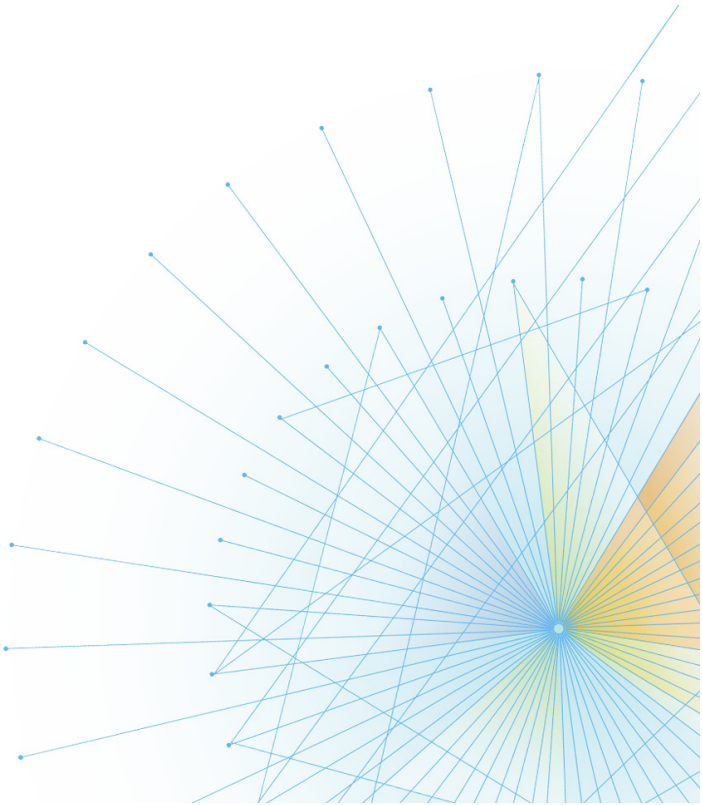
Compuware | The Mainframe Software Partner For The Next 50 Years

Enterprise Common Components Installation and Configuration Guide

Release 17.02



Please direct questions about Enterprise Common Components
or comments on this document to:

Compuware Customer Support

https://go.compuware.com/

Doc. APR2019
March 28, 2019

Contents

Introduction

This manual provides information about how to install, customize, and maintain Enterprise Common Components.

Intended Audience

Enterprise Common Components installers, z/OS system programmers and administrators.

System Environment

See the *Enterprise Common Components Release Notes* for the most current system environment information.

Related Publications

An RFN order email also includes a copy of the *Compuware Installer Mainframe Products SMP/E Installation Guide*, which should be used to perform the installation of Enterprise Common Components. Preparation for installation and post-installation configuration should be done according to this *Guide*.

The documents in the following list are available on Compuware Support Center.

- *Compuware Installer Mainframe Products SMP/E Installation Guide*
- *Enterprise Common Components Advanced Configuration Guide*
- *Enterprise Common Components Messages and Codes*
- *Compuware Shared Services User/Reference Guide*
- *Enterprise Common Components Release Notes*

Documentation Availability

Online Documentation

The Enterprise Common Components product installation package does not include the product documentation. Access the Enterprise Common Components documentation from the Compuware Support Center website at **https://go.compuware.com** in the following electronic formats:

- Release Notes in HTML format
- Product manuals in PDF format
- Product manuals in HTML format.

The product documentation is available for viewing or downloading:

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

Icons

The icons found in this guide include:

A note or tip providing additional information.

Information important to remember.

If a particular milestone or task does not apply to your site—or your site is not licensed for a particular option—you can skip ahead to the next milestone or task by clicking the icon.

Caution. Failure to follow these instructions can cause problems.

Indicates which skill set is most likely needed to perform the following task(s).

Customer Support

Visit the Compuware Support Center, **https://go.compuware.com**, to find product documentation, knowledge articles, and other technical resources. You can open a case with the Customer Solutions team, order products, and much more.

Contact Customer Solutions by phone:

- USA and Canada: 1-800-538-7822 or 1-313-227-5444.
- All other countries: Contact your local Compuware office. Contact information is available at **https://go.compuware.com**.

Visit Compuware on the web at **http://www.compuware.com** for additional product information.

Information for Customer Support

If problems arise, please check your manual for assistance. If problems persist, please obtain the following information before calling Compuware for assistance. This information will help determine the exact cause of the problem as quickly as possible.

1. Identify the release number of Compuware product(s) in use.
2. Identify the operating system.
3. Identify the release of CICS Transaction Server that is being used.
4. If an abend occurs, note the displacement and the module in which it occurs. If possible, obtain a copy of the system dump.

5. Note the sequence of steps (including all commands issued) that resulted in the problem. Also note any variable data types and programming languages involved.
6. To receive product fixes electronically, be ready to provide your email address.

ECC Overview

Enterprise Common Components (ECC) is a product delivered as a single install image for the following Compuware facilities:

- Compuware Mainframe Services Controller (CMSC)
- License Management System (LMS)
- Host Communications Interface (HCI)
- Compuware Shared Services (CSS)
- Base Services

All components of ECC are installed together using the single install image. ECC consists of a single FMID (MLCX*nnn*) for all components, operating from one authorized load library (SLCXAUTH) and one non-authorized load library (SLCXLOAD). CMSC, LMS, HCI, CSS, and Base Services are all on the same release level.

Compuware recommends that you activate and configure all the ECC components at the same time to avoid compatibility issues between different release levels. Which components of ECC you need to activate and configure depends on which Compuware products are licensed by your organization.

ECC is installed using SMP/E, facilitated by the Compuware Installer. Each ECC component can be configured individually, depending on your site's requirements.

Integration of ECC with Other Compuware Products

Compuware Mainframe Services Controller (CMSC) is required for any Compuware mainframe product, release 17.02 or higher, as those products use Compuware PARMLIB.

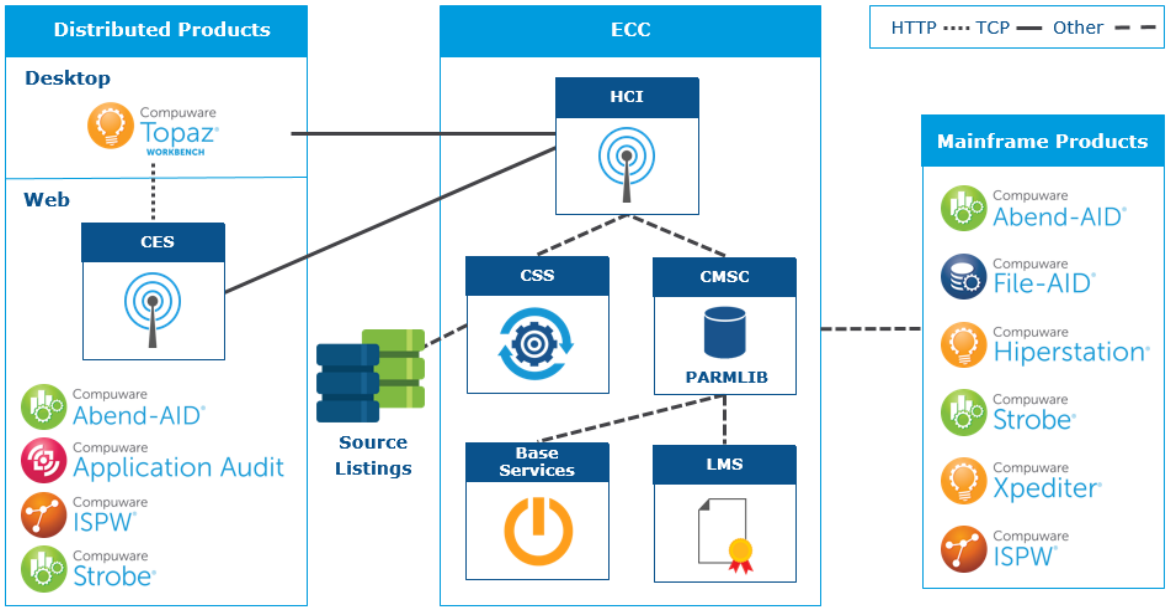
License Management System (LMS) is required for any Compuware mainframe product, in order to verify that the corresponding License Certificate exists, enabling use of the product.

Host Communication Interface (HCI) is required to facilitate communications between a distributed Compuware product (for example, Topaz Workbench) and a mainframe Compuware product(s).

Compuware Shared Services (CSS) is required for sharing compiled program information for Compuware products (for example, Xpediter).

Base Services is required to support CMSC and Abend-AID.

Product Architecture



Compuware Mainframe Services Controller (CMSC) Overview

The CMSC address space is a centralized facility providing license management and common parameter library services. The CMSC can also automatically start HCI address spaces. HCI address spaces started by the CMSC will automatically restart in the event of a failure.

As the service provider for the new Compuware's Common Parameter Library feature (Compuware PARMLIB), the CMSC reads the user's parameter libraries, stores parameters for Compuware products in memory, and initializes the parameter retrieval service used by Compuware products. Operator commands are provided to refresh the parameters in memory in the event of a change to one or more of the parameter library members.

ECC enables Compuware mainframe products to use PARMLIBs for storing configuration data. Instructions are provided to guide you through any necessary migration to using PARMLIBs.

Compuware's mainframe products, with a release of 17.02 or greater, use a common PARMLIB for specifying configuration and customization parameter values, serviced through the CMSC address space. Parameter values are stored in a PDS or concatenation of PDS libraries. The PARMLIB members are human readable and editable, allowing administrators to view or change values using an operator command. z/OS System Symbolics are supported by Compuware PARMLIB members, which can be helpful in simplifying, for example, multiple LPAR deployments.

Compuware PARMLIB to all Compuware mainframe products release 17.02 and higher, similar product information can be expressed to more than one Compuware product. For example, Strobe information can be defined for both Strobe and another product, like HCI.

Users can also specify different PARMLIB members for different invocations of individual products. For example: HCI*nn,* where *nn* identifies the PARMLIB member to be used.

For further information regarding the Common Parameter Library, review *Enterprise Common Components Advanced Configuration Guide*.

License Management System (LMS) Overview

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. LMS now executes in the CMSC address space.

Host Communications Interface (HCI) Overview

The HCI 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.

The HCI now uses a single simplified parameter member that supports all functionality in the HCI for any Compuware product including Topaz Workbench, DevEnterprise, and Strobe.

Compuware Shared Services (CSS) Overview

CSS is an enterprise-wide tool for sharing compiled program information. CSS is used by many Compuware products spanning the application debugging, application performance management, and fault management 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 used, which provide improved efficiency in maintaining information and activity within these files. CSS also provides language processing support for COBOL, PL/I, Assembler, and C.

Base Services Overview

Base Services is an application framework providing batch, server, subsystem and presentation components. Functions include dynamic allocation, storage management, transaction scheduling, message formatting, operator command routing, recovery and termination management, and screen, menu and message presentation in ISPF, VTAM, CICS and web browser environments. This framework is used by CMSC, Abend-AID and Abend-AID for CICS.

Planning

This section provides information related to planning an installation or update to Enterprise Common Components (ECC).

Steps Involved

1. Order ECC and its companion products, including the latest maintenance, via Compuware's Product Ordering web page or by calling *Compuware's License Management at 800-538-7822*. Before ordering, you should understand the system requirements and prerequisites. See Prerequisites.
2. Install ECC per the instructions provided in the order confirmation email. See Milestone 1: Install ECC.
3. Follow the instructions in this guide to configure and deploy ECC. See Milestone 2: Configuration Preparation before beginning an configuration steps. Also, for configuration information for an upgrade see Milestone 3: Configuration for an Upgrade.

Compuware Simple Deploy

When organizations have multiple environments, it necessitates constant customization when copying product datasets from one LPAR to another. Compuware's Simple Deploy allows systems programmers to do less of this manual customization after installing Compuware mainframe products.

Compuware Simple Deploy uses an API to resolve dataset names while still giving you complete flexibility of what they are called. In short, it is no longer necessary to manually edit Compuware install-dataset names in REXX and CLIST when upgrading and deploying certain Compuware mainframe products[1].

Each CLIST has been redesigned to allocate datasets using the dataset names from the DDSN*nnnn* member. If you have fully configured the DDSN*nnnn* member with all of the Compuware product DDNAME / DATASET NAMEs, you need make no changes to the CLIST in order for it to launch the product. If you do not have all of the Compuware mainframe product DDNAME / DATASET NAMES, then invoking the CLIST will flag the products that are missing. You can then modify the CLIST to resolve the offending product.

