==> E-COBOL      Select From The List Below          <-Opt 2
(COB/COB2/COB370/COBMVS/COB390/E-COBOL/CAOPT/HASM/HLASM/PLI/PLI370) <-Opt 1
(VAPLI/E-PLI/CMVS/C390/CZOS)

SEL Options:
D - Display settings          S - Select the DB2 Precompile libraries
S - Process only              and/or the CICS Translator libraries
SEL      STEPS                DB2 Precompile          CICS Translator
-----
- 1. DB2 Precompile          - D810S1 - D710S1          - CTS31A - CTS31B <-Opt 3&4
- 2. CICS Translator         - D610S1 - D810S2          - CTS31C - CTS21A
- 3. Compile                 - D710S2 - D610S2          - CTS21B - CTS21C
- 4. Linkedit                - D510S1 - D510S2          - CTS11A - CTS11B

```

Note: The *option* references refer to the option number on the Install Menu screen.

If you select all of the compilers, this is the list that is generated (option 1). The user will select the language he wants to use for his compile by placing one of the names from the list in the Language field. The name entered tells the compile facility which libraries to use during the compile (option 2).

If you used eight unique DB2 identifiers and libraries, then the DB2 Precompile list will be similar to this list (option 3). If you supplied less than eight unique identifiers, then they will appear in a left to right, top to bottom sequence.

If you used eight unique CICS identifiers and libraries, then the CICS Translator list will be similar to this list (option 4). If you supplied less than eight unique identifiers, then they will appear in a left to right, top to bottom sequence.

If you set the Foreground Compile value to Y, then the Foreground literal appears in the Preparation list (option 5). If you set the value to N, then the literal does not appear.

The following figure shows the Compile Menu with fewer values supplied during installation. In addition, the Foreground Compile option is set to N.

Step 5. Activate Compuware ISPF Dialogs (Required)

You should perform this step if you plan to use CSS Utilities.

Compuware ISPF dialogs include CSS Utilities. This section covers two different ways to activate the CSS Utilities libraries—dynamically, or at TSO startup.

Dynamically allocating the libraries is discussed in Option 1 below. Allocating the libraries at TSO startup is discussed in “Option 2 — Allocate the Libraries at TSO Startup” on page 5-16. For more information about CSS Utilities functionality, see the *Compuware Shared Services User/Reference Guide*.

Option 1 — Dynamically Allocate Libraries when CSS Utilities are Executed

To dynamically allocate the CSS Utilities libraries, you must determine how to make the REXX EXECs and CLISTs accessible.

CSS provides sample REXX EXECs and CLISTs for invoking CSS Utilities in SLCXCNTL sample library. You can allocate this library at TSO logon time, or make the CLIST or REXX EXEC accessible in a user-defined library allocated at TSO logon.

IMPORTANT

If CSS REXX EXECs or CLISTs are currently used to invoke the CSS utilities, you must replace them with the latest versions in order to use the new Abend-AID features and Abend-AID for CICS utility features. With release 8.3, the Compile Facility has requirements for a new table library and a new skeleton library. Also with release 8.3, the CSS utilities have requirements for the Abend-AID for CICS load library when use with CICS type DDIO files is anticipated.

Complete the following steps:

1. Determine whether to use REXX EXECs or CLISTs.
Compuware recommends using REXX EXECs; you may find them easier to customize and maintain.
2. Determine whether Japanese and English language support is required, or just English language support.
3. If using REXX EXECs, choose the appropriate REXX EXEC for CSS Utilities using Table 5-2 as a guide. If using CLISTs, select one CLIST for CSS Utilities.
4. Change the library names in the REXX EXECs or CLISTs selected in Step 3 to match your site standards.

Note: Effective with CSS Release 8.1, an optional ISPSLIB can be used with On-the-Fly processing. See Appendix D in the *Compuware Shared Services User/Reference Guide* for details about On-the-Fly processing. You may use an existing skeleton library allocated at TSO logon time, or allocate one at viewer startup by updating the REXX EXECs or CLISTs selected in “Step 3. Make New CSS Load Modules Accessible to Compuware Products (Required)” on page 5-4.

Note: Effective with CSS Release 8.3, the CSS Online Utilities require the SLCXSLIB and SLCXTABL datasets to be concatenated to ISPSLIB and ISPTLIB respectively. The

SLCXTABL dataset was created in “Step 4. Create the Compile Information Table (Required)” on page 5-4.

Table 5-2. REXX EXECs or CLISTs Used to Invoke CSS utilities

Use this REXX EXEC	Or this CLIST	For
CWUTREXE	CWUTCLSE	CSS Utilities, English-only language support, set up or run XPEDITER utilities
CWUTREXJ	CWUTCLSJ	CSS Utilities, English and Japanese language support, set up or run XPEDITER utilities

5. Allocate the CSS SLCXCNTL library at TSO logon time, or copy the edited REXX EXECs/CLISTs to a library allocated at TSO logon:
 - **To allocate the CSS SLCXCNTL library at TSO logon**
Add the library to the SYSPROC concatenation in the TSO logon PROC.
 - **To make specific REXX EXECs/CLISTs accessible in a library allocated at TSO logon**
Copy the REXX EXECs or CLISTs to an existing SYSPROC library allocated at TSO logon.
6. Users can invoke the CSS Utilities by selecting an option from an ISPF panel (recommended), or by executing a REXX EXEC or CLIST.
 - **To enable users to select an ISPF menu option**
Add one or more of the following lines to an ISPF panel. Note that ‘o’ represents the option to be entered at the menu command prompt.

To invoke CSS Utilities	o,'CMD (%CSS Utilities CLIST name or CSS Utilities REXX EXEC name)
-------------------------	--

- **To invoke CSS Utilities by executing a REXX EXEC or CLIST**
Enter the appropriate command at the TSO command prompt:

To invoke CSS Utilities	TSO %CSS Utilities CLIST name or CSS Utilities REXX EXEC name
-------------------------	---

Option 2 — Allocate the Libraries at TSO Startup

Allocating the load, panel, and message libraries at TSO logon simplifies maintenance. All TSO users have access to CSS Utilities after logging on to TSO, and upgrades that affect all TSO users can be made in one central location. Allocating these libraries at TSO logon may be inappropriate, however. In that case, dynamically allocating the libraries may be a better solution.

To provide access to the load, panel, and message modules at TSO logon time, complete the following steps.

1. Concatenate each library as shown in Table 5-3:

Table 5-3. Allocating CSS Libraries

Load library SLCXLOAD	<p>Place the CSS load library in the link list (see “Step 3. Make New CSS Load Modules Accessible to Compuware Products (Required)” on page 5-4 for instructions)</p> <p>OR</p> <p>Place a DD statement for the load library in the concatenation for STEPLIB or ISPLLIB in your TSO logon PROC</p> <p>OR</p> <p>Execute a CLIST or REXX EXEC at logon time that will concatenate the load library to the appropriate DD statement before starting ISPF</p>
Load library SLHCLOAD HCI — required for Abend-AID or Abend-AID for CICS	<p>Place the HCI load library in the link list (see “Step 3. Make New CSS Load Modules Accessible to Compuware Products (Required)” on page 5-4 for instructions)</p> <p>OR</p> <p>Place a DD statement for the load library in the concatenation for STEPLIB or ISPLLIB in your TSO logon PROC</p> <p>OR</p> <p>Execute a CLIST or REXX EXEC at logon time that will concatenate the load library to the appropriate DD statement before starting ISPF</p> <p>Note: Abend-AID for CICS users must place a DD statement for the load library in the concatenation for FDBDRPL in your server startup JCL.</p>
Message library SLCXMENU	<p>Concatenate the library to ISPMLIB in the TSO logon PROC</p> <p>OR</p> <p>Concatenate the library to ISPMLIB in a CLIST or REXX EXEC before starting ISPF</p>
Panel library SLCXPENU	<p>Concatenate the library to ISPLLIB in the TSO logon PROC</p> <p>OR</p> <p>Concatenate the library to ISPLLIB in a CLIST or REXX EXEC before starting ISPF</p>
EXEC library SLCXEXEC	<p>Concatenate the library to SYSPROC in the TSO logon PROC</p> <p>OR</p> <p>Concatenate the library to SYSPROC in a CLIST or REXX EXEC before starting ISPF</p>
Skeleton Library SLCXSENU	<p>Due to the Compile Facility's requirement for this library, as of CSS release 8.3, this DD is required.</p> <p>Concatenate the library to ISPSLIB in the TSO logon PROC</p> <p>OR</p> <p>Concatenate the library to ISPSLIB in a CLIST or REXX EXEC before starting ISPF.</p> <p>Optional - concatenate the library for On-the-Fly processing. See Appendix D in the Compuware Shared Services User/Reference Guide for details.</p>
Table Library SLCXTABL	<p>Concatenate the library to ISPTLIB in the TSO logon PROC</p> <p>OR</p> <p>Concatenate the library to ISPTLIB in a CLIST or REXX EXEC before starting ISPF.</p> <p>Note: This file is created in “Step 4. Create the Compile Information Table (Required)” on page 5-4.</p>

Notes:

- a. Replace any existing CSS libraries in these concatenations.
 - b. The low level qualifiers are listed below their respective library names.
 - c. When testing a new install of Abend-AID, if the new Abend-AID load and customization libraries have not yet been placed in the link list, they must be added as explained in the instructions for the SLCXLOAD library. Doing this will prevent text merge error messages when viewing an Abend-AID report.
 - d. When testing a new install of Abend-AID for CICS, if the new Abend-AID for CICS load library has not yet been placed in the link list, it must be added as explained in the instructions for the SLCXLOAD library. Doing this will prevent error messages when using the CSS Utilities with Abend-AID for CICS DDIO files.
2. Add one or more of the following lines to an ISPF panel. Note that 'o' represents the option to be entered at the ISPF menu command prompt.

To invoke CSS Utilities	o,'PGM(CWDDIUFE) NEWAPPL(AAUT)'
-------------------------	---------------------------------

Step 6. Link CSS Utilities Tutorials to Your Main Tutorial Panel (Optional)

CSS SLCXCNTL sample library contains two sample tutorial main menu panels for ISPF and ISPF/PDF:

- CXTUTOR
- CXR00003

To link the CSS Utilities tutorial panels to your tutorial main menu panel (TTUTOR or ISR00003), use the applicable member as an example.

Step 7. Prepare the DDIO Files

You must perform this step if you are a new CSS user or have never created DDIO files and now need to use them. If you are an Abend-AID user you will need to allocate a Report Shared Directory and one or more Report databases.

CSS DDIO files are used to store Abend-AID reports, transaction dump information, and source listings. The term *DDIO file* is a generic term used to refer to the datasets that store the reports and listings from Compuware Products that use CSS. This step describes the various types of DDIO files available and who needs to allocate which type.

Types of DDIO Files

There are two main categories of DDIO files: report or source listing files and report or source listing *databases* which are grouped into pools that are attached to Shared Directories. CSS uses the Shared Directory architecture to manage the available pool space.

Within these two categories of DDIO files, there are three different types of DDIO files characterized by the type of information that is stored in them: Abend-AID reports, source listings, and CICS transaction dump information.

Note: If you use the Shared Directory architecture for your DDIO files, different database types cannot be attached to the same Shared Directory (for example, you cannot attach a source listing database to a report shared directory, and vice versa). Starting with CSS release 8.3, you can now attach transaction databases and report databases to the same Shared Directory.

Note: Non-database report or source listing files cannot be attached to a Shared Directory until they have been converted using the batch utility CONVERT command in CWDDALLU, or CWAASDUT (report only) and CWDDLPUT (source listing only). See the "Batch File Utility" chapters in the *Compuware Shared Services User/Reference Guide* for more information, specifically Chapter 8 "Batch File Utility CWDDALLU" for detailed command and parameter descriptions.

1. Abend Reports:
 - a. **Report Shared Directories and Databases.** This type of file is used to store abend reports created by Abend-AID. *This Report Shared Directory architecture is only supported for Abend-AID Release 9.4 and later.* Report databases are attached to a report shared directory. The report databases are managed as a POOL of space belonging to the report shared directory to which they are attached.
 - b. **Report DDIO files.** This type of DDIO file is used to store Abend Reports created by Abend-AID. Prior to Abend-AID Release 9.4, this was the only type of report file available. For compatibility purposes, Abend-AID Release 9.4 can also use this type of report file. A conversion utility is available to run when the user is ready to migrate the report file to the report database format.
2. Source Listings:
 - a. **Source Listing Shared Directories and Databases.** This type of DDIO file is used to store compiler source listings for use by various Compuware Products. Source listing databases are attached to a source listing shared directory. The source listing databases are managed as a POOL of space belonging to the source listing shared directory to which they are attached. *The Source Listing Shared Directory architecture can now be used with all supported language compilers. However, XPEDITER users should see notes and exceptions below.*
 - b. **Source Listing DDIO Files.** This type of DDIO file is used to store compiler source listings. A conversion utility is available to run when the user is ready to migrate the listing file to the source listing database format.

The same source listing file, database, or source listing member can be used by multiple Compuware Products. It is not necessary to create duplicate source listing files, databases, or source listing members for each Compuware Product.

3. **Transaction Databases and Shared Directory for Abend-AID for CICS.** Transaction databases are used by Abend-AID for CICS to store CICS transaction dump information. Abend-AID for CICS catalogs both CICS transaction dumps and imported Region Dumps into the shared directory.

The following table lists the types of DDIO files created by the CSS utilities and which products use/require them:

Table 5-4. DDIO File Formats

Compuware Product	Report File	Source Listing File	Transaction Databases & Shared Dir.	Report Shared Dir. & Database	Source Listing Shared Dir.	Source Listing Database
Abend-AID	x	x		x	x	x
Abend-AID for CICS		x	x		x	x
Xpediter/TSO		x			x ¹ See Note	x
Xpediter/IMS		x			x ¹ See Note	x
Xpediter/CICS		x			See Note	x
Xpediter/Eclipse		x			See Note	x
Strobe		x			x	x

¹ If using Xpediter/TSO and /IMS with the C Language Processor, VisualAge PL/I, or Enterprise PL/I, and you use Long Program Names, the Source Listing Shared Directory must be used.

Refer to the “Source Listing Databases and Shared Directories” section of Chapter 2 in the *Compuware Shared Services User/Reference Guide* for more information.

File Access Methods

You may use either sequential or VSAM files for DDIO files. Shared directories must be VSAM files. If you are using VSAM files for your DDIO files, you can copy them using the IDCAMS REPRO command. You cannot, however, change any of the file attributes such as CISIZE without reallocating a DDIO/DB file. Allocation and format requirements, along with the steps for preparing the various DDIO files, will vary according to the type of file. Source listing files can be shared between all Compuware products that use CSS, while report files, transaction databases, work files, and profile files, cannot. Source listing files used by XPEDITER/CICS must be VSAM.

CAUTION:

CWDDSUTL, CWFXSDUT, CWAASDUT, CWDDLPUT, and CWDDALLU are the only utilities recommended for use with Compuware DDIO files.

File-AID/MVS and other VSAM-related vendor products (including vendor packages that compress or enhance buffer management on VSAM files) should NOT be used against Compuware DDIO files. Doing so can result in corrupting the contents of the DDIO file. Refer to “Chapter 2. Allocating and Formatting DDIO Files” in the *Compuware Shared Services User/Reference Guide* for more information.

Step 7a. Allocate and Format Abend-AID Report Shared Directories and Databases for Abend-AID Release 9.4 and Later Users

This step should only be performed by Abend-AID users who have installed, are installing, or are migrating to Abend-AID Release 9.4 or later. The purpose of this step is to allocate and format a Report Shared Directory and one or more Report Databases attached to that Shared Directory. Note that Abend-AID 9.4 is the minimum release able to use these types of DDIO file.

Typically, this step would be performed during Abend-AID release 9.4 or later installation. For more information, see the *Abend-AID Installation and Customization Guide*.

The CSS online utilities can be used to allocate and format Abend-AID report DDIO files. Refer to "Chapter 3. CSS Utilities" in the *Compuware Shared Services User/Reference Guide* for more information.

Create the Report Shared Directory

Use the sample JCL in the CSS SLCXCNTL dataset to allocate and format the Report Shared Directory (member is CXALLDAA). For additional options refer to the "Batch File Utility CWDDALLU" chapter in the *Compuware Shared Services User/Reference Guide*. The shared directory CREATE command will both ALLOCATE and FORMAT the shared directory. If the file currently exists, you must specify parameter REALLOCATE=YES or the allocate will fail with a duplicate file error.

Create the Report Databases

You must have already created the Report Shared Directory before attempting to create the Report Database. You **can** create both in the same job as long as the Report Shared Directory create is done first. The database CREATE command will both ALLOCATE and FORMAT the database. If the file currently exists, you must specify parameter REALLOCATE=YES or the allocate will fail with a duplicate file error.

VSAM Report Databases can be created using the sample JCL in the CSS SLCXCNTL dataset (member is CXALLVAA). To allocate and create a database that spans multi-volumes, use member CXDCRADB. To create a report shared directory in the same job, use member CXDCRASD.

Sequential Report Databases can be created using the sample JCL in the CSS SLCXCNTL dataset (member is CXALLBAA).

Step 7b. Allocate and Format Source Listing Shared Directories and Databases

This step may be performed by all users who use Source Listing support. However, releases prior to XPEDITER/TSO 7.3, XPEDITER/CICS 7.6, and Abend-AID for CICS 4.5 do not support Source Listing Shared Directories. Instead, individual source databases may be specified for lower levels. Source Listing Databases used by XPEDITER/CICS must be VSAM.

The CSS online utilities can be used to allocate and format source listing DDIO files. Refer to "Chapter 3. CSS Utilities" in the *Compuware Shared Services User/Reference Guide* for more information.

Create the Source Listing Shared Directory.

Use the sample JCL in the CSS SLCXCNTL dataset to allocate and format the Source Listing Shared Directory (member is CXALLDLP). For additional options refer to the "Batch File Utility CWDDALLU" chapter in the *Compuware Shared Services User/Reference Guide*. The shared directory CREATE command will both ALLOCATE and FORMAT the

shared directory. If the file currently exists, you must specify parameter REALLOCATE=YES or the allocate will fail with a duplicate file error.

Create the Source Listing Databases

You must have already created the Source Listing Shared Directory before attempting to create the Source Listing Database. You **can** create both in the same job as long as the Source Listing Shared Directory create is done first. The database CREATE command will both ALLOCATE and FORMAT the database. If the file currently exists, you must specify parameter REALLOCATE=YES or the allocate will fail with a duplicate file error.

VSAM Source Listing Databases can be created using the sample JCL in the CSS SLCXCNTL dataset (member is CXALLVLP). To allocate and create a database that spans multi-volumes, use member CXDCRLDB. To create a source shared directory in the same job, use member CXDCRLSD.

Sequential Source Listing Databases can be created using the sample JCL in the CSS SLCXCNTL dataset (member is CXALLBLP).

Step 7c. Allocate and Format Source Listing Files

You must perform this step if you did **not** perform “Step 7b. Allocate and Format Source Listing Shared Directories and Databases” **and** you do not have an existing source listing file or database. Source listing files used by XPEDITER/CICS must be VSAM.

Allocate the Source Listing Files

VSAM Source Listing files can be allocated using the sample JCL in the CSS SLCXCNTL dataset member CXALLVS. To allocate and format a source DDIO file that spans multi-volumes, use member CXDFMSL).

Sequential Source Listing files can be allocated using the sample JCL in the CSS SLCXCNTL dataset member CXALLDS.

Format the Source Listing Files

After the source listing files have been allocated, they must be formatted before they can be used. The formatting process initializes the directory structure and specifies the file options.

CSS SLCXCNTL library member CXFMTSL contains sample JCL to format a source listing file with appropriate defaults for the sample files allocated using members CXALLVS or CXALLDS. Adjust the formatting parameters to meet your DDIO file needs. Do not execute the CXFMTSL member, if you have already run the CXDFMSL job.

Step 7d. Allocate and Format Abend-AID for CICS Shared Directories and Transaction Databases

Normally this would be performed during the Abend-AID for CICS install procedure. For additional information, see the *Abend-AID for CICS Installation and Customization Guide*.

Create Abend-AID for CICS Shared Directory (Allocate & Format)

Use the sample JCL to create the shared directory. CSS SLCXCNTL sample library member CXALLMC contains the sample JCL for creating a Abend-AID for CICS shared directory.

Create Abend-AID for CICS Transaction Database (Allocate & Format)

You must create the shared directory before proceeding to this step. Use one of the following two sets of sample JCL to create each Abend-AID for CICS database as needed:

VSAM Abend-AID for CICS Database: CSS SLCXCNTL sample library member CXALLVSD contains sample JCL for creating a Abend-AID for CICS VSAM transaction database. To allocate and create a database that spans multi-volumes, use member CXDCRTDB. To create an Abend-AID for CICS shared directory in the same job, use member CXDCRTSD.

Sequential Abend-AID for CICS Database: CSS SLCXCNTL sample library member CXALLBSD contains sample JCL for creating a Abend-AID for CICS sequential transaction database.

Step 8. Implement the Language Processor JCL (Required)

See the sub-steps below to determine when you must perform this step.

The language processors support two methods of processing your programs: preprocessing and postprocessing. Both of these choices may involve modifications to your compile JCL.

You may use the preprocessor or post-processor (whichever is best for your site as needed.) This section discusses the benefits of the preprocessor and the post-processor and describes when to use them.

Refer to “CSS Language Processor (LP)” on page 2-4 for descriptions of preprocessing and postprocessing and a summary of the benefits provided by each method.

To use the language processor, you must have previously completed installation of the CSS load library and also prepared a DDIO file. For each language you are using, you should modify the corresponding compiler procedure to provide for the type of processing you will be implementing.

When running the language processor as a preprocessor, you can specify assembler/compiler options in the CWPPRMO dataset or in the PARM statement passed to the preprocessor. The CWPPRMO is an input dataset that contains the language processor compiler options. Depending on the method used, it may be necessary to include the LANGPARM command with these options.

If you are currently using the post-processor, and you will be converting to the preprocessor, you are not required to spool your SYSPRINT to a temporary dataset.

Note: For each of the following steps, it is possible that you will receive a return code other than zero. Should this occur, note the accompanying language processor error message and refer to the *Enterprise Common Components Messages and Codes* guide for further information.

Step 8a. Modify the JCL to Run the COBOL Language Processor

You must perform this step if either of the following is true:

- You are installing XPEDITER/CICS for the first time.
- You are installing Abend-AID for the first time or you are upgrading a current Abend-AID installation with Abend-AID source support **and** you are licensed for the Compuware COBOL language processor.

You may optionally perform this step if you are installing Abend-AID for CICS or XPEDITER/TSO **and** you are licensed for the Compuware COBOL language processor. XPEDITER/TSO customers can either perform this step now in batch mode, or online as part of the XPEDITER/TSO installation procedure.

This step contains instructions for both the COBOL language preprocessor and post-processor. You should follow the instructions that apply to the type of processing you will be performing.

Preprocessor

SLXCNTL sample library member CXCOBPRES contains sample JCL necessary to run the COBOL language preprocessor.

Modify the compile step of your COBOL JCL:

- On the EXEC statement, change the name of your compiler to CWPCMAIN and add the following to any existing COBOL compiler options:

