CA OPS/MVS® Event Management and Automation

Installation Guide

Release 12.2

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2014 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA products:

- CA 1® Tape Management (CA 1)
- CA 7™ Workload Automation (CA 7 WA)
- CA ACF2™ for z/OS (CA ACF2)
- CA Automation Point
- CA Common Services for z/OS (CCS for z/OS)
- CA Dynam®/TLMS Tape Management (CA Dynam/TLMS)
- CA Examine® Auditing (CA Examine)
- CA Hyper-Buf®VSAM Buffer Optimizer (CA Hyper-Buf)
- CA Jobtrac®Job Management (CA Jobtrac)
- CA MIC Message Sharing (CA MIC)
- CA NSM
- CA NSM System Status Manager CA OPS/MVS®Option (CA NSM SSM CA OPS/MVS Option)
- CA OPS/MVS® Event Management and Automation (CA OPS/MVS)
- CA Scheduler®Job Management (CA Scheduler)
- CA SYSVIEW® Performance Management (CA SYSVIEW)
- CA Top Secret®for z/OS (CA Top Secret)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

Note: In PDF format, page references identify the first page of the topic in which a change was made. The actual change may appear on a later page.

- Updated the HWS Parameters (see page 180) section.
- Updated to CA CSM 6.0 (see page 55).
- Added the How to Install and Configure CA OPS/MVS RESTful Web Services (see page 169) section.
- Standardized SMP/E installation.
- Removed tape installation.
- Added the OPSVASRV OPS/REXX Function (see page 201) section.
- Added the Interface to Hardware Interface (see page 201) Services section.
- Added the Interface to Linux Connector (see page 202) section.

Contents

Chapter 5: Starting Your Product 55

Chapter 6: Configuring and Installing Optional Components 105

Chapter 1: Overview

This section contains the following topics:

Introduction to CA OPS/MVS

CA OPS/MVS manages critical resources by status across systems and includes automated applications that simplify the deployment of powerful and complex automation to manage the environment. CA OPS/MVS is a critical component for automating the disaster recovery process and end-to-end automation.

CA OPS/MVS provides efficient synchronous automation, and includes user efficiency tools and utilities that help you create and deploy automation of complex systems and processes. CA OPS/MVS integrates with CA's automation, performance management, and workload automation products.

Base Product Components

CA OPS/MVS provides tools that streamline data center operations, which let you unify and simplify the management of your IT environment for greater business results.

The CA OPS/MVS base product, which is a formal z/OS subsystem, runs in a number of z/OS address spaces. An alphabetical list of base product components follows:

Automated Operations Facility (AOF)

Automated Operations Facility (AOF) lets you program a response to a system event, such as a message or the passage of time. AOF rules are specially structured OPS/REXX programs that support automated operations by taking advantage of extensions made to the OPS/REXX language.

Enhanced Console Facility (ECF)

The Enhanced Console Facility (ECF) is intended for use when TSO (and therefore OPSVIEW) is down. It lets you log on to a z/OS or JES console and conduct a line-mode interactive TSO session. From this session, you can issue TSO commands or invoke TSO CLISTs or OPS/REXX programs, including those that issue prompts for additional input. By logging on to the ECF, the operator can perform tasks such as repairing members of SYS1.PROCLIB required for TSO operation.

External Product Interface (EPI)

The External Product Interface (EPI) permits CA OPS/MVS systems that are running under VTAM to communicate with any VTAM application that supports IBM 3270 (SLU2) type virtual terminals. The EPI appears to VTAM as a real 3270 terminal that can emulate any number of 3270 type virtual terminals that are connected to any number of VTAM applications.

Using EPI, you can automate issuing commands to and fetching data from VTAM applications and you can share VTAM sessions between OPS/REXX programs.

Operator Server Facility (OSF)

An integral part of CA OPS/MVS, the Operator Server Facility (OSF) lets users schedule OPS/REXX programs, TSO commands, and TSO/E REXX programs or CLISTs for CA OPS/MVS to execute. Various CA OPS/MVS components use the OSF services, including the AOF, ECF, IOF, and MSF.

OPS/REXX Language

REXX (Restructured EXtended eXecutor) is the standard command language for all of the IBM environments under its Systems Application Architecture (SAA).

Because a product such as CA OPS/MVS must be programmable in some language, CA OPS/MVS comes with its own implementation of REXX, called OPS/REXX. This provides users with long-term stability for their investments in CA OPS/MVS. OPS/REXX provides SAA compatibility with added functions to help you write programs for system automation tasks.

OPSVIEW Interface

OPSVIEW is a full-screen, menu-driven operations interface that both data processing professionals and end users can use. OPSVIEW provides panels for performing various z/OS system functions, and it is the primary vehicle for controlling CA OPS/MVS itself.

Programmable Operations Interface (POI)

The Programmable Operations Interface (POI) consists of TSO command processors and REXX functions. The POI provides a programmable interface to both the z/OS console and to CA OPS/MVS facilities. You can use the command processors and functions to build custom operations automation and productivity enhancement applications. OPSVIEW, the CA OPS/MVS operations interface, is one example of an application that was built using the POI.

Relational Data Framework (RDF)

The Relational Data Framework (RDF) facility lets you use Structured Query Language (SQL) statements to manage the large amounts of system information that are required by automation rules and OPS/REXX programs. Instead of using large sets of variables, use the RDF to collect system information, organize it into a relational table containing rows and columns of related data, and retrieve related system information by selecting it from a particular row or column.

We chose SQL to manage automation data because of the wide popularity of SQL with mainframe and PC users. The RDF consists of relational SQL tables plus a subset of the SQL language that conforms to American National Standards Institute (ANSI) standards. If you already know SQL, you can able to use its CA OPS/MVS subset right away.

System State Manager (SSM)

The System State Manager (SSM) monitors and controls the status of the hardware and software resources on your system.

Using information from the RDF relational tables, SSM maintains a model of the proper state of your system resources. When the actual state of a resource deviates from that model (for instance, when a tape drive that should be online goes offline), SSM dispatches an OPS/REXX program to restore the resource to its proper state.

VM Guest Support (VMGS)

VM Guest Support (VMGS) lets CA OPS/MVS issue a CP command to anywhere that a z/OS command could be issued. This support means that if your site runs z/OS under VM, you can coordinate the z/OS activities with those of VM.

Automated Operations Facility

Automated Operations Facility (AOF) lets you program a response to a system event, such as a message or the passage of time. AOF rules are specially structured OPS/REXX programs that support automated operations by taking advantage of extensions made to the OPS/REXX language.

Enhanced Console Facility

The Enhanced Console Facility (ECF) is intended for use when TSO (and therefore OPSVIEW) is down. It lets you log on to a z/OS or JES console and conduct a line-mode interactive TSO session. From this session, you may issue TSO commands or invoke TSO CLISTs or OPS/REXX programs, including those that issue prompts for additional input. By logging on to the ECF, the operator can perform tasks such as repairing members of SYS1.PROCLIB required for TSO operation.

External Product Interface

The External Product Interface (EPI) permits CA OPS/MVS systems that are running under VTAM to communicate with any VTAM application that supports IBM 3270 (SLU2) type virtual terminals. The EPI appears to VTAM as a real 3270 terminal that can emulate any number of 3270 type virtual terminals that are connected to any number of VTAM applications.

Using EPI, you can automate issuing commands to and fetching data from VTAM applications and you can share VTAM sessions between OPS/REXX programs.

Operator Server Facility (OSF)

An integral part of CA OPS/MVS, the Operator Server Facility (OSF) lets users schedule OPS/REXX programs, TSO commands, and TSO/E REXX programs or CLISTs for CA OPS/MVS to execute. Various CA OPS/MVS components use the OSF services, including the AOF, ECF, IOF, and MSF.

OPS/REXX Language

REXX (*R*estructured *EX*tended e*X*ecutor) is the standard command language for all of the IBM environments under its Systems Application Architecture (SAA).

Because a product such as CA OPS/MVS must be programmable in some language, CA OPS/MVS comes with its own implementation of REXX, called OPS/REXX. This provides users with long-term stability for their investments in CA OPS/MVS. OPS/REXX provides SAA compatibility with added functions to help you write programs for system automation tasks.

OPSVIEW Interface

OPSVIEW is a full-screen, menu-driven operations interface that both data processing professionals and end users can use. OPSVIEW provides panels for performing various z/OS system functions, and it is the primary vehicle for controlling CA OPS/MVS itself.

Programmable Operations Interface

The Programmable Operations Interface (POI) consists of TSO command processors and REXX functions. The POI provides a programmable interface to both the z/OS console and to CA OPS/MVS facilities. You can use the command processors and functions to build custom operations automation and productivity enhancement applications. OPSVIEW, the CA OPS/MVS operations interface, is one example of an application that was built using the POI.

Relational Data Framework

The Relational Data Framework (RDF) facility lets you use Structured Query Language (SQL) statements to manage the large amounts of system information required by automation rules and OPS/REXX programs. Instead of using large sets of variables, use the RDF to collect system information, organize it into a relational table containing rows and columns of related data, and retrieve related system information by selecting it from a particular row or column.

We chose SQL to manage automation data because of the wide popularity of SQL with mainframe and PC users. The RDF consists of relational SQL tables plus a subset of the SQL language that conforms to American National Standards Institute (ANSI) standards. If you already know SQL, you will be able to use its CA OPS/MVS subset right away.

System State Manager

The System State Manager (SSM) monitors and controls the status of the hardware and software resources on your system.

Using information from the RDF relational tables, SSM maintains a model of the proper state of your system resources. When the actual state of a resource deviates from that model (for instance, when a tape drive that should be online goes offline), SSM dispatches an OPS/REXX program to restore the resource to its proper state.

VM Guest Support

VM Guest Support (VMGS) allows a CP command to be issued by CA OPS/MVS anywhere that a z/OS command could be issued. This means that if your site runs z/OS under VM, you can coordinate the z/OS activities with those of VM.

Optional Features

CA OPS/MVS has a number of facilities that are not necessarily applicable to every environment. For this reason, these facilities are packaged as optional features. A list of these optional features follows:

- CICS Operations Facility (COF)
- Critical Path Monitoring (CPM)
- Expert Systems Interface (ESI)
- IMS Operations Facility (IOF)
- Multi-System Facility (MSF)
- Switch Operations Facility (SOF)

CICS Operations Facility

The CICS Operations Facility (COF) is an interface between CA OPS/MVS and CICS that extends the capability for AOF rule processing to CICS messages, which are written only to CICS transient data queues. This additional message traffic expands the number of automatable events that you can use to control CICS subsystems. Events that are visible to AOF rules using the COF include terminal failures, the logon and logoff activities of the user, and journal switches.

With the COF interface installed, a single copy of CA OPS/MVS can handle an unlimited number of CICS address spaces.

Critical Path Monitoring

CA Critical Path Monitoring (CA CPM) Version 3 monitors the performance of groups of batch jobs (flows) against user-defined deadlines. CA CPM Version 3 works in conjunction with any of the CA scheduling products (CA 7 WA, CA Scheduler, and CA Jobtrac) to provide this functionality. By interfacing CA OPS/MVS with CA CPM Version 3, information on monitored flows can be viewed using a web-enabled or Windows user interface on a CA NSM SSM CA OPS/MVS Option workstation.

Expert Systems Interface

The Expert Systems Interface (ESI) Application Programming Interface, or OPSLINK, accesses selected CA OPS/MVS facilities from an application written in either a high-level language or assembler language.

Specific uses of the ESI include executing operator commands, executing TSO commands (when running under TSO TMP interactively or in batch), and accessing and updating the CA OPS/MVS global variables.

IMS Operations Facility

The IMS Operations Facility (IOF) is an interface between CA OPS/MVS and IMS that extends the CA OPS/MVS facilities to IMS. For example, you can write AOF rules that process IMS messages, and you can use OPSVIEW to operate IMS.

A single copy of CA OPS/MVS can handle up to 32 copies of IMS. If you run multiple copies of IMS under the control of one copy of CA OPS/MVS, the copies of IMS may be any combination of IMS levels that CA OPS/MVS supports.

Multi-System Facility

The Multi-System Facility (MSF) extends the facilities of CA OPS/MVS into the multiple-CPU and multiple-site environment. The MSF establishes VTAM, XCF, or TCP/IP sessions between copies of CA OPS/MVS, permitting any copy to issue a command to any other copy and to receive its response.

The MSF also facilitates the connection to CA Automation Point through a TCP/IP connection using the CCI services of CCS for z/OS.

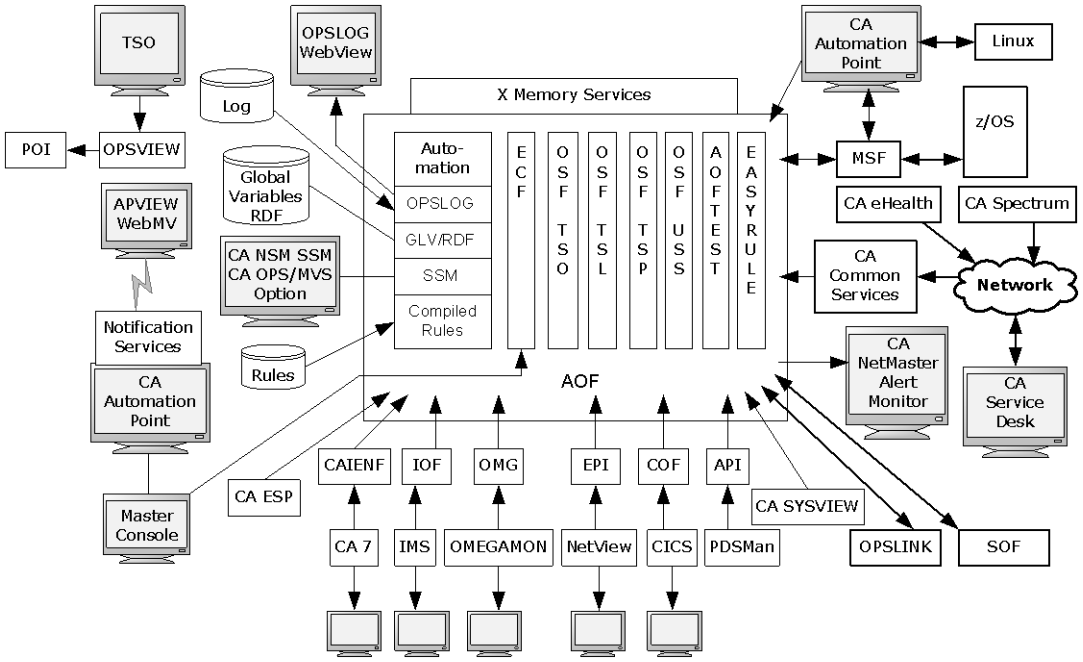
Switch Operations Facility

The Switch Operations Facility (SOF) automates I/O configuration management through the following features:

- Automatic discovery
- Automatic continuous monitoring
- Automatic cross-system resolution
- Single point of display and control
- ISPF interface
- OPS/REXX host environment
- Saved switch configurations

Overview of CA OPS/MVS

The following illustration presents an overview of CA OPS/MVS and how it fits into the z/OS operating system:



For information on CA products that integrate with CA OPS/MVS, see the *Integration Guide*.