See "DDSN Configuration Parameters for Simple Deploy" section in the *Enterprise Common Components Advanced Configuration Guide* for more information on the DDSN*nnnn* member parameters

IMPORTANT:

For product-specific instructions on how configure the DDSN*nnnn* member, refer to the comments contained in the CLISTs themselves.

Member DDSN00 is shipped in the SLCXCNTL dataset andcontains DD_INFO entries for all Compuware product libraries. See Examples of DDSNnnnn Member Syntax for the DDSNnnnn member parameters and specific DD_INFO syntax examples.

1. Consult your respective Compuware mainframe product's installation and configuration guide to see if Simple Deploy is available.

Benefits of Compuware Simple Deploy

Here are four benefits of simplified deployment of Compuware mainframe products:

- **API Automation**
  You no longer have to edit Compuware install-dataset names for each LPAR every time you change the dataset name, whether for upgrading or deploying products. We're helping to automate the selection of Compuware datasets by using an API call into a PARMLIB member.
- **Centralization**
  Having Compuware file product names and DD Names in a CMSC member gives you the advantage of having a single point of knowledge and a single point of control, allowing changes to propagate everywhere. In a way, this helps break down silos of information and helps your organization drive DevOps.
- **Flexibility**
  Compuware's simple deploy will give you as much flexibility as possible by allowing the use of system symbols to provide an easily customizable and configurable deployment of Compuware products, especially across LPARs or product upgrades.
- **Cross-product Configuration**
  Currently, cross-product configuration can be difficult because different products are often installed by different people, and products are not always at the same version levels. With Compuware's simple deploy, current versions are automatically picked up and used, regardless of which person or team installed the individual products.

Examples of DDSNnnnn Member Syntax

General Syntax

```
...
DD_INFO
DDNAME=CXVJCLIB
DSNAME=SYS2.CW.VJR17A.CXVJCLIB
END

DD_INFO
DDNAME=SXVJCLIB
DSNAME=SYS2.CW.VJR17A.SXVJCLIB
END
...
```

Syntax using Symbolics

```
...
DD_INFO
DDNAME=CXVJCLIB
DSNAME=&CPWR.CXVJCLIB
END

DD_INFO
DDNAME=SXVJCLIB
DSNAME=&CPWR.SXVJCLIB
END
...
```

Syntax using the FMID parameter

```
...
DD_INFO
DDNAME=SXVJCLIB
DSNAME=SYS2.CW.VJR16A.CXVJCLIB
END

DD_INFO
DDNAME=SXVJCLIB
DSNAME=SYS2.CW.VJR17A.SXVJCLIB
FMID=MXVJ170
END
...
```

Milestones and Roles

Table 1 identifies whether a milestone is involved during a new installation and/or an upgrade. It also identifies the role or skill set required for each of those milestones. Knowing the persons involved, with some proper planning, you may be able to have certain milestones or tasks performed at the same time.

Table 1. People required for each milestone.

Milestone	New Install	Upgrade	z/OS Systems Programmer	z/OS Security Administrator
Milestone 1: Install ECC	1	1	1	
Milestone 2: Configuration Preparation	1	1	1	1
Milestone 3: Configuration for an Upgrade		1	1	
Milestone 4: Configure LMS for a New Install	1		1	
Milestone 5: Configure HCI for a New Install	1		1	
Milestone 6: Configure CMSC for a New Install	1		1	
Milestone 7: Configure CSS for a New Install	1		1	
Milestone 8: Configure Topaz Workbench Integration	1		1	
Milestone 9: Deployment	1		1	
Milestone 10: Configure Global Parameters	1	1	1	

Checklist of Milestones and Tasks

Consult the checklist of the milestones and tasks involved in the install. For convenience, you may want to print this page and check off the milestones and tasks to mark your progress through the install.

Preparation

The following are considerations to take notice of, and to remember, when doing the customization of ECC or one of its components.

ECC Considerations

OMVS Segments

- The CMSC and HCI started tasks require that an OMVS segment be defined for the user ID associated with each of these started tasks to perform various TCP/IP services.

TCP/IP Security

- The CMSC and HCI started tasks require access to the TCP/IP stack through which they communicate. This requires that these started tasks have read authority to the EZB.STACKACCESS resource class.

ECC Security

The CMSC and HCI started tasks require specific security rights. The following tables list the requirements needed for each component:

CMSC started task owner ID requires access to the following files and libraries:

Compuware PARM library	Update
Compuware license VSAM file	Update
Compuware license checkpoint VSAM file	Update
ECC authorized library (SLCXAUTH)	Read
ECC load library (SLCXLOAD)	Read

HCI started task owner ID requires access to the following files and libraries:

Compuware PARM library	Read
HCI journal VSAM files	Update
SSAS journal files (if using File-AID under Topaz Workbench)	Update
ECC authorized library (SLCXAUTH)	Read
ECC load library (SLCXLOAD)	Read
Compuware product load or authorized libraries in the STEPLIB DD	Read

Note: HCI spawns one and possibly two started subtasks that create HCI TSO address spaces instead of using the user's TSO address space for specific tasks

CXSSAS started task runs unauthorized — Primarily used for DDIO processing and Xpediter/Code Coverage analysis

CXSS0000 started task runs authorized — Primarily used by File-AID, Strobe and Topaz TSO Command processing under Topaz Workbench.

These started tasks are initiated as needed by HCI and run under a given user IDs authority (no additional access rights are required). The rights of the started task are inherited from the user ID running it.

HCI Considerations

TCP/IP

To avoid any unpredictable connection issues, make sure all current IBM maintenance is applied to TCP/IP.

Before you install the HCI on a z/OS LPAR, you must be able to run TCP/IP traffic from the workstation to that LPAR or host.

CMSC PARMLIB Information

Enterprise Common Components Installer/Administrator

As the Enterprise Common Components Installer/Administrator the following information should be recorded, as it will be needed in order to configure other Compuware mainframe product parameters.

- The name of the CMSC task running on your system when updating the CMSC PARMLIB values using the z/OS MODIFY (F) command.
- The file name of the Compuware Mainframe Services Controller (CMSC) PARMLIB dataset. You will need this name for configuring the other Compuware mainframe products.
- The name of CMSC PARMLIB member(s) DDSN*nnnn* (default is DDSN00, or the specified default in your site's CMSC start-up) with the DDNAMEs and DATASET NAMEs of all Compuware products, such as File-AID, Xpediter/TSO, Abend-AID, etc. A sample is located in *hlq*.SLCXCNTL(DDSN00). DDSN*nnnn* provides the names of Compuware run-time libraries installed at your site. You will need the DDSN*nnnn* name(s) for configuring your Compuware mainframe products.

Prerequisites

System Requirements

Hardware Platforms for Mainframe Systems

- z14, z14 ZR1
- z13, z13s
- zEC12, zBC12
- z196, z114
- z10-EC/BC
- z9-EC/BC
- z900, z990
- z800, z890

IBM Specialty Processors Enablement

- Language processor is zIIP

Operating Systems

- IBM z/OS V2.2, 2.3
- IBM ISPF for the supported z/OS releases

Languages

- IBM Enterprise COBOL for z/OS V4.2, 5.1, 5.1.1, 5.2, 5.3, 6.1, 6.2
- IBM Enterprise PL/I for z/OS V4.5, 5.1, 5.2
- IBM High Level Assembler for z/OS, z/VM, and z/VSE V1.6
- IBM z/OS XL C/C++ for the supported z/OS releases

Co-requisites – Compuware Products and Components

- Abend-AID 17.02
- Abend-AID for CICS 17.02
- Compuware Program Analyzer 18.03
- DevEnterprise 18.03
- File-AID 17.02
- Strobe 17.02, 18.02
- Xpediter/CICS 17.02

- Xpediter/TSO 17.02
- Topaz Workbench 17.02, 18.02, 18.03, 19.01, 19.02

Notes:

1. All 17.02 Compuware mainframe products require a minimum of ECC 17.02, including all maintenance.
2. Pre-16.05 releases of the ECC components (which includes LMS 4.0, HCI 3.0, and CSS 9.0) will not be supported by any Compuware mainframe products release 17.02 or greater.
3. Supported releases of Abend-AID, Compuware Program Analyzer and DevEnterprise require ECC 17.02.
4. Topaz for Total Test 17.02 or higher requires ECC 17.02.
5. If you wish to use the **REST API** and **Zero Administration** functions implemented in the CMSC, you must be using z/OS V2.2 or higher.
   - If your operating system is z/OS V2.1 (unsupported), you can apply IBM PTF UA79089 (APAR OA46622) and PTF UA76726 (APAR OA46575). These PTFs enable the IBM Web Enablement Toolkit HTTP support and JSON parsing.

Central Licenses Facility discontinued

Beginning January 1, 2019, Compuware no longer supports or issues licenses for the Central License Facility (CLF).

Milestone 1: Install ECC

ECC is a mainframe product delivered using Receive From Network (RFN) and is packaged for installation using System Modification Program Extended (SMP/E). RFN and SMP/E are IBM methods developed to transfer and install software products in the z/OS environment.

This milestone provides information and instructions for transferring the Compuware product to your mainframe and completing the SMP/E process to install the product into Global, target, and distribution zones, in preparation for configuring the product for your site's use.

Role involved: z/OS Systems Programmer

Caution: Abend-AID, Xpediter/TSO, Xpediter/CICS, and Strobe make extensive use of ECC. ECC releases are downward compatible but not upward compatible. You must ensure that the products are NOT pointing to a release of ECC that is lower than the release of ECC used during the step which loads the listing into the DDIO (whether it is part of the compile process or a standalone task). We suggest you review your product configurations to guarantee they are running with the desired release of ECC.

Task 1.1 RFN and SMP/E Install

1. Follow the instructions in the *Compuware Installer Mainframe Products SMP/E Installation Guide* to transfer and install ECC.
2. Once completed, return to this document to configure and deploy ECC.

Caution: ECC 16.05 or higher cannot be in the same Target and Distribution zones as ECC 3.3 and prior releases, Abend-AID 12.4 and prior releases, Compuware Program Analyzer 5.3, or DevEnterprise 5.3 and prior releases. However, they can co-exist in the same Global zone.

To keep your ECC software current, download and apply the quarterly cumulative maintenance file delivered through Compuware Support Center. Follow the instructions in the Compuware Installer Mainframe Products SMP/E Installation Guide to apply maintenance.

Milestone 2: Configuration Preparation

Milestone 2 tasks are required to begin the configuration for ECC, whether it is for an upgrade to an existing installation, or for a new product installation. This milestone will assist you to prepare for either upgrading your existing ECC to the current release, or to install the current release of ECC.

Role involved: z/OS Systems Programmer

Task 2.1 Verify APF-Authorize SLCXAUTH Load Library

Compuware recommends that the ECC SLCXAUTH library be APF-authorized to z/OS.

Certain ECC components require some modules to be in an APF-authorized library. Compuware has provided a load library, SLCXAUTH, that contains those modules. APF-authorize only the SLCXAUTH load library.

Instructions:

Follow IBM procedures to complete this task.

Task 2.2 Define Compuware's Common Parameter Library Dataset

Compuware's Common Parameter Library (PARMLIB) feature requires a partitioned dataset to be allocated on your system to keep PARMLIBs for those Compuware products that employ this configuration tool.

At least one Compuware PARMLIB dataset must exist in order to contain Compuware product PARMLIB members, which are accessed by CMSC. **If you have already allocated the Compuware PARMLIB dataset, skip this task. Otherwise, define a library.**

Instructions:

Modify the sample JCL member CMSCALCP, located in library SLCXCNTL, to allocate the PARMLIB dataset.

1. Update the JCL:
   a. Add a job card.
   b. Specify an appropriate dataset name for the CPWRPARM DD statement by changing the CPWR.PARMLIB name. Compuware recommends you add a high-level qualifier to the provided dataset name.
2. Submit the job and review the output to verify successful completion.

Task 2.3 Define HCI's Default User ID

The DEFAULT_USER parameter specifies the name of a valid RACF (ACF/2 or TOPSECRET) user ID to be used by the HCI when TP jobs are submitted or started and when there is no other user ID available. The user ID must be assigned within the security system and granted authority to perform

any processing the TP requires until the TP issues one of the set security calls. Until that time, the new user ID is in effect.

Example: Ensure that the default user ID has either ACCESS(READ) or ACCESS(EXECUTE) to the //STEPLIB DD used in the TP JCL and that it has appropriate access to any other datasets that it processes under the default user ID.

Once the TP has issued a security call, the default user ID is no longer used for access because the TP starts running under the user ID passed in the security call.

Task 2.4 Upgrade or New Installation

If you are **upgrading** an existing release of ECC, continue with Milestone 3: Configuration for an Upgrade.