```
LANGUAGE(compiler version)
```

Notes:

- The LANGUAGE command may be specified in the CWPPRMO command stream rather than on the EXEC statement.
 - The preprocessor accepts VSCOBOLII, VSCOBOLIIREL3, VSCOBOLIIREL4, COBOL/370, COBOL/390, COBOL/MVS, and COBOLZ/OS as being equivalent because the program name is the same. The compiler that is actually executed depends on the library specified in the STEPLIB, JOBLIB, and/or LINKLIST concatenation.
- Add the CSS SLCXLOAD DSN to the STEPLIB or JOBLIB concatenation if it is not in the link list.
 - Add the CWPDDIO DD statement to specify the name of the source listing DDIO file you will be using. The source listing DDIO file may be in standard DDIO, shared directory, or database format. Releases prior to XPEDITER/TSO 7.3, XPEDITER/CICS 7.6, and Abend-AID for CICS 4.5 do not support source listing shared directories. Instead, individual source databases may be specified for lower levels.

Note: If the source member is to be used by XPEDITER/CICS, the source listing file specified here must be VSAM.

- Determine the language processor options you will use as input and add them to the EXEC parameter, or add a CWPPRMO DD statement that points to the appropriate options. Sample options are contained in SLCXCNTL sample library members CXLPCOBB (for batch programs) and CXLPCOBC (for CICS programs) to provide the smallest output to the source listing file (for minimizing space) and generate a printout of the COBOL enhanced listing. When using these options, a printout of the listing from the DDIO file contains only the source statements. The XREF and other portions of the listing are not written to the DDIO file. If you use the sample options, note the following:
 - If you do not have XPEDITER/TSO, remove DDIO(OUTPUT(FIND)).
 - If the program will be used by XPEDITER/CICS or Abend-AID for CICS, use the options in sample CXLPCOBC.
- The SYSPRINT file will be LRECL=133, RECFM=FBA.

Post-processor

You can use the post-processor to process compile listings that were previously stored, or you can modify your COBOL compile JCL to pass the compiler listing to the post-processor as described below.

- Check the COBOL compiler options.
- Change the COBOL compiler JCL to write the compiler listing (SYSPRINT) to a sequential file that can be used as input to the COBOL language post-processor. A sample of the necessary JCL is shown below.

```
//SYSPRINT DD UNIT=SYSDA,SPACE=(TRK,(25,20)),DCB=BLKSIZE=16093,
//          DISP=(MOD,PASS)          <===== CA-OPTIMIZER NEEDS 'MOD'
```

The above example is a temporary file. You can save the JCL as a permanent file to be reused as shown in the example below:

```
//SYSPRINT DD DSN=CX,STORED.LISTINGS(MEMBER)
```

- The SYSPRINT file will be LRECL=133, RECFM=FBA.
- Add SLCXCNTL sample library member CXCOB1 as the post-processor step following your COBOL compile step.
- Determine the language processor options you will use as input and add them to the EXEC parameter, or add a CWPPRMO DD statement that points to the appropriate options. Sample options are contained in SLCXCNTL sample library members CXLPCOBB (for batch programs) and CXLPCOBC (for CICS programs) to provide the smallest output to the source listing file (for minimizing space) and generate a printout of the COBOL listing. When using these options, a printout of the listing from the DDIO file will contain only the source statements. The XREF and other portions of the listing will not be written to the DDIO file. If you use the sample options, note the following:
 - If you do not have XPEDITER/TSO, remove DDIO(OUTPUT(FIND)).
 - If the program will be used by XPEDITER/CICS or Abend-AID for CICS, use the options in sample CXLPCOBC.

The SLCXCNTL sample library members that are supplied for the COBOL language post-processor invoke the language processor in different ways:

Table 5-5. SLCXCNTL Sample Library Members for COBOL Language Processor

Member	Purpose
CXCOB1	Processes compiler listings through the language post-processor. Add this JCL after your COBOL compile step and before your link edit step (if one exists).
CXCOB2	Postprocesses compiler listings stored from a previous compile job. Use this JCL to process listings that have been previously compiled and stored in machine-readable format.
CXCOB99	Compiles COBOL programs and feeds the compiler output into the COBOL post-processor.

- The CWPPRTO file will be LRECL=133, RECFM=FBA.
- The CWPLOAD file must be allocated using DISP=OLD or DISP=SHR.
- The object module that is processed by the post-processor must be used as input to the linkedit step.

Step 8b. Modify the JCL to Run the PL/I Language Processor

You must perform this step if the following is true:

- You are installing XPEDITER/CICS or XPEDITER/TSO and you are licensed for the Compuware PL/I language processor.

This step contains instructions for both the PL/I language preprocessor and post-processor. You should follow the instructions that apply to the type of processing you will be performing.

Preprocessor

SLCXCNTL sample library member CXPLIPRE contains sample JCL necessary to run the PL/I language preprocessor.

Modify the compile step of your PL/I JCL:

- On the EXEC statement, change the name of your compiler to CWPPMAIN and add the following to any existing PL/I compiler options:

```
LANGUAGE(compiler version)
```

Refer to the *Compuware Shared Services User/Reference Guide* for a list of the compiler versions that can be specified with the LANGUAGE command. The default compiler for PL/I is the compiler currently in use on your system.

Note: The LANGUAGE command may be specified in the CWPPRMO command stream rather than on the EXEC statement.

- Add the CSS SLCXLOAD DSN to the STEPLIB or JOBLIB concatenation if it is not in the link list.
- Add the CWPDDIO DD statement to specify the name of the source listing DDIO file you will be using. The source listing DDIO file may be in standard DDIO, shared directory, or database format. Releases prior to XPEDITER/TSO 7.3, XPEDITER/CICS 7.6, and Abend-AID for CICS 4.5 do not support source listing shared directories. Instead, individual source databases may be specified for lower levels.

Note: If the source member is to be used by XPEDITER/CICS, the source listing file specified here must be VSAM.

- Determine the language processor options you will use as input and add them to the EXEC parameter, or add a CWPPRMO DD statement that points to the appropriate options. Sample options are contained in SLCXCNTL sample library member CXLPLI to provide the smallest output to the source listing file (for minimizing space). If you use these options, and you will be using the PL/I language preprocessor, add LANGUAGE(xxx) to the EXEC PARM or to the CWPPRMO DD statement.

See the *Compuware Shared Services User/Reference Guide* for more information on this JCL and PL/I language processor options.

Post-processor

You can use the post-processor to process compile listings that were previously stored, or you can modify your PL/I compile JCL to pass the compiler listing to the post-processor as described below.

- Check the PL/I compiler options.
- Change the PL/I compile JCL to write the compiler listing (SYSPRINT) to a sequential file for use as input to the PL/I language post-processor. A sample of the necessary JCL is shown below.