Audience

Readers of this book should have knowledge in the following areas:

- JCL
- TSO/ISPF
- z/OS environment and installing software in this environment
- Your organization's IT environment, enterprise structure, and region structure

You may need to work with the following personnel:

- Systems programmer for z/OS and VTAM definitions
- Storage administrator, for DASD allocations

How the Installation Process Works

CA Technologies has standardized product installations across all mainframe products. Installation uses the following process:

- Acquisition—Transports the software to your z/OS system.
- Installation using SMP/E—Creates an SMP/E environment and runs the RECEIVE, APPLY, and ACCEPT steps. The software is untailored.
- (For CA CSM Release 5.1 and earlier only) Deployment—Copies the target libraries to another system or LPAR.

  Note: This step is optional for CA CSM Version 6.0. For more information, see the scenario *Configuring Products Using CA CSM* that is available in the CA CSM Version 6.0 bookshelf at http://ca.com/support.
- Configuration—Creates customized load modules, bringing the software to an executable state.
- (For staging system configurations in CA CSM Version 6.0 only) Deployment—Makes configured run-time libraries available to a remote location where that software can be activated, bringing it to an executable state.

CA Chorus™ Software Manager (CA CSM) - formerly known as CA Mainframe Software Manager™ (CA MSM) - is an intuitive web-based tool that can automate and simplify many CA Technologies product installation activities on z/OS systems. This application also makes obtaining and applying corrective and recommended maintenance easier. A web-based interface enables you to install and maintain your products faster and with less chance of error. As a best practice, we recommend that you install mainframe products and maintenance using CA CSM. Using CA CSM, someone with limited knowledge of JCL and SMP/E can install a product.

Note: If you do not have CA CSM, you can download it from the Download Center at http://ca.com/support. Follow the installation instructions in the CA Chorus Software Manager documentation bookshelf on the CA Chorus Software Manager product page.

You can also complete the standardized installation process manually using pax files that are downloaded from http://ca.com/support or a product DVD.

To install your product, do the following tasks:

1. Prepare for the installation by confirming that your site meets all installation requirements.
2. Verify that you acquired the product using one of the following methods:
   - Download the software from http://ca.com/support using CA CSM.
   - Download the software from http://ca.com/support using Pax-Enhanced Electronic Software Delivery (Pax ESD).
   - Order a product DVD. To do so, contact your account manager or a CA Technologies Support representative.
3. Perform an SMP/E installation using one of the following methods:
   - If you used CA CSM to acquire the product, start the installation process from the SMP/E Environments tab in CA CSM.
   - If you used Pax ESD to acquire the product, you can install the product in the following ways:
     - Install the product manually.
     - Complete the SMP/E installation using the Add Product option in CA CSM.
   - If you used a DVD, install the product manually.

   Note: If a CA Recommended Service (CA RS) package is published for your product, install it before proceeding.
4. (For CA CSM Release 5.1 and earlier only) Deploy the target libraries.

   Note: This step is optional for CA CSM Version 6.0. For more information, see the scenario *Configuring Products Using CA CSM* that is available in the CA CSM Version 6.0 bookshelf at http://ca.com/support.
5. Configure your product using CA CSM or manually.
6. (For staging system configurations in CA CSM Version 6.0 only) Deploy configured run-time libraries, and activate your product.

   Note: Configuration is considered part of starting your product (see page 55).

Chapter 2: Preparing for Installation

This section describes what you need to know and do before you install the product.

This section contains the following topics:

Hardware Requirements

CA OPS/MVS r12.2 can be installed on hardware that supports the software described in the section Software Requirements.

Software Requirements

Review these requirements to run CA OPS/MVS depending on the software or operating system you are using. If your site does not have the correct software levels, contact Technical Support at http://ca.com/support.

- Operating system support:

  z/OS

  Release 1.10 and higher

  JES2

  Any IBM-supported release.

  JES3

  Any IBM-supported release. An FMID of the format HJS*nnnn* should be in the SMP/E target zone of the system where you plan to install CA OPS/MVS.

TSO

Any IBM-supported release of TSO/E.

IMS

Versions 9.1 and higher if you are installing the IMS Operations Facility (IOF).

You can have any mixture of supported IMS releases; CA OPS/MVS adjusts automatically to differences between IMS version and release levels.

CICS

CICS Transaction Server for z/OS Versions 2.3 and higher.

You can have any combination of IBM-supported CICS versions.

- Software level support:

z/OS Security Server (RACF)

Any IBM-supported release.

IBM Communications Server (VTAM)

Any IBM-supported release.

CA ACF2

Any CA-supported release.

CA Top Secret

Any CA-supported release.

CCS for z/OS

Any CA-supported release.

Important! These supported software levels are valid as of the CA OPS/MVS r12.2 GA date. For verification of the supported levels, see the “Upgrade Information” link on the CA OPS/MVS Product Home page at http://ca.com/support.

CA Common Services Requirements

The following CA Common Services are used with CA OPS/MVS:

- CAICCI
- CAIRIM
- CAISSF
- CA LMP
- CA GSS
- CA Health Checker Common Service

Note: If other CA products are installed at your site, some of these services may already be installed.

Note: For information on CA Common Services FMIDs, see the appendix "CCS for z/OS Component Requirements (see page 195)."

CAIRIM

Prepares your operating system environment for all CA applications and starts them. The common driver for a collection of dynamic initialization routines eliminates the need for user SVCs, SMF exits, subsystems, and other installation requirements commonly encountered when installing systems applications.

Integral parts of CAIRIM are CAISSF and CA LMP.

CAISSF

Provides an external security mechanism for controlling and monitoring access to all system and application resource processes. CAISSF is integrated into many CA enterprise applications and is also used by other CCS for z/OS services. CAISSF provides security services for user logon, resource access control, process use control, and recording and monitoring of violation activity.

CA LMP

Provides a standardized and automated approach to the tracking of licensed software and is provided as an integral part of CAIRIM. After CAIRIM is installed, you have access to Technical Support for all CA LMP-supported products.

CAICCI

Provides CA enterprise applications with a common communications software layer that insulates the applications from dealing with protocol specifics, error recovery, and system connection establishment.

CA GSS

CA GSS is part of CA Common Services for z/OS and is installed with it.

To make full use of some of your product features, you must have CA GSS installed at your site

CA Health Checker

Provides a simple and consistent method for CA products to create health checks to run under the IBM Health Checker for z/OS. The IBM Health Checker for z/OS helps you identify potential problems in your z/OS environment by checking system or product parameters and system status against recommended settings. CA has joined other vendors in creating checks for CA z/OS products. CA OPS/MVS health checks are automatically activated on the target system when the product is started on a system where the following components are installed and configured:

- CA Health Checker Common Service
- IBM Health Checker for z/OS

For more information on installing the CA Health Checker Common Service, see the *CA Common Service Installation Guide.*

For more information about the IBM Health Checker for z/OS, see the *IBM Health Checker for z/OS User Guide*.

Security Requirements

To complete the tasks in this guide, you need security privileges described in the following sections.

Note: For detailed security product logon ID requirements, see the section Create Product Security Product Logon IDs (see page 86).

The following table summarizes the access requirements for CA OPS/MVS. If you develop applications that update your own databases, then they also need access. After you have started to use the product and written your own applications, you will need to provide access to your own REXX, CLIST, OPSEXEC, and possibly user ISPF data sets.

Data Set Name	Access	User IDs
OPS.*xxx*.RULES	Read, write	OPSMAIN and authorized TSO users
OPS.CCLXLOAD	Execute	OPSMAIN, OPSOSF, OPSECF, and all TSO users
OPS.CCLXEXEC	Read	OPSMAIN, OPSOSF, OPSECF, and all TSO users

Data Set Name	Access	User IDs
OPS.CCLXCLS00	Read	OPSMAIN, OPSOSF, OPSECF, and all TSO users
OPS.OPSLOG	Read, write	OPSMAIN
OPS.SYSCHK1	Read, write	OPSMAIN
Logical Parmlib Concatenation	Read	OPSMAIN
OPS.CCLXHENU	Read	All authorized TSO users
OPS.CCLXPENU	Read	All authorized TSO users
OPS.CCLXMENU	Read	All authorized TSO users
OPS.CCLXSENU	Read	All authorized TSO users
OPS.CCLXTENU	Read	All authorized TSO users
OPS.CCLXOPEX	Read	All authorized TSO users and possibly OPSOSF

Note: If you are using SSMGA, you must also allocate the OPS.CCLXOPEX data set to the OPSMAIN procedure.

Important! Running CA OPS/MVS without giving its various address spaces enough authorization to access their data sets is the most common installation problem.

TSO OPER Authority

Provide TSO OPER authority through your security package to all user IDs, including the OPSMAIN and OPSOSF user IDs, that issue ADDRESS OPER commands, enter commands from the OPSVIEW 6, or opslog OPSVIEW 1 panels.

To determine if you have the appropriate TSO OPER authority, run the OPS/REXX program OPSIVP.

Similarly, this must be done for user IDs requiring the use of TSO submit, status, and cancel commands.

Export Declaration

The U.S. government has completed a technical review on the encryption capabilities within CA OPS/MVS and has provided the Commerce Classification of CCATS #G050502 and 5D002 ENC. The product's use of encryption is described in the OPSLOG WebView feature.

Storage Requirements

This section describes storage needed to install and run CA OPS/MVS.

ECSA Usage and Storage Requirements

CA OPS/MVS uses a minimal amount of below-the-line CSA storage; at most, the main CA OPS/MVS address space uses about 2 KB of storage below the 16-MB line. CA OPS/MVS achieves this low CSA usage by using z/OS cross-memory services extensively, so the extended private area of the main CA OPS/MVS address space can store globally used data areas. However, much of the CA OPS/MVS code and some data areas reside in extended CSA (ECSA) storage.

We strongly recommend that you reserve *500 KB of ECSA* for the use of CA OPS/MVS. Most sites have so much ECSA allocated that earmarking 500 KB for CA OPS/MVS does not require you to increase the amount of ECSA. However, if your site runs with a limited amount of ECSA, increase it by 500 KB before you install CA OPS/MVS.

Notes:

- The bottom of ECSA is rounded to a 1 MB boundary, so you may have more ECSA than you think.
- If CA OPS/MVS terminates abnormally, there may not be enough ECSA to load all necessary modules and restart CA OPS/MVS. Should this occur, IPL your system to free an appropriate amount of common storage. If you think that this could happen at your data center, ensure that the second value of your IEASYS*xx* CSA parameter is sufficiently large (for example, CSA=(x,40000)). If you use the CA OPS/MVS module reload facility to reload ECSA resident modules, you will find that ECSA usage of CA OPS/MVS increases. Replaced modules are not deleted until product shutdown.

DASD Space for Distribution, Target Libraries and Data Areas

CA OPS/MVS requires DASD space for a variety of purposes, including distribution, target libraries and various data areas, such as those for OPSLOG, global variables, and RDF and System State Manager variables.

The following outlines the amount of DASD space to allocate for these purposes:

- SMP/E-controlled Distribution and Target Libraries

  Number of 3390 cylinders: 340

- For ESD Installations only additional Relfiles will be temporarily allocated for install and can be cleaned up post installation using the Cleanup JCL in the OPS.SAMPJCL

  Number of 3390 cylinders: 110

- Default size of data areas, which is sufficient for initial product users

  Number of 3390 cylinders:270

Note: For information on calculating how much DASD you need, see the appendix "DASD Calculation Chart (see page 203)" in this guide.

USS Space Requirements

Ensure that you have sufficient free space in the USS file system that you are using for Pax ESD to hold the directory that the pax command and its contents create. You need approximately 3.5 times the pax file size in free space.

If you do not have sufficient free space, you receive error message EDC5133I.

Concurrent Releases

You can install this release of your product and continue to use an older release in another SMP/E environment. If you plan to continue to run a previous release, consider the following points:

- When you install the product into an existing SMP/E environment, this installation deletes previous releases in that environment.
- If you acquired your product with Pax ESD, select different target and distribution zones for your new release from where your current release is installed. The new zones use different libraries than your current release.

  Note: CA CSM installs a product into a new SMP/E environment by default. You can select an existing SMP/E environment from your working set. For more information, see the online help that is included in CA CSM.
- Define DDDEF entries in your new zones to point SMP/E to the proper libraries for installation. Ensure that they point to the new release libraries.

Tailor Installation of CA OPS/MVS for Your Site

The installation of CA OPS/MVS r12.2 is broken into five separate SMP/E functions, or FMIDs. To use SSM, you are only required to install the CA OPS/MVS Base function (FMID CCLXC320). Installation of these other functions (FMIDs) is optional:

1. CA OPS/MVS OPSLOG WebView and Web Services (CCLXC21)
2. CA Network and Systems Management (NSM) System State Manager (SSM) CA OPS/MVS Option (CCLXC22)

   This option is a CA OPS/MVS agent that interfaces with CA NSM to display SSM tables and resources in the CA NSM World View Map.
3. CA OPS/MVS CICS Operations Facility (CCLXC23)
4. CA OPS/MVS OPSHMC REXX (CCLXC24)

Note: Functions 2 and 3 require a separate user license agreement.

During installation, you are required to select what functions to install in addition to the Base function.

During step 4 of the CA CSM installation, choose Custom Installation to select the functions to install. A Full Installation option is provided as a convenience to install all functions.

During Native SMP/E JCL installation, manually edit the SAMPJCL members to SMP/E receive, apply, and accept only those functions which you select to install.