If you are **installing** ECC for the first time, continue with Milestone 4: Configure LMS for a New Install.

Milestone 3: Configuration for an Upgrade

If you are upgrading from a previous release of ECC, follow these instructions.

This milestone will assist you in upgrading from an existing release of ECC at your site to the current release.

Role involved: z/OS Systems Programmer

Older releases of ECC—which comprised of a suite of products—must migrate some artifacts in order to be compatible with ECC 17.02. Use the following table to determine which tasks need to be completed based on the release of ECC from which you are upgrading.

Table 2. Determine which instructions to upgrade to 17.02

ECC Release	Instructions to Follow	Comments
pre-16.05	Task 3.1 Upgrade from pre-16.05 to 17.02	Use conversion utility to create PARMLIB members. Refresh of product libraries required.
16.05	Task 3.2 Upgrade from 16.05 to 17.02	Refresh of product libraries required.

Task 3.1 Upgrade from pre-16.05 to 17.02

This task will assist you in upgrading your existing pre-release 16.05 ECC software to the current release of ECC.

Older releases of ECC—which comprised a suite of products—must migrate some artifacts in order to be compatible with ECC 17.02. Use the following table to determine which tasks need to be completed based on the release of ECC from which you are upgrading

1. Starting with ECC release 16.05, LMS runs in the CMSC address space. The initialization job(s) no longer run independently.
2. ECC requires the use of Compuware PARMLIB. You should copy your pre-16.05 LMS parameters from member LMINPARM in library SLMSCNTL to member LMCL00 in the Compuware PARMLIB dataset.
3. You must delete the pre-16.05 LMS subsystem before you start the CMSC for the current LMS release. To delete the earlier LMS subsystem, specify FUNCTION(DELETE) in the LMS client PARMLIB member. **This must be accomplished using the job LMSINIT from your pre-16.05 LMS library using the specified SYSIN PARMLIB, possibly located in member LMINPARM from library SLMSCNTL.**
4. For testing ECC 17.02, after upgrading from LMS 4.0, you can override the installation of LMS 4.0 by changing the DEFAULT parameter in ECC 17.02 from DEFAULT=YES to DEFAULT=FORCE. You will also need to change the LMS subsystem name, checkpoint dataset, and exit PROC. If during testing of ECC 17.02 you encounter any issues, stop ECC 17.02 and reissue the UPDATE for LMS 4.0.

Any products that were checked out or running when the earlier LMS's subsystem was deleted should be shutdown and restarted. Also, any products that are started between the time the earlier LMS's subsystem was deleted and LMS is started in the CMSC will fail with a license error.

5. Add the LICENSE_DSNAME parameter to the newly created LMCL00 PARMLIB member. Set the value for the parameter to the name of the current license file. This parameter setting will avoid

an enqueue reserve on the license file when attempting to update a license.If you are upgrading from HCI 3.0 to ECC 17.02, Compuware provides a migration utility that converts HCI 3.0 XML parameters and their associated CSS TP configuration parameters into a PARMLIB member for use with ECC 17.02.

Modify the sample JCL member HCIJMGRT, located in the SLCXCNTL library, to execute the migration utility. The utility writes its output into the specified PARMLIB member, for example HCI00, replacing any data in the member. The output includes the parameters and their values.

a. Update the JCL:

– Add a job card.

– On the HCIXML DD statement, specify the HCI 3.0 member that contains the HCI XML parameters to be converted.

– On the TPCONFIG DD statement, specify the CSS 9.0 library that contains the TP configuration members.

– On the HCIPARM DD statement, specify the pre-allocated PARMLIB dataset and the member to which we will write the converted data.

– On the SYSTSIN DD statement, specify the library member HCIRMGRT in the EXEC statement that contains the modified JCL.

b. Submit the job and review the output PARMLIB member.

c. Make any additional changes to the new PARMLIB member according to your site's requirements for the current release of ECC.

If you are upgrading from an HCI release 2.1 and prior, you will have to manually update the PARMLIB member with your site's values.

6. Prior to ECC 16.05, Base Services was part of Abend-AID Common Components. Base Services was removed from Abend-AID 16.05 and became part of ECC beginning with release 16.05. If you are upgrading to ECC 17.02, but ***not*** upgrading Abend-AID to release 16.05 or 17.02, then the following procedures must be implemented to allow your existing Abend-AID installation to function correctly with ECC 17.02.

   You may continue to use Abend-AID 12.4 (and below) with your prior version of CSS. However, if you want to change existing Abend0-AID servers to use the ECC 17.02 load libraries then these changes will be needed:

   a. BDCAS – in the step that executes program FDBMMPLU:

      - Add the ECC 17.02 SLCXAUTH library as the first library on the STEPLIB DD statement.
      - Add the ECC 17.02 SLCXLOAD library as the first library on the FDBDRPL DD statement.
      - Remove the existing SLCXLOAD library from the FDBDRPL DD statement.

   Note: If you have steps to execute program FDBASUBS to start or stop an Abend-AID subsystem, update the STEPLIB DD statement to remove the Abend-AID Common library SKAZAUTH and replace it with the ECC 17.02 SLCXAUTH library.

   b. TDCAS – in the step that executes program FDBMMPLU:

      - Add the ECC 17.02 SLCXAUTH library as the first library on the STEPLIB DD statement.
      - Remove the HCI library SLHCAUTH library from the STEPLIB DD statement.
      - Add the ECC 17.02 SLCXLOAD library as the first library on the FDBDRPL DD statement.

- Remove the existing SLCXLOAD library from the FDBDRPL DD statement.
- Remove the HCI SLHCLOAD library from the FDBDRPL DD statement.

If you have steps to execute program FDBASUBS to start or stop an Abend-AID subsystem, update the STEPLIB DD statement to remove the Abend-AID Common library SKAZAUTH and replace it with the ECC 17.02 SLCXAUTH library.

c. VIEWING SERVER – in the step that executes program FDBMMPLU:

- Add the ECC 17.02 SLCXAUTH library as the first library on the STEPLIB DD statement.
- Remove the HCI library SLHCAUTH library from the STEPLIB DD statement.
- Add the ECC 17.02 SLCXLOAD library as the first library on the FDBDRPL DD statement.
- Remove the existing SLCXLOAD library from the FDBDRPL DD statement.
- Remove the HCI SLHCLOAD library from the FDBDRPL DD statement.

7. If you have steps to execute program FDBASUBS to start or stop an Abend-AID subsystem, update the STEPLIB DD statement to remove the Abend-AID Common library SKAZAUTH and replace it with the ECC 17.02 SLCXAUTH library.
8. If you installed CSS with your ECC pre-16.05 release, copy your existing CSS Compile Information Table to the ECC 17.02 library. The Compile Information Table is member AAUTCTBL in library SLCXTABL.
9. Review and modify started tasks (PROCs) or batch jobs to use 17.02 libraries.
    - HCIPROC
    - LMSEXIT
    - CXSS0000
    - CXSSAS
    - Any others that may have older ECC library names.
10. Because your ECC pre-16.05 release did not include the CMSC component, you need to also complete Milestone 6: Configure CMSC for a New Install to finish the upgrade.

Task 3.2 Upgrade from 16.05 to 17.02

This task will assist you to upgrade your existing ECC software, release 16.05, to the current release of ECC.

Instructions:

1. Modify started tasks (PROCS) or batch jobs to use 17.02 libraries.
    - CMSC
    - HCIPROC (only if using the HCI)
    - LMSEXIT
    - CXSS0000
    - CXSSAS
2. Remove CWLF0000 DD statement in CMSC started task.
3. Authorize the 17.02 SLCXAUTH library.

4. Execute the LMSPREP CLIST to update the CWLMA CLIST to use the 17.02 libraries. Copy the CWLMA CLIST to an appropriate CLIST system library, if desired.
5. Update the LMCLxx PARMLIB member:
   a. Change DEFAULT parameter value to FORCE.
   b. Change CHKPT_DSNAME value to a new library name.
   c. Change EXIT_PROC value to the name of the new PROC.
   d. Add LICENSE_DSNAME parameter with the value of the current license file, to avoid enqueue reserve on license file when attempting to update a license.
6. Delete the existing LMS subsystem:
   a. Update LMCLxx parameter FUNCTION to DELETE.
   b. Refresh the LMCLxx PARMLIB member in storage.
      `f cmsc,parmlib` refresh LMCL*xx* where *cmsc* is the CMSC started task name (if modified) and *xx* is the suffix for the PARMLIB member (default is LMCL00)
   c. Shutdown the LMS subsystem.
      ```
      f cmsc,lmshut client
      ```
7. Add the 17.02 LMS subsystem:
   a. Update LMCLxx parameter FUNCTION to UPDATE.
   b. Refresh the LMCLxx PARMLIB member in storage.
      ```
      f cmsc,parmlib refresh LMCLxx
      ```
   c. Shutdown the LMS subsystem.
      ```
      f cmsc,shutdown immed
      ```
   d. Restart the CMSC
      ```
      s cmsc
      ```
8. If your HCI is controlled by the CMSC, then shutdown the HCIPROC and restart the CMSC. This will execute LMS to load your Compuware licenses into storage and will start the HCIPROC. Otherwise, stop and restart your HCI started task or batch job.
9. Copy your existing CSS Compile Information Table (member AAUTCTBL) from the 16.05 SLCXTABL library to the 17.02 SLCXTABL library.

Base Services requires no customization for an upgrade other than the refresh to the current releases libraries.

Task 3.3 Configure CMSC

If CMSC had not been previously installed on your system, you will have to configure it using the instructions found in Milestone 6: Configure CMSC for a New Install. When finished, return here and continue with Task 3.4 Verification.

Instructions for Simple Deploy Users:

1. Copy the DDSN00 sample from hlq.SLCXCNTL(DDSN00) into the CPWR common PARMLIB dataset.
2. Uncomment the DD_INFO blocks for products that are installed or will be installed.
3. Update the DSNAME parm in the DD_INFO block to match the actual dataset names on your LPAR. Reminder that you can use System Symbols and the CMSC will resolve the names.

Task 3.4 Verification

This task is used to verify that your LMS subsystem lists your Compuware License Certificates and the HCIPROC was started.

Instructions:

1. **Verification of LMS subsystem**
   a. Modify the sample JCL member LMLFCRPT, located in library SLCXCNTL, to list the contents of the LMS License File.
      - Add a job card.
      - On the IMPORT step's STEPLIB DD statement, update the SLCXAUTH library to your installation's fully qualified dataset name.
      - Modify 'CPWR.LICENSE.FILE' to the name of your LMS License File.
      - Submit the job and review the output to verify successful completion. Also, verify that all your Compuware License Certificates are listed.
2. **Verification of HCI**
   a. Modify the sample JCL member HCITEST, located in library SLCXCNTL, to execute the verification job. The job will attempt to ping the HCI to validate the connection.
      - Add a job card.
      - On the IVP step's STEPLIB DD statement, update the SLCXLOAD library to your installation's fully qualified dataset name.
      - On the IVP step's SYSIN DD for both the HOSTIP and PORT:

        HOSTIP – Must contain the value of the host address in which the HCI is executing. It can be the host DNS name or the host IP address.

        PORT – Must represent the port on which the HCI is listening.
      - Submit the job and review the output.

        If a connection was established, the output should resemble:

```
Compuware Shared Services TP Connection Verification Utility

Control statement specification:
HOSTIP=MVS1.XYZCORP.COM
PORT=16196

OPEN connection     RC = 00, ERRNO=00000
SEND data           RC = 00, ERRNO=00000
RECV data           RC = 00, ERRNO=00000
SEND data           RC = 00, ERRNO=00000
RECV data           RC = 00, ERRNO=00000

Connection to the port was successfully validated.
Valid Compuware mainframe product license verified.
```

Task 3.5 Post-upgrade Considerations

Once you have completed the verification for the upgrade of ECC, you have completed the ECC upgrade. You should NOT perform Milestones 4, 5, 6, and 7, which are instructions strictly for configuring new installations. If you plan to integrate with Topaz Workbench, you should proceed

directly to Milestone 8: Configure Topaz Workbench Integration. Otherwise, you should proceed directly to Milestone 9: Deployment.

Compuware recommends using Compuware PARMLIB version 2 (PARMLIB V2). A utility is available to assist in migrating your parameters from version 1 to version 2. See Migration Utility for more information.

If you plan to integrate with Topaz Workbench, you should proceed directly to Milestone 8: Configure Topaz Workbench Integration.

If you DO NOT plan to integrate with Topaz Workbench, you should proceed directly to Milestone 9: Deployment.