```
//SYSPRINT DD UNIT=SYSDA,SPACE=(CYL,(100,10)),DCB=BLKSIZE=19069,
//          DISP=(MOD,PASS)
```

The above example is a temporary file. You can save the JCL as a permanent file to be reused as shown in the example below:

```
//SYSPRINT DD DSN=CX,STORED.LISTINGS(MEMBER)
```

- The SYSPRINT file will be LRECL=133, RECFM=FBA.

Table 5-6. SLCXCNTL Sample Library Members for PL/I Language Processor

Member	Purpose
CXPLI	Processes compiler listings through the language post-processor. Add this JCL after your PL/I compile step and before your link edit step (if one exists). CXPLI can be customized to meet your site's requirements.
CXPLI2	Postprocesses compiler listings stored from a previous compile job. Use this JCL to process listings that have been previously compiled and stored in machine-readable format.

- The CWPLOAD file must be allocated using DISP=OLD or DISP=SHR.
- The object module that is processed by the post-processor must be used as input to the linkedit step.

See the *Compuware Shared Services User/Reference Guide* for more information on this JCL and PL/I language processor options.

Step 8c. Modify the JCL to Run the Assembler Language Processor

You must perform this step if you are licensed for the Assembler language processor option for any product.

You can optionally perform this step if you are installing XPEDITER/TSO **and** you are licensed for the Compuware Assembler language processor. XPEDITER/TSO customers can either perform this step now in batch mode, or online as part of the XPEDITER/TSO installation procedure.

This step contains instructions for both the Assembler language preprocessor and post-processor. You should follow the instructions that apply to the type of processing you will be performing.

SLCXCNTL sample library member CXASMPRE contains sample JCL necessary to run the Assembler language preprocessor.

Preprocessor

Modify the assembly step of your Assembler JCL:

- On the EXEC statement, change the name of your compiler to CWPAMAIN and add the following to any existing assembler options:

```
LANGUAGE(assembler version)
```

Refer to the *Compuware Shared Services User/Reference Guide* for a list of the assembler versions that can be specified with the LANGUAGE command. The default assembler is Assembler H.

Note: The LANGUAGE command may be specified in the CWPPRMO command stream rather than on the EXEC statement.

- Add the CSS SLCXLOAD DSN to the STEPLIB or JOBLIB concatenation if it is not in the link list.
- Add the CWPDDIO DD statement to specify the name of the source listing DDIO file you will be using. The source listing DDIO file may be in standard DDIO, shared directory, or database format. Releases prior to XPEDITER/TSO 7.3, XPEDITER/CICS 7.6, and Abend-AID for CICS 4.5 do not support source listing shared directories. Instead, individual source databases may be specified for lower levels.

Note: If the source member is to be used by XPEDITER/CICS, the source listing file specified here must be VSAM.

- Determine the language processor options you will use as input and add them to the EXEC parameter, or add a CWPPRMO DD statement that points to the appropriate options. Sample options are contained in SLCXCNTL sample library member CXLPSM to provide the smallest output to the source listing file (for minimizing space). When using these parameters, a printout of the listing from the DDIO file will contain only the source statements. The XREF will not be written to the DDIO file. If you use these options and you will use the Assembler language preprocessor, add LANGUAGE(xxx) to the EXEC PARM or to the CWPPRMO DD statement.
- Specify the compiler (and release) you will use by including it as a parameter of the LANGUAGE command in your EXECUTE parameter or CWPPRMO DD statement.

Refer to the *Compuware Shared Services User/Reference Guide* for more information on the LANGUAGE command, for a list of available compilers, and for more information on this JCL and language processor options.

Post-processor

You can use the post-processor to process compile listings that were previously stored, or you can modify your Assembler compile JCL to pass the compiler listing to the post-processor as described below.

- Check the assembler options.
- Change your assembler JCL to write the compiler listing (SYSPRINT) to a sequential file for use as input to the Assembler language post-processor. A sample of the necessary JCL is shown below.

```
//SYSPRINT DD UNIT=SYSDA,SPACE=(TRK,(25,20)),DCB=BLKSIZE=16093,
//          DISP=(MOD,PASS)
```

The above example is a temporary file. You can save the JCL as a permanent file to be reused as shown in the example below:

```
//SYSPRINT DD DSN=CX,STORED.LISTINGS(MEMBER)
```

- The SYSPRINT file will be LRECL=133, RECFM=FBA.
- Add SLCXCNTL sample library member CXASM as the post-processor step following your assembler step. CXASM can be customized to meet your site's requirements.

Note: The CWPLOAD file must be allocated using DISP=OLD or DISP=SHR.

- The object module that is processed by the post-processor must be used as input to the linkedit step.

Refer to the *Compuware Shared Services User/Reference Guide* for more information on this JCL and language processor options.

Step 8d. Modify the JCL to Run the C Language Processor

You must perform this step if you are installing XPEDITER/TSO and you are licensed for the Compuware C language processor.

This step contains instructions for both the C language preprocessor and post-processor. You should follow the instructions that apply to the type of processing you will be performing.

Preprocessor

SLCXCNTL sample library member CXCPRE contains sample JCL necessary to run the C language preprocessor.

Modify the compile step of your C JCL:

- On the EXEC statement, change the name of your compiler to CWPZMAIN and add the following to any existing C compiler options:

```
LANGUAGE(compiler version)
```

Refer to the *Compuware Shared Services User/Reference Guide* for a list of the compiler versions that can be specified with the LANGUAGE command. The default compiler for C is the compiler currently in use on your system.

Note: The LANGUAGE command may be specified in the CWPPRMO command stream rather than on the EXEC statement.

- Add the CSS SLCXLOAD DSN to the STEPLIB or JOBLIB concatenation if it is not in the link list.
- Add the CWPDDIO DD statement to specify the name of the source listing DDIO file you will be using. The source listing DDIO file may be in standard DDIO, shared directory, or database format. Releases prior to XPEDITER/TSO 7.3 do not support source listing shared directories. Instead, individual source databases may be specified for lower levels.
- Determine the language processor options you will use as input and add them to the EXEC parameter, or add a CWPPRMO DD statement that points to the appropriate options. Sample options are contained in SLCXCNTL sample library member CXLPC to provide the smallest output to the source listing file (for minimizing space). If you use these options, and you will be using the C language preprocessor, add LANGUAGE(xxx) to the EXEC PARM or to the CWPPRMO DD statement.

Refer to the *Compuware Shared Services User/Reference Guide* for more information on this JCL and C language processor options.

Post-processor

You can use the post-processor to process compile listings that were previously stored, or you can modify your C compile JCL to pass the compiler listing to the post-processor as described below.

- Check the C compiler options.
- Change the C compile JCL to write the compiler listing (SYSCPRT) to a sequential file for use as input to the C language post-processor. A sample of the necessary JCL is shown below.

```
//SYSCPRT DD UNIT=SYSDA,SPACE=(CYL,(100,10)),
//          DCB=(RECFM=VB,BLKSIZE=16096),
//          DISP=(MOD,PASS)
```

For more information on C language processor JCL, see the “C Language Processor” chapter in the *Compuware Shared Services User/Reference Guide*.

The above example is a temporary file. You can save the JCL as a permanent file to be reused as shown in the example below:

```
//SYSRINT DD DSN=CX,STORED.LISTINGS(MEMBER)
```

- The SYSRINT file will be LRECL=133, RECFM=FBA.
- Add SLCXCNTL sample library member CXC as the post-processor step following your C compile step. CXC can be customized to meet your site’s requirements.

Note: The CWPLOAD file must be allocated using DISP=OLD or DISP=SHR.

- The object module that is processed by the post-processor must be used as input to the linkedit step.

See the *Compuware Shared Services User/Reference Guide* for more information on this JCL and C language processor options.

Step 9. Initialize the zIIP Enablement Service (optional)

Overview

IBM has provided a mechanism for Independent Software Vendors (ISV), such as Compuware, to execute portions of ISV product code on zIIP processors on the customer's system. Work executed on a zIIP processor is not charged for usage as with general use processors. As a result, when portions of Compuware product code are offloaded to execute on a zIIP processor, it can result in reduced costs for the customer.

The CSS zIIP Enablement Service provides a common method to enable selected portions of Compuware product code to be eligible to run on a zIIP processor. If available on your system, performance of Compuware software products can be enhanced by executing selected portions of the product on a zIIP processor. Execution on a zIIP processor can result in less billable CPU time used and significantly fewer I/O operations for normal product operation.

The CSS Cobol, PL/I, and Assembler language processors are particularly well-suited for zIIP processing. Executing the language processors with zIIP processing enabled can reduce CPU time used by 35 to 65% depending on the size and nature of the compiler listing being processed.

Enabling the Service

Before the service can be used, it must be started. Starting the service enables its API for use by other Compuware products. The service should be started once after system IPL; it remains active as long as the system remains running. The service can optionally be stopped, if desired.

In order to start the service, run the initialization utility program CXZPINIT, which just takes a moment. Once this is completed, Compuware products can begin using zIIP processing. The default JCL used to start the service is below.

```
//GO          EXEC PGM=CXZPINIT
//STEPLIB DD DSN=COMPWARE.CX.SLCXAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//
```

It is recommended that the initialization utility be executed in an automated fashion after each system IPL. The utility can be executed as a job (just add an appropriate job card to the JCL above), or as a start command (if the JCL above is placed into a system PROCLIB).

Upon successful initialization, the following message will appear in the job log:

```
CXZPINIT010I Initialization successful for ID CPWR
```

Other error messages are possible, and these are documented in the ECC Messages and Codes.

Disabling the Service

This service can be temporarily disabled on a job-by-job basis, or completely disabled for the entire system.

Job disablement

To disable the service for a specific job, use the CXZPIGNR DD statement:

```
//CXZPIG NR DD DUMMY
```

This statement will cause the job to execute without using the service. Note that the portions of the product that were designated for zIIP processing will still be executed normally, but not on a zIIP processor.

System-wide disablement

If you wish to disable the service for all jobs, use the initialization utility with an EXEC PARM field to specify the keyword "DELETE", as shown in the example below:

```
//GO          EXEC PGM=CXZPINIT,PARM=DELETE
//STEPLIB DD DSN=COMPWARE.CX.SLCXAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//
```

After running this JCL, the service will no longer be active in the system. The initialization utility can be re-executed to re-enable the service, if desired.

Step 10. Continue with ECC Customization

If you have any Compuware products that have been licensed for use by your organization and you have installed the LMS component, you must continue with the procedures in **Chapter 6, "LMS Customization"**.

Note: If you have installed the Host Communications Interface (HCI) components, you should refer to the appropriate product manuals for specific customization/configuration instructions.

Chapter 6.

LMS Customization

This chapter provides the basic procedures for setting up and customizing Compuware's License Management System (LMS) for use with the Compuware products licensed by your organization. It includes instructions for:

- Transferring a License Certificate to your host system.
- Setting up and using the License Administration Utility (LAU) to create a License File and define system nodes.
- Importing a Certificate into a License File.
- Initializing the LMS runtime environment.

This chapter provides only the most basic elements needed to establish a Compuware LMS environment. To gain a more complete understanding of the LMS features consult the *License Management System User/Reference Guide—MVS* and the *License Management System User/Reference Guide—Centralized Licensing Facility*. Both of these LMS manuals can be found on Frontline and on the product distribution media.

Preliminary Considerations

Which tasks you need to perform will depend on whether LMS has already been enabled at your site and whether you want to manage your Compuware license certificate from a central location or manage them individually on each of your LPARs.

The following is a list of items you might want to consider:

- Managing your licenses from a centralized location
- Running more than one license management system
- Managing more than one Compuware customer number

Managing your licenses from a centralized location

Please refer to the *License Management System User/Reference Guide—Centralized Licensing Facility* for instructions on how to install a centralized licensing facility (CLF). The remainder of this document will discuss only the local maintenance of license certificates on each of your LPARs.

Running More than One License Management System

Under certain circumstances, a site may decide to run more than one instance of LMS on an MVS system. For instance, you might want both test and production LMS environments on the same MVS system. Compuware supports the environment of multiple LMS systems on a particular LPAR and information for establishing such an environment can be found in the User/Reference Guide (named previously), though for all but the most unusual implementations it is not necessary to run more than one LMS subsystem on any particular LPAR. One possible exception would be the installation of a complete new release of LMS where it is desired that existing Compuware products continue to execute normally during the time that the new release is installed and tested.

Managing More than One Compuware Customer Number

A site that manages multiple Compuware customer numbers, for example a service bureau, will have a License File for each Compuware customer number. The details for establishing a licensing environment consisting of multiple Compuware customer numbers can be found in the LMS User/Reference Guide.

Migration Considerations

The following items should be considered when upgrading from an earlier release of LMS to LMS release 4.0 or later:

- An LMS subsystem cannot be updated with an LMS 4.0 or later load library. You must delete the earlier version's subsystem and then create the LMS subsystem. By specifying FUNCTION(DELETE) in the LMSINIT SYSIN parms the older subsystem will be deleted.

Note: Any products that were checked out or running when the earlier version's subsystem was deleted should be shutdown and restarted.

- The LMS 2.0 checkpoint dataset cannot be updated with an LMS 4.0 or later load library. A new checkpoint that uses the LMS 4.0 or later file attributes must be created on each system running LMS 4.0 or later. LMS will automatically create the checkpoint data set using the name you supplied in the LMSINIT SYSIN parameters. Choose a data set name that is different from the one you used with the older release.
- Ensure that the LMS EXIT_PROC passes the correct parameter format:

```
EXEC PGM=LMSNMGR,PARM=&LMSSYS
```

Be especially careful to insure that the //STEPLIB DD in the LMS EXIT PROC points to the new release 4.0 (or later) data sets. Often this step is overlooked and results with the LMS EXIT PROC for the older release unable to update the checkpoint data set created by LMS 4.0.

Which Steps Do I Need to Perform?

The choice of which steps to perform depends on whether LMS has already been installed and initialized as part a previous Compuware product installation. For this reason, the procedures in this chapter are tailored for two possible scenarios:

- You need to install and set up LMS for the first time, then enable it to work with the Compuware product being installed.
- You are an established LMS user who only needs to enable it to work with the Compuware product being installed.

As shown in Table 6-1, a first-time user must perform every step. Most established LMS users can skip Steps 3, 4, and 5.

Table 6-1. LMS Steps Required

First-Time User	Step	Established User
x	"Step 1. Transfer License Certificate to Host"	x
x	"Step 2. Set up License Administration Utility (LAU)"	x
x	"Step 3. Create License File" on page 6-7	
x	"Step 4. Verify License File" on page 6-9	
x	"Step 5. Define Nodes" on page 6-10	
x	"Step 6. Import License Certificate" on page 6-12	^a x
x	"Step 7. Prepare an APF Authorized LMSINIT" on page 6-14	x

Table 6-1. LMS Steps Required

First-Time User	Step	Established User
x	"Step 8. Run LMSINIT" on page 6-15	x
x	"Step 9. Verify Proper LMS Function" on page 6-16	x
x	"Step 10. Continue ECC Customization" on page 6-17	x

a. This step is required only if new license certificates have been received.

Installing and Enabling LMS

Follow the instructions below that apply to your situation.

Step 1. Transfer License Certificate to Host

LMS uses License Certificate files to configure access to Compuware products licensed by your organization. A License Certificate is a text file typically sent to your site via e-mail by Compuware's Worldwide License Management team. To be used by LMS, a License Certificate must be accessible to MVS.

1. Locate the License Certificate for the Compuware products being installed. A License Certificate file can contain licensing for more than one Compuware product.
2. Allocate a target dataset on the host system for the License Certificate. Use the DCB parameters RECFM=FB, LRECL=80, and whatever BLKSIZE you prefer.
3. Transfer the License Certificate to the host using File Transfer Protocol (FTP), IND\$FILE, cut and paste, or any other method desired.

Note: If you open the License Certificate in an ISPF editor or transfer it using cut and paste, make sure NUMBERS is set to OFF.

4. If you have more than one License Certificate file, repeat the process as required.

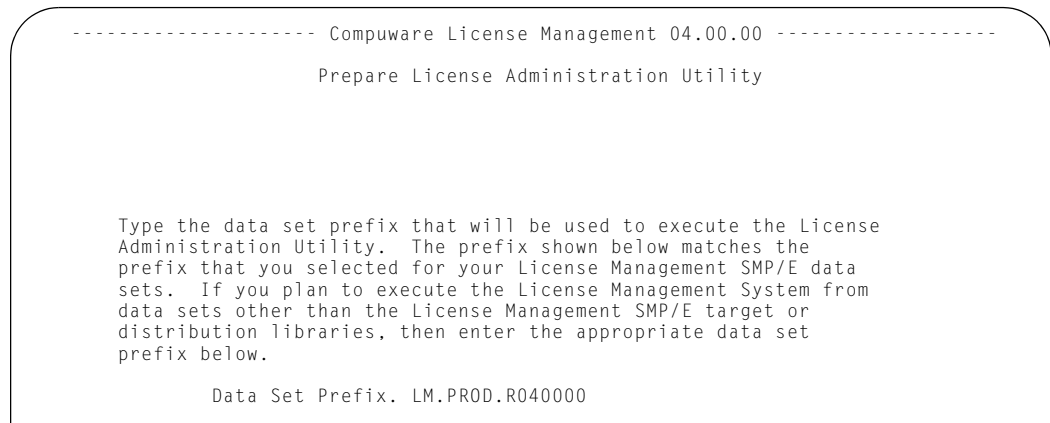
Step 2. Set up License Administration Utility (LAU)

This step creates the CLIST for starting the LAU, prepares the LAU to run, and makes it accessible to the License Administrators at your site.

1. Enter the following command to execute the LMSPREP EXEC from the SLMSCNTL library.

```
TSO EX ' your SMP/E dataset prefix.SLMSCNTL(LMSPREP)'
```

Figure 6-1. Prepare License Administration Utility Screen



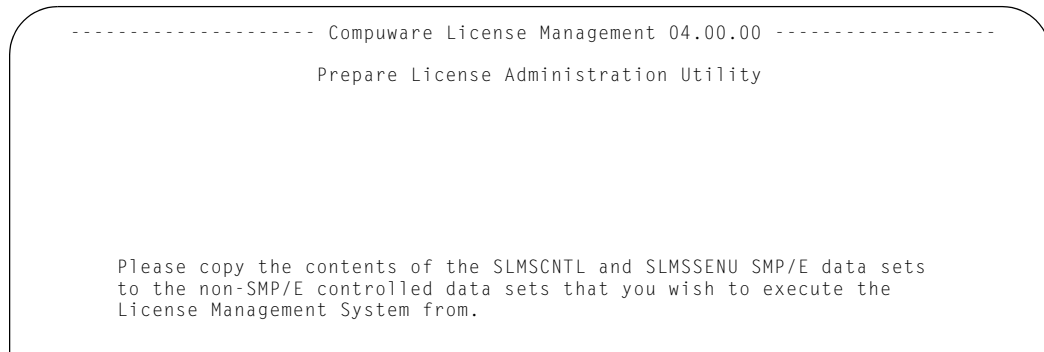
The following messages will notify you that you have successfully tailored the SLMSCNTL screen. Message LMP100I will be shown for each successfully tailored data set. Error messages will be displayed for any unsuccessful execution.

An example of the list of messages is shown below. You may have more messages in your particular environment-based upon features chosen:

```
LMP100I SLMSCNTL(CWLMA) SUCCESSFULLY TAILORED
LMP100I SLMSCNTL(CWLMQURY) SUCCESSFULLY TAILORED
LMP100I SLMSCNTL(CWLMDMSG) SUCCESSFULLY TAILORED
LMP100I SLMSCNTL(CWLMCHKP) SUCCESSFULLY TAILORED
LMP100I SLMSCNTL(LMUSER) SUCCESSFULLY TAILORED
LMP100I SLMSCNTL(ROIMAIN) SUCCESSFULLY TAILORED
LMP100I SLMSENU(LMASTPLB) SUCCESSFULLY TAILORED
```

2. The second screen will be displayed only when the dataset prefix field displayed in the first screen was changed to a value that did not match the dataset prefix of the SLMSCNTL library that contained LMSPREP. This is to accommodate customers who do not want to execute the LAU directly from the SMP/E target libraries for the License Management System.

Figure 6-2. Prepare License Administration Utility - second screen



3. Perform either of the following:
 - Add the SLMSCNTL library to your SYSPROC concatenation, or
 - Copy member CWLMA from the SLMSCNTL library into a dataset already concatenated to your SYSPROC DD.
4. Add the following line to an ISPF selection panel of your choice:

```
L, 'CMD(%CWLMA) NOCHECK'
```

Note: Adding CWLMA to ISPF is optional, and you may execute CWLMA as a CLIST.

5. If your site has elected to secure the ISPF Command Table via an external security manager (i.e., RACF, TOPSECRET, or ACF2), then you must take appropriate steps to identify the command LMAMAIN to your external security manager.
6. If you are an established user, go to “Step 4. Verify License File” on page 6-9.

Step 3. Create License File

If you have previously installed Compuware's License Management System software, go to "Step 4. Verify License File" on page 6-9.

In this step, you will use the LAU to create a License File.

1. Start the LAU by selecting the ISPF menu item that was added as part of "Step 2. Set up License Administration Utility (LAU)" on page 6-5. The License File Selection screen (Figure 6-3) will be displayed.

Figure 6-3. License File Selection Screen

```

----- Compuware License Management 04.00.00 -- Row 1 to 2 of 2
Command ==>                                     SCROLL ==> PAGE
                                     License File Selection

Current Selection:
Enter New DSN . . .
          (fully qualified without quotes)
Delete/Define . N (Y|N)

OR select below:-

Action      DSN                                     Added by
***** Bottom of data *****

```

2. Type the name you want to use for your License File (fully qualified without quotes) in the Enter New DSN field.
3. Type Y in the Delete/Define field.
4. Press Enter. The Delete/Define and Initialize License File screen (Figure 6-4) will be displayed, and the file name you specified above will be shown in the New License File field.

Figure 6-4. Delete/Define and Initialize License File Screen

```

----- Compuware License Management 04.00.00 -- Row 1 to 2 of 2
Command ==>                                     SCROLL ==> PAGE
                                     Delete/Define and Initialize License File

New License File . . : TS0ID01.License.File.New

Unit . . . . . _____ (required for JES3 only)
Volume Serial . . . _____ (blank for system determined volume)

Edit JCL . . . . . Y (Y-Yes,N-No)

Press END Key to skip this process

Jobcard:
//Job Card information line 1
//Job Card information line 2
//Job Card information line 3
//Job Card information line 4

Select      Node      Description
***** Bottom of data *****

```

5. Type Y in the Edit JCL field.
6. Type the information for your jobcard in the four lines of the Jobcard field.

Note: If you are not using all four lines of the job card, comment out the unused lines.

7. Press Enter. The Edit Delete/Define JCL screen (Figure 6-5) will be displayed.

Figure 6-5. Edit Delete/Define JCL Screen

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT      SYS98274.T095731.RA000.TS0ID01.R0111344          Columns0000100080
Command ==>                                           Scroll ==>CSR
***** Top of Data *****
000001 //Job Card information line 1
000002 //Job Card information line 2
000003 //Job Card information line 3
000004 //Job Card information line 4
000005 /**
000006 /*******
000007 /** DELETE/DEFINE COMPUWARE LICENSE FILE
000008 /**
000009 /** JCL GENERATED BY TS0ID01 ON 2001-05-17 AT 15:22
000010 /*******
000011 /**
000012 //LFDELDEF EXEC PGM=IDCAMS
000013 /**
000014 //SYSPRINT DD SYSOUT=*
000015 //SYSIN DD *
000016 DELETE (TS0ID01.License.File.New)
000017 SET MAXCC = 0
000018 DEF CL(NAME(TS0ID01.License.File.New)
000019      IXD

```

8. Make any necessary edits, then enter the END command. A Confirm Submission window will be displayed over the Delete/Define and Initialize License File screen.
9. Type **Y** in the Confirm Submission window and press Enter. The JCL to create your License File will be submitted.
10. Enter the END command. The License File Selection screen (Figure 6-3) will be displayed.
11. After your job has completed with a good return code, type **S** in the Action field next to your new License File and press Enter. Your License File name will appear in the Current Selection field.
12. Enter the END command.
 - If you are a first-time user, the main Compuware License Management screen (Figure 6-8 on page 6-10) will appear.
 - If you are an established user, the Parameter Option screen (Figure 6-9 on page 6-10) will appear. To access the main Compuware License Management screen (Figure 6-8 on page 6-10), enter the END command.

Step 4. Verify License File

In this step, you will use the LAU to verify which License File you will be using.

1. Start the LAU by selecting the ISPF menu item that was added as part of “Step 2. Set up License Administration Utility (LAU)” on page 6-5. Because the LAU has been run before and a License File already exists, the main Compuware License Management screen shown in Figure 6-8 on page 6-10 will be displayed. If the dataset name of the License file that you want to use is displayed next to the Browse option, go to “Step 5. Define Nodes” on page 6-10.
2. Type 0 in the Option field and press Enter. The Parameter Option screen (Figure 6-6) will be displayed.

Figure 6-6. Parameter Option Screen

```

----- Compuware License Management 04.00.00 -----
Option ==>

1)  Select      License File - TS0ID01.LICENSE.FILE
2)  Node        Specify System Nodes

X) Exit          Return to previous panel

(C) Copyright 1998, Compuware Corp. All Rights Reserved.

```

3. Type 1 in the Option field and press Enter. The License File Selection screen (Figure 5-7) will be displayed.

Figure 6-7. License File Selection Screen

```

----- Compuware License Management 04.00.00 -- Row 1 to 2 of 2
Command ==>                               SCROLL ==> PAGE

                               License File Selection

Current Selection:
Enter New DSN . . .
                (fully qualified without quotes)
Delete/Define . N (Y|N)

OR select below:-

Action   DSN                               Added by
-        LICENSE.FILE1                     USERID1 2000-08-07
-        LICENSE.FILE2                     USERID1 2000-08-07
***** Bottom of data *****

```

4. Type S in the Action field next to the file that you want to use and press Enter. Your selected file should appear in the current selection field.