Tailor External Library Names for Your Site

Installation of CA OPS/MVS r12 can require the use of libraries that are installed by other products at your site. Customize the names of these "external" product libraries for your specific environment. These library names are defined to SMP/E as DDDEFs.

During step 4 of CA CSM installation, provide the name of any “external” product library that is required to install the functions you have selected.

During Native SMP/E JCL installation, manually update the SAMPJCL member CLXSEDIT with the name of any “external” product library that is required to install your selected functions.

A list follows of the default “external” product library names that can be required, along with the associated CA OPS/MVS function that requires that library for installation. Specify the actual name of the library at your site when you select the listed CA OPS/MVS function for installation.

MQM.SCSQLOAD

IBM WebsSphere MQ-Series load library.

CA OPS/MVS Base function (FMID CCLXC20) requires this library.

Important! If IBM WebSphere MQ-Series is not installed at your site, you must specify the name of the CA OPS/MVS sidedeck SMP/E Target library that is installed by CA OPS/MVS. For example, hlq.CCLXSIDE.

SYS1.SIEASID

IBM Cryptographic Services Secure Sockets Layer sidedeck library.

CA OPS/MVS OPSLOG Webview function (FMID CCLXC21) requires this library.

CAI.CNSMSDF

CA Common Services (CCS) Agent Technologies sidedeck library.

CA OPS/MVS NSM SSM Option function (FMID CCLXC22) requires this library.

This library is installed as CAI.CNSMSDF at CCS r14.

This library is installed as CAI.CAISDF at CCS r12.

This library is installed as CAI.EXP at CCS r11.

CICTS.SDFHLOAD

IBM CICS load library.

CA OPS/MVS CICS Operations Facility function (FMID CCLXC23) requires this library.

/user/dll/

IBM Hardware Console Management API DLL USS directory.

CA OPS/MVS OPSHMC REXX function (FMID CCLXC24) requires this USS directory.

Download the DLL file named HWMCAAPI from the IBM website into the file HWMCAAPI of this USS directory. The file is available for download from http://www.ibm.com/servers/resourcelink; click Services, then API, then z/OS. For more information, see *IBM manual System z Application Programming Interfaces* (SB10-7030-15).

USER.HWMCAEXP

IBM Hardware Console Management API sidedeck library.

CA OPS/MVS OPSHMC REXX function (FMID CCLXC24) requires this library.

This library must be RECFM=FB/LRECL=80.

Download the sidedeck file named HWMCAAPI.x from the IBM website into the member HWMCAAPI of this library. The file is available for download from http://www.ibm.com/servers/resourcelink; click Services, then API, then z/OS. For more information, see *IBM manual System z Application Programming Interfaces* (SB10-7030-15).

Tailor CA OPS/MVS OPSLOG WebView and Web Services zFS Mount for Your Site

Installation of CA OPS/MVS OPSLOG WebView and Web Services functions (FMID CCLXC21) allocates and mounts a zFS at your site. Customize the high-level qualifier of the allocated zFS, as well as the USS mount point created. The zFS name and mount point are defined to SMP/E as DDDEFs.

During step 4 of CA CSM installation, provide the high-level qualifier of the zFS to be allocated and the USS mount point to be created.

During Native SMP/E JCL installation, manually update the SAMPJCL members CL11ALLU, CL12MKD, and CL13CSIU with the high-level qualifier of the zFS to be allocated and the USS mount point to be created.

Chapter 3: Installing Your Product Using CA CSM

This section contains the following topics:

How to Install Your Product Using CA CSM

As a system programmer, your responsibilities include acquiring, installing, maintaining, deploying, and configuring CA Technologies mainframe products on your system.

CA CSM is an application that simplifies and unifies the management of your CA Technologies mainframe products on z/OS systems. As products adopt the CA CSM services, you can install your products in a common way according to industry best practices.

If you do not have CA CSM installed, download it from the Download Center at http://ca.com/support. This web page also contains links to the complete documentation for CA CSM.

You can use the following scenarios to guide you through the product installation process (see page 19) using CA CSM:

- Acquiring Products Using CA CSM
- Installing Products Using CA CSM
- Maintaining Products Using CA CSM
- Configuring Product Using CA CSM

These scenarios are available in the CA CSM Version 6.0 bookshelf at http://ca.com/support. For additional information about how to use CA CSM, use the online help.

Chapter 4: Installing Your Product Using Pax ESD or DVD

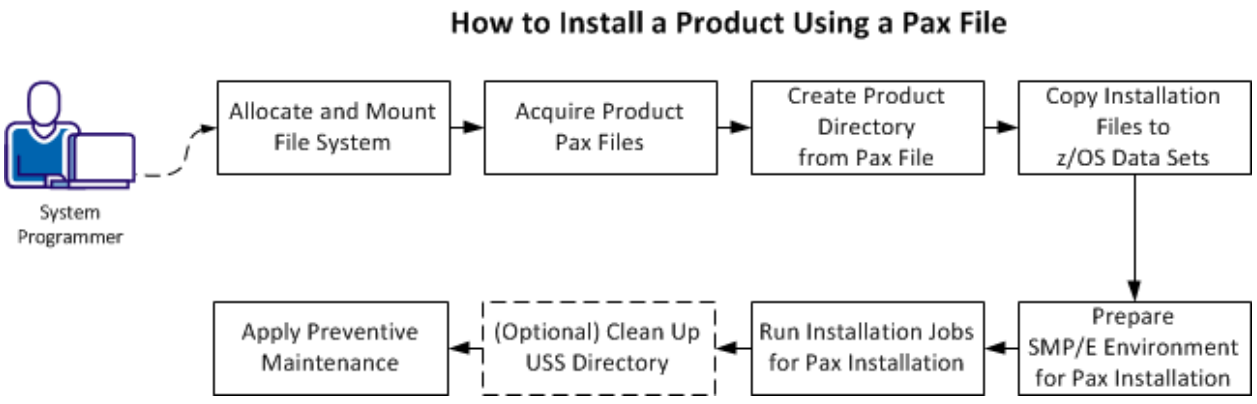
This section contains the following topics:

How to Install Your Product Using a Pax File

As a system programmer, your responsibilities include installing products on your mainframe system. With this option, you acquire a product pax file from http://ca.com/support or from a product DVD.

The DVD contains a folder that includes the pax file for the product. Product updates may have occurred after you acquired the product DVD. The files on the online site always have the most current product updates. To determine if you have the latest updates, go to http://ca.com/support and click Download Center.

You perform the following tasks to install a product with a pax file:



1. Allocate and mount the file system (see page 35).
2. Acquire the product pax files (see page 37).

3. Create a product directory from the pax file (see page 42).
4. Copy the installation files to z/OS data sets (see page 43).
5. Prepare the SMP/E environment for a pax installation.
6. Run the installation jobs for a pax installation (see page 47).
7. (Optional) Clean up the USS directory (see page 48).
8. Apply preventive maintenance.

USS Environment Setup

You need a UNIX System Services (USS) directory and a file system with adequate space to perform the following tasks:

- Receive product pax files from http://ca.com/support.
- Perform utility functions to unpack the pax file into MVS data sets that you can use to complete the product installation.

We recommend that you allocate and mount a file system that is dedicated to Pax ESD. The amount of space that you need for the file system depends on the following variables:

- The size of the pax files that you intend to download.
- Whether you plan to keep the pax files after unpacking them. We do not recommend this practice.

We recommend that you use one directory for downloading and unpacking pax files. Reusing the same directory minimizes USS setup. You need to complete the USS setup only one time. You reuse the same directory for subsequent downloads. Alternatively, you can create a directory for each pax download.

Important! Downloading pax files for the SMP/E installation as part of the Pax ESD process requires write authority to the UNIX System Services (USS) directories that are used for the Pax ESD process. In the file system that contains the Pax ESD directories, you also need free space approximately 3.5 times the pax file size to download the pax file and unpack its contents. For example, to download and unpack a 14 MB pax file, you need approximately 49 MB of free space in the file system hosting your Pax ESD directory.

Allocate and Mount a File System

The product installation process requires a USS directory to receive the pax file and to perform the unpack steps. We recommend that you allocate and mount a file system that is dedicated to the product acquisition and create the directory in this file system.

You can use the zSeries File System (zFS) or hierarchical file system (HFS) for product downloads.

This procedure describes how to perform the following tasks:

- Allocate a zFS or an HFS.
- Create a mount point in an existing maintenance USS directory of your choice.
- Mount the file system on the newly created mount point.

  Note: You must have either SUPERUSER authority, or the required SAF profile setting to allow you to issue the USS mount command for the file system.
- Optionally, permit write access to anyone in the same group as the person who created the directory.

  Important! USS commands are case-sensitive.

Follow these steps:

1. Allocate the file system by customizing one of the following samples to your site requirements:

   - On a zFS, use the following sample:

```
//DEFINE   EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//AMSDUMP  DD   SYSOUT=*
//SYSIN    DD *
  DEFINE CLUSTER ( +
  NAME(your_zFS_data_set_name) +
  STORAGECLASS(class) +
  LINEAR +
  CYL(primary secondary) +
  SHAREOPTIONS(3,3) +
  )
/*
//FORMAT   EXEC PGM=IOEAGFMT,REGION=0M,
// PARM=('-aggregate your_zFS_data_set_name -compat')
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//STDOUT   DD   SYSOUT=*
//STDERR   DD   SYSOUT=*
//CEEDUMP  DD   SYSOUT=*
//*
```

- On an HFS, use the following sample:

```
//ALCHFS EXEC PGM=IEFBR14
//CAPAX  DD   DSN=yourHFS_data_set_name,
//            DISP=(NEW,CATLG,DELETE),UNIT=3390,
//            DSNTYPE=HFS,SPACE=(CYL,(primary,secondary,1))
```

The file system is allocated.

Note: Ensure that the zFS or HFS data set name that you use conforms to your data set naming conventions for USS file systems. If the allocation of the file system data set fails, it is because of environmental settings not allowing for the allocation. On an HFS, try using the ISPF 3.2 Data Set Utility to allocate your HFS data set.

2. Create a mount point for the file system. This example shows how to create a /CA/CAPAX directory in an existing directory, /u/maint. From the TSO OMVS shell, enter the following commands:

```
cd /u/maint/
mkdir CA
cd CA
mkdir CAPAX
```

Note: This document refers to this structure as *yourUSSpaxdirectory*.

The mount point is created.

3. Mount the file system by customizing one of the following samples to your site requirements:

- On a zFS, use the following sample:

```
MOUNT FILESYSTEM('your_zFS_data_set_name')
      MOUNTPOINT('yourUSSpaxdirectory')
      TYPE(ZFS)  MODE(RDWR)
      PARM(AGGRGROW)
```

- On an HFS, use the following sample:

```
MOUNT FILESYSTEM('your_HFS_data_set_name')
      MOUNTPOINT('yourUSSpaxdirectory')
      TYPE(HFS)  MODE(RDWR)
```

The file system is mounted.

4. (Optional) Set security permissions for the directory. You can use the chmod command to let other users access the Pax ESD directory and its files. For example, to allow write access to the Pax ESD directory for other users in your USS group, from the TSO OMVS shell, enter the following command:

```
chmod -R 775 /yourUSSpaxdirectory/
```

Write access is granted.

Note: For more information about the chmod command, see the IBM *z/OS UNIX System Services User Guide* (SA22-7802).

Acquire the Product Pax Files

To begin the CA Technologies product installation procedure, copy the product pax file into the USS directory that you set up.

Important! Downloading pax files for the SMP/E installation as part of the Pax ESD process requires write authority to the UNIX System Services (USS) directories that are used for the Pax ESD process. Also, you must have available USS file space before you start the procedures in this guide.

Use one of the following methods:

- Download the product pax file from http://ca.com/support to your PC (see page 38), and then upload it to your USS file system.

  If you download a zip file, you must unzip it before uploading to your USS file system.
- Download the pax files from http://ca.com/support directly to your USS file system (see page 38).
- Download the pax file from the product DVD to your PC, and then upload the pax files to your USS file system. (see page 41)

This section includes the following information:

- A sample batch job to download a product pax file from the CA Support Online FTP server directly to a USS directory on your z/OS system
- Sample commands to upload a pax file from your PC to a USS directory on your z/OS system

Important! The FTP procedures vary due to local firewall and other security settings. Consult your local network administrators to determine the appropriate FTP procedure to use at your site.

Ensure that sufficient free space is available in the USS file system that you are using to hold the product pax file. If you do not have sufficient free space, error messages similar to the following appear:

```
EZA1490I Error writing to data set
EZA2606W File I/O error 133
```

When the download finishes, the pax file size in your USS directory matches the value in the Size column for the corresponding pax file on the CA Technologies Products Download window.

Download Files to a PC Using Pax ESD

You can download product installation files from http://ca.com/support to your PC.

Follow these steps:

1. Log in to http://ca.com/support, and click Download Center.

   The Download Center web page appears.

2. Under Download Center, select Products from the first drop-down list, and specify the product, release, and gen level (if applicable), and click Go.

   The CA Product Download window appears.

3. Download an entire CA Technologies product software package or individual pax files to your PC. If you download a zip file, you must unzip it before continuing.

   Note: For traditional installation downloads, see the *Traditional ESD User Guide*. For information about download methods, see the Download Methods and Locations article. Go to http://ca.com/support, log in, and click Download Center. Links to the guide and the article appear under the Download Help heading.

Download Using Batch JCL

You download a pax file from http://ca.com/support by running batch JCL on the mainframe. Use the sample JCL attached to the PDF file as CAtoMainframe.txt (see page 40) to perform the download.

Important! The PDF version of this guide includes sample JCL jobs that you can copy directly to the mainframe. To access these jobs, click the paper clip icon at the left of the PDF reader. A window displaying attachments opens. Double-click a file to view a sample JCL. We recommend that you use the latest version of Adobe Reader for viewing PDF files.

Note: We recommend that you follow the preferred download method as described on http://ca.com/support. This JCL procedure is our preferred download method for users who do not use CA CSM. We also include the procedure to download to the mainframe through a PC in the next section.

Follow these steps:

1. Replace *ACCOUNTNO* with a valid JOB statement.
2. Replace *yourTCPIP.PROFILE.dataset* with the name of the TCP/IP profile data set for your system. Consult your local network administrators, if necessary.

   The job points to your profile.

3. Replace *YourEmailAddress* with your email address.

   The job points to your email address.