Milestone 4: Configure LMS for a New Install

This milestone will describe tasks for setting up and customizing LMS for use by the Compuware products licensed to your organization.

The tasks describe only the most basic elements required to establish a Compuware LMS environment.

Role involved: z/OS Systems Programmer

Task 4.1 Transfer License Certificate to Host System

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 email by Compuware's Worldwide License Management team. In order to be used by LMS, a License Certificate must be accessible to z/OS.

Instructions:

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 dataset in an IPSF 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 many times as required.

Task 4.2 Create a License File

This task will assist you to create the VSAM License File required to store your License Certificates for all licensed Compuware software. The License File is fully compatible across all releases of LMS.

If you prefer to use the License Administration Utility (LAU) to manage your License File, directions to do so can be found in the *Enterprise Common Components Advanced Configuration Guide*.

Instructions:

Modify the sample JCL LMDELDEF, located in library SLCXCNTL, to allocate and initialize the VSAM License File.

1. Update the JCL as required:
   a. Add a job card.
   b. Modify all instances of CPWR.LICENSE.FILE to the name you prefer. We recommend just changing the high-level qualifier, CPWR.
   c. Specify a DASD volume or use VOL(*) to let the system assign one for you.
   d. On the INITLF step's STEPLIB DD statement, update the SLCXAUTH library to your installation's fully qualified dataset name.
2. Submit the job and review the output to verify successful completion.

This job performs an IDCAMS DELETE on the specified LICENSE FILE. Use with caution.

Task 4.3 Import License Certificates

This task will assist you to import your Compuware license certificates into the VSAM LMS License File.

Instructions:

Modify the sample JCL member LMIMPORT, located in library SLCXCNTL, to import Compuware software license certificates (transferred in Task 4.1 Transfer License Certificate to Host System) to the License File (created in Task 4.2 Create a License File).

1. Update the JCL:
   a. Add a job card.
   b. On the IMPORT step's STEPLIB DD statement, update the SLCXAUTH library to your installation's fully qualified dataset name.
   c. Modify 'CPWR.LICENSE.FILE' to the name specified in Task 4.2 Create a License File. Refer to LMDELDEF JCL.
   d. Modify 'CPWR.CERTIF.FILE' to the name specified in Task 4.1 Transfer License Certificate to Host System.
2. Submit the job and review the output to verify successful completion.

If you have more than one License Certificate file, repeat until all License Certificate files have been imported to the LMS License File.

Task 4.4 Verify the License File

This task will assist you to verify that the contents of the License File lists License Certificates for each Compuware software licensed to you.

Instructions:

Modify the sample JCL member LMLFCRPT, located in library SLCXCNTL, to list the contents of the LMS License File.

1. Update the JCL:
   a. Add a job card.

b. On the LMAREPT step's STEPLIB DD statement, update the SLCXAUTH library to your installation's fully qualified dataset name.

c. Modify 'CPWR.LICENSE.FILE' to the name specified in Task 4.2 Create a License File. Refer to LMDELDEF JCL.

2. Submit the job and review the output to verify successful completion. Also, verify that all your Compuware License Certificates are listed.

Task 4.5 Create LMS Exit Procedure

This task will assist you to copy, to a system PROCLIB, the LMS Exit procedure and to modify the PROC as required.

The Exit PROC is used by LMS to establish with z/OS a request for it to be notified when any of the following events occur on the LPAR:

- A dynamic CPU upgrade.
- A change to the defined capacity of the LPAR.
- The first SMF interval to expire each midnight.

Each of these system events causes LMS to be notified via a standard z/OS Event Manager Exit. This exit code examines the state of the LPAR and determines whether it should invoke the PROC named in the EXIT_PROC parameter of the LMS client parameters.

You will notice that the EXIT_PROC is invoked once each day, just after midnight (based upon your installation's SMF Interval). It is necessary for LMS to update the checkpoint dataset with values consistent with the change of date.

Instructions:

1. Copy the sample JCL member LMSEXIT, located in library SLCXCNTL, to a system PROCLIB.
2. Update the JCL: On the EVENT step's STEPLIB DD statement, update the SLCXAUTH library to your installation's fully qualified dataset name.

Task 4.6 Customize LMS Parameters

Compuware provides LMS PARMLIB member (LMCL00), located in SLCXCNTL, that include the core parameters for getting a viable LMS subsystem loaded. Compuware provides other LMS PARMLIB member (LMCLALL) that include all the possible parameters defined for use by LMS, and may include parameters with default values.

Instructions:

1. Copy the sample PARMLIB member LMCL00, located in library SLCXCNTL, to your Compuware PARMLIB dataset (created in Task 2.2 Define Compuware's Common Parameter Library Dataset.)
2. Edit the copied member (LMCL00) to provide values for the following parameters:

   – **FUNCTION=UPDATE**
   This parameter, with the UPDATE value, will define a new SUBSYSTEM_ID value to the table of subsystems.

   – **LICENSE_DSNAME**
   This parameter specifies the license file dataset name to be used.

   – **SITE**
   This parameter specifies a site number that is required for LMS execution. This number tells LMS which site to load from the license file into the license cache.

DD statements must be included in the LMS JCL named CWLFxxxx DD, where xxxx can be any alphanumeric characters.

- **SUBSYSTEM_ID**
  This parameter specifies the subsystem identifier that this invocation of LMS will process. This value can contain only uppercase letters and numbers.

  Although the operating system allows subsystem names to contain lowercase letters and special characters, LMS does not because the subsystem name may need to be included in the DDNAME of a `CWIDssss DD DUMMY` statement. Therefore, the value chosen for this parameter must also be valid in a DDNAME.

  You must coordinate the use of the subsystem identifier, because duplicate names are not allowed, either within the set of License Management subsystems, or across all subsystems defined. Verify that the subsystem value chosen is not already in use.

- **DEFAULT**
  This parameter specifies whether this SUBSYSTEM_ID will be declared the default subsystem. Your choices are NO|YES|FORCE.

- **CHKPT_DSNAME**
  This parameter specifies the name of a checkpoint dataset. LMS requires a checkpoint dataset to be available for LMS and to certain operating system exits at all times. This dataset is created automatically by LMS if it does not already exist, or it is updated if it does exist. You must specify the 1- to 38-character name of this dataset.

  Note: The CMSC user ID (used to run LMS) must have ALTER access to the entity named by this parameter. The IDCAMS utility is dynamically invoked to define this dataset. Note that the same user ID is used for LMSEXIT.

- **EXIT_PROC**
  This parameter specifies the 1- to 8-character member name in your system PROCLIB, of the PROC which LMS automatically starts whenever an event occurs which drives the event notification exit. The events which will cause this PROC to be started are:
  — A dynamic CPU upgrade (CPU upgrade on demand).
  — A change to the defined capacity of the LPAR.
  — The first SMF interval to expire each midnight.

  Note: Refer to Task 4.5 Create LMS Exit Procedure for additional information.

- **SMF_ID**
  This parameter specifies the SMF Record ID number to be used on all SMF records written by LMS. The SMF_ID must be a three-digit number in the range of 128 through 255.

  If SMF_ID is not specified, then SMF recording will not take place regardless of the specification on any license file concerning this recording. Therefore, this parameter is required before any SMF recording can occur.

  You must coordinate the use of SMF identifiers so that no duplicate numbers are defined to any program or system writing SMF records. Verify to ensure that the SMF_ID value chosen is not already in use.

  Note: Specifying an SMF_ID does not automatically start SMF recording. SMF recording is performed on a product-by-product basis and is activated via an indicator within the file itself. This parameter and its value only state which SMF_ID will be used when SMF recording is activated.

3. To ensure access to Compuware products is automatically enabled, establish a procedure in your system PROCLIB to launch your CMSC as a started task during IPL and IML processing.

4. Specify the license file dataset name in the LMCL00 module, under the LICENSE_DSNAME parameter.
5. If you changed your LMS PARMLIB member name from the default LMCL00 (for example: LMCL0005), update your CMSC PARMLIB member to contain the suffix of the new LMS PARMLIB member, **LMCL=0005**.

Note: For the complete list of LMS parameters, see the *Enterprise Common Components Advanced Configuration Guide*.

If you have any Compuware products that use Host Communication Interface (HCI), continue with the tasks in Milestone 5: Configure HCI for a New Install—otherwise, skip to Milestone 6: Configure CMSC for a New Install.

Milestone 5: Configure HCI for a New Install

This milestone will assist you to customize the mainframe components needed to support the host communication from the mainframe. You do not need to configure an HCI or its started tasks unless you are using Strobe, Topaz, or any of Compuware's Eclipse-based plug-ins.

Skip this milestone if you do not have Compuware products that require HCI on this LPAR. Continue configuration with Milestone 6: Configure CMSC for a New Install.

Role involved: z/OS Systems Programmer

Task 5.1 Define Journals

This task will assist you to allocate HCI and SSAS journal datasets. These journals are used by the HCI and SSAS jobs to log processing requests and to provide trace information that may be required by Compuware for problem determination and resolution. The SSAS journals are only necessary if you are going to deploy the SSAS address space, to enable Compuware DDIO file support in Topaz or the Code Coverage/Eclipse plug-in for Topaz Workbench. See Milestone 8: Configure Topaz Workbench Integration.

Instructions:

Modify the sample JCL member HCIJOURN, located in library SLCXCNTL, to define the VSAM journals. The job will use IDCAMS to define and initialize the journal datasets. Compuware recommends defining three journals for HCI use and two journals for SSAS use, as specified in the sample JCL.

1. Update the JCL:
   a. Add a job card.
   b. On the ALLOCJH step's DEFINE statements, provide fully qualified dataset names. Compuware recommends adding a high-level qualifier to the specified name.
   c. On the ALLOCJC step's DEFINE statements, provide fully qualified dataset names. Compuware recommends adding a high-level qualifier to the specified name.
2. Submit the job and review the output to verify successful completion.

Task 5.2 Implement HCI Procedures

This task will create the PROCs required to start the HCI and to provide support to the HCI. The three PROCs that will be created are:

- The HCIPROC — is the HCI started task procedure. Optionally, the HCI can be run via a batch job, typically to verify configuration. Compuware recommends that the HCIPROC run as a started task that is managed by the CMSC.
- The CXSSAS PROC — is the Shared Services (CSS) Address Space procedure. The CXSSAS PROC is primarily used for DDIO processing and for Xpediter/Code Coverage analysis. This PROC is designed to execute unauthorized.

- The CXSS0000 PROC — is the CSS TP procedure. The CXSS0000 PROC is used to support File-AID and Strobe processing. This PROC is designed to execute authorized.

Instructions:

Modify the sample JCL member HCIPROCS, located in library SLCXCNTL, to create three PROCS.

1. Update the JCL:
   a. Add a job card.
   b. On the HCIPROC step's SYSUT2 DD statement, specify a system PROCLIB where the job will write the three PROCs. In the sample JCL, SYS1.PROCLIB is specified, but can be changed to an appropriate PROCLIB for your installation.
   c. For the HCIPROC member, on the STEP1 step's STEPLIB DD statement, update the SLCXAUTH library to your installation's fully qualified dataset name and specify a proc name.
   d. For the CXSSAS member, on the IEFPROC step's STEPLIB DD statement, update the SLCXLOAD library to your installation's fully qualified dataset name and specify a proc name.
   e. For the CXSSAS member, the PARM &TPNAME is a value generated by the CSS TP internally and should not be modified.
   f. For the CXSS0000 member, on the IEFPROC step's STEPLIB DD statement, update the SLCXAUTH library and proc to your installation's fully qualified dataset name.
2. Submit the job and review the output to verify successful completion.

Task 5.3 HCI Parameter Customization

This task will assist you to customize those parameters required to successfully start the HCI.

Compuware provides a default HCI PARMLIB member (HCI00), located in SLCXCNTL, that includes the core parameters for getting a viable HCI started. Compuware provides a second HCI PARMLIB member (HCIALL) that includes *all* the possible parameters defined for use by HCI, and may include parameters with default values.

Instructions:

1. Copy the sample HCI PARMLIB member (HCI00), located in library SLCXCNTL, to your Compuware PARMLIB dataset (created in Define Compuware's Common Parameter Library Dataset).
2. Edit the copied member (HCI00) to provide values for the following parameters:

   SYSID=HCI0

   This parameter specifies your subsystem ID. *SYSID* must be four characters in length and start with an alpha character or #, $, @. The *SYSID* is used by the HCI initialization routines to dynamically create a z/OS subsystem. It must be unique among all subsystems running on this LPAR and must not be defined in a system PARMLIB.