Step 5. Define Nodes

If your installation does not consist of multiple nodes in a JES network, or if you do not require the ability to route License Management work (Import, Export, or Reporting) from one JES node to another in your network, go to “Step 6. Import License Certificate” on page 6-12.

In this step, you will use the LAU to define (to LMS) the network nodes on which License Management maintenance or reporting tasks could be dispatched.

Figure 6-8. Main Compuware License Management Screen

```

----- Compuware License Management 04.00.00 -----
Option ==>

0) Parameters Specify System Parameters
1) Browse License File - TS0ID01.License.File
2) Update License File
3) IMPORT Import License Certificate
4) EXPORT Export License Certificate
5) Reports Run Reports

6) Disaster Enable Disaster Site
7) Emergency Emergency Password

X) Exit License Management

Copyright (c) 1999 Compuware Corporation. All Rights Reserved.
Unpublished rights reserved under the Copyright Laws of the United States.
Enter HELP for Copyright/Trade Secret Notice information.

```

1. Type **0** in the Option field and press Enter. The Parameter Option screen (Figure 6-9) will be displayed.

Figure 6-9. Parameter Option Screen

```

----- Compuware License Management 04.00.00 -----
Option ==>

1) Select License File - TS0ID01.LICENSE.FILE
2) Node Specify System Nodes

X) Exit Return to previous panel

(C) Copyright 2010, Compuware Corp. All Rights Reserved.

```

2. Type **2** in the Option field and press Enter. The Maintain Nodes Table screen (Figure 6-10) will be displayed.

Figure 6-10. Maintain Nodes Table Screen

```

----- Compuware License Management 04.00.00 -- Row 1 to 6 of 6
Command ==>                                SCROLL ==> PAGE

                                Maintain Nodes Table

Enter new node . . . _____
Description . . . . _____

OR

(C-Change,D-Delete)
Action      Node      Description      Added by
-           Node1     Production      TS0ID01 2001-05-06 08:51
-           Node2     MVS Test1      TS0ID01 2001-05-05 10:44
-           Node3     QA Test        TS0ID01 2001-05-22 10:38
***** Bottom of data *****

```

3. In the Enter new node field on the Maintain Nodes Table (Figure 6-10), specify the name of the network node you want to add to the display list. This name identifies your installation's local JES in a network of systems or system complexes being used for network job entry (NJE) tasks. The node name can be up to eight alphanumeric characters.
4. Describe the new node briefly in the Description field. This description can be up to 20 alphanumeric characters.
5. Press Enter. The screen will be redisplayed with the new node in the selection list at the bottom of the screen. Repeat tasks step 3 through step 5 until you have defined all of the required nodes.
6. Enter the END command. The Parameter Option screen (Figure 6-9) will be displayed.
7. Enter the END command again. The main Compuware License Management screen (Figure 6-8) will be displayed.

Step 6. Import License Certificate

In this step the License Certificate for the Compuware product being installed will be imported into your site's License File.

1. On the main Compuware License Management screen (Figure 6-8 on page 6-10), type 3 in the Option field and press Enter. The first of two Import License Certificate screens (Figure 6-11) will be displayed.

Figure 6-11. First Import License Certificate Screen

```

----- Compuware License Management 04.00.00 -----
Command ==>

                IMPORT License Certificate

Process DSN   . : TS0ID01.License.File
IMPORT From   . . LICENSE.CERTIF_____
                (fully qualified without quotes)
Preview Only  . . N (Y|N)
Reminder      . . N (Y|N)

```

2. Type the name of the dataset used for your License Certificate file (fully qualified without quotes) in the IMPORT From field.
3. Type N in the Preview Only field.

Note: If you prefer to generate and review a report detailing the implications of the License Certificate importation, enter Y in the Preview Only field. You then need to repeat this step with Preview Only set to N.

4. If you type Y in the Reminder field you will see the message LMA104I changes to the license file will not take effect until LMSINIT has been executed. If you type N, the reminder message will not appear.
5. Press Enter. The second Import License Certificate screen (Figure 6-12) will be displayed.

Figure 6-12. Second Import License Certificate Screen

```

----- Compuware License Management 04.00.00 -- Row 1 to 5 of 6
Command ==>                                SCROLL ==> PAGE

                IMPORT License Certificate

Process DSN   . : TS0ID01.License.File
IMPORT From   . . LICENSE.CERTIF
Preview Only  . . N
Reminder      . . N

Edit JCL . . . . Y (Y-Yes,N-No)

Jobcard:
//Job Card information line 1
//Job Card information line 2
//Job Card information line 3
//Job Card information line 4

Select      Node      Description
-           Node1     Production
-           Node2     MVS Test1
-           Node3     QA Test
***** Bottom of data *****

```

6. Type **Y** in the Edit JCL field.
7. Type the information for your job card in the four lines of the Jobcard field.

Note: If you are not using all four lines of the job card, comment out the unused lines.

8. Type **S** next to the nodes on which you want the Import job to run. If no node is selected, the job is submitted without any specific routing.

Note: If you are running on only one node, do not make a selection here.

9. Press Enter. The Edit JCL screen (Figure 6-13) will be displayed.

Figure 6-13. Edit JCL Screen

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT SYS98274.T095731.RA000.TS0ID01.R0111344          Columns0000100080
Command ==>                                         Scroll ==>CSR
***** Top of Data *****
000001 //Job Card information line 1
000002 //Job Card information line 2
000003 //Job Card information line 3
000004 //Job Card information line 4
000005//*
000006//*****
000007//* IMPORT COMPUWARE LICENSE CERTIFICATE
000008//*
000009//* JCL GENERATED BY TS0ID01 ON 2001-07-08 AT 15:32:17
000010//*****
000011//*
000012//IMPORT EXEC PGM=LMAFIM,
000013// PARM='UPDATE'
000014//STEPLIB DD DSN=LM.DEVL.LOAD,
000015// DISP=SHR
000016//*
000017//CWLFO000 DD DSN=TS0ID01.License.File,
000018// DISP=SHR
000019//CWLFIIMP DD DSN=LICENSE.CERTIF,

```

10. Make any necessary edits, then enter the END command. A Confirm Submission window will be displayed over the second Import License Certificate screen.
11. Type **Y** in the Confirm Submission window and press Enter. The JCL to create your License File will be submitted.
12. Enter the END command. The main Compuware License Management screen (Figure 6-8 on page 6-10) will be displayed.
13. Type **X** in the Option field and press Enter to exit the LAU.
14. Proceed as follows:
 - If you are a first-time user, continue with “Step 7. Prepare an APF Authorized LMSINIT”.
 - If you are an established user, go to “Step 8. Run LMSINIT” on page 6-15.

Step 7. Prepare an APF Authorized LMSINIT

If you are an established user, please ensure that the modules listed below are copied from your License Management SMP/E target load library into your License Management APF authorized load library and go to “Step 8. Run LMSINIT” on page 6-15.

In this step, you will ensure that the LMS modules will be loaded from an APF authorized library, and you will also create the customized proc, JCL, and parameters dataset that will be used by LMSINIT to build the LMS runtime environment.

1. In order to function correctly, the LMS load modules must reside in an MVS APF authorized library. Consult your site’s MVS system programmer for details. This can be accomplished in either of the following ways:
 - APF authorize the library in which all the load modules for LMSINIT currently reside.

Because modules can be added over time via PTF, the most current list of modules that must reside in an APF-authorized data set can be found in the SAMPLE data set member name #APFMODES. Please refer to this member for a current list of modules.

Note that modules suffixed with two asterisks must be bound with AC=1. All other modules must be bound with AC=0 or allowed to default to this value.

Parameters used by LMSINIT are read from a SYSIN dataset. Edit the sample dataset provided in SLMSCNTL(LMINPARM) (see Figure 6-13).

- The operand and its associated value can start in any column and can be continued on more than one statement.
- Columns 1 through 71 are available for use.
- Column 72 is reserved.
- Columns 73 through 80 can be used for sequence numbers. They are ignored by LMSINIT.
- To continue an operand on a second statement, complete the first statement through column 71, and continue the operand on the next statement starting in column 1.
- Only the value specified for SUBSYSTEM_ID is case-sensitive. All other operands and their values are folded to upper case by LMSINIT.
- Only one occurrence of each of the operands is allowed. If duplicates exist, the last one encountered will be used.
- An asterisk ‘*’ in column 1 causes the entire statement to be a comment.
- Line level comments are supported using the ‘/*’ to start a comment and ‘*/’ to end the comment. Imbedded comments are supported.

For more information about specific parameters, refer to the chapter entitled “Creating Runtime Environment” in the *License Management System User/Reference Guide*.

Note: The following parameters are required: FUNCTION(UPDATE), SITE(), SUBSYSTEM_ID(), CHKPT_DSNAME(), EXIT PROC().

Sample LMSINIT Parameters Dataset

You will find a sample LMSINIT parameter data set in the sample data set shipped to you and installed with LMS. The member name is LMINPARM and should be used as a skeleton for parameters for your particular installation. Since this member can change with PTF maintenance, it is not listed here.

- To ensure access to Compuware products is automatically enabled, establish a procedure in your SYS1.PROCLIB to launch LMSINIT as a started task during IPL and IML processing. Consult your site's MVS system programmer for details. Edit the sample proc provided in SLMSCNTL(LMINPROC) (Figure 6-14). Change the data set names to match those used at your site.

Note: This proc must run before any other Compuware product procs.

Figure 6-14. Sample LMSINIT Proc

```
//LMSINIT PROC
//LMSSTEP EXEC PGM=LMSINIT,PARM='LANGUAGE=EN'
//STEPLIB DD DISP=SHR,DSN=USER.LMS.AUTHLOAD <===== CHANGE DSN
//SYSPRINT DD SYSOUT=*
//CWLFO000 DD DISP=SHR,DSN=YOUR.VSAM.LICENSE.FILE <===== CHANGE DSN
//SYSIN DD DISP=SHR,
// DSN=<SMPE.HLQ>.SLMSCNTL(LMINPARM) <===== CHANGE DSN
```

- Edit the sample JCL provided in SLMSCNTL(LMINJCL) (Figure 6-15). Change the dataset names to match those used at your site.

Figure 6-15. Sample LMSINIT JCL

```
/**
/** YOUR JOBCARD GOES HERE
/**
//LMSSTEP EXEC PGM=LMSINIT,PARM='LANGUAGE=EN'
//STEPLIB DD DISP=SHR,DSN=USER.LMS.AUTHLOAD <===== CHANGE DSN
//SYSPRINT DD SYSOUT=*
//CWLFO000 DD DISP=SHR,DSN=YOUR.VSAM.LICENSE.FILE <===== CHANGE DSN
//SYSIN DD DISP=SHR,
// DSN=<SMPE.HLQ>.SLMSCNTL(LMINPARM) <===== CHANGE DSN
```

Step 8. Run LMSINIT

Submit your site's customized LMSINIT JCL created from the sample provided in SLMSCNTL(LMINJCL) (Figure 6-15). This will establish the runtime environment that enables LMS to function.

Step 9. Verify Proper LMS Function

In this step, you will use the LAU to generate a License Verification Report. You can then use that report to verify that the procedures in this chapter were performed correctly and LMS is operating properly.

1. Start the LAU by selecting the ISPF menu item that was added as part of “Step 2. Set up License Administration Utility (LAU)” on page 6-5. The main Compuware License Management screen shown in Figure 6-8 on page 6-10 will be displayed.
2. Type **5** in the Option field and press Enter. The Reports screen (Figure 6-16) will be displayed.

Figure 6-16. Reports Screen

```

----- Compuware License Management 04.00.00 -- Row 1 to 3 of 7
Command ==>                                SCROLL ==> PAGE
                                     Report Selection
Select "D" for Detail, or "S" for Summary
                                     Expire Date   Process DSN
- License Verification
- Current Cache Report
- VSAM License File
- SAM License File
- SMF License Records
- Product Activity
- Activity Extract

Edit JCL . . . . N
Jobcard:
//Job Card information line 1
//Job Card information line 2
//Job Card information line 3
//Job Card information line 4

Select      Node      Description
-           Node1     Production
-           Node2     MVS Test1
-           Node3     QA Test

```

3. Type **S** next to License Verification.
4. Type **Y** in the Edit JCL field.
5. Type the information for your jobcard in the four lines of the Jobcard field.

Note: If you are not using all four lines of the job card, comment out the unused lines.

6. Type **S** next to the nodes on which you want the Report job to run. If no node is selected, the job is submitted without any specific routing.

Note: If you are running on only one node, do not make a selection here.

7. Press Enter. An Edit JCL screen will be displayed.
8. Make any necessary edits, then enter the END command. A Confirm Submission window will be displayed over the Reports screen.
9. Type **Y** in the Confirm Submission window and press Enter. The JCL to create your License Verification Report will be submitted.
10. Enter the END command. The main Compuware License Management screen (Figure 6-8 on page 6-10) will be displayed.
11. Type **X** in the Option field and press Enter to exit the LAU.

12. Use SDSF to examine the License Verification Report and ensure that the Compuware product being installed is recognized and enabled by LMS.

Step 10. Continue ECC Customization

The ECC installation and customization procedures are now complete. Please return to your product's installation guide and proceed with the installation instructions provided there.

Notes:

1. If you have installed the Host Communications Interface (HCI) components, you should refer to the appropriate product manual for specific customization/configuration instructions.
2. If you are installing Abend-AID or Abend-AID for CICS, refer to the respective installation and customization guides for configuration instructions concerning the HCI libraries.

Resolving Problems

If any error messages were returned while performing steps in this chapter, refer to the *Enterprise Common Components Messages and Codes* guide.

If you encountered any other difficulties related to License Management System, please consult the *License Management System User/Reference Guide* for more information. If problems persist, contact Compuware Technical Support.

Appendix A.

Compuware Maintenance Utility

The Compuware Maintenance Utility (CMU) is a program executed through JCL that, by default, generates reports listing all PTFs currently applied to a particular system, for all specified Compuware mainframe products. With the addition of an optional control file (available from Frontline (<http://go.compuware.com>)), the CMU can produce reports that include a list of PTFs that have *not* been applied or are otherwise missing. The contents of the control file (Figure A-2) are placed into a dataset or in stream JCL for the CMU to scan.

The CMU is distributed in the License Management System loadlib (SLMSLOAD).

Note: The CMU was introduced in License Management System (LMS) via a PTF in May 2013. Since then, the CMU control file format was adjusted and minor enhancements added along with more supported mainframe products. PTF LMF0068 is now required. Make sure all current maintenance is applied to LMS before attempting to use the CMU. Refer to the Compuware products listed in the "z/OS, Compatibility and Supported Releases Table" on FrontLine for any additional product PTF requirements.

The CMU processes a simple list of dataset names that it will scan and report on. The utility is able to scan the system link list for CMU supported Compuware product datasets. It will also scan the STEPLIB DD concatenation of the JCL as well as the datasets specified for SYSIN. The combined scan results from these sources will be presented in the output reports.

JCL Required to Execute the CMU

Several DD statements are required to execute the CMU and some are optional. Sample JCL is provided in the LMS samples library (SLMSCNTL) member LMMNTUTL. The general JCL structure required is shown below:

Figure A-1. General JCL Structure