4. Replace *yourUSSpaxdirectory* with the name of the USS directory that you use for Pax ESD downloads.

   The job points to your USS directory.

5. Locate the product component to download on the CA Support Product Download window.

   You have identified the product component to download.

6. Click Download for the applicable file.

   Note: For multiple downloads, add files to a cart.

   The Download Method window opens.

7. Click FTP Request.

   The Review Download Requests window displays any files that you have requested to download.

   Note: We send you an email when the file is ready to download or a link appears in this window when the file is available.

8. Select one of the following methods:

   Preferred FTP

   Uses CA Technologies worldwide content delivery network (CDN). If you cannot download using this method, review the security restrictions for servers that company employees can download from that are outside your corporate network.

   Host Name: ftp://ftpdownloads.ca.com

   Alternate FTP

   Uses the original download servers that are based on Long Island, New York.

   Host Name: ftp://scftpd.ca.com for product files and download cart files and ftp://ftp.ca.com for individual solution files.

   Both methods display the host, user name, password, and FTP location, which you then can copy into the sample JCL.

   Note: The following links provide details regarding FTP: the FTP Help document link in the Review Download Requests window and the Learn More link available in the Download Methods window.

9. Submit the job.

   Important! If your FTP commands are incorrect, it is possible for this job to fail and still return a zero condition code. Read the messages in the job DDNAME SYSPRINT to verify the FTP succeeded.

   After you run the JCL job, the pax file resides in the mainframe USS directory that you supplied.

Example: CAtoMainframe.txt, JCL

The following text appears in the attached CAtoMainframe.txt JCL file:

```
//GETPAX   JOB (ACCOUNTNO),'FTP GET PAX ESD PACKAGE',
//          MSGCLASS=X,CLASS=A,NOTIFY=&SYSUID
//*********************************************************************
//* This sample job can be used to download a pax file directly from  *
//* CA Support Online to a USS directory on your z/OS system.         *
//*                                                                   *
//* When editing the JCL ensure that you do not have sequence numbers *
//* turned on.                                                        *
//*                                                                   *
//* This job must be customized as follows:                           *
//* 1. Supply a valid JOB statement.                                  *
//* 2. The SYSTCPD and SYSFTPD JCL DD statements in this JCL may be   *
//*    optional at your site. Remove the statements that are not      *
//*    required. For the required statements, update the data set     *
//*    names with the correct site-specific data set names.           *
//* 3. Replace "Host" based on the type of download method.           *
//* 4. Replace "YourEmailAddress" with your email address.            *
//* 5. Replace "yourUSSpaxdirectory" with the name of the USS         *
//*    directory used on your system for Pax ESD downloads.           *
//* 6. Replace "FTP Location" with the complete path                  *
//*    and name of the pax file obtained from the FTP location        *
//*    of the product download page.                                  *
//*********************************************************************
//GETPAX   EXEC PGM=FTP,PARM='(EXIT TIMEOUT 120',REGION=0M
//SYSTCPD  DD   DSN=yourTCPIP.PROFILE.dataset,DISP=SHR
//SYSFTPD  DD   DSN=yourFTP.DATA.dataset,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//OUTPUT   DD   SYSOUT=*
//INPUT    DD   *
Host
anonymous YourEmailAddress
lcd yourUSSpaxdirectory
binary
get FTP_location
quit
/*
```

Download Files to Mainframe through a PC

You download the product installation files to your PC and transfer them to your USS system.

Follow these steps:

1. Download the product file to your PC using one of the following methods:
   - Pax ESD (see page 38). If you downloaded a zip file, first unzip the file to use the product pax files.
   - DVD. Copy the entire product software package (or individual pax files) to your PC.

   The pax file resides on your PC.

   Note: Do *not* change the format of the pax.Z.

2. Open a Windows command prompt.

   The command prompt appears.

3. Customize and enter the following FTP commands:

```
FTP mainframe
userid
password
bin
lcd C:\PC\folder\for\thePAXfile
cd /yourUSSpaxdirectory/
put paxfile.pax.Z
quit
exit
```

mainframe

Specifies the z/OS system IP address or DNS name.

userid

Specifies your z/OS user ID.

password

Specifies your z/OS password.

C:*PC\folder\for\thePAXfile*

Specifies the location of the pax file on your PC.

Note: If you specify a location that has blanks or special characters in the path name, enclose that value in double quotation marks.

yourUSSpaxdirectory

Specifies the name of the USS directory that you use for Pax ESD downloads.

***paxfile*.pax.Z**

Specifies the name of the pax file to upload.

The pax file is transferred to the mainframe.

Create a Product Directory from the Pax File

The pax command performs the following actions:

- Extracts the files and directories that are packaged within the pax file.
- Creates a USS directory in the same directory structure where the pax file resides.
- Automatically generates a product and level-specific directory name.

Set the current working directory to the directory containing the pax file, and create a directory in your USS directory by entering the following command:

```
pax -rvf pax-filename
```

Use the sample JCL that is attached to the PDF file as Unpackage.txt (see page 43) to extract the product pax file into a product installation directory.

Important! The PDF version of this guide includes sample JCL jobs that you can copy directly to the mainframe. To access these jobs, click the paper clip icon at the left of the PDF reader. A window displaying attachments opens. Double-click a file to view a sample JCL. We recommend that you use the latest version of Adobe Reader for viewing PDF files.

Follow these steps:

1. Replace *ACCOUNTNO* with a valid JOB statement.
2. Replace *yourUSSpaxdirectory* with the name of the USS directory that you use for product downloads.

   The job points to your specific directory.
3. Replace *paxfile.pax.Z* with the name of the pax file.

   The job points to your specific pax file.
4. Submit the job.

   The job creates the product directory.

Note: If the PARM= statement exceeds 71 characters, uncomment and use the second form of UNPAXDIR instead. This sample job uses an X in column 72 to continue the PARM= parameters to a second line.

Example: JCL File, Unpackage.txt, to Customize

The following text appears in the attached Unpackage.txt JCL file:

```
//ESDUNPAX JOB (ACCOUNTNO),'UNPAX PAX ESD PACKAGE',
// MSGCLASS=X,CLASS=A,NOTIFY=&SYSUID
//*********************************************************************
//* This sample job can be used to invoke the pax command to create   *
//* the product-specific installation directory.                      *
//*                                                                   *
//* This job must be customized as follows:                           *
//* 1. Supply a valid JOB statement.                                  *
//* 2. Replace "yourUSSpaxdirectory" with the name of the USS         *
//*    directory used on your system for Pax ESD downloads.           *
//* 3. Replace "paxfile.pax.Z" with the name of the pax file.         *
//* NOTE: If you continue the PARM= statement on a second line,       *
//*       start entering characters in column 16 and make sure        *
//*       the 'X' continuation character is in column 72.             *
//*********************************************************************
//UNPAXDIR EXEC PGM=BPXBATCH,
// PARM='sh cd /yourUSSpaxdirectory/; pax -rvf paxfile.pax.Z'
//*UNPAXDIR EXEC PGM=BPXBATCH,
//* PARM='sh cd /yourUSSpaxdirectory/; pax                            X
//*             -rvf paxfile.pax.Z'
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
```

Copy Installation Files to z/OS Data Sets

Use this procedure to invoke the SMP/E GIMUNZIP utility to create MVS data sets from the files in the product-specific directory.

The file UNZIPJCL in the product directory contains a sample job to GIMUNZIP the installation package. You edit and submit the UNZIPJCL job to create z/OS data sets.

Follow these steps:

1. Locate and read the product readme file or installation notes, if applicable, which resides in the product-specific directory that the pax command created. This file contains the product-specific details that you require to complete the installation procedure.

   You have identified the product-specific installation details.

2. Use ISPF EDIT or TSO ISHELL to edit the UNZIPJCL sample job. You can perform this step in one of the following ways:

   - Use ISPF EDIT. Specify the full path name of the UNZIPJCL file.
   - Use TSO ISHELL. Navigate to the UNZIPJCL file and use the E line command to edit the file.

   The job is edited.

3. Change the SMPDIR DD PATH to the product-specific directory created by the pax command.

   Your view is of the product-specific directory.

4. If ICSF is not active, perform the following steps:

   a. Change the SMPJHOME DD PATH to your Java runtime directory. This directory varies from system to system.

   b. Perform one of the following steps:

      - Change the SMPCPATH DD PATH to your SMP/E Java application classes directory, typically /usr/lpp/smp/classes/.
      - Change HASH=YES to HASH=NO on the GIMUNZIP parameter.

   One of the following occurs: ICSF is active or you are using Java.

5. Change all occurrences of y*ourHLQ* to the high-level qualifier (HLQ) for z/OS data sets that the installation process uses. We suggest that you use a unique HLQ for each expanded pax file to identify uniquely the package. Do *not* remove CAI after *yourHLQ*. Do *not* use the same value for y*ourHLQ* as you use for the SMP/E RELFILEs.

   All occurrences of y*ourHLQ* are set to your high-level qualifier for z/OS data sets.

6. Submit the UNZIPJCL job.

   The UNZIPJCL job completes with a zero return code. Messages GIM69158I and GIM48101I in the output and IKJ56228I in the JES log are acceptable.

   GIMUNZIP creates z/OS data sets with the high-level qualifier that you specified in the UNZIPJCL job. You use these data sets to perform the product installation. The pax file and product-specific directory are no longer needed.

   Note: For more information, see the IBM *SMP/E for z/OS Reference (SA22-7772)*.

Prepare the SMP/E Environment for a Pax Installation

The following steps describe the process to install products using native SMP/E JCL:

1. Download external HOLDDATA.
2. Allocate product data sets and SMP/E data sets.
3. Create an SMP/E environment.
4. Receive base functions.
5. Apply base functions.
6. Accept base functions.
7. Configure the product according to your site requirements.

The members that are used in this procedure prepare the data sets, initialize the zones, and create the DDDEFs for your product.

Determine if you are going to install the CA OPS/MVS OPSLOG WebView and Web Services function (FMID CCLXC21), before starting this procedure.

For information about the members, see the comments in the JCL.

Follow these steps:

1. Customize the macro CLXSEDIT with your site-specific information and then copy the macro to your SYSPROC location. Replace the rightmost parameters for each ISREDIT CHANGE command. Each time you edit an installation member, type CLXSEDIT on the command line, and press Enter to replace the defaults with your specifications.

   The macro is ready to customize the *yourHLQ*.SAMPJCL members.

   Note: Set the DASD HLQ to the same value specified for *yourHLQ* within the JCL that is used to unzip the pax file.

   Note: The following steps include instructions to execute the CLXSEDIT macro each time you open a new SAMPJCL member. To edit all SAMPJCL members simultaneously, read and follow the instructions in the CLXAREAD member, and submit the CLXEDALL member.

2. Open the SAMPJCL member CLX1HOLD in an edit session and execute the CLXSEDIT macro from the command line.

   CLX1HOLD is customized.

3. Submit CLX1HOLD.

   This job downloads the error and FIXCAT HOLDDATA from http://ca.com/support.

4. Open the SAMPJCL member CLX2ALL in an edit session and execute the CLXSEDIT macro from the command line.

   CLX2ALL is customized.

5. Submit CLX2ALL.

   This job produces the following results:

   - The target and distribution data sets for your product are created.
   - Unique SMPLTS, SMPMTS, SMPSCDS, and SMPSTS data sets for this target zone are created.

6. If you choose to install the CA OPS/MVS OPSLOG WebView and Web Services function (FMID CCLXC21), complete the following substeps.

   Note: If your site requires it, you can customize the supplied HFS JCL to zFS.

   a. Open the SAMPJCL member CL12ALLU in an edit session and execute the CLXSEDIT macro from the command line.

      Make supplementary modifications to the JCL as instructed by comments in the member.

      CL12ALLU is customized.

   b. Submit CL12ALLU.

      This job allocates your HFS data sets.

   c. Open the SAMPJCL member CL13MKD in an edit session and execute the CLXSEDIT macro from the command line.

      Make supplementary modifications to the JCL as instructed by comments in the member.

      CL13MKD is customized.

   d. Submit CL13MKD.

      This job creat*es* all directories and mounts the file system.

7. Open the SAMPJCL member CLX3CSI in an edit session and execute the CLXSEDIT macro from the command line.

   CLX3CSI is customized.

8. Submit CLX3CSI.

   This job produces the following results:

   - The CSI data set is defined.
   - The SMPPTS and SMPLOG data sets are allocated.
   - The global, target, and distribution zones are initialized.
   - The DDDEF entries for your product are created.
   - The DDDEFs for the required SMP/E data sets are created.

9. If you choose to install the CA OPS/MVS OPSLOG WebView and Web Services function (FMID CCLXC21), complete the following substeps:

   Note: If your site requires it, you can customize the supplied HFS JCL to zFS.

   a. Open the SAMPJCL member CL13CSIU in an edit session and execute the CLXSEDIT macro from the command line.

      CL13CSIU is customized.

   b. Submit CL13CSIU.

      This job customizes the CSI *by* adding the DDDEFs associated with the directory.

Run the Installation Jobs for a Pax Installation

Submit and run these SAMPJCL members in sequence. Do not proceed with any job until the previous job has completed successfully.

Note: The following steps include instructions to execute the CLXSEDIT macro each time you open a new SAMPJCL member. To edit all SAMPJCL members simultaneously, read and follow the instructions in the CLXAREAD member, and submit the CLXEDALL member.

Follow these steps:

1. Open the SAMPJCL member CLX4RECD in an edit session, and execute the CLXSEDIT macro from the command line.

   Remove any unwanted FMIDs.

   CLX4RECD is customized.

2. Submit CLX4RECD to receive SMP/E base functions.

   CA OPS/MVS is received and now resides in the global zone.

3. Open the SAMPJCL member CLX5APP in an edit session, and execute the CLXSEDIT macro from the command line.

   Remove any unwanted FMIDs.

   CLX5APP is customized.

4. Submit CLX5APP to apply SMP/E base functions.

   CA OPS/MVS is applied and now resides in the target libraries.

5. Open the SAMPJCL member CLX6ACC in an edit session, and execute the CLXSEDIT macro from the command line.

   Remove any unwanted FMIDs.

   CLX6ACC is customized.

6. Submit CLX6ACC to accept SMP/E base functions.

   CA OPS/MVS is accepted and now resides in the distribution libraries.

Clean Up the USS Directory

This procedure is optional. If you decide to perform the procedure, do so after you complete the installation process and when you do not need the installation files anymore.