   JOURNAL*n*=HCI.&SYSNAME..JOURNAL*n*

   These parameters (one for each HCI journal defined in Define Journals) specify journal datasets that allow for logging of HCI activity. The HCI uses the Journaling Facility to write out diagnostic information that may then be provided to Compuware.

   Example: `JOURNAL1=HCI.&SYSNAME..JOURNAL1`

Journal datasets cannot be shared across multiple HCI instances. Each HCI instances requires its own set of journal datasets.

HCI_CONNECTION

– PORT=16196

  This parameter specifies a port number in which the HCI listens for incoming connection requests. Check with your network administrator for an available port number.

PORT_CONFIG=16196

This parameter denotes the start of a CSS TP port configuration and must come *after* any required HCI parameter statements.

All parameters added to the HCIxx PARMLIB member need to come *after* the cooresponding PORT_CONFIG for which they pertain.

NOTIFY=YES

This parameter enables the job notification feature. This feature provides for end-of-job notifications that are generated by the NOTIFY=*userid* JCL parameter to be sent to the user. The NOTIFY=*userid* JCL parameter is used on JOB statements.

Normally, end-of-job notifications are sent to the TSO user. This remains in effect, regardless of whether the NOTIFY feature is enabled or disabled. Enabling the feature allows a copy of the same end-of-job notification to be sent to the user's Topaz Workbench instance, if started.

NOTIFY_PURGE=2

This parameter provides an automatic mechanism to purge messages that cannot be delivered to a Workstation user after a set period of days. Messages might be undeliverable because the userID associated with the message never uses Topaz Workbench, or the userID has not used Topaz Workbench for a period longer than the purge duration period. Purging undeliverable messages reduces overhead and keeps the storage required to maintain the undelivered messages to a reasonable level. The NOTIFY_PURGE duration value is in days.

STASK=CXSS0000

This parameter specifies the CSS TP Started Task procedure name. The STASK is required for CA-Endevor, File-AID Data Privacy and/or File-AID/Eclipse plug-in support.

SSAS_PROC=CXSSAS

This parameter is the Shared Services Address Space procedure name. The SSAS_PROC parameter identifies the JCL procedure name used by the Compuware Shared Services Address Space (SSAS).

SSAS_NAME=CXSSAS00

This parameter is the Shared Services Address Space. The SSAS_NAME identifies the address space name to be used by the Compuware Shared Services Address Space (SSAS). This address space is intended to run unauthorized functions on behalf of Topaz Workbench.

LOCAL_ENV

– PORT=16196

  This parameter identifies the port number of the HCI that owns this configuration section.The port number example of 16196 is the port number of the TP assigned to service Topaz Workbench plug-in connections for this HCI image. This should be the same port number that is specified on the corresponding PORT_CONFIG statement.

– HOST=&SYSNAME..COMPUWARE.COM

This parameter identifies the host name or the IP address of the HCI that owns this configuration section. If you use an IP address it must be in IPV4 format.

– ZIIP=N

This parameter can enable the zIIP processor of the HCI that owns this configuration section. This parameter is set to **N** by default, but can be changed to **Y** to enable the CSS TP to execute selected portions of TP activity on an available zIIP processor.

HCI_ENV

– PORT=16196

This parameter is the port number of the CSS TP for this HCI image.

You must code only one HCI_PORT statement in order to support Host Explorer connections to this LPAR, and it is the same HCI as that coded for LOCAL_* statements.

– HOST=&SYSNAME..COMPUWARE.COM

This parameter specifies the host name or IP address of the operating system image where the HCI executes. If you use an IP address it must be in IPV4 format.

– SYSNAME=CPWR-TP-&SYSNAME

This parameter is the user-specified description for the HCI connection point for a given LPAR.
You must code only one HCI_SYSNAME statement in order to support Host Explorer connections to this LPAR, and it is the same HCI as that coded for LOCAL_* statements.

– LPAR=&SYSNAME

This parameter is the name of the LPAR that this HCI image will be executing.

You must code only one HCI_LPAR statement in order to support Host Explorer connections to this LPAR, and it is the same HCI as that coded for LOCAL_* statements.

For a complete list of HCI parameters, see the *Enterprise Common Components Advanced Configuration Guide*.

3. If you changed your HCI PARMLIB member name from the default HCI00 (for example: HCI0005), update your CMSC PARMLIB member to contain the suffix of the new HCI PARMLIB member, HCI=0005.

Figure 1. Example HCI PARMLIB member configured for *two* ISPW instances

```
*************************************************************************
*         HOST COMMUNICATIONS INTERFACE (HCI) PARAMETER FILE            *
*************************************************************************
SYSID=HCIO                              HCI SUBSYSTEM NAME
JOURNAL1=HCI.&SYSNAME..JOURNAL1         JOURNAL1 DATASET
JOURNAL2=HCI.&SYSNAME..JOURNAL2         JOURNAL2 DATASET
JOURNAL3=HCI.&SYSNAME..JOURNAL3         JOURNAL3 DATASET
*************************************************************************
*                    HCI CONNECTION DEFINITIONS                         *
*************************************************************************
HCI_CONNECTION                          HCI CONNECTION DEFINITION ENTRY
  PORT=17100
END
*************************************************************************
*                    HCI CONNECTION DEFINITIONS                         *
*************************************************************************
HCI_CONNECTION                          HCI CONNECTION DEFINITION ENTRY
  PORT=17200
END
*************************************************************************
*                  PORT DEFINITION CUSTOMIZATION                        *
*************************************************************************
PORT_CONFIG=17100                       PORT CUSTOMIZATION DEFINITION
NOTIFY=YES                              JOB NOTIFICATIONS (YES/NO)
NOTIFY_PURGE=2                          DAYS TO PURGE UNSENT MESSAGES
STASK=CXSS0000                          PROCNAME: TP STARTED TASK
SSAS_PROC=CXSSAS                        PROCNAME: SHARED SERVICES A/S
SSAS_NAME=CXSSAS00                      SHARED SERVICES A/S NAME
*************************************************************************
*                      LOCAL SPECIFICATIONS                             *
*************************************************************************
LOCAL_ENV
  PORT=17100                            LOCAL PORT
  HOST=&SYSNAME..COMPUWARE.COM          LOCAL HOST
  ZIIP=NO                               ZIIP ENABLEMENT (YES/NO)
END
*************************************************************************
*                      SERVICE DEFINITIONS                              *
*************************************************************************
HCI_ENV
  PORT=17100                            HCI PORT
  HOST=&SYSNAME..COMPUWARE.COM          HCI HOST NAME
  SYSNAME=CPWR-HCI-&SYSNAME             20 CHARACTER SYSNAME
  LPAR=&SYSNAME                         HCI LPAR NAME
END
*************************************************************************
*                     ISPW CONFIGURATION                                *
*************************************************************************
ISPW_ENV
   HOST=CW09.COMPUWARE.COM
   PORT=3500
END
*************************************************************************
*                     PORT CONFIGURATION=17200                          *
*************************************************************************
PORT_CONFIG=17200                       PORT CUSTOMIZATION DEFINITION
NOTIFY=YES                              JOB NOTIFICATIONS (YES/NO)
NOTIFY_PURGE=2                          DAYS TO PURGE UNSENT MESSAGES
STASK=RJA10000                          PROCNAME: TP STARTED TASK
SSAS_PROC=RJA1SSAS                        PROCNAME: SHARED SERVICES A/S
SSAS_NAME=RJA1SSAS                      SHARED SERVICES A/S NAME
*************************************************************************
*                      LOCAL SPECIFICATIONS                             *
*************************************************************************
LOCAL_ENV
  PORT=17200                            LOCAL PORT
  HOST=&SYSNAME..COMPUWARE.COM          LOCAL HOST
  ZIIP=NO                               ZIIP ENABLEMENT (YES/NO)
END
*************************************************************************
*                      SERVICE DEFINITIONS                              *
*************************************************************************
HCI_ENV
  PORT=17200                            HCI PORT
  HOST=&SYSNAME..COMPUWARE.COM          HCI HOST NAME
  SYSNAME=CPWR-HCI-&SYSNAME             20 CHARACTER SYSNAME
  LPAR=&SYSNAME                         HCI LPAR NAME
END
*************************************************************************
*                     ISPW CONFIGURATION                                *
*************************************************************************
ISPW_ENV
  HOST=CW09.COMPUWARE.COM
  PORT=3600
END
```

Task 5.4 HCI Verification (HCITEST)

This task verifies whether HCI was configured correctly by testing the port for connectivity.

If you plan is to have CMSC control the HCIPROC[1], then complete Milestone 6: Configure CMSC for a New Install now, then return to this task to complete the HCI verification.

Instructions:

1. Modify the sample JCL member HCITEST, located in library SLCXCNTL, to execute the verification job. The job will attempt to ping the HCI to validate the connection.
   a. Add a job card.
   b. On the IVP step's STEPLIB DD statement, update the SLCXLOAD library to your installation's fully qualified dataset name.
   c. On the IVP step's SYSIN DD for both the HOSTIP and PORT:
      - HOSTIP – Must contain the value of the host address in which the HCI is executing. It can be the host DNS name or the host IP address.
      - PORT – Must represent the port on which the HCI is listening.
2. Submit the job and review the output.

- If a connection was established, the output should resemble:

```
Compuware Shared Services TP Connection Verification Utility
Control statement specification:
HOSTIP=MVS1.XYZCORP.COM
PORT=16196

OPEN connection     RC = 00, ERRNO=00000
SEND data           RC = 00, ERRNO=00000
RECV data           RC = 00, ERRNO=00000
SEND data           RC = 00, ERRNO=00000
RECV data           RC = 00, ERRNO=00000

Connection to the port was successfully validated.
Valid Compuware mainframe product license verified.
```

- If a connection cannot be established, or there are error conditions present, the output resembles:

```
Compuware Shared Services TP Connection Verification Utility

Control statement specification:
HOSTIP=MVS1.XYZCORP.COM
PORT=16196

OPEN connection     RC = 00, ERRNO=00000
SEND data           RC = 00, ERRNO=00000
RECV data           RC = 28, ERRNO=00054

The connection was unexpectedly terminated. Possible causes:
* HCI configuration macros HCICNPCB or HCICNTPT are not correctly set.
* The PORT number specified is a secure port via AT-TLS.
* The PORT number specified is an active port, but is in use by another kind of
application.
```

- Different messages could be in the output depending on the exact return code received during execution.

1. Compuware does NOT recommend that the CMSC control the HCI when an automated procedure such as a JES3 reader file, or an automated operations product, is being used to shut down both the CMSC and HCI.

- If a connection to the HCI is established, messages will be issued and will be visible in the HCI job log. The messages might resemble:

```
CXTPMAI009I 0001 Init CXTP01   HOBB24DD
CXTPCFG019I 0001 Configuration file CXTP01   successfully processed
CXTPMAI043I 0001 Ping from 10.10.0.200
CXTPMAI007I 0001 Term RC=00000000 FDBK=0 Reason=00000000
```

It is recommended that once the HCI is implemented, it should be set up to run as a started task and included in the IPL startup sequence.

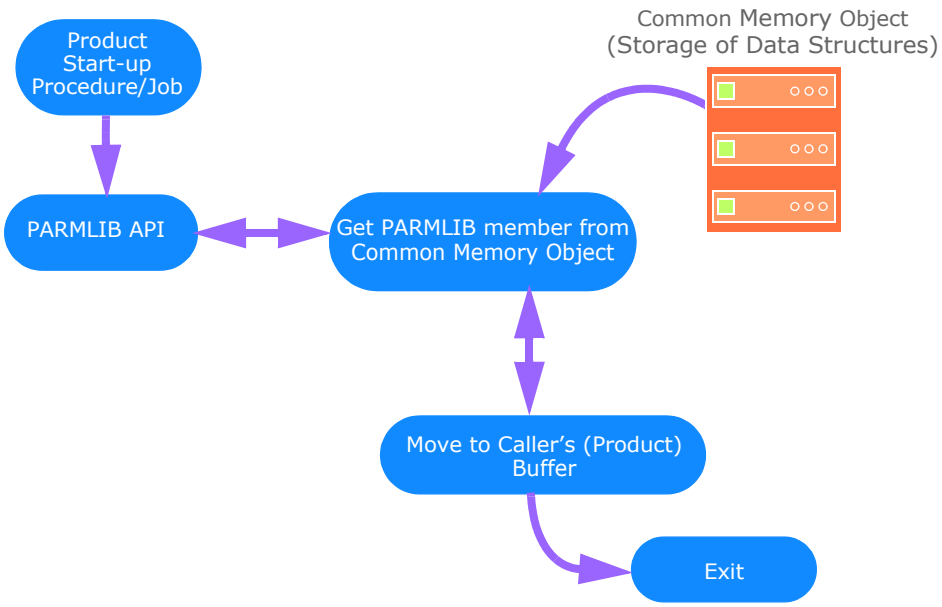
Milestone 6: Configure CMSC for a New Install

This milestone will assist you to customize the CMSC, which will provide support to all the other Compuware mainframe products that require the use of the Compuware PARMLIB.