```
//JOB CARD JOB CLASS=A
//MAINT EXEC PGM=LMMNTUTL
//STEPLIB DD DISP=SHR,DSN=COMPWARE.MLMSNN.SLMSLOAD
//SYSPRINT DD SYSOUT=*
//REPORT DD SYSOUT=*
//ALLPTFS DD SYSOUT=*
//PRODUCTS DD DISP=SHR,DSN=COMPWARE.LM.MAINTCTL(LMMNTCTL)
//SYSIN DD *
product.dataset.name1
product.dataset.name2
product.dataset.name3
/*
//
```

Required Datasets

STEPLIB — this dataset must point to the License Management System SLMSLOAD library where the CMU resides. Any other Compuware product dataset may

optionally be specified in the STEPLIB DD concatenation for scanning and reporting on their contents as well.

SYSPRINT — a required SYSOUT dataset. This output dataset will contain any error messages resulting from the scan.

REPORT — a required SYSOUT dataset. This output dataset will contain the summary "one-page" report for each product dataset found in STEPLIB, SYSIN, or the system link list.

ALLPTFS — a required SYSOUT dataset. This output dataset will contain the full list of applied or missing PTFs for each product dataset found in STEPLIB, SYSIN, or the system link list.

PRODUCTS — this optional input dataset contains the control file information for the CMU. The CMU uses the information in this dataset to determine the highest levels of maintenance that are available for each product. When this dataset is available and processed successfully, the CMU will produce reports based on PTFs that are not yet applied (missing), by product.

When the PRODUCTS DD statement is not present in the JCL, or if the associated dataset was not in the proper format and could not be processed, the CMU will produce the default report based on all of the PTFs that are applied, by product.

The control file input specified by PRODUCTS can be in stream data in the JCL, or be a PDS dataset member. If the control information resides in a physical dataset, the dataset must be 80-byte fixed length records, any valid blocksize.

The format of the control file data is discussed below in "Control File Format"

SYSIN - this is an optional, but recommended, input dataset. This dataset specifies a list of Compuware product dataset load libraries that are to be scanned. If this dataset is coded, the product dataset names should be specified in columns 1 through 44, one dataset per line.

Control File Format

The control file, provided on Frontline (<http://go.compuware.com>), under License Management System, contains a list of Compuware products and release numbers, which includes a list of all the currently available PTFs for each.

The control file is used by the CMU to determine which PTFs have not yet been applied. That determination is based on what PTFs are actually applied compared to the inclusion list of all the PTFs that are currently available.

The control file will be updated on Frontline each time maintenance is released for the participating products. Be sure to check for updates before running the utility to ensure the reporting is current.

An example of the control file input is in Figure A-2.

Figure A-2. Example of the Control File input

```

*****
**** PTF-INCLUDE-LIST ****
*****
UPDATED 2013 SEP 23

** Compuware Shared Services
*
PRODUCT CX 08.05.00
  CXM0001-CXM0097 CXM0099-CXM0101 CXM0103-CXM0141 CXM0143-CXM0408
PRODUCT CX 08.06.00
  CXN0001-CXN0088 CXN0090-CXN0526 CXN0528-CXN0559 CXN0561-CXN0575
  CXN0577-CXN0593
PRODUCT CX 08.07.00
  CX00001-CX00071 CX00073-CX00080 CX00082-CX00105 CX00108-CX00418
  CX00420-CX00480 CX00482-CX00741 CX00743
PRODUCT CX 08.08.00
  CXP0001-CXP0119 CXP0121-CXP0180 CXP0182-CXP0287 CXP0289-CXP0446
  CXP0448
PRODUCT CX 09.00.00
  CXQ0001-CXQ0046 CXQ0048-CXQ0049 CXQ0052

** License Management System
*
PRODUCT LM 04.00.00
  LMF0000-LMF0061

** Abend-AID Common
*
PRODUCT AC 12.03.00
* FMID MKAZ231
  ACB001Y-ACB002Y ACB004Y ACB006Y ACB008Y-ACB009Y ACB012Y ACB014Y-ACB016Y
  ACB020Y ACB022Y ACB027Y-ACB030Y ACB032Y ACB034Y ACB037Y ACB043Y
  ACB045Y ACB050Y ACB053Y ACB055Y ACB057Y ACB060Y-ACB061Y ACB065Y
* FMID OKAZ231
  ACB004Z ACB006Z ACB008Z-ACB009Z ACB020Z ACB022Z ACB028Z-ACB030Z

** Abend-AID
*
PRODUCT AA 12.03.00
* FMID MPAA230
  AAB005Y AAB007Y AAB010Y AAB013Y AAB015Y AAB019Y-AAB023Y AAB026Y
  AAB028Y AAB031Y-AAB032Y AAB035Y AAB045Y-AAB048Y AAB050Y-AAB052Y
  AAB054Y AAB060Y AAB065Y-AAB066Y
* FMID OPAA230
  AAB020Z AAB022Z AAB032Z

** Abend-AID for CICS
*
PRODUCT CF 12.03.00
* FMID MKFX231
  CFB003J CFB011J CFB015J CFB018J CFB020J-CFB022J CFB025J CFB044J
  CFB050J-CFB051J CFB065J
* FMIDS NKFX232-NKFX236
  CFB017*-CFB018* CFB021* CFB024* CFB033* CFB036* CFB038* CFB041*
  CFB049*-CFB050* CFB065*
* FMID OKFX231
  CFB020A CFB022A

** Fault Analytics
*
PRODUCT MF 12.03.00
  MFB002J

*****
*** END-PTF-INCLUDE-LIST ***
*****

```

The control file specifies all CMU supported Compuware mainframe products back to a reasonable release level, typically no more than two releases back from current ($n-2$).

Lines within the control file that contain an asterisk (*) in column 1 are considered comments and are ignored.

If the CMU is unable to parse the control file information because the format is incorrect or because the information was garbled as a result of download, it will report an error message to SYSPRINT, then produce the default report containing only applied PTFs.

Report Output

Figure A-3 shows the output for the summary report when the CMU is executed *without* a control file (no PRODUCTS DD statement). Following this summary report, is an example of the output showing the list of PTFs applied (Figure A-4).

Further examples are shown providing a sample of the CMU output *with* a control file present (Figure A-5 and Figure A-6).

In the reports, if a product dataset was specified in one of the input sources but it could not be identified as a CMU Supported Compuware product dataset (no xxRELSMP-like module), a double-question mark (??) will appear in the report for the product code and an N/A will appear in the description and release.

If a product dataset was scanned successfully, but that product was not represented in the control file input, then a double-question mark (??) will appear in the highest available PTF column because no control information was available. When this condition occurs, the CMU will also be unable to determine the number of missing PTFs.

Figure A-3. Summary report without a PRODUCTS DD control file

Compuware Product Maintenance at a Glance for specified datasets					
Product code and name	Product Release	Highest PTF Applied	# PTFs Applied	Source	Library
CX Shared Services	08.07.00	CX00486	481	STEPLIB	CX.PROD.R080700.SLCXLOAD
HC Host Communic. Interface	02.50.00	HCA0045	44		SYS2.HC250.S1.AUTHLIB
?? N/A	N/A				SYS2.HC250.S2.AUTHLIB
CX Shared Services	08.08.00	CXP0187	186		CX.PROD.R080800.ALLPTFS.SLCXLOAD
CX Shared Services	08.05.00	CXM0413	404	EXPLICIT	CX.PROD.R080500.SLCXLOAD
FA File-AID/MVS	09.03.01	FA9A011	9		FA.PUBLIC.V9R3M1T.MXFA931.SXFAAUTH
XT Xpediter/TSO	09.01.00	XTI0056	56		XT.PROD.MLXT910.ALLPTFS.SLXTLOAD
XT Xpediter/TSO	07.06.00	XTF0164	163		XT.PROD.MLXT760.ALLPTFS.SLXTLOAD
XD Xpediter/CICS	09.01.00	XDNL068	66		XD.R090100.NMXD910.SMXD065L
XD Xpediter/CICS	08.03.00	XDM1207	205		XD.R080300.MMXD839.SMXD063L
XG Xpediter/Xchange	05.00.00	XGB0014	14		XG.PROD.MLXG500.ALLPTFS.SLXGLOAD
XG Xpediter/Xchange	04.00.00	XGA0077	71		XG.PROD.MLXG400.ALLPTFS.SLXGLOAD
XV Xpediter/Code Coverage	03.01.00	XVC0041	40		XV.PROD.MLXV310.ALLPTFS.SLXVLOAD
XV Xpediter/Code Coverage	03.00.00	XVB0033	33		XV.PROD.MLXV300.ALLPTFS.SLXVLOAD
CX Shared Services	08.08.00	CXP0142	140	LINKLIST	
FA File-AID/MVS	09.03.01	FA9A028	20		

Figure A-4. Full report without a PRODUCTS DD control file

```

Compuware Product Maintenance Applied
                        for specified datasets

CX Shared Services           08.07.00
-----
From: STEPLIB   DSN: CX.PROD.R080700.ALLPTFS.SLCXLOAD

PTFs applied:
CX00001 CX00002 CX00003 CX00004 CX00005 CX00006 CX00007 CX00008 CX00009
CX00010 CX00011 CX00012 CX00013 CX00014 CX00015 CX00016 CX00017 CX00018
CX00019 CX00020 CX00021 CX00022 CX00023 CX00024 CX00025 CX00026 CX00027
CX00028 CX00029 CX00030 CX00031 CX00032 CX00033 CX00034 CX00035 CX00036
CX00037 CX00038 CX00039 CX00040 CX00041 CX00042 CX00043 CX00044 CX00045
CX00046 CX00047 CX00048 CX00049 CX00050 CX00051 CX00052 CX00053 CX00054
CX00055 CX00056 CX00057 CX00058 CX00059 CX00060 CX00061 CX00062 CX00063
CX00064 CX00065 CX00066 CX00067 CX00068 CX00069 CX00070 CX00071
CX00073 CX00074 CX00075 CX00076 CX00077 CX00078 CX00079
CX00082 CX00083 CX00084 CX00085 CX00086 CX00087 CX00088 CX00089 CX00090
CX00091 CX00092 CX00093 CX00094 CX00095 CX00096 CX00097 CX00098 CX00099
CX00100 CX00101 CX00102 CX00103 CX00104 CX00105
CX00109 CX00110 CX00111 CX00112 CX00113 CX00114 CX00115 CX00116 CX00117
CX00118 CX00119 CX00120 CX00121 CX00122 CX00123 CX00124 CX00125 CX00126
CX00127 CX00128 CX00129 CX00130 CX00131 CX00132 CX00133 CX00134 CX00135
..
..                               these lines omitted for brevity
..
CX00460 CX00461 CX00462 CX00463 CX00464 CX00465 CX00466 CX00467 CX00468
CX00469 CX00470 CX00471 CX00472 CX00473 CX00474 CX00475 CX00476 CX00477
CX00478 CX00479 CX00480 CX00481 CX00482 CX00483 CX00484 CX00485 CX00486

```

Figure A-5. Summary report with a Product DD control File

```

Compuware Product Maintenance at a Glance
                        for specified datasets

Product code and name      Product   Highest   Highest   #
                           Release    PTF      PTF      PTFs
                           -----
                           Applied   Avail   Missing  Source   Library
-----
CX Shared Services        08.07.00  CX00488  CX00488      2  STEPLIB  CX.PROD.R080700.SLCXLOAD
HC Host Communic. Interf 02.50.00  HCA0045  HCA0045      0
?? N/A                    N/A      ??
CX Shared Services        08.08.00  CXP0171  CXP0189     23
CX Shared Services        08.05.00  CXM0408  CXM0413     15  EXPLICIT  CX.PROD.R080500.LOADLIB
FA File-AID/MVS          09.03.01  FA9A027  FA9A028      2
XT Xpediter/TSO          09.01.00  XT10057  XT10042      0
XT Xpediter/TSO          07.06.00  XTF0164  XT10164      1
XD Xpediter/CICS         09.01.00  XDNL068  XD*L070      4
XD Xpediter/CICS         09.01.00  XDML068  XD*L070      4
XD Xpediter/CICS         08.03.00  XDMI207  XD*I207      2
XW DevEnterprise         05.03.00  XWE0026  ??
XG Xpediter/Xchange      05.00.00  XGB0015  XGB0015      0
XG Xpediter/Xchange      04.00.00  XGA0077  XGA0077      6
XV Xpediter/Code Coverage 03.01.00  XVC0041  XVC0041      1
XV Xpediter/Code Coverage 03.00.00  XVB0033  XVB0033      0
CX Shared Services        08.08.00  CXP0142  CXP0189     49  LINKLIS
FA File-AID/MVS          09.03.01  FA9A028  FA9A028      1
}

```

Figure A-6. Full report with a PRODUCTS DD control file