To free file system disk space for subsequent downloads after downloading and processing the pax files for your CA Technologies product, we recommend removing the files from your USS directory and deleting unnecessary MVS data sets. You can delete the following items:

- Pax file
- Product-specific directory that the pax command created and all of the files in it
- SMP/E RELFILEs, SMPMCS, and HOLDDATA MVS data sets

  These data sets have the HLQ that you assigned in the UNZIPJCL job.

Note: Retain non-SMP/E installation data sets such as *yourHLQ*.INSTALL.NOTES for future reference.

Follow these steps:

1. Navigate to your Pax ESD USS directory.

   Your view is of the applicable USS directory.

2. Delete the pax file by entering the following command:

   `rm paxfile`

   paxfile

   Specifies the name of the CA Technologies pax file that you downloaded.

   The pax file is deleted.

3. Delete the product-specific directory by entering the following command:

   `rm -r product-specific_directory`

   product-specific_directory

   Specifies the product-specific directory that the pax command created.

   The product-specific directory is deleted.

Note: You can also use TSO ISHELL to navigate to the pax file and product-specific directory, and delete them using the D line command.

Apply Preventive Maintenance

Important! We strongly recommend that you use CA CSM to maintain your CA Technologies z/OS-based products. The procedure that is discussed in this section is fully automated when you use CA CSM.

CA Support Online at http://ca.com/support has maintenance and HOLDDATA published since the installation data was created. After the maintenance process completes, the product is ready to deploy.

Use this procedure during product installation and for ongoing preventive maintenance in non-installation use cases according to your maintenance strategy.

Note: To review the CA Technologies mainframe maintenance philosophy, see your *Best Practices Guide* or visit the CA Next-Generation Mainframe Management page.

This procedure directs you to use the CAUNZIP utility. The CAUNZIP utility processes ZIP packages directly on z/OS without the need for an intermediate platform, such as a Microsoft Windows workstation. If you are not familiar with this utility, see the *CA Common Services for z/OS Administration Guide*. This guide includes an overview and sample batch jobs. To use this utility, you must be running CA Common Services for z/OS Version 14.0 with PTF RO54887 or CA Common Services for z/OS Release 14.1 with PTF RO54635 and RO58216. These PTFs are included in CA Common Services for z/OS Release 14.1 at the S1401 Service Update level.

Follow these steps:

1. Check the Download Center at http://ca.com/support for PTFs that have been published since this release was created. If the base release was created recently, no PTFs will have been published yet. If PTFs exist, add published solutions for your product to your Download Cart, and click Checkout.
2. Specify that you want a complete package.

   When processing completes, a link appears on the Review Download Requests page. You also receive an email notification.

3. Click the Alternate FTP link for your order to obtain FTP login information and the ZIP file location. Download the ZIP file into a USS directory on your z/OS system.
4. Run the CAUNZIP utility.

   CAUNZIP unzips the package of published solutions and creates a SMPNTS file structure that the SMP/E RECEIVE FROMNTS command can process. For sample JCL to run the utility that is located in *yourHLQ*.CAW0JCL(CAUNZIP), see the *CA Common Services for z/OS CAUNZIP Administration Guide*. After execution completes, the ZIPRPT data set contains the summary report. The summary report does the following:

   - Summarizes the content of the product order ZIP file.
   - Details the content of each data set and the z/OS UNIX files produced.
   - Provides a sample job to receive the PTFs in your order.
5. Review the sample job that is provided in the CAUNZIP output ZIPRPT file. Cut and paste the JCL into a data set, specify your SMP/E CSI on the SMPCSI DD statement and submit the job to receive the PTFs in your order.
6. Verify that you have the values from the base installation in the CLXSEDIT macro that was customized in the installation steps.
7. Open the SAMPJCL member CLX1HOLD in an edit session and execute the CLXSEDIT macro from the command line.

   Note: Update CLX1HOLD SAMPJCL to download the HOLDDATA file.

   CLX1HOLD is customized.
8. Submit CLX1HOLD.

   The job downloads the external HOLDDATA file.
9. Open the SAMPJCL member CLX7RECH in an edit session and execute the CLXSEDIT macro from the command line.

   CLX7RECH is customized.
10. Submit CLX7RECH.

    The job receives the external HOLDDATA file.

11. (CA Recommended Service (CA RS)) installation only) Do the following:
    a. Determine which ASSIGN statements to download.
        - The yearly CA RS ASSIGN statements are stored in the following file:

          `ftp.ca.com/pub/ASSIGN/YEARLY/CAR`*yyyy*`.TXT`
        - The quarterly CA RS ASSIGN statements are stored in the following file:

          `ftp.ca.com/pub/ASSIGN/CAR`*yymm*`.TXT`

    b. Open the SAMPJCL member CLX7CARS in an edit session, update CLX7CARS SAMPJCL to download ASSIGN statements from http://ca.com/support, and execute the CLXSEDIT macro from the command line.

    CLX7CARS is customized.
12. (CA RS installation only) Submit CLX7CARS.

    The job downloads the CA RS ASSIGN statements.
13. (CA RS installation only) Open the SAMPJCL member CLX7RECP in an edit session, manually add the data set that contains the ASSIGN statements to the SMPPTFIN DD, and execute the CLXSEDIT macro from the command line.

    CLX7RECP is customized.
14. (CA RS installation only) Submit CLX7RECP.

    The job receives the external HOLDDATA file and CA RS ASSIGN statements.
15. Open the SAMPJCL member CLX8APYP in an edit session and execute the CLXSEDIT macro from the command line.

    CLX8APYP is customized.
16. Submit CLX8APYP.

    The PTFs are applied.
17. (Optional) Open the SAMPJCL member CLX9ACCP in an edit session and execute the CLXSEDIT macro from the command line.

    CLX9ACCP is customized.
18. (Optional) Submit CLX9ACCP.

    The PTFs are accepted.

    Note: You do not have to submit the job at this time. You can accept the PTFs according to your site policy.

HOLDDATA

When you apply maintenance, you typically encounter SMP/E HOLDDATA. We use HOLDDATA to notify your SMP/E system of SYSMODs that have errors or special conditions. We support system and external HOLDDATA.

System HOLDDATA

System HOLDDATA indicates data that is an in-stream part of the SYSMOD, informing you of special conditions. The following reasons are used with SYSTEM HOLDDATA for your product:

ACTION

Indicates that you must perform special processing before or after you apply this SYSMOD.

AO

Affects automated operations. It changes either the message identifier or the displacement of a field inside the message.

DDDEF

Indicates that data sets and DDDEFs are being added or modified.

DELETE

Deletes the SYSMOD load module. You cannot reverse this type of SYSMOD with the SMP/E RESTORE command.

DEP

Indicates a dependency for this SYSMOD that you must externally verify.

DOC

Indicates a documentation change with this SYSMOD.

DYNACT

Describes the steps to dynamically activate this fix without performing an IPL.

EC

Indicates that this SYSMOD requires a hardware engineering change. An EC hold SYSMOD usually does not affect the product unless the EC is present on the hardware device.

ENH

Introduces a small programming enhancement. The hold contains the instructions to implement the enhancement. If no action is needed to implement the enhancement, give a summary of the enhancement.

EXIT

Indicates that changes delivered by this SYSMOD require reassembly of user exits.

EXRF

Indicates that the SYSMOD must be installed in both the Active and Alternate Extended Recovery Facility Systems.

IPL

Indicates that an IPL is required for this SYSMOD to take effect. This is used only when there is no alternative for dynamic activation.

MSGSKEL

Indicates that the SYSMOD contains internationalized message versions that must be run through the message compiler for each language.

MULTSYS

Apply this SYSMOD to multiple systems for either pre-conditioning, coexistence, or exploitation.

RESTART

Indicates that after applying this SYSMOD, the site must perform a special restart as opposed to a routine restart.

SQLBIND

Indicates that a bind is required for a database system other than DB2.

DOWNLD

Indicates that some or all of the elements that this SYSMOD delivers are to be downloaded to a workstation.

Code a BYPASS(HOLDSYS) operand on your APPLY command to install SYSMODs that have internal holds. Code the BYPASS(HOLDSYS) operand only after you have performed the required action, or if you are performing the action after the APPLY, if that is appropriate.

External HOLDDATA

External HOLDDATA is not part of the PTF. The HOLDDATA resides in a separate file and contains both error and FIXCAT HOLDDATA. The error HOLDDATA is used for SYSMODs that have been distributed and later are discovered to cause problems. The FIXCAT HOLDDATA helps identify maintenance that is required to support a particular hardware device, software, or function.

Download the external HOLDDATA from http://ca.com/support to a DASD file, and allocate the file to the SMPHOLD DD statement. To take care of the external HOLDDATA, receive it into your SMP/E environment. SMP/E receives the HOLDDATA from CA-supplied jobs.

You can find JCL to download the external HOLDDATA in your SAMPJCL member. Open CLX1HOLD in an edit session and execute the CLXSEDIT macro on the command line. Then, submit the JCL.

Error HOLDDATA

If a SYSMOD has unresolved error HOLDDATA, SMP/E does not install it unless you add a bypass to your APPLY command. You can bypass error HOLDDATA in situations that are not applicable to you. Error HOLDDATA that is not applicable to you can include a problem that happens only with a hardware device that you do not have or in a product feature that you do not use.

When CA Technologies publishes a SYSMOD that resolves the error HOLDDATA, the resolving SYSMOD supersedes the error HOLDDATA. This action lets you apply the original SYSMOD in conjunction with the fixing SYSMOD.

The only manual task is running a REPORT ERRSYSMODS. This report identifies the following:

- Any held SYSMODs already applied to your system
- Any resolving SYSMODs that are in RECEIVE status

SMP/E identifies the SYSMOD to apply to correct the situation.

FIXCAT HOLDDATA

CA Technologies provides FIXCAT HOLDDATA to help identify maintenance that is required to support a particular hardware device, software, or function. Fix categories are supplied as SMP/E FIXCAT HOLDDATA statements. Each FIXCAT HOLDDATA statement associates an APAR and its related fixing PTF to one or more fix categories.

Chapter 5: Starting Your Product

This section describes what you need to do to start CA OPS/MVS.

This section contains the following topics:

The procedures in this chapter prepare CA OPS/MVS to start the base components.

Note: For information on configuring optional CA OPS/MVS components, see the chapter "Configuring and Installing Optional Components (see page 105)."

Introduction

CA CSM can install, deploy, and configure CA OPS/MVS.

Installation

For more information see, Installing Your Product Using CA CSM (see page 31).

Deployment

For more information, see the How to Deploy a Product.

Configuration

CA CSM has the ability to configure product software. CA OPS/MVS takes advantage of this to a degree by doing some of the necessary configuration and customization steps within CA CSM. At this point in time, not all configuration and customization has been implemented in CA CSM so the required manual steps are provided (see page 63). Optional configuration and customization are documented in Configuring and Installing Optional Components (see page 105).

How to Prepare for Deployment

This section contains topics that describe the manual tasks you need to perform before beginning the deployment process.

Apply IBM APARs (Optional)

CA Technical Support has identified a number of IBM APARs that, if missing, may impact the operation or performance of CA OPS/MVS. We recommend that you review our current list of IBM APARs and apply only those that are appropriate to your environment.

For a current list of IBM APARS, see the appendix "IBM APARS that Impact CA OPS/MVS" or Contact CA Support.

Note: This step is optional but recommended.

Identify SMP/E Installed Target Data Sets

The following table lists the SMP/E Target data sets that are created during installation. SMP/E installs the CA OPS/MVS base product and maintenance into these data sets. You may customize the installation to use a high-level qualifier other than 'CAI' for the data set names. You may not change the low-level qualifier.

You will want to deploy a runtime copy of these data sets for configuration and execution of CA OPS/MVS.

DSNAME	SMPE Element Type	Contents
CAI.CCLXASM	++MAC	Assembler source and macros
CAI.CCLXCLS0	++CLIST	TSO CLISTs
CAI.CCLXCNTL	++SAMPENU	For example, Sample JCL PROCs, REXX, data, and scripts.
CAI.CCLXEXEC	++EXEC	OPS/REXX (not compiled)
CAI.CCLXHFS.ZFS		Mounted zFS
/usr/lpp/CAI/CCLXHFS	++HFS ++SHELLSCR	Mountpoint of zFS; contains OPSLOG WebView and Web Services USS components

DSNAME	SMPE Element Type	Contents
CAI.CCLXLOAD	++MOD	Executable load library (may be PDS)
CAI.CCLXMENU	++MSGENU	ISPF messages
CAI.CCLXMIB	++DATA	SNMP MIBs
CAI.CCLXOPEX	++USER5	OPS/REXX (compiled)
CAI.CCLXPENU	++PNLENU	ISPF panels
CAI.CCLXPLD	++MOD	Executable load library (must be PDS/E)
CAI.CCLXRULB	++USER3	BASE AOF rules
CAI.CCLXRULM	++USER3	STATEMAN AOF rules
CAI.CCLXRULS	++USER3	SAMPLE AOF rules
CAI.CCLXSAMP	++SAMP	Sample REXX
CAI.CCLXSENU	++SKLENU	ISPF skeletons
CAI.CCLXSIDE	++UTIN	LE sidedeck files
CAI.CCLXTENU	++TBLENU	ISPF tables
CAI.CCLXXML	++PRODXML	CSM/SDS and CSM/SCS XML metadata

How to Deploy the Product

You can deploy CA OPS/MVS with or without CA CSM.

How to Complete Deployment With CA CSM

The topics in this section describe the manual tasks that you perform when deploying your product using CA CSM.

You can use CA CSM to deploy a runtime copy of all of the CA OPS/MVS SMP/E-installed Target libraries to one or all of the systems at your site.

You can choose to deploy CA OPS/MVS checkpoint files as custom data sets using CA CSM.

For more information, see the How to Deploy a Product.

How to Deploy Without CA CSM

The topics in this section describe the manual tasks that you perform if you are not deploying your product using CA CSM.

Deploy a runtime copy of all of the CA OPS/MVS SMP/E-installed Target libraries to one or all of the systems at your site, using either ISPF or batch JCL. Follow the procedures in How to Configure Without CA CSM (see page 58) before starting the product.

Important! Do not directly edit the CA OPS/MVS SMP/E installed Target libraries. Do not execute CA OPS/MVS directly from the CA OPS/MVS SMP/E installed Target libraries. You should deploy a runtime copy of the CA OPS/MVS SMP/E-installed Target libraries for editing and execution.