Role involved: z/OS Systems Programmer

The CMSC address space is a centralized facility providing license management and Common Parameter Library Services for Compuware mainframe products. The CMSC also has the ability to automatically start HCI address spaces. HCI address spaces started by the CMSC will automatically restart in the event of a failure

Figure 2. Compuware Mainframe Product Parameter Retrieval flowchart



Task 6.1 Preparing for Simple Deploy

For Compuware Simple Deploy feature users only (recommended).

1. Copy the DDSN00 sample from *hlq*.SLCXCNTL(DDSN00) into the CPWR common PARMLIB dataset.
2. Uncomment the DD_INFO blocks for products that are installed or will be installed.
3. Update the DSNAME parm in the DD_INFO block to match the actual dataset names on your LPAR. Reminder that you can use System Symbols and the CMSC will resolve the names.

4. Modules MSCUDDRX and MSCUDDCL must be made available to the TSO user executing the Compuware CLISTs and REXX execs. This can be accomplished with one of the following options.

   Option 1: (*Recommended)* Put the SLCXLOAD dataset (or modules MSCUDDRX and MSCUDDCL) in the STEPLIB concatenation in the TSO LOGON procedure for users of Compuware CLISTs and REXX execs.

   Option 1 is specifically recommended if your system is set up with LNKAUTH=LNKLST (which is the default setting) in SYS1.PARMLIB member IEASYSxx.

   Option 2: Copy modules MSCUDDRX and MSCUDDCL to a library that is on LNKLST.

   Option 3: Put the SLCXLOAD dataset on LNKLST.

Task 6.2 Customize CMSC Parameters

Compuware provides a default CMSC PARMLIB member (CMSC00), located in library SLCXCNTL, that includes the core parameters for getting a viable CMSC started. Compuware provides a second CMSC PARMLIB member (CMSCALL), also located in library SLCXCNTL, that includes all the possible parameters defined for use by the CMSC, and may include parameters with default values.

Instructions:

1. Copy the sample PARMLIB member CMSC00, located in library SLCXCNTL, to your Compuware PARMLIB dataset (created in Task 2.2 Define Compuware's Common Parameter Library Dataset).
2. Edit the copied member (CMSC00) to provide values for the following parameter:

   CMSC_ID=CMSC
   This parameter specifies the name of the CMSC. This parameter must be a 1 to 4-character string. The string must consist of letters (A-Z), digits (0-9), or national symbols (@, #, $). It must begin with a letter or national symbol.

   CMSC_ID=CMSC must be specified for the primary CMSC.

   Note: For other optional CMSC parameters, see the *Enterprise Common Components Advanced Configuration Guide*.

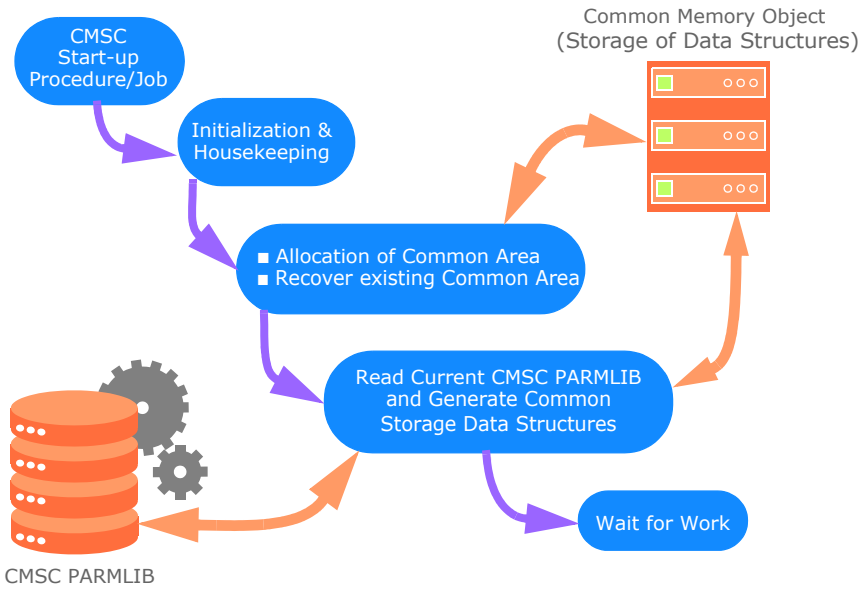
If you defined each of the LMCL, HCI, and CSS PARMLIB members with the default suffix 00 (LMCL00, HCI00, and CSS00), then there is no need to add these PARMLIB member names to the CMSCxx PARMLIB member, because CMSC automatically assumes these default member names.

Task 6.3 Start the CMSC

This task will assist you to copy, to a system PROCLIB, the sample CMSC JCL. This JCL, once customized, will be used to start the CMSC. The CMSC can be submitted as a batch job, however, Compuware recommends that the PROC be setup as a started task and added to your installation's IPL process.

Member CMSC in the SLCXCNTL library contains sample JCL to run the CMSC. Copy the sample JCL to your system PROCLIB, follow the instructions in the member and modify the JCL to your site's requirements. The JCL can be submitted as a batch job, or can be executed as a started task.

Figure 3. CMSC Initialization Process



Instructions:

1. Copy the sample JCL member CMSC, located in library SLCXCNTL, to a system PROCLIB.
2. Update the JCL:
   a. On the CMSC00 EXEC statement, update the PARM:

      `PARM='name,CMSC,ssss'`

      – *name* is the name of the CMSC. This can be any unique name of your choosing, up to eight characters in length. The only restrictions are that it must begin with a letter or national character, and contain letters, digits, or national characters.

      – *CMSC* identifies the address space being initialized by this JCL as a CMSC.

      – *ssss* specifies the one to four-character suffix for the CMSC PARMLIB member, discussed in *Task 6.1*. This parameter will be concatenated with the prefix 'CMSC' to construct the name of the PARMLIB member to be used for the CMSC startup parameters. The suffix must follow the rules for PDS members.

   b. On the CMSC00 step's STEPLIB DD statement, update the SLCXAUTH library to your installation's fully qualified dataset name.

   c. On the CMSC00 step's FDBDRPL DD statement, update the SLCXLOAD library to your installation's fully qualified dataset name.

   d. On the CMSC00 step's CWPARM statement, update the CPWR.PARMLIB to your installation's fully qualified Compuware PARMLIB dataset name.
3. Start the PROC as a started task or for testing purposes, submit as a batch job.

Task 6.4 Verifying CMSC

This task will assist you to verify that the CMSC is executing, without errors, and that the LMS subsystem was started.

Instructions:

1. View the executing CMSC's job output:
   a. Look for the following messages in the FDBDLOG:

```
FDBRC1999I REGION INITIALIZATION COMPLETE FOR SERVER CMSC
```

   This message signifies that the CMSC Server was initialized and is ready.

```
MSCSC2512I THE LMS TASK HAS TERMINATED. DETACHING.
```

   This message indicates the LMS task that started the LMS subsystem has ended and is detaching.

   b. Look for the following messages in the LMCPRINT:

```
LM6012I UPDATE OF CHECKPOINT DATA SET COMPLETED
LM5020I SUBSYSTEM LMS1 HAS BEEN UPDATED ON SYSTEM CW99
LM5003I HIGHEST RETURN CODE WAS 00000000 AND REASON CODE WAS 0
```

   These messages signify that the LMS subsystem was successfully loaded into the CMSC address space and is available for license validation by the Compuware product requiring it.

If you completed this milestone as part of Task 3.3 Configure CMSC please continue your ECC upgrade with Task 3.4 Verification.

If the CMSC controls your HCIPROC, return to Task 5.4 HCI Verification (HCITEST)to complete the verification of HCI. Once the HCI verification is complete, continue with Milestone 7: Configure CSS for a New Install.

Compuware does NOT recommend that the CMSC be used to control the HCI when an automated procedure such as a JES3 reader file, or an automated operations product, is being used to shut down the CMSC and HCI.

Milestone 7: Configure CSS for a New Install

This milestone will assist you to configure CSS for use with Compuware products that utilize the CSS functions.

Skip this milestone if you do not have Compuware products that require CSS on this LPAR. Products include:

- Abend-AID
- Abend-AID for CICS
- Xpediter/CICS
- Xpediter/TSO and Xpediter/IMS
- Xpediter/Code Coverage
- Strobe
- Topaz Workbench plug-ins dependent upon the above mainframe products.

Continues with Milestone 8: Configure Topaz Workbench Integration

Role involved: z/OS Systems Programmer

Task 7.1 Make CSS Load Modules Accessible to Compuware Products

This task allows other Compuware products that need the CSS modules to have access to them.

Instructions:

Ensure that the CSS load modules can be accessed by Compuware products by placing the ECC load library in the STEPLIB or ISPLLIB for individual TSO logon PROCs, and in the STEPLIB or JOBLIB for batch jobs. The ECC load library (SLCXLOAD or a copy of it) should be concatenated **in front** of any Compuware product libraries.

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 ECC load library in front of the Compuware product in that library concatenation.

You can dynamically allocate the ECC load library as part of the ISPLLIB concatenation. Please refer to **Option 1** listed in Create the Compile Information Table for more information.

Task 7.2 Create the Compile Information Table

This task will create the compile information table, which is stored in SLCXTABL and used by the CSS Utilities. This table contains information about your compilers, compiler libraries, DB2 pre-compile libraries, CICS translator libraries and other 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 in Activate Compuware ISPF Dialogs.

This task is only required if

- Your Compuware product run-time libraries currently being configured are in a different DDSN*nnnn* member than the default member, as specified in your site's CMSC start-up
- Or you are not using the CMSC Simple Deploy facility.

If you are not required to create a Compile Information Table, skip to the next task: Activate Compuware ISPF Dialogs

Instructions:

1. Edit member CXUECIN2 in the SLCXCNTL library. To avoid having your modifications overlaid by future maintenance, you should copy this 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 ECC panel library, SLCXPENU, and your ECC REXX EXEC/CLIST library, SLCXEXEC.
2. If you placed the member CXUECIN2 in a library that is in your SYSPROC concatenation, then you may invoked the REXX EXEC as follows:

   ```
   TSO %CXUECIN2
   ```

   Otherwise, you may invoke it as follows:

   ```
   TSO EXEC 'your.dataset.library(CXUECIN2)'
   ```

Figure 4. The first panel displayed is the 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 the information entered will be saved in this table. If the file does not exist it will be allocated for you. The second field is used during the COPY command. See Option C. Copy a Prior Installation's Information to a New Library.

Option 1. Select Compilers

This option displays the available compilers that the Compile Facility supports. In the Select Compilers screen (Figure 5 and Figure 6), specify **Y** next to each compiler that will to be used with the Compile Facility.

This option must be specified first, with at least one compiler selected.