```

                                Compuware Product Maintenance Requirements
                                for specified datasets

CX Shared Services              08.07.00
-----
From: STEPLIB   DSN: CX.PROD.R080700.ALLPTFS.SLCXLOAD

PTFs missing:
  CX00080 CX00081

HC Host Communic. Interface  02.50.00
-----
From: STEPLIB   DSN: SYS2.HC250.S1.AUTHLIB

PTFs missing:
  NONE

CX Shared Services              08.08.00
-----
From: STEPLIB   DSN: CX.PROD.R080800.SLCXLOAD

PTFs missing:
  CXP0086 CXP0164 CXP0168 CXP0169 CXP0170 CXP0172 CXP0173 CXP0174 CXP0175
  CXP0176 CXP0177 CXP0178 CXP0179 CXP0180 CXP0181 CXP0182 CXP0183 CXP0184
  CXP0185 CXP0186 CXP0187 CXP0188 CXP0189

CX Shared Services              08.05.00
-----
From: EXPLICIT  DSN: CX.PROD.R080500.LOADLIB

PTFs missing:
  CXM0013 CXM0070 CXM0119 CXM0120 CXM0121 CXM0122 CXM0142 CXM0236 CXM0294
  CXM0298 CXM0409 CXM0410 CXM0411 CXM0412 CXM0413
FA File-AID/MVS                09.03.01
-----
From: EXPLICIT  DSN: FA.PUBLIC.V9R3M1T.MXFA931.SXFAAUTH \