Configure Your Product

The topics in this section describe the manual task you perform whether you are configuring using CA CSM or manually.

How to Complete Configuration With CA CSM

The following articles describe the manual tasks you perform when configuring your product using CA CSM:

How to Begin Configuration With CA CSM

You can use CA CSM to configure a usable copy of CA OPS/MVS. Currently, not all of the parameters possible to customize CA OPS/MVS have been integrated into CA CSM. So, configuration with CA CSM is now targeted to give you only a simple starting point. Further customization may be necessary using the provided OPSSXP00 parameter file. Any of the parameters that are documented to work from OPSSPA00 work exactly the same in file OPSSXP00.

Important! If you choose to configure using CA CSM now, you *must* customize parameters beyond those parameters that CA CSM supports. You add these parameters to member OPSSXP00 and *not* OPSSPA00.

Configuring Using CA CSM

CA CSM creates data sets after completing each step successfully, that is the SMP/E installation, deployment, and configuration procedures.

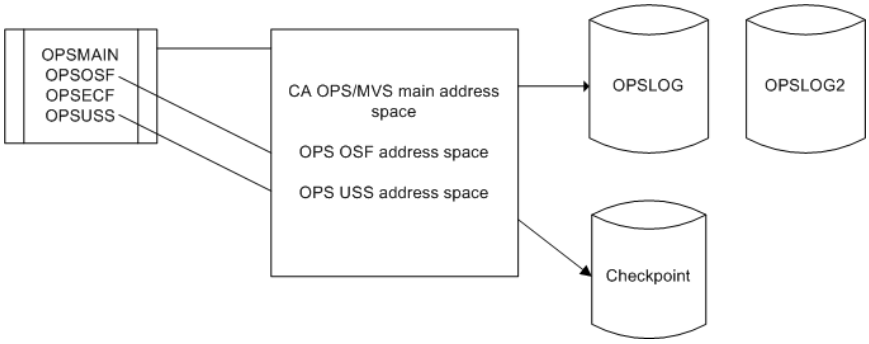
Note: CA Technologies strongly advises that you perform the initial installation and configuration of CA OPS/MVS and its components in a test environment as a precaution. This testing lets you detect any possible conflicts with other vendor products.

CA OPS/MVS Simple Configuration Example

This example shows a simple CA OPS/MVS environment that you can configure with CA CSM. This environment has a single copy of CA OPS/MVS running in one LPAR and two defined OPSLOGs. CA CSM does not presently contain support to configure CA OPS/MVS parameters files containing multiple system capability.

Note: This environment would be useful for testing the initial implementation of CA OPS/MVS in a situation where the user has little or no experience implementing and configuring CA OPS/MVS.

Note: Configuration with CA CSM automatically executes some of the steps that are otherwise documented as manual steps elsewhere in this guide.



CA OPS/MVS Configuration Options Using CA CSM

You can configure the CA OPS/MVS Base operations and the following optional components using CA CSM.

COF

CICS Operations Facility; a separate license code is required.

HWS

Hardware Services

MSF

Multi-System Facility; a separate license code is required.

USS

Unix System Services; a separate license code is required.

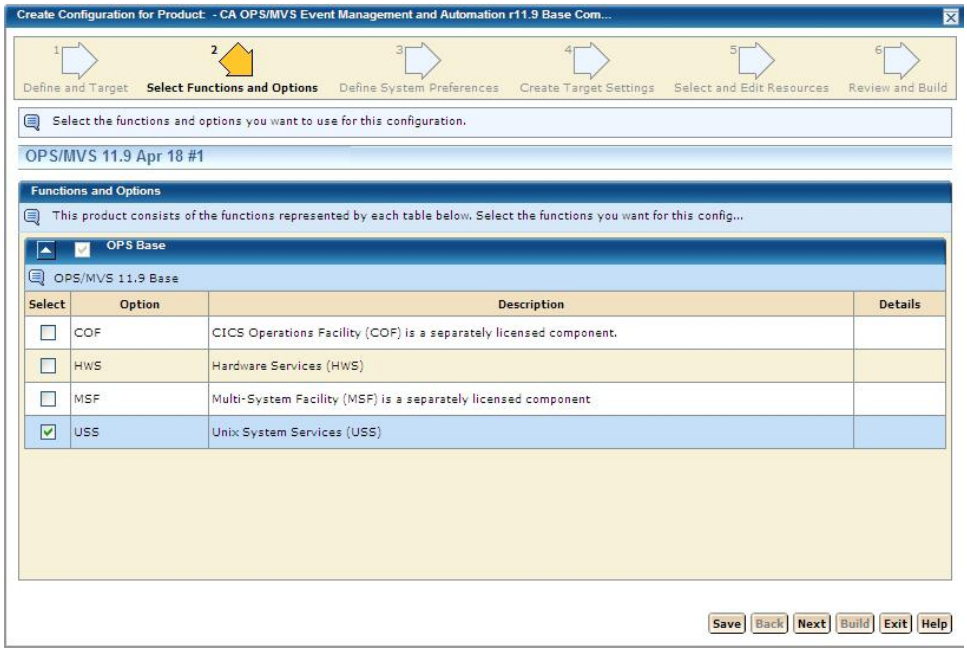
To configure the Base operations only:

Do not select any options.

To configure the Base operations and any options:

Select the options that you want to configure.

Note: Other components and facilities not shown are unavailable for configuration under CA CSM. They can still be configured manually by adding custom parameters to the OPSSXP00 member in your *hlq*.CCLXCNTL data set.



For more information on how to configure using CA CSM see the section How to Configure a Product.

Once you have configured CA OPS/MVS using CA CSM, you must complete the following manual procedures. There are exceptions to this; some steps will have been partially or fully completed by CA CSM depending on selections you have made in CA CSM. This will be noted in each subsection as appropriate.

Note: Configuration with CA CSM automatically executes some of the steps that are otherwise documented as manual steps elsewhere in this manual.

Startup JCL Procedures Customized by CA CSM

If you specified a user proclib in CA CSM for the PROCLIB, you supplied a value for the PROCLIB variable in CA CSM, the PROCs were generated and are ready to the specified user proclib by the implementation step of CA CSM.

If the user proclib *was specifie*d in PROCLIB during CA CSM configuration, then the value for the PROCs were generated and are in the user proclib specified.

If the user proclib w*as not specif*ied *bec*ause NONE was selected, then copy the CA CSM generated PROCs to a user proclib of your choice.

To deploy the JCL PROCs configured by CA CSM.

Note:. The value of @OPSPFX@ is a variable in CA CSM configuration that you supplied earlier during the CA CSM configuration. You can rename all of these PROCs.

1. Copy SCSMAIN to OPSMAIN

   Copy @OPSPFX@.CCLXCNTL(SCSMAIN) to any JCL PROCLIB that is automatically searched as part of z/OS START command processing, such as SYS2.PROCLIB(OPSMAIN)..

2. Copy SCSECF to OPSECF

   Copy @OPSPFX@.CCLXCNTL(SCSECF) to any JCL PROCLIB that is automatically searched as part of z/OS START command processing, such as SYS2.PROCLIB(OPSECF).

3. Copy SCSOSF to OPSOSF

   Copy @OPSPFX @.CCLXCNTL(SCSOSF) to any JCL PROCLIB that is automatically searched as part of z/OS START command processing, such as SYS2.PROCLIB(OPSOSF).

4. Copy SCSUSS (optional – if it exists).

   Copy @OPSPFX@.CCLXCNTL(SCSUSS) to any JCL PROCLIB that is automatically searched as part of z/OS START command processing, such as SYS2.PROCLIB(OPSUSS).

Note: The SCSUSS PROC is not created unless the USS option was selected.

Verify Your Installation

Diagnosing problems that an incomplete CA OPS/MVS installation causes are difficult to detect. Use the following checklists before starting the product to avoid such problems.

To verify your installation

1. Make sure the CA CSM generated user PROCLIB contains members OPSMAIN, OPSECF, and OPSOSF (and, optionally, OPSUSS).

   If you specified ‘NONE’ for the variable PROCLIB, the procedures that are generated by CA CSM need to be manually copied into your proclib before starting the product. Follow the procedure "To deploy the JCL PROCs configured by CA CSM” in section Startup JCL Procedures Customized by CA CSM (see page 61) for directions about how to accomplish this task manaually.

2. Make sure the OPSMAIN, OPSOSF, and OPSECF started tasks either have a STEPLIB that is authorized and contains all the load modules that are distributed with CA OPS/MVS or that all these modules are available in a LNKLSTxx load library or an LPALSTxx load library.

3. Verify that the Logical Parmlib Concatenation contains members OPSSSC00 and OPSSXP00.
4. Make sure the library that is allocated to SYSPROC contains the OPSTART1 CLIST and OPSLOGON CLIST. OPSTART2 may be located in the SYSPROC or SYSEXEC.
5. If you are using the CA ACF2 command limiting feature, check that the entries listed in the table in Provide a CA ACF2 Command Limiting List in the chapter "Configuration Tasks for the Base Component" are present.

Complete the Configuration

The parameter file OPSSXP00 found in hlq.CCLXCNTL can be used to specify additional parameters manually or override existing ones. Do *not* modify file OPSSSC00 as CA CSM maintains this file automatically.

Important! You *must* follow the procedures under How to Complete the Required Manual Configuration to complete the configuration process. Also, read and review Configuring and Installing Optional Components (see page 105) before proceeding to start the product for the first time.

How to Configure Without CA CSM

The topics in this section describe the manual tasks you perform if you are not configuring your product using CA CSM.

This section is intended only for the steps that need to be followed if you choose NOT to begin the CA OPS/MVS configuration process using CA CSM.

1. Customize Startup JCL PROCs (see page 63).
2. Tailor the Startup JCL (see page 64) and Startup Procedures (see page 65).
3. Define OPSLOG and Checkpoint VSAM Linear Data Sets (see page 69).
4. Verify Your Installation (see page 74).
5. How to Complete the Required Manual Configuration.

Customize Startup JCL PROCs

To customize JCL PROCs used to start CA OPS/MVS components on each system, follow these procedures.

Tailor the Startup JCL

The SYS1.OPS.CCLXCNTL data set contains four JCL members that required to run the CA OPS/MVS started tasks:

- OPSMAIN (main CA OPS/MVS address space)
- OPSOSF (TSO server address spaces)

  For more information, see Regulating OSF Servers in the Technical Notes in the *CA OPS/MVS Administration Guide*.
- OPSECF (Enhanced Console Facility address spaces)
- OPSUSS (UNIX System Services server address space)

To tailor the startup JCL

1. Copy the OPSMAIN, OPSOSF, OPSECF, and OPSUSS members into your started task procedures library. (which must be SYS1.PROCLIB if you intend to start OPSMAIN under the master subsystem).

   These members can now be tailored as described in the following steps.
2. Change the OPSMAIN Member JCL as follows:
   - Change the LOADLIB parameter in the PROC statement to the name of the deployed runtime OPS.CCLXLOAD library. If you placed the CA OPS/MVS load library (OPS.CCLXLOAD) into the linklist or LPALST, remove the STEPLIB statement and the LOADLIB parameter from the PROC statement.
   - Change the SYSPROC data set name to the name of the deployed runtime OPS.CCLXCLS0 data set.
   - Change the SYSEXEC data set names to the name of the deployed runtime OPS.CCLXEXEC library, along with your *hlq*.USER.REXX library.
- If your installation does not support VIO data sets, change the UNIT=VIO in the SYSTSPRT DD statement to a valid unit name. Ensure that VIO OPSPARM has a defined, esoteric name such as SYSDA, or properly allocate your OPSTSO DD.
- If use of above the bar 64-bit storage is automatically restricted by your installation using SM*FP*RMxx parmlib members or the SMF exit IEFUSI, then you may have to add the parameter 'MEMLIMIT=4G' to the OPSMAIN EXEC JCL statement or insure that CA OPS/MVS is not subject to any MEMLIMIT restrictions.

Note: The CAHBEXCL DD statement prevents the CA Hyper Buf product from interfering with VSAM processing requests within the CA OPS/MVS address space.

1. Change the OPSOSF and OPSECF JCL as follows:

   - Change the DSN parameter in the STEPLIB statement to the name of the deployed runtime OPS.CCLXLOAD library. Remove the STEPLIB statement completely if you placed the CA OPS/MVS load library (OPS.CCLXLOAD) into the linklist or LPALST.
   - Change the SYSPROC data set name to the name of the deployed runtime OPS.CCLXCLS0 data set.
   - Change the SYSEXEC data set names to the name of the deployed runtime OPS.CCLXEXEC library, along with your *hlq*.USER.REXX library.

   The startup JCL members are tailored and ready to run the CA OPS/MVS started tasks.

Tailor the Startup Procedures

At startup, CA OPS/MVS invokes customizable procedures that control startup and set the CA OPS/MVS parameters.

To tailor the startup procedures

1. Tailor the OPSTART1 initialization CLIST.

   The OPSTART1 initialization CLIST resides within the de*p*loyed runtime *hlq*.CCLXCLS0 data set that is allocated within the //SYSPROC concatenation of the OPSMAIN procedure. This CLIST executes within a TMP that is internally created during product start-up, and its primary purpose is to invoke the OPS/REXX program that sets up various CA OPS/MVS parameters. The OPSTART1 CLIST invokes this OPS/REXX program using the following statement:

   ```
   OX 'SYS1.PARMLIB(&SUBSYSNAME.PA&MEMBER)'
   ```

   If the residing location of SYS1.PARMLIB is not desired, then change this statement accordingly. Additionally, the SUBSYSNAME and MEMBER substitution variables are set within the OPSMAIN procedure or can optionally be overridden with the START command of OPSMAIN (S OPSMAIN, MEMBER=99).

   Default SUBSYSNAME setting: OPSS

   Default MEMBER setting: 00

   Thus, the default start-up OPS/REXX program that is called by the OPSTART1 CLIST is named OPSSPA00. For new installations of CA OPS/MVS we recommended using the default SUBSYSNAME and MEMBER settings.

2. Copy and tailor the supplied start-up OPSSPA00 OPS/REXX program.

   Copy member OPSSPA00 of the *hlq*.CCLXCNTL data set into the data set that was specified within the OPSTART1 CLIST (step 1).