Figure 5. 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 6. 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
```

Enter Y 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.

Option 2. Enter Compile Libraries

This option allows you to supply the library names for the compiler you selected in Option 1. Select Compilers. The compilers that you selected will have their corresponding arrow indicators highlighted and the field names will be modifiable.

Figure 7. Compile Information Table - Enter Compiler Libraries (1 of 5)

```
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 8. Compile Information Table - Enter Compiler Libraries (2 of 5)

```
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      ===>
                       ===>

  PLI/370              ===>
                       ===>
  Linkedit SYSLIB      ===>

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

Figure 9. Compile Information Table - Enter Compiler Libraries (3 of 5)

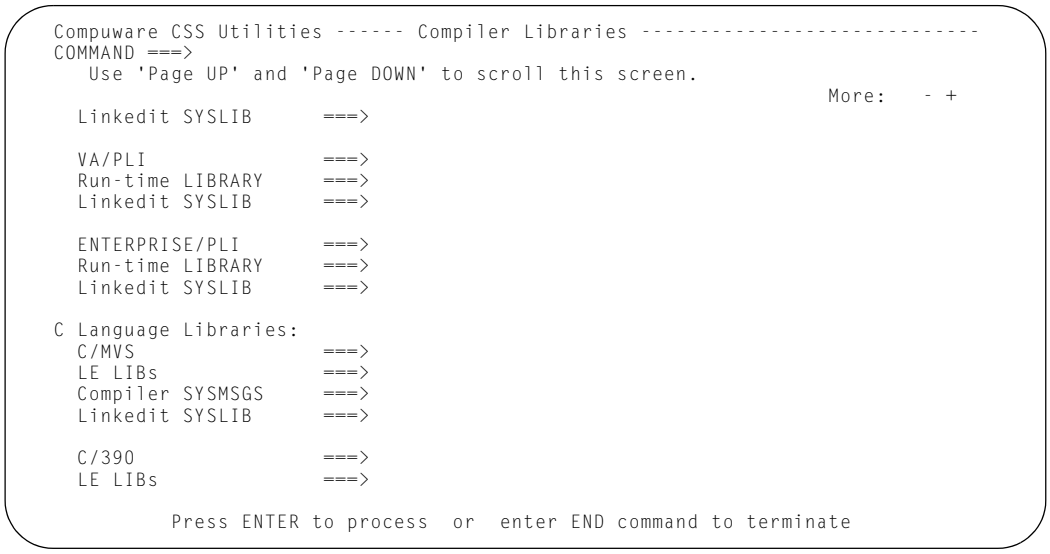


Figure 10. Compile Information Table - Enter Compiler Libraries (4 of 5)

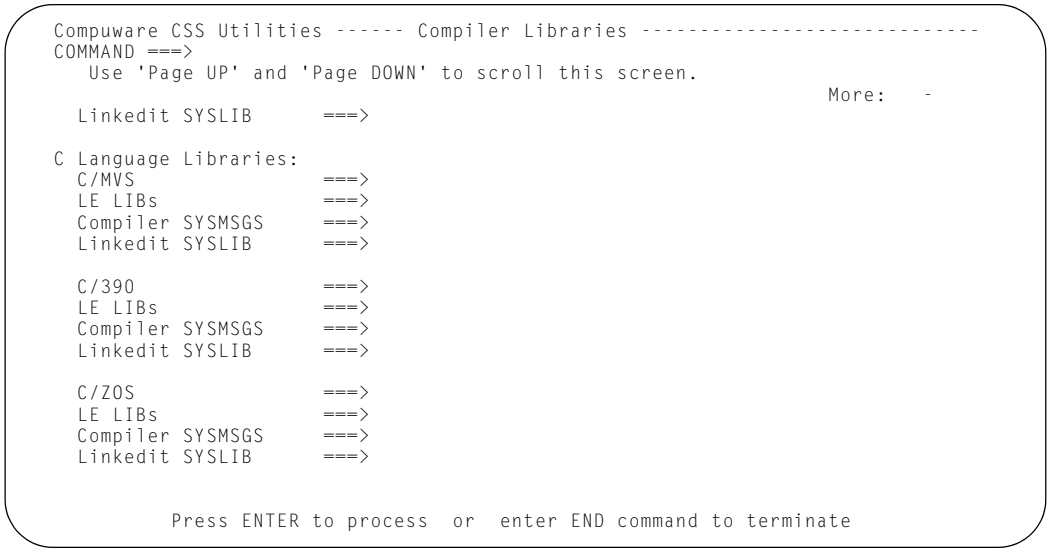


Figure 11. Compiler Libraries – screen 5 of 5

```
Compuware CSS Utilities ------ Compiler Libraries --------------------------
COMMAND ===>Use 'Page UP' and 'Page DOWN' to scroll this screen.
   Compiler libraries appear in the STEPLIB DD of compile JCL.
   SYSLIB libraries appear in the SYSLIB DD of the Linkedit step.
                                                              More:     +

  High Level Assembler===>

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


PLI/370               ===>
                      ===>
Linkedit SYSLIB       ===>

VA/PLI                ===>
Run-time LIBRARY      ===>
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.

Option 3. Enter DB2 IDs and Pre-compile Libraries

If your site has DB2, this option will allow you to define DB2 identifiers and associate the identifiers with DB2 libraries. When an identifier is selected, the associated datasets will appear in the STEPLIB DD of the pre-compile step and in the SYSLIB DD of the linkedit step. The DB2 IDs will appear on the Compile Menu.

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 in Figure 12 type in the DB2 dataset names that are required to perform the DB2 pre-compile step.

Figure 12. Compile Information Table - Enter DB2 IDs and Pre-compile Libraries

```
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
```

For example:

```
(1) ===> D910S1 ===> 'SYS1.DSN910.SDSNLOAD'
(2) ===> D910S1 ===> 'SYS1.DSN910.SDSNEXIT'
(3) ===> DA10S1 ===> 'SYS1.DSNA10.SDSNLOAD'
(4) ===> DA10S1 ===> 'SYS1.DSNA10.SDSNEXIT'
(5) ===> DB10S1 ===> 'SYS1.DSNB10.SDSNLOAD'
(6) ===> DB10S1 ===> 'SYS1.DSNB10.SDSNEXIT'
```

There are two D910S1 identifiers. That is the maximum for a single identifier. Also, there are three unique identifiers: D910S1, DA10S1, and DB10S1. You may have up to eight of these.

Option 4. Enter CICS IDs and Translator Libraries

If your site has CICS, this option will allow you to define CICS identifiers and associate the identifiers with CICS libraries. 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.

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:

```
(1) ===> CTS530 ===> 'SYS1.CICS530.LOADLIB1'
(2) ===> CTS530 ===> 'SYS1.CICS530.LOADLIB2'
(3) ===> CTS520 ===> 'SYS1.CICS520.LOADLIB1'
(4) ===> CTS520 ===> 'SYS1.CICS520.LOADLIB2'
(5) ===> CTS510 ===> 'SYS1.CICS510.LOADLIB1'
(6) ===> CTS510 ===> 'SYS1.CICS510.LOADLIB2'
```

There are two CTS530 identifiers. That is the maximum for a single identifier. Also, there are three unique identifiers: CTS530, CTS520, and CTS510. You may have up to eight of these.

Option 5. Enter Miscellaneous Information

Information that did not fall within the previous option categories (1 through 4) are found on the Miscellaneous Settings panel (Figure 13).

Figure 13. Miscellaneous Settings

```
Compuware CSS Utilities ---- Miscellaneous Settings ------------------------
COMMAND ===>
   To prohibit foreground compiles, set the following value to 'N'.
   Foreground Compile ===> Y

   Hold Queue Sysout Class ===> X (0-9, A-Z)

   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.

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.

Option A. Process Options 1 through 5

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

Option 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.

During the installation, you will need the dataset name of the compile information table.

1. Specify `C` on the command line.
2. In the first field, enter the dataset name that you want to contain the compile information.
3. In the second field, enter the dataset name of an existing compile information table.
4. Press *Enter* to copy the information. The Compile Menu is displayed (Figure 14) with the values that you supply during this installation.

Figure 14. Compile Menu – Compile Facility

```
Compuware CSS Utilities -------- Compile Menu -------------------- CSS 17.2.0
COMMAND ===>

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

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

Source Dsname ===> 'user ID.COBOL.SOURCE(CWXTCOB)'
  Preparation ===> EDITJCL (Batch/Editjcl/Foreground)
    Language  ===> E-COBOL Select From The List Below
  (COB/COB2/COB370/COBMVS/COB390/E-COBOL/COB5/COB6/CAOPT/HASM/HLASM/PLI/PLI370)
  (VAPLI/E/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            _ D910S1 _ DA10S1        _ CTS530 _ CTS520
   _ 2. CICS Translator           _ DB10S1 _ D910S2        _ CTS510 _ CTS420
   _ 3. Compile                   _ DA10S2 _ DB10S2        _ CTS410 _ CTS320
   _ 4. Linkedit                  _ DA10S3 _ DB10S3        _ CTS310 _ CTS230
```

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

If you set the Foreground Compile value to **Y** in the Miscellaneous Settings screen (Figure 13), then the Foreground literal appears in the parenthetical list to the right of the Preparation field. If you set the Foreground Compile value to **N**, then the literal does not appear.

If you selected all the compilers, the parenthetical list under the Language field is generated. The user will select the language they want to use for their 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.

If you used eight unique DB2 identifiers, the list under the DB2 Precompile field would be similar. If you supplied less than eight unique identifiers, then any supplied would appear in a left to right, top to bottom sequence.

If you used eight unique CICS identifiers, the list under the CICS Translator field would be similar. If you supplied less than eight unique identifiers, then any supplied would appear in a left to right, top to bottom sequence.

Task 7.3 Activate Compuware ISPF Dialogs

This task discusses two different ways to activate the CSS Utilities libraries, included with the Compuware ISPF Dialogs—dynamically, or at TSO startup.

If you do not plan on using the CSS Utilities, continue with Link CSS Utilities Tutorials to Your Main Tutorial Panel.

Dynamically allocating the libraries is discussed in Option 1. Allocating the libraries at TSO startup is discussed in Option 2. 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 when CSS Utilities are executed, you must determine how to make the REXX EXECs and CLISTs accessible.

CSS provides sample REXX EXECs and CLISTs for invoking CSS Utilities in library SLCXCNTL. 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.

Instructions:

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 the following table as a guide. If using CLISTs, select one CLIST for 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.

4. Change the library names in the REXX EXECs or CLISTs selected in step 3 to match your site standards.

An optional ISPSLIB can be used with On-the-Fly processing invoked in the Abend-AID Viewer. See the *Enterprise Common Components Advanced Configuration Guide* for details about support for On-the-Fly processing CLISTS. 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.

The CSS Utilities require the SLCXSLIB and SLCXTABL datasets to be concatenated to ISPSLIB and ISPTLIB respectively. The SLCXTABL dataset was created in Create the Compile Information Table

5. Allocate the ECC SLCXCNTL library at TSO logon time, or copy the edited REXX EXECs/CLISTs to a library allocated at TSO logon:
   - *To allocate the ECC 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.

```
o 'CMD (%CSS Utilities CLIST name or REXX EXEC name)
```

– To invoke CSS Utilities by executing a REXX EXEC or CLIST:

  Enter the appropriate command at the TSO command prompt.

```
TSO %CSS Utilities CLIST name or 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. However, allocating these libraries at TSO logon may be inappropriate. In that case, dynamically allocating the libraries may be a better solution.

Steps to provide access to the load, panel, and message modules at TSO logon time:

1. Concatenate each library as shown:

Table 3. ECC Libraries

Load Library SLCXLOAD	Place a DD statement for the load library in the concatenation for STEPLIB or ISPLLIB in your TSO logon PROC OR Execute a CLIST or REXX EXEC at logon time that will concatenate the load library to the appropriate DD statement before starting ISPF **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.
Message Library SLCXMENU	Concatenate the library to ISPMLIB in the TSO logon PROC OR Concatenate the library to ISPMLIB in a CLIST or REXX EXEC before starting ISPF
Panel Library SLCXPENU	Concatenate the library to ISPPLIB in the TSO logon PROC OR Concatenate the library to ISPPLIB in a CLIST or REXX EXEC before starting ISPF
EXEC Library SLCXEXEC	Concatenate the library to SYSPROC in the TSO logon PROC OR Concatenate the library to SYSPROC in a CLIST or REXX EXEC before starting ISPF
Skeleton Library SLCXSENU	Due to the Compile Facility's requirement for this library this DD is required. Concatenate the library to ISPSLIB in the TSO logon PROC OR Concatenate the library to ISPSLIB in a CLIST or REXX EXEC before starting ISPF

Table 3. ECC Libraries

Table Library SLCXTABL	Concatenate the library to ISPTLIB in the TSO logon PROC OR Concatenate the library to ISPTLIB in a CLIST or REXX EXEC before starting ISPF

Notes:

- Replace any existing CSS libraries in these concatenations.
- 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 reports.
- 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:
   /
   o 'PGM(CWDDIUFE) NEWAPPL(AAUT)'

Task 7.4 Customize CSS Parameters

Compuware provides a CSS PARMLIB member (CSSALL), which contains a single, non-core parameter, ESS. As a non-core parameter, it is not required.

While there is a default CSS PARMLIB member named CSS00, it does not contain any parameters. All of Compuware's default PARMLIB members (the members with a 00 suffix) contain only core parameters—CSS does not have required core parameters.

ESS=NO

The ESS parameter, Embedded Source Support, controls whether the source is embedded in the program object or not. **NO** specifies the source listing to not be embedded in the program object. **YES** specifies the source listing to be embedded in the program object. The default is NO.

Note: Abend-AID retrieves the source listing at abend view time, not capture time.

Task 7.5 Link CSS Utilities Tutorials to Your Main Tutorial Panel

This task provides two sample tutorial main menu panels for ISPF and ISPF/PDF. The sample panels, CXTUTOR and CXR00003, are located in the SLCXCNTL library.

Instructions:

To link the CSS utilities tutorial panels to your tutorial main menu panel (TTUTOR or ISR00003), use the applicable member (either CXTUTOR or CXR00003) as an example.