PTFs missing:
  FA9A007 FA9A028
    
```

Glossary

Abend-AID. The Compuware product for fault diagnosis that provides an immediate, detailed analysis of application program failures in a comprehensive report format.

Abend-AID Fault Analytics. The Compuware Abend-AID option that assists managers in monitoring, measuring, controlling, and improving the fault resolution process in the IT environment. Abend-AID Fault Analytics (formerly Abend-AID Fault Manager) offers comprehensive fault management of the IBM OS/390, IBM z/OS, Windows 2000/NT, IBM/AIX, HP/UX, and Sun Solaris UNIX environments

Abend-AID for CICS. The Compuware product for fault diagnosis that provides a full range of analysis functions for managing abnormal CICS transaction terminations (abends) and CICS region outages.

Abend-AID for WebSphere (MQ). The Compuware product that enables developers to clearly define reason, return, and error codes, obtain MQ environment information whenever an application fails, and quickly solve enterprise and e-business systems problems. Comprehensive support is provided for WebSphere MQ/MQSeries in batch, IMS, and CICS environments.

Abend-AID products. A generic name for the Abend-AID product family including Abend-AID (MVS), Abend-AID for CICS, Abend-AID Fault Manager, and Abend-AID for WebSphere (MQ).

Abend-AID Report. (1) A set of records containing data extracted at abend time from the affected MVS region and associated control blocks that are stored in the Abend-AID DDIO report file. (2) A readable report produced at view time using the records stored in the Abend-AID DDIO report file.

Abend time. The time when an abend occurs and Abend-AID products perform their analyses.

Allocation group. A set of blocks assigned to a DDIO file member when the member requires additional storage.

APF Authorization. APF can be used to restrict access to system functions and can require that all modules loaded by an authorized program be loaded from an authorized load library. All modules loaded by LMSINIT must be loaded from

authorized load libraries or LMSINIT will fail. Only individuals with proper RACF authority will have permission to update authorized load libraries.

Automatic lock. Automatic locks are created whenever a member is added to a DDIO file format using the AUTODELETE=DUPS or AUTODELETE=STAGED option. The most recent member is automatically locked. See also **manual lock**.

AUTODELETE. An attribute of a DDIO file, specified during formatting of the file, that determines the action to be taken when the file becomes full and an attempt is made to add a new member to the file.

Base Services. Former ECC component now distributed with Abend-AID Common Components of the Abend-AID product line. It provides basic viewing server-related services enabling the Distributed Viewing Support.

Batch File Utility. A set of CSS utilities used to prepare, maintain, and manipulate DDIO files, shared directories, and databases.

CEC. Central Electronic Complex designates the physical computer which can contain one or more LPARs. CECs can be of the older type designated as G1 through G6 or can be the new z/800 or z/900.

CEC Licensing. The number of MSUs provided by that CEC across all LPARs.

CGI. Common Gateway Interface. A way of interfacing computer programs with HTTP or WWW servers, so that a server can offer interactive sites instead of just static text and images.

CGI-bin. Common Gateway Interface-BINaries. A special directory where common gateway interface (CGI) scripts are kept.

Compuware Shared Services (CSS). A set of components used by several Compuware products to provide storage, retrieval, and maintenance for source listings and diagnostic reports.

Compuware Viewing Facility (Compuware/VF). An obsolete online ISPF application found in earlier releases of CSS. VF provided menu-driven access to DDIO files such as Abend-AID reports, databases, shared directories and source listings. Compuware/VF supported online viewing, printing, locking, unlocking and deleting of members in any DDIO file. Most batch utility functions were also available, including creating and deleting databases or shared directories.

Source listings on remote files can be also be viewed. The Compuware/VF was phased out after the inception of the Abend-AID Viewer.

Customer Modification Facility (CMF). A method for activating specialized code modifications to the CSS base code. This facility can retrofit your restricted zaps from a release prior to CSS 7.4 to the SMP/E environment.

Customized Translation Table. An optional table provided by Compuware customers for translating non-printable characters. It is used by the batch file utility for printing or displaying lines containing storage displayed by Abend-AID products at abend time.

CWAASDUT. The batch file utility used for Abend-AID shared directories and report databases. CWAASDUT allows you to manage the files used by an Abend-AID shared directory and its associated abend report files. The shared directory is used to process all directory requests related to an abend report file and maintain information about database activity.

CWASSECU. The name of the user-coded Security Exit program.

CWDDALLU. The batch file utility program that can manage all source listing and report file DDIO types previously handled separately by CWAASDUT, CWDDSUTL, CWFXSDUT and CWDDLPUT.

CWDDLPUT. The batch file utility program used to manage source shared directories and databases for all of the supported languages. The shared directory maintains information about source listing processing and database activity.

CWDDSUTL. The program for the batch file utility used by Abend-AID, Xpediter/CICS, and Xpediter/TSO. CWDDSUTL allows you to print, delete, lock, unlock, copy, move, import, and export DDIO file members, and to list DDIO file directory entries. It also allows you to initially format a DDIO file.

CWFXSDUT. The batch file utility used by Abend-AID for CICS. CWFXSDUT allows you to manage the files used by a shared directory and its associated transaction databases. The shared directory is used to process all directory requests related to a transaction database.

CWPDDIO. The DDIO file that contains the source member output generated during language processing.

CWPERRM. Dynamically allocated dataset generated during language processing containing all the CSS messages.

CWPPRMO. A language processor input dataset that contains all of the language processor options. This DD statement can also be used to specify in-stream options.

CWPPRTI. Compiler or assembler listing input to the CSS postprocessor.

CXTRACE. A report that CSS technical support requests to assist in problem resolution. This report traces processing through CSS code and can be a useful tool for determining the cause of an abend.

DBMODEL. The name of an OBJECT that contains the specifications required to create a new source listing database during dynamic database creation (DYNCREATE). The information is stored in the specified source listing shared directory. It is used during language processing to dynamically create a new source listing database to contain the source member if all attached databases are full.

DDIO (Dump Dataset Input Output). A proprietary file access method that stores Abend-AID reports, transaction dump information, and source listings within specially formatted DDIO files for several Compuware products that use CSS. You can allocate DDIO files as VSAM or sequential datasets depending on specific product requirements.

DDIOCALC. An online calculator provided to calculate optimal allocation and formatting parameters for DDIO files. DDIOCALC cannot be used for shared directories.

DDIO File. A generic term that refers to the DDIO database or dataset used to store Abend-AID reports, transaction dump information, or source listings from Compuware products that use CSS.

DDIO File Member. A generic name for an individual diagnostic report in an Abend-AID report file, a source listing in a source listing file, or transaction dump information in a transaction database.

Directory entry. A record in the fixed portion of the DDIO file (Directory) that contains information specific to a member. Each member has a corresponding directory entry. The number of directory entries is specified during formatting of the DDIO file and cannot be changed without reformatting the file.

DIRX Report. A report produced by the DIRX command of the batch file utility or the "I" line command of the Compuware V/F utility panel. This report contains attribute and allocation information about the specified DDIO file and a listing of the directory entries.

Distributed Viewing Support (DVS). Allows Abend-AID and Abend-AID for CICS users to view both merged and base reports, where either the base report or listing file resides on a remote system. Abend-AID users can have source support across remote MVS systems that don't share DASD.

DYNCREATE. A term used to denote the "dynamic database creation" process that can be invoked when using the CSS Language Processor with a source shared directory listed for the compiler output (CWPDDIO). A DBMODEL must first be defined in the desired source shared directory using the batch utility CWDDLPUT or CWD-DALLU with DYNCREATE=YES specified to allow invoking of the dynamic database creation process. If enough space is not found in any of the attached source listing databases, then the DYNCREATE process is invoked. A new source listing database is created and attached, and the new source member is written.

Enhanced Listing. A convenient source of quick reference information and program documentation that merges DMAP and CLIST information, in addition to error and diagnostic messages, with a COBOL source listing.

Enterprise Common Components (ECC). ECC is the packaging method for the following Compuware facilities:

- Compuware Shared Services (CSS)
- License Management System (LMS)
- Host Communications Interface (HCI)

Entry. A generic name for a transaction diagnostic report in a transaction database for Abend-AID for CICS.

Formatting a DDIO File. The preparation of an allocated VSAM or sequential file to be used as a DDIO file.

Full DDIO File. A DDIO file for which all allocation groups and/or all directory entries are allocated to members. This scenario invokes the AUTODELETE process.

Host Communications Interface (HCI). A facility that provides connectivity between mainframe-based programmer productivity software and peer node software running on other platforms in a network. It allows application programs to communicate over any one of several protocols without knowledge of which protocol is in use at any given time.

Installation JCL Customization Facility. A CSS facility consisting of a set of screens that prompt you to enter installation information. This infor-

mation is used to build the jobs necessary to perform the CSS SMP/E installation.

Language Processor (LP). The CSS processor that analyzes and builds Assembler, C, COBOL, or PL/I compiler output into sort work records. The C and COBOL processors then sort and merge these records with the listing. The Assembler and PL/I processors build an incore symbol table of all the identifiers. Processor control blocks are then generated and written to a source listing DDIO file. These DDIO members can then be used as input for Compuware products that require this information.

License Administration Utility (LAU). The LAU is the license administration control center for your organization's IT professional who is responsible for managing your access to Compuware products. The LAU is an ISPF application that enables

- the creation of a License File
- the import of License Certificates into a License File
- the maintenance and export of a License File
- the reporting and analysis of your License File and License Certificate information in virtual storage

Your organization's License Administrator will also set up License Management System parameters and system operation options using the LAU.

License Certificate. English-like readable electronic records that contain a portion of the information from your license agreement for a product release's use at a particular site. The License Certificate is used to update your License File.

License File. Dataset containing imported License Certificate information for all licensed releases of Compuware products. This file is in ASCII text format.

License Management System (LMS). Facility that enables you to centrally administer Compuware product License Certificates and manage access to Compuware products at your site. The LMS includes several components that together enable you to establish, maintain, diagnose, and upgrade access to the Compuware products licensed by your enterprise. The LMS replaces the customer profile utility provided with earlier versions of Compuware mainframe products.

Line command. A command entered next to the line to be processed. A line command is executed only for the specified member.

LMVERIFY. See **Verification Report.**

Locked member. A DDIO file member that was manually locked using the batch file utility or automatically locked as a result of AUTODELETE=DUPS or AUTODELETE=STAGED processing. A manually locked member is identified by an M in the directory report or directory screen while an automatically locked member is identified by an L in the directory report or directory screen. When manually locked, a member cannot be automatically deleted when the DDIO file becomes full.

LPAR. Logical Partition is one operating system image, such as OS/390, z/OS, or z/OS-e.

LPAR Licensing. Refers to the defined capacity or “capped capacity” of the LPAR itself.

Manual lock. A member that was locked by using the LOCK batch command. Members that have been manually locked retain their locks during the autodeletion process. See also **automatic lock**.

MIPS. Millions of Instructions per Second. A unit of measure of processing performance equal to one million instructions per second.

MSU. Millions of Service Units per Hour.

Oldest member. (1) For report files, the diagnostic report with the earliest abend date and time. (2) For source listing files, the program with the earliest compile date and time.

Primary Command. A command entered in the COMMAND INPUT field while in TSO.

Problem Documentation Utility. A CSS utility that captures the documentation needed by Compuware for resolving technical issues and problems. This utility allows you to collect the documentation — such as copybooks, SYSPRINT, and DDIO members — that CSS Technical Support requires for problem resolution.

Report File. An Abend-AID DDIO file containing diagnostic reports.

Security Exit Program. A user-coded program that allows or denies execution of a requested command for a selected member by a specific user against particular DDIO files.

Server Grace. The amount of time an LPAR can run without being connected to the server.

Shared Directory. A variable-length record VSAM RRDS that maintains information about abends and language processing along with the attached database activity. A shared directory can contain Abend-AID for CICS directory records for

each region and transaction dump known to a server, Abend-AID directory records for abend report processing, or source listing shared directory records necessary to process source listing database members.

SMP/E. System Modification Program/Extended is IBM’s standard facility for installing and maintaining software modifications in the MVS environment.

Source Listing. A compiled or enhanced listing and additional information about a program, that is stored in a source listing DDIO file.

Source Listing Database. A specially formatted source listing DDIO file, attached to and managed by a source shared directory.

Source Listing File. A DDIO file containing source listings generated by the Language Processor.

STROBE. The STROBE MVS Application Performance Measurement System is a product that determines where and how application time is spent in online regions and batch processing programs and how system resources are used. STROBE collects several types of data as it tracks activity within an MVS environment and produces a collection of reports that helps you determine where to revise applications to improve their performance.

Suppressed statements. Statements that are not displayed in the compiler listing. Suppressed statements in a DDIO source member contain an internal flag that the Xpediter/TSO product can use to display or suppress source statements using the GEN command.

Transaction Database. A DDIO file, containing transaction dump information, that is managed by a shared directory in Abend-AID for CICS. Multiple entries created from CICS transaction terminations are stored in a single transaction database.

Unlocked Member. A DDIO file member that is not currently locked. The locked member can be unlocked either manually, using the batch file utility or automatically as a result of AUTODELETE=DUPS or AUTODELETE=STAGED processing. An unlocked member in a full DDIO file formatted with AUTODELETE=YES, AUTODELETE=DUPS, or AUTODELETE=STAGED can be automatically deleted when another member is added to the file.

Verification Report (LMVERIFY). Displays the result of a verification program that checks out each product version in cache. This data is exactly what a product would receive when it runs. Run-

ning this report will prove that the License Management System environment is properly set up for the customer.

View time. The time when a diagnostic report or source listing is presented in a readable format for online viewing or printing.

Index

A

abend time, G-1
 Abend-AID, G-1
 executing, 1-4
 Abend-AID Fault Manager, G-1
 Abend-AID for CICS, G-1
 executing, 1-4
 Abend-AID for WebSphere (MQ), G-1
 Abend-AID products, G-1
 Abend-AID report, G-1
 Abend-AID report shared directories
 allocating and formatting, 5-21
 Acrobat PDF online documentation, xii
 allocating
 Abend-AID for CICS transaction shared directories,
 5-22
 Abend-AID report shared directories and databases,
 5-21
 CSS libraries, 5-17
 dynamically allocating libraries, 5-15
 libraries at TSO startup, 5-16
 source listing files, 5-22
 source listing shared directories and databases, 5-21
 allocation group, G-1
 APF authorization
 CSS, 2-3
 definition, G-1
 LMS, 2-12
 LMSINIT, 6-14
 Assembler Language Processor
 modify JCL for installation, 5-28
 postprocessor, 5-29
 postprocessor DD statement, 5-29
 preprocessor, 5-28
 PRINT NOGEN, 2-5
 PRINT OFF, 2-5
 AUTODELETE, G-1
 automated compiler options, 2-5
 automatic lock, G-1

B

Base Services, G-1
 batch file utilities, 1-3
 batch file utility
 definition, G-1

C

C Language Processor
 modify JCL for installation, 5-29
 NOSHOWINC, 2-5
 postprocessor, 5-30

 preprocessor, 5-29
 capturing suppressed source code, preprocessor, 2-5
 CD
 contents of, 2-2
 CEC, G-1
 licensing, G-1
 CGI, G-1
 CGI-bin, G-1
 CICS startup JCL, 2-3
 CLISTS, 5-15
 COBOL Language Processor
 COPY SUPPRESS, 2-5
 modify JCL for installation, 5-24
 postprocessor, 5-25
 postprocessor compiler JCL, 5-25
 preprocessor, 5-24
 SLCXCNTL installation library members, 5-26–5-27
 compile errors, handling
 preprocessor, 2-5
 components of CSS, 1-2
 Compuware
 Frontline web site, xiii
 phone number and address, xiii
 Compuware Shared Services
 definition, G-1
 description of, 1-1
 Compuware/VF
 definition, G-1
 creating
 Abend-AID for CICS databases, 5-22
 Abend-AID for CICS shared directories, 5-22
 source listing databases, 5-22
 source listing shared directories, 5-21
 CSS
 allocating libraries, 5-17
 APF authorization, 2-3
 CICS startup JCL, 2-3
 common files and utilities, 1-2
 components, 1-2
 customization, 4-1, 5-1
 executing, 1-4
 installation library members, 2-7
 load modules, 5-4
 sharing DDIO files with multiple CPUs, 2-4
 SLCXCNTL sample library tutorials, 5-18
 SLCXEXEC EXEC library, 5-17
 SLCXLOAD load library, 5-17
 SLCXMENU message library, 5-17
 SLCXPENU panel library, 5-17
 system environment, 1-2
 TSO logon PROCs, 2-4
 CSS Utilities, 1-3
 allocating libraries at TSO startup, 5-16
 dynamically allocating libraries, 5-15
 installing, 5-15
 linking tutorials to main tutorial panel, 5-18
 Customer Modification Facility
 definition, G-2
 customer number, more than one, 6-2
 customer support web site, xii
 customized translation table, G-2
 CWAASDUT, G-2
 CWASSECU, G-2
 CWCMTRE load module, 5-3

CWCMTRHT load module, 5-3
 CWCMTRHU load module, 5-3
 CWCMTRVE load module, 5-3
 CWCMTRVT load module, 5-3
 CWCMTRVU load module, 5-3
 CWDDLPUT, G-2
 CWDDSUTL, G-2
 CWFXSDUT, G-2
 CWLMA copy member, 6-6
 CWPDDIO, G-2
 CWPERRM, G-2
 CWPPRMO, G-2
 CWPPRMO command
 C Language Processor, 5-29
 CWPPRTI, G-2
 CXTRACE, G-2

D

DBMODEL, G-2
 DD statements
 C Language Processor postprocessor, 5-30
 Language Processor, 2-6
 DDIO, G-2
 DDIO file, G-2
 DDIO file member, G-2
 DDIO files
 debugging production programs, 2-7
 definition, xii
 description of, 1-2
 formats, 5-19
 preparing, 5-19
 sharing with multiple CPUs, 2-4
 testing and debugging programs, 2-7
 writing listings to, 2-7
 DDIOCALC, G-2
 debugging
 production programs, 2-7
 define nodes, 6-10
 directory entry, G-2
 DIRX report, G-2
 Distributed Viewing Support
 definition, G-3
 distribution library, 2-8
 documentation, related, xii
 DVS
 customization, 4-1, 5-1
 dynamically allocated files, 2-5
 DYNCREATE, G-3

E

ECC
 component prefixes and FMIDs, 2-2
 description of, 1-1
 installation considerations, 2-3
 installation, overview of, 2-1
 overview, 1-1
 procedures, 2-1
 required components, 2-1
 selecting components, 2-1
 table of components used by products, 2-1
 ECC FMIDs, 2-2
 edit JCL screen, 6-13

enhanced listing, G-3
 Enterprise Common Components, G-3
 entry, G-3
 environment
 CSS system, 1-2
 executing
 Abend-AID, 1-4
 Abend-AID for CICS, 1-4
 CSS, 1-4
 Xpediter/CICS, 1-4
 Xpediter/TSO, 1-4

F

file access method, 5-20
 file formats
 DDIO, 5-19
 FMIDs, ECC, 2-2
 formatting
 Abend-AID for CICS transaction shared directories, 5-22
 Abend-AID report shared directories and databases, 5-21
 DDIO file, G-3
 source listing files, 5-22
 source listing shared directories, 5-21
 FrontLine support web site, xii
 full DDIO file, G-3

G

GRS considerations, 2-4

H

HCI
 APF authorization, 2-14
 description, 1-6
 installation considerations, 2-14
 help
 getting help, xiii
 horizontal translation table
 CWCMTRHE load module, 5-3
 CWCMTRHT load module, 5-3
 CWCMTRHU, 5-3
 Host Communications Interface, G-3
 host system
 transferring a License Certificate, 6-1, 6-4
 HTML documentation, xii

I

import license certificate, 6-12
 installation
 Compuware/VF, 5-15
 CSS considerations, 2-3
 CSS Utilities, 5-15
 ECC considerations, 2-3
 ECC installation overview, 2-1
 HCI considerations, 2-14

- installing and enabling LMS, 6-4
- library members, 2-7
- link lists, 2-3
- LMS considerations, 2-12
- PDS/E support, 2-3
- Security Exit program, 5-2
- translation tables, customized, 5-3
- Installation JCL Customization Facility, G-3
- Internet, Compuware WWW address, xii
- ISPF dialogs, activating, 5-15

J

- JCL
 - modifying LP JCL, 5-24
- JCL customization, 7-1

L

- Language Processor (LP), G-3
 - DD statements, 2-6
 - description of, 1-3, 2-4
 - modifying the LP JCL, 5-24
 - postprocessor, 2-6
 - preprocessor, 2-4
 - supported languages, 1-3
- LAU, G-3
 - define nodes, 6-10
 - preparing LAU screen, 6-5
 - set up, 6-5
 - using, 6-1
 - verify license file, 6-9
- License Administration Utility
 - define nodes, 6-10
 - set up, 6-5
 - using, 6-1
 - verify license file, 6-9
- license certificate, 1-5, G-3
 - importing, 6-12
 - transfer to host, 6-4
- license file, G-3
 - creating, 6-7
 - selection screen, 6-7, 6-9
 - system nodes, 6-9
 - verify, 6-9
- License Management System, G-3
 - description of, 1-1
- license verification report, 6-16
- line command, G-3
- link list considerations, 2-3
- LMINJCL sample, 6-15
- LMINPARM sample dataset, 6-14
- LMINPROC sample, 6-15
- LMS
 - administration, 2-13
 - advantages, 1-5
 - APF authorization, 2-12
 - customization, 6-1
 - establishing runtime environment, 2-13
 - execution sequence, 2-12
 - functions, 2-12
 - importing a license certificate, 6-1
 - information gathering, 1-5
 - installation, 1-6

- installation considerations, 2-12
- installation steps (table), 6-2
- installing and enabling, 6-4
- license certificate, 1-5
- license verification report, 6-16
- LMSINIT, 6-14
- main screen, 6-10
- more than one customer number, 6-2
- multiple instances, 6-1
- nodes table, 6-11
- operating environment, 1-5
- overview, 1-5
- parameter option screen, 6-10
- preliminary considerations, 6-1
- process, 1-5
- running more than one LMS, 6-1
- runtime environment, 6-1
- service bureaus, 6-2
- system nodes, 6-11
- transferring a License Certificate to host system, 6-1
- validating product access requests, 2-13
- verify proper function, 6-16

- LMSINIT
 - prepare an APF authorize LMSINIT, 6-14
 - run, 6-15
- LMVERIFY, G-3
- load modules, 5-4
 - CWCMTRHE, 5-3
 - CWCMTRHT, 5-3
 - CWCMTRHU, 5-3
 - CWCMTRVE, 5-3
 - CWCMTRVT, 5-3
 - CWCMTRVU, 5-3
- locked member, G-4
- LPAR, G-4
 - licensing, G-4

M

- manual lock, G-4
- manuals, related, xii
- MIM considerations, 2-4
- MIPS, G-4
- MSU, G-4

N

- nodes table, LMS, 6-11

O

- oldest member, G-4
- overview
 - ECC, 1-1
 - LMS, 1-5

P

- parameter option screen, 6-9
 - LMS, 6-10

PDF documentation, xii
 PDS/E support, 2-3
 PL/I Language Processor
 %NOPRINT, 2-5
 modify JCL for installation, 5-26
 postprocessor, 5-27
 preprocessor, 5-26
 postprocessor, 2-4
 Assembler Language Processor, 5-29
 Assembler Language Processor DD statement, 5-29
 C Language Processor, 5-30
 COBOL Language Processor, 5-25
 COBOL Language Processor compiler JCL, 5-25
 description of, 2-6
 PL/I Language Processor, 5-27
 preprocessor, 2-4
 Assembler Language Processor, 5-28
 automated compiler options, 2-5
 benefits, 2-5
 C Language Processor, 5-29
 capturing suppressed source code, 2-5
 COBOL Language Processor, 5-24
 description of, 2-4
 dynamically allocated files, 2-5
 handling compile errors, 2-5
 PL/I Language Processor, 5-26
 simplified JCL, 2-5
 steps, 2-5
 primary command, G-4
 Problem Documentation Utility, G-4
 PROCs, TSO logon, 2-4

R

related documentation, xii
 report databases
 creating, 5-21
 report file, G-4
 report shared directory
 creating, 5-21
 reports screen, 6-16
 REXX EXECs, 5-15
 runtime environment
 establishing LMS, 2-13

S

Security Exit Program, G-4
 description of, 1-4
 implementing, 5-2
 sequential source listing database
 sample JCL, 5-23
 server grace, G-4
 shared directory, G-4
 simplified JCL, 2-5
 SLCXCNTL
 COBOL Language Processor, 5-26–5-27
 installation library members, 5-3, 5-27
 sample library tutorials, 5-18
 SLCXCNTL installation library members, 5-3
 SLCXEXEC EXEC library, 5-17
 SLCXLOAD load library, 5-4, 5-17
 SLCXMENU message library, 5-17
 SLCXPENU panel library, 5-17

SMP/E, G-4
 overview of, 2-2
 softcopy documentation, xii
 source listing, G-4
 source listing database, G-4
 source listing databases
 creating, 5-22
 source listing file, G-4
 source listing files
 allocating and formatting, 5-22
 source listing shared directories
 allocating and formatting, 5-21
 creating, 5-21
 startup JCL for CICS, 2-3
 STROBE, G-4
 support, obtaining, xiii
 suppressed statements, G-4
 system environment
 CSS, 1-2
 system nodes
 LMS, 6-11

T

target library, 2-8
 technical support, xiii
 testing and debugging programs, 2-7
 TP subsystem, 1-6
 transaction database, G-4
 transfer license certificate to host, 6-4
 translation tables, installing custom, 5-3
 TSO logon PROCs
 CSS, 2-4
 tutorials
 Compuware/VF, 5-18
 CSS Utilities, 5-18

U

unlocked member, G-4

V

validating LMS product access requests, 2-13
 verification report (LMVERIFY), G-4
 verify license file, 6-9
 vertical translation table
 CWCMTRE load module, 5-3
 CWCMTRE load module, 5-3
 CWCMTRE load module, 5-3
 view time, G-5
 VSAM Source Listing Database
 sample JCL, 5-23

W

World Wide Web, Compuware address, xii

X

Xpediter/CICS
 executing, 1-4
Xpediter/TSO
 executing, 1-4