   This sample provides the logic to set various control parameters within CA OPS/MVS and also allocate the SYSCHK1 DIV data set and any OPSLOG DIV data sets that were created using the DEFDIV utility during installation step 1. Follow the detailed implementation steps located within the beginning comments of this member to successfully allocate these DIV data sets, and to override default values for these specific CA OPS/MVS parameters:

   RULEPREFIX

   Prefix name of the AOF rulesets

   RULESUFFIX

   Suffix name of the AOF rulesets

   OSFCHAR

   Override default command character of '!' for OPS/MVS servers

   OSFSTC

   Name of JCL procedure for OPS/MVS servers if not using the default of OPSOSF

   For first-time users, the default settings for the remaining CA OPS/MVS control parameters let you quickly start and begin using the product. In the future, specific automation and environmental requirements may have you updating the default values of other CA OPS/MVS control parameters. Some of the most commonly updated parameters include the following:

   GLOBAL* parms

   Sets global variable parameter control.

   OSF*

   Controls all aspects of OPSOSF.

   OCCONSOLENAME, EXTENDED*, EXTRA*

   Controls the count and names of consoles used from within automation to issue commands to the system.

   SSICMD, SSIMSG

   Determines how command and WTO hooks are to be set.

   STATEMAN, SSM*

   Controls the System State Manager Component.

For specific details on these common parameters as well as all other CA OPS/MVS control parameters, see the Parameter and Reference. Additionally, this manual describes how to set these parameters outside of installation using the programmatic OPSPRM() OPS/REXX function, or manually using the OPSVIEW facility.

3. (Optional) Tailor the OPSTART2 OPS/REXX program

   When the main CA OPS/MVS address space completes its internal initialization (this does not mean that the AOF is completely active) and before any OSF address spaces are started, CA OPS/MVS schedules the OPSTART2 OPS/REXX program for execution in the first OSF TSO server that is ready by sending the following command to the OSF TSO execute queue:

   ```
   OI OPSTART2
   ```

   The command OI OPSTART2 is the first OSF TSO server command that is executed. The distributed OPSTART2 program is designed to run only as an OPS/REXX program. If it is invoked as a TSO/E REXX program, it issues a highlighted warning message and terminates. The OPSTART2 OPS/REXX program must be in either the SYSEXEC (source) or OPSEXEC (compiled) concatenation of the OPSOSF procedure.

   The OPSTART2 program can include any OPS/REXX functions, host commands, or TSO commands that you want to execute after CA OPS/MVS startup. OPSTART2 calls an external procedure, MSFINIT, which then calls the InitMSF internal procedure. The InitMSF procedure contains sample ADDRESS OPSCTL MSF control statements to start sessions between the current copy of CA OPS/MVS and two remote CA OPS/MVS copies.

4. Copy OPSTART2 from *hlq*.CCLXSAMPto your *hlq*.*USE*R..REXX.

   Note: You can define the MSF in the OPSTART2 program; however if you do, ensure that VTAM is running before you attempt to start it.

   OI OPSTART2 is the default initial OSF server command. You may change it using the BEGINCMD parameter during product initialization (for details, see Tailor the OPSSPA00 REXX Program in this chapter). You may also set the BEGINCMD parameter to execute a different OPS/REXX program or even a CLIST or TSO/E REXX program.

   For example:

   ```
   T = OPSPRM(“SET”,"BEGINCMD","OI FIRSTPGM")
   ```

5. (Optional) Implement an AOF initialization OPS/REXX program.

   Using the AOFINITREXX product parameter, you can specify the name of an OPS/REXX program to be executed during AOF initialization. This special OPS/REXX program executes before the product enables all auto-enabled rules, allowing you to logically control your AOF rules environment. You can use OPS/REXX language facilities to control your AOF environment based on SMF ID, time of day, or whatever criteria make sense for each system.

Almost all host command environments, like ADDRESS AOF, are available in this program. The only exception is ADDRESS TSO, where TSO commands are not allowed in the main product address space. ADDRESS TSO host commands will be treated like ADDRESS OSF host commands -they will be queued for execution in a server. At this point in CA OPS/MVS startup, the servers have not been started; the queued commands execute later when the servers are started. Access to existing global variables and relational tables is also available, which is useful for retaining information from a previous IPL or to pass information to automation routines that will execute later.

Notes:

- Any function call or host command that causes a WAIT will cause the AOF initialization of the product to be delayed. The OPSWAIT REXX function is an example.
- If the REXX program whose name you specify as the value of AOFINITREXX RETURNs or EXITs with a value of 8, the automatic enablement of all rules during AOF initialization is bypassed. All other return codes allow automatic enablement.

6. (Optional) Tailor OSFSTART OPS/REXX program.

   The OPSOSF procedure, which creates a CA OPS/MVS server address space, always invokes the OSFSTART TSO/E REXX EXEC as its first command. You can customize this REXX EXEC, which is found in the *hlq*.CCLXCLS0 data set.

   CA OPS/MVS can preallocate the data set used to capture the output of commands addressed to TSO in a server through the ADDRESS TSO host environment of OPS/REXX. To do this use the OPSTSO DD allocated in the OSFSTART REXX EXEC. After you specify this DD, the preallocation is used for all commands instead of allocating a data set for every REXX program. The ALOPSTSO subroutine in the OSFSTART REXX dynamically allocates a uniquely named OPSTSO data set for each server.

   A typical use for the OSFSTART REXX EXEC is allocating ISPF data sets for use by the server. ISPF requires a unique profile data set name for each server, which you can revise the REXX EXEC to provide.

   Note: We recommend that you use ALLOCSPF, located in the OPS.CCLXSAMP library, to allocate ISPF data sets for use by a server. This sample shows you two different ways of allocating a unique ISPF profile data set for each server. You should read the comments in this sample carefully before customizing and using it.

   Your startup procedures are defined and you are ready to verify your installation.

Define OPSLOG and Checkpoint VSAM Linear Data Sets

The CA OPS/MVS OPSLOG component and the global variable checkpoint facility require the allocation of unique VSAM linear data sets, which are also called data-in-virtual or DIV data sets. If you are a new user of CA OPS/MVS, then perform the following steps.

To define OPSLOGs and checkpoint VSAM linear data sets

1. Review detailed comments and then tailor member DEFDIV of the deployed runtime *hlq*.CCLXCNTL data set. This member contains the IDCAMS DEFINE commands that are needed to create a primary OPSLOG DIV data set, an optional secondary or backup OPSLOG DIV data set, and the SYSCHK1 DIV data set, which is used for the global variable checkpoint facility.

   Note: The placement of the DIV data sets should be based on information in How You Place the DIV Data Sets (see page 73).

   If you are installing CA OPS/MVS on multiple systems, then incorporate either the SMFID or the system name of the system into the data set names as the sample JCL illustrates. This action lets you share a common CA OPS/MVS startup member. For more information, see the section Tailor the Startup Procedures (see page 65) in this chapter.

   The DEFDIV member is tailored.

2. Use the tailored DEFDIV member either as a SYSIN statement in a batch job or execute it as a REXX program under TSO. To execute DEFDIV under TSO, enter the following command from ISPF option 6 or at the TSO command prompt:

   ```
   EXEC 'hlq.CCLXCNTL (DEFDIV)'
   ```

3. If you are not using DFSMS, define these data sets in the master catalog to allow CA OPS/MVS to start under the master subsystem.

4. Note the names that you created for the OPSLOG and SYSCHK1 data sets because you refer to them when you perform the step Tailor the OPSSPA00 REXX Program in this chapter.

   The OPSLOG and checkpoint VSAM linear data sets are defined.

Installation for Existing Customers

If you are an existing CA OPS/MVS customer and you are installing a new release of the product, then do *one* of the following tasks:

- Review the migration issues in *Release Notes*. Address any issues that pertain to your site. You can then use the current OPSLOG and SYSCHK1 data sets that you created in a previous release of CA OPS/MVS with this release.
- Create new OPSLOG and SYSCHK1 data sets by following the steps for new users of CA OPS/MVS described in the previous topic. Use the same allocation specifications for the new data sets that you specified in your current ones. Also, determine if the allocation size of your SYSCHK1 data set has to be increased. For more information about how to increase it, see the appendix "DASD Calculation Chart (see page 203)."

  If you have implemented the global variable backup utilities, invoke OPSSGVRS after starting this release to copy data into your new SYSCHK1 data set. If the utilities are not implemented, copy your existing SYSCHK1 data set to the new SYSCHK1 data set before starting this release. You can use a system utility such as IDCAMS REPRO to copy the data set.

Note: You cannot run two releases of CA OPS/MVS concurrently while allocating the same VSAM linear data sets. Additionally, you may be unable to convert from a new release of CA OPS/MVS to an older release.

Global Variable Backup Methods

Use one of the following methods to back up global variables:

- RDF tables and GLOBALx stem variables (see page 71)
- Back up only the GLOBALx variables (see page 72)
- Back up and restore specific RDF tables (see page 72).

Back Up RDF Tables and GLOBALx Stem Variables

The OPSSGVBK (backup) and OPSSGVRS (restore) sample procedures let you back up both RDF tables and GLOBALx stem variables. You can locate the sample procedures in the CA OPS/MVS CNTL library.

Note: For more information about these sample procedures and parameters, and about the backup and restore procedures, see the *Administration Guide*.

Follow these steps for the OPSSGVBK Backup:

1. Allocate the Generation Data Group (GDG) data sets.
2. Review the GVBKGDG sample JCL in the CA OPS/MVS CNTL library to allocate the required GDG and MDSCB data sets.
3. Set the appropriate GLOBALBACKUP parameters.
4. You can trigger the backup procedure on request, or on a specific interval.

   After the backup procedure executes, the generated GDG data sets contain a complete backup of all GLOBALx stem variables and all RDF tables.

Follow these steps for the OPSSGVRS Restore:

1. Update and start this procedure when you need a restore.
2. Review the JCL sample in the CA OPS/MVS CNTL library.
3. You can customize the sample to allocate the desired GDG data set that the OPSSGVBK backup procedure created, or it can allocate the most recent generation (+0).

   Note: You can start this restore procedure while CA OPS/MVS is active. Using this method, you cannot restore only one global variable or one RDF table.

   All GLOBALx stem variables reset to their backed-up values. All RDF tables are replaced with all tables that were part of the last OPSSGVRS backup.

   After the restore procedure executes, the global variable tree rebuilds. If the SSM engine is being utilized, it restarts to resynch with newly loaded or restored tables.

Back Up GLOBALx Variables Only

The OPSVIEW 7.5 option lets you take a quick backup of all GLOBALx variables or a particular stem. This online option creates a REXX member which contains the necessary OPSVALUE() instructions to back up (reset) the desired variables. You can also use this method if you want to transfer some particular GLOBALx variables quickly from one system to another.

Follow these steps:

1. Back up the desired variables to create a REXX exec.
2. Execute this REXX exec on another system to reset and initialize these desired variables.

Back Up and Restore Specific RDF Tables

Use the supplied WRITETBL and READTBL programs as backup and restore procedures for one or more RDF tables. The WRITETBL (create seq dsn copy of an RDF table) and READTBL (create a table from a dsn that WRITETBL created) OPS/REXX programs are in the opsmvshlq.SAMPLE.REXX library.

Follow these steps:

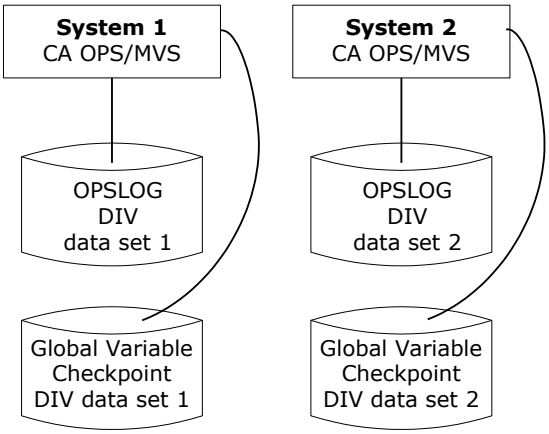
1. Invoke these programs when you want a backup of a table, such as making a copy of the SSM STCTBL before you make new changes.

   You can use this backup for unsuccessful changes, or to back out the changes.
2. Invoke the WRITETBL exec from within an OPS/MVS TOD rule to create a more specific point-in-time backup of a specific table.
3. Refer to comments within both the WRITETBL and READTBL OPS/REXX programs for execution details.

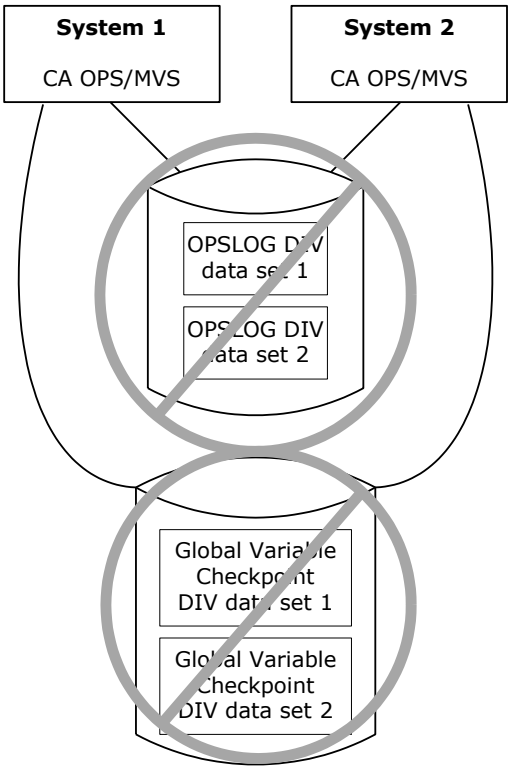
How You Place the DIV Data Sets

The placement of the DIV data sets for the CA OPS/MVS OPSLOG Browse function and REXX global variable checkpoint facility should be as though they are page data sets. These data sets should never be placed on shared DASD, thus avoiding cross-system lockouts, and they should also never be on volumes that have page data sets or other data sets with high levels of I/O, RESERVE activity, or both.

Following is a diagram of *correct* placement of the DIV data sets:



Following is a diagram of *incorrect* placement of the DIV data sets:



Important! Disregarding the above may result in degraded performance for the entire system, the eventual need to re-IPL the system, or both!

Verify Your Installation

Diagnosing problems caused by incomplete installation of CA OPS/MVS is difficult. Use the following checklists before starting the product to avoid such problems.