Additional Configuration

Additional configuration considerations can be found in the *Enterprise Common Components Advanced Configuration Guide*.

CSS Configuration Considerations

This section discusses the additional CSS configuration items that will be detailed in the *Enterprise Common Components Advanced Configuration Guide*.

Implement the Security Exit Program

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.

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.

If, after reviewing the above-mentioned chapter, you want to implement the security exit, please refer to the *Enterprise Common Components Advanced Configuration Guide* for configuration instructions.

Install Customized Translation Tables

Customized translation tables can be created and are used in other Compuware products, like Abend-AID.

The default translation tables (mixed-case English for both horizontal and vertical display) are automatically used and unless you are using Japanese support with Abend-AID, you do not have to implement customized translation tables.

If you want to implement customized translation tables, please refer to the *Enterprise Common Components Advanced Configuration Guide* for configuration instructions.

Milestone 8: Configure Topaz Workbench Integration

This milestone contains instructions necessary to integrate with Topaz Workbench.

Task 8.1 Activating Xpediter/Eclipse Support

This task is necessary only if you plan to use the Xpediter/Eclipse plug-in, which is used in conjunction with the Xpediter products.

Instructions:

1. Update the HCIxx PARMLIB member for Xpediter/TSO, if necessary:
   a. Add the CSPF parameter to specify the fully qualified name for the Compuware Shared Profile File dataset, as created and populated by the Xpediter/TSO installer.
   b. Add the TSO_* parameters (TSO_SYSNAME, TSO_APPLID, TSO_LOGMODE, and TSO_LPAR) to specify the corresponding system name, APPLID, LOGMODE, and LPAR name values for the four parameters.
   c. Add the APPLID parameter(s) to specify one or more VTAM APPLIDs defined for use by Xpediter/Eclipse.
2. Update the HCIxx PARMLIB member for Xpediter/CICS, if necessary:
   a. Add the CICS_SYSNAME parameter to specify the CICS region system name.
   b. Add the CICS_SOCK parameter to specify
   c. Add the CICS_PORT parameter to specify
   d. Add the CICS_HOST parameter to specify
3. Refresh the HCIxx PARMLIB
4. Add the Xpediter/TSO Authorized Load Library, SLXTAUTH, to the HCI PROC, HCIPROC.
5. Restart the HCI.
6. Install the TSO Logon Procedure.
7. Verify integration by configuring and running an Xpediter Debug Session.

Task 8.2 Activating Hiperstation/Eclipse Support

This task is necessary only if you plan to use the Hiperstation/Eclipse plug-in, which is used in conjunction with Hiperstation, to create audit files.

Instructions:

1. Modify the HCI PROC (job HCIPROC) to add the fully qualified Hiperstation library (SQQFLOAD) in the STEPLIB ??? concatenation.
2. Modify the HCIxx PARMLIB member to add the HPERPLIB parameter with a value of the fully qualified Hiperstation panel library, SQQFPENU.
3. Restart the HCI.
4. Use the **Compuware** menu to start the Hiperstation/Eclipse perspective.

5. If the perspective launches and displays audit files, then you have properly activated Hiperstation/Eclipse support.

Task 8.3 Activating Code Coverage/Eclipse Support

This task is necessary only if you plan to use the Code Coverage/Eclipse plug-in, which is used in conjunction with Xpediter/Code Coverage.

Instructions:

1. Modify the SSAS PROC (job CXSSAS) to uncomment the Xpediter/Code Coverage library (SLXVLOAD) in the STEPLIB concatenation and to update with the fully-qualified name.
2. Restart HCI.
3. Use the **Compuware** menu to start the Code Coverage/Eclipse perspective. Use the Cheat Sheet provided to help you get started.
4. If the perspective launches and displays Coverage Analysis results from an Xpediter/Code Coverage repository, then you have properly activated Code Coverage/Eclipse support.

Task 8.4 Activating ISPW Support

This task only necessary if you plan to activate the ISPW interface in Topaz Workbench release 18.2.1 or higher. Also required is ISPW release 17.02 or higher with cumulative maintenance applied.

To connect to multiple ISPW instances from a HCI, the ISPW parameters (ISPW_PORT, ISPW_HOST, and ISPW_CTNAME) must be specified in the HCIxx PARMLIB member after a PORT_CONFIG statement corresponding to that instance. Each ISPW instance requires a separate PORT_CONFIG statement in the HCIxx member (see Figure 1, in Milestone 5: Configure HCI for a New Install).

Instructions:

1. Modify the HCI*xx* PARMLIB member by adding the following ISPW Configuration parameters:
   a. **ISPW_HOST=*ip_addr***
   where *ip_addr* is the IP address, or DNS, of the ISPW server.
   b. **ISPW_PORT=*port_num***
   where *port_num* is the ISPW server port and must match the value of WZCMXPRT in the ISPW start parms.
   c. **ISPW_CTNAME=*ct_server_name***
   where *ct_server_name* is the name of the ISPW/CT server that should be used by Topaz Workbench clients to access ISPW life cycle datasets. This parameter is required when the CTLOCAL value defined to ISPW does not have access to the life cycle datasets used by the Topaz Workbench clients that are connecting through this HCI. This parameter is used in multi-site implementations with a single ISPW server.
2. Refresh the HCI*xx* PARMLIB within the CMSC address space.
3. Restart the HCI.

Task 8.5 Activating CA-Endevor Support

This task is necessary only if you plan to activate the CA Endevor interface (the appearance of the Endevor navigation path in Topaz Workbench's Host Explorer perspective).

Instructions:

1. Modify the HCIxx PARMLIB member to add the ENDEVOR parameter with a value of Y.
2. Add the Endevor APF-authorized library to the STEPLIB DD in the CSS TP started task (job CXSS0000 in your system PROCLIB).
3. Restart the HCI.

4. In the Host Explorer perspective, select a mainframe LPAR and expand to see the paths.
5. Right-click the Endevor path and select *Add Filter*.
6. In the *New Endevor Filter* dialog box, use the Environment menu to view available environments. The appearance of a list of accessible environments validates that the CA-Endevor interface was properly activated.

Milestone 9: Deployment

This milestone provides information to assist in deploying ECC to other LPARs.

Role involved: z/OS Systems Programmer

Task 9.1 Deployment Possibilities

Instructions:

Compuware recommends using one of the following two methods to deploy ECC to another LPAR.

Method 1

As ECC components are critical for most other Compuware mainframe products, you may want to conduct a full installation to ensure a smooth and correct ECC installation for the LPAR.

Method 2

If you do not require, or want to conduct a complete SMP/E installation on another LPAR, you can instead copy the existing SMP/E Target libraries from one LPAR to the other LPAR, then follow the configuration tasks for each of the components.

If there is an existing release of ECC (or existing older releases of the ECC components), you can follow the “Migration” path. Otherwise, use the “New Installation” path.

Milestone 10: Configure Global Parameters

The Compuware Global Parameter member (CWGL*xxxx*) is a single location for specification of parameters that are consumed across Compuware products. This member must be at PARMLIB version 2. A sample member is located in hlq.SLCXCNTL(CWGL00).

zAdviser

zAdviser is a collaborative service that uses machine learning to find correlations between developer behaviors and key performance indicators (KPIs) based on DevOps toolchain data and Compuware product-usage data. Optionally, zAdviser records may be streamed directly to Compuware.

The ECC zAdviser services provide the following components to Compuware products:

- An interface through License Management to write zAdviser records to SMF, previously License Management ROI support.
- A CMSC service for streaming zAdviser records directly from z/OS to zAdviser, a subdomain of the Compuware API endpoint. All communications are outbound, over a secure HTTPS connection and are GDPR compliant.

Configuring zAdviser

Disabling zAdviser streaming support

Set the ZADVISER parameter to NO. Records will not be written to zAdviser, but will continue to be written to SMF.

```
ZADVISER=NO
```

Enabling zAdviser current-version records

Set the ZADVISER_SMF_VERSION parameter to 17.02.07 and ensure all products are current on maintenance. This specification will tell Compuware products to write the latest-supported version of zAdviser records.

```
ZADVISER_SMF_VERSION=17.02.07
```

Enabling zAdviser record streaming

Set the ZADVISER parameter to YES. The CMSC parameter CES_SSL_KEYRING must specify a keyring that contains all four trusted Root CAs from AWS Trust Store. This allows the CMSC to Communicate outbound to the Compuware API, `api.compuware.com`, over HTTPS.

```
ZADVISER=YES
```

The LMS parameters SMF_ID and SMF_LOGALL parameters must be specified for zAdviser records to be written for all products.

The ROI_CAPTURE parameter is no longer supported and must be omitted from the LMS client PARMLIB member.

For the complete list of **LMS parameters**, see the "License Management System (LMS) Configuration and Administration" chapter in the *Enterprise Common Components Advanced Configuration Guide*.

For the complete list of **zAdviser parameters**, see the "Compuware Mainframe Services Controller (CMSC) Configuration and Administration" chapter in the *Enterprise Common Components Advanced Configuration Guide*.

Migration Utility

Compuware PARMLIB has been enhanced to provide a centralized point of validation prior to product startup, a more consistent and simplified syntax, and the means to migrate to a newer version at your discretion.

To enable the latest enhancements, you must be running ECC 17.2 with *all* cumulative maintenance along with the most current version of the Compuware product(s) with *all* cumulative maintenance. Once all ECC instances that share the same Compuware PARMLIB member have been upgraded and the Compuware product(s) the require those ECC instances have been upgraded, you are now ready to migrate the Compuware PARMLIB member at your convenience.

Migration Utility

The migration utility converts all prior CMSC-compatible PARMLIB members to their latest format. The member MIGDRIVR is included in SLCXCNTL and provides sample JCL to execute the migration driver. This utility appends the second version (“Version 2 “or “PARMLIB V2”) of the parameter delimiter text $$$ V2 END $$$ to the migrated member, and before the previous member where applicable. Various options may be specified on the *PARM=operand* to further refine the migration process.

The CMSC PARMLIB member must not contain the Version 2 delimiter text.

EXEC Parameter Options

MEMBER= Specifies a 1- to 8-character member name filter, or explicit member name. Specifying * will migrate all valid Compuware PARMLIB members, NAME* will migrate all valid Compuware PARMLIB members prefixed with NAME, and NAME will migrate a valid Compuware PARMLIB member prefixed with NAME.

READONLY= Specifies whether members are to be written to their respective output member or the FDBDLOG DD. Specifying YES will allow for validating migration output prior to writing the member our permanently. Options are YES or NO.

VERSION= Specifies the PARMLIB version for which the migration driver will output. Specifying BOTH will output the new PARMLIB member data, the version two delimiter text, and the version one member. Specifying 2 will omit the version one member.

DDs

//CWPARM This DD is the concatenation of Compuware PARMLIB datasets that will be used as input to the migration driver. Only valid CMSC PARMLIB members will be processed. If a duplicate member name is found only migration of the first V1 section and the first V2 section will be written to CWPOUT.

//CWPOUT Optional output dataset to which migrated members are written. This DD is optional but if not specified, and if READONLY=NO is in effect, member updates will be written in place,

Checklist of Milestones and Tasks

- ❑ Milestone 1: Install ECC
  - ❑ Task 1.1 RFN and SMP/E Install
- ❑ Milestone 2: Configuration Preparation
  - ❑ Task 2.1 Verify APF-Authorize SLCXAUTH Load Library
  - ❑ Task 2.2 Define Compuware's Common Parameter Library Dataset
  - ❑ Task 2.3 Define HCI's Default User ID
  - ❑ Task 2.4 Upgrade or New Installation
- ❑ Milestone 3: Configuration for an Upgrade
  - ❑ Task 3.1 Upgrade from pre-16.05 to 17.02
  - ❑ Task 3.2 Upgrade from 16.05 to 17.02
  - ❑ Task 3.3 Configure CMSC
  - ❑ Task 3.4 Verification
  - ❑ Task 3.5 Post-upgrade Considerations
- ❑ Milestone 4: Configure LMS for a New Install
  - ❑ Task 4.1 Transfer License Certificate to Host System
  - ❑ Task 4.2 Create a License File
  - ❑ Task 4.3 Import License Certificates
  - ❑ Task 4.4 Verify the License File
  - ❑ Task 4.5 Create LMS Exit Procedure
  - ❑ Task 4.6 Customize LMS Parameters
- ❑ Milestone 5: Configure HCI for a New Install
  - ❑ Task 5.1 Define Journals
  - ❑ Task 5.2 Implement HCI Procedures
  - ❑ Task 5.3 HCI Parameter Customization