To verify your installation

1. Make sure the SYS1.PROCLIB (or the procedure library you copied the started task JCL procedures to) contains members OPSMAIN, OPSECF, and OPSOSF.
2. Make sure the OPSMAIN, OPSOSF, and OPSECF started tasks either have a STEPLIB that is authorized and contains all the load modules distributed with CA OPS/MVS or that all these modules are available in a LNKLSTxx load library or an LPALSTxx load library.
3. Make sure the Logical Parmlib Concatenation contains member OPSSPA00. Also check that the parameter values set by the OPSSPA00 member meet your requirements.
4. Make sure the library allocated to SYSPROC contains the OPSTART1 CLIST and OPSLOGON CLIST. OPSTART2 may be located in the SYSPROC or SYSEXEC.
5. If you are using the CA ACF2 command limiting feature, check that the entries listed in the table in Provide a CA ACF2 Command Limiting List in the chapter “Configuration Tasks for the Base Component” are present.

When the above items are verified, you are ready to start the product.

How to Complete the Required Manual Configuration

This section contains steps you must perform manually, whether or not you used CA CSM to configure CA OPS/MVS.

1. Provide APF authorization for the load libraries (see page 75).
2. Place license keys in the CA Common Services PPOPTION data set (see page 75).
3. Set up product licensing (see page 75).
4. Grant data set access (see page 79).
5. Complete the configuration tasks for the base component (see page 79).

Provide APF Authorization for the Load Libraries

There are two CA OPS/MVS load libraries:

.CCLXLOAD

Contains the majority of the product modules. By default, this is a standard PDS and must always be APF authorized.

.CCLXPLD

Contains those load modules that must reside in a PDSE. This load library must also be APF authorized if you are using the CA NSM SSM CA OPS/MVS Option or the Switch Operations Facility (SOF). It is recommended that you give this library permanent APF authorization regardless of the features that are currently installed.

As stated above, the CA OPS/MVS .CCLXLOAD load library must be APF authorized. When you put it in your LNKLST or LPALIB, the CA OPS/MVS load library automatically has this authority if LNKAUTH=LNKLST is specified (or allowed to default) in your appropriate IEASYSxx member *of* the Logical Parmlib Concatenation. If not, assign the load libraries APF authority by putting their names, and the volume serial number of the disk on which they reside, in the appropriate IEAAPFxx *member of* the Logical Parmlib Concatenation. Next, IPL your system to make the change effective.

If you do not want to IPL to authorize CA OPS/MVS, you can use either of these z/OS commands to dynamically allocate APF-authorized libraries:

```
SET PROG=xx
SETPROG APF,ADD...
```

You can also use an existing authorized library or use any one of the major online z/OS performance and operations enhancement tools such as CA SYSVIEW, Tivoli OMEGAMON XE on z/OS, or RESOLVE/MVS to add an entry for a new authorized library.

Place License Keys in the CA Common Services PPOPTION Data Set

During startup, CA OPS/MVS license validation is performed by calling CA LMP service of the CAIRIM component of CCS. For information about installing CAIRIM, activating CA LMP, and coding CA LMP keys, see the CA Common Services for z/OS documentation.

Place CA LMP keys for each of the CA OPS/MVS components (MSF, USS, etc) that you intend to activate in the KEYS member of the PPOPTION data set, found in the CAS9 JCL procedure.

Set Up Product Licensing

This section shows you how you use CA LMP to set up your license key and unlock the features for your product.

CA LMP Key Certificate

Examine the CA License Managed Program (CA LMP) key certificate. Your certificate contains the following information:

Product Name

Defines the trademarked or registered name of your product as licensed for the designated site and CPUs.

Product Code

Defines a two-character code that corresponds to the product.

Supplement

Defines the reference number of your license for a particular facility and has the following format:

nnnnnn-nnn

This format differs slightly inside and outside North America and, in some cases, the reference number is *not* provided at all.

CPU ID

Defines the code that identifies the specific CPU for which installation of this product is valid.

Execution Key

Defines an encrypted code that CA LMP requires for installing your product. During the installation, it is referred to as the LMP code.

Expiration Date

Defines the date when your license expires and has the following format:

ddmmmyy

Example: 21Mar16

Technical Contact

Defines the name of the designated technical contact at your site who is responsible for the installation and maintenance of your product. CA addresses all CA LMP correspondence to this person.

MIS Director

Defines the name of the Director of MIS or the person who performs such a function at your site. If the title but not the name of the individual is indicated on the certificate, supply the actual name when correcting and verifying the certificate.

CPU Location

Defines the address of the building in which the CPU is installed.

How CA LMP Statements Are Coded

Before you start this product, code CA LMP statements for product license authorization.

To code CA LMP statements, take the following steps:

1. Install CAIRIM.
2. Activate LMP.
3. Add your product license codes to the LMP statements.
4. Place the LMP statements in the KEYS member of the CAW0OPTN data set.

Note: The KEYS member of the CAW0OPTN data set is specified in the CAS9 JCL procedure. For more information, see the *CA Common Services for z/OS Administration Guide*.

KEYS Member—Add Execution Key

You must add the CA LMP execution key, provided on your product key certificate, to the CAIRIM parameters to ensure proper initialization.

To define a CA LMP execution key to the CAIRIM parameters, modify the KEYS member.

This sample parameter structure for KEYS member has the following format:

```
PROD(pp) DATE(ddmmmyy) CPU(tttt-mmmm/sssss)
LMPCODE(kkkkkkkkkkkkkkkk)
```

Parameter definitions are as follows:

PROD(*pp*)

Specifies the two-character product code. This code agrees with the product code already in use by the CAIRIM initialization parameters for any earlier releases (if applicable).

Valid values for *pp* are as follows:

A0 - CA OPS/MVS JES 2

CG - CA OPS/MVS JES 3

CI - CA OPS/MVS CICS Operations Facility (COF)

CJ - CA OPS/MVS Switch Operations Facility (SOF)

CN - CA OPS/MVS IMS Operations Facility (IOF)

CU - CA OPS/MVS Multi-system Facility (MSF)

CV - CA OPS/MVS Expert System Interface (ESI)

DATE(*ddmmmyy*)

Specifies the CA LMP licensing agreement expiration date, for example, 13MAR12.

CPU(*tttt-mmmm/ssssss*)

tttt

Specifies the CPU type on which CA LMP is to run, for example, 3090.

-mmmm

Specifies the CPU model on which CA LMP is to run, for example, 600.

Note: If the CPU type and or model require fewer than four characters, blank spaces are inserted for the unused characters.

/ssssss

Specifies the serial number of the CPU on which CA LMP is to run.

LMPCODE(*kkkkkkkkkkkkkkkk*)

Specifies the execution key (*kkkkkkkkkkkkkkkk*) needed to run CA LMP. The key certificate shipped with each CA LMP software solution provides this CA LMP execution key.

Example: Add CA LMP Execution Key

The following example shows a control statement for the CA LMP execution software parameter:

```
PROD(Y7) DATE(27JUN12) CPU(2096-E26 /370623)
LMPCODE(52H2K06130Z7RZD6)
```

In this example, with your product running on the specified CPU, the CA LMP licensing agreement will expire on June 27, 2012. The product code and execution key values are different when you install your product at your site.

Note: For a full description of the procedure for defining the CA LMP execution key to the CAIRIM parameters and further details about the features and associated utilities of CAIRIM, see the *CA Common Services for z/OS Administration Guide*.

Grant Data Set Access

Before you start the CA OPS/MVS startup JCL PROC, make sure that it has the required security access to CA OPS/MVS product libraries created in previous installation steps.

The CA OPS/MVS startup JCL PROC requires UPDATE access to the following data sets:

- OPSLOG files created in Allocate OPSLOG Files in this chapter
- Checkpoint files created in Allocate Checkpoint Files in this chapter

The CA OPS/MVS startup JCL PROC requires READ access to the deployed runtime copies of the following data sets:

- opspfx.CCLXCNTL
- opspfx.CCLXLOAD
- opspfx.CCLXPLD

Note: The CA OPS/MVS startup JCL PROC will fail if it is not granted appropriate mainframe security access to CA OPS/MVS product libraries.

Configuration Tasks for the Base Component

The following sections describe tasks that you perform before starting CA OPS/MVS.

Define z/OS Consoles

To enable CA OPS/MVS to issue z/OS (and subsystem) commands and receive responses, specify some combination of subsystem and extended consoles.

Extended Consoles

CA OPS/MVS controls extended consoles using the following initialization parameters:

- EXTENDEDCONSOLES
- EXTCONSPREFIX

Note: For more information, see the *CA OPS/MVS Parameter Reference*.

Define Subsystem Consoles

If you are running a product that does not support extended consoles, then you may need to use subsystem consoles to issue commands to that product. If you have no subsystem *co*nsoles defined in your CONSOLnn members of the Logical Parmlib Concatenation, then you must add them.

How many subsystem consoles you allocate determines the maximum number of concurrent z/OS commands that CA OPS/MVS can issue on behalf of its users.

To define subsystem consoles

1. Tailor the following sample console definition in the member CONSOL00 of the SYS1.OPS.CCLXCNTL data set:

   ```
   CONSOLE DEVNUM(SUBSYSTEM),AUTH(ALL),NAME(OPSSSC01)
   ```

2. Perform an IPL.

   Your subsystem console is defined.

Important! Automation that is dependent on specific subsystem console IDs may fail in a sysplex environment because the IDs are dynamically assigned by z/OS and they may change from IPL to IPL.

Ensure Availability of a System Linkage Index

CA OPS/MVS requires a system linkage index (LX) in the system function table. If CA OPS/MVS terminates normally or abnormally and you restart it, it reuses this system linkage index. If you plan to run multiple copies of CA OPS/MVS, then you need a system linkage index for each copy.

Typically, the system linkage index should contain enough entries to accommodate CA OPS/MVS. If it does not, then increase the number by modifying the NSYSLX value in the appropriate IEASYS*xx* member of the Logical Parmlib Concatenation.

Note: You can determine whether the system linkage index contains enough entries to accommodate CA OPS/MVS if you are running CA SYSVIEW product. For information about how you can do this, see the LXATABLE command in the CA SYSVIEW command help. Conversely, without CA SYSVIEW, you cannot determine whether such a condition exists before CA OPS/MVS startup because the z/OS operating system does not provide the capability to check for this data.

For more information about the system linkage index, see the IBM documentation.

Replace ASVT Entries

Because CA OPS/MVS owns space switch entry tables, z/OS marks the ASVT entry used by the main product address space as nonreusable after the product terminates. If CA OPS/MVS is stopped and started repeatedly, there is a small chance that you might run out of usable address spaces.

To allow for the replacement of these non-reusable ASVT entries, increase the RSVNONR parameter in the appropriate IEASYS*xx* member in the Logical Parmlib Concatenation by a small number (5 for example).

Add Command Processors in LPA with ISPF

If you run CA OPS/MVS out of LPA and you use ISPF, you must add the names of the CA OPS/MVS command processors to the ISPF TSO command table module.

To add the names of the command processors

1. Review the sample in member OPISPTCM of the SYS1.OPS.CCLXCNTL data set of the modifications that you need to make to ISPTCM.

   For details about adding names to ISPTCM, see your ISPF installation guide.

2. Enter the names in the ISPTCM module, which contains a list of TSO command names.

   The modifications to this table module are complete.

3. Reassemble it and link it into an appropriate load library.

   The command processors are added.

More information:

Provide a CA ACF2 Command Limiting List (see page 87)

Other Command Processor Considerations

The ISPTCM module that contains the CA OPS/MVS command processors can be loaded from a STEPLIB, linklist, or LPALIB. However, if your CA OPS/MVS command processors are loaded from a STEPLIB or a linklist, they must *not* also be in the ISPTCM load module used by the system.

However, different scenarios could exist, specifically for testing a new release of CA OPS/MVS. For instance, you could establish the following scenarios:

- An ISPTCM load module *without* CA OPS/MVS command processors in a STEPLIB, and new CA OPS/MVS command processors in a STEPLIB for testing.
- The ISPTCM module that your system actually uses, in a linklist, and the CA OPS/MVS command processors in an LPALIB.

Note: Because the ISPTCM table can be in a STEPLIB, a linklist, or an LPALIB, a different version of it could exist in each of these places at the same time. If this occurs and you attempt to load a TSO command processor under ISPF, then the ISPTCM version that is found first in a search will be the controlling ISPTCM.

LPA Usage Efficiency

To achieve the most benefit from LPA usage, copy CA OPS/MVS module OPSAEX into an LPALST library. By doing this, you are enabling OPSVIEW and all of the CA OPS/MVS command processors to execute more quickly and to share common code. This can reduce the overall demand for real storage and paging.

If you decide to place the OPSAEX load module in the LPA, you can allow the main address space to share many of the modules that OPSAEX contains. If you choose to do so, add the following DD card to the OPSMAIN JCL:

```
//OPSAEX DD DUMMY
```

Doing this reduces the amount of ECSA used by the main product address space by approximately 400 KB when OPSAEX resides in the LPA. If you omit this ddname, then the main address space makes no attempt to share OPSAEX, even if it resides in the LPA.

Provide Access to the ISPF Interface Modules

CA OPS/MVS request rules can use ISPF services, so make the ISPF interface modules, ISPLINK and ISPEXEC available to the CA OPS/MVS main address space. If they are already in the LPALST concatenation or an APF-authorized library in the LNKLST concatenation, do nothing. Otherwise, copy these two modules or link-edit them into the CA OPS/MVS load library or an APF-authorized library concatenated to it, as a STEPLIB, in the CA OPS/MVS procedure. Member ISPFLINK in the SYS1.OPS.CCLXCNTL data set contains sample JCL to link-edit the ISPF interface modules into the CA OPS/MVS load library.

Establish Data Set Naming Standards

CA OPS/MVS assumes that its data set names start with the characters SYS1.OPS. You can change the names to conform to your data set naming conventions.

Create a User REXX Library

User modifiable programs are contained within the *hlq*.CCLXSAMP library. The user modifiable programs support various sample automated applications as well as specific CA OPS/MVS components that are customized by the end-user (System State Manager ,OPSVIEW command option, and so on) .

Important: CA OPS/MVS programs contained within the *hlq*.CCLXEXEC library are needed for CA OPS/MVS base component functionality and should never be modified without the direction of CA support.

To create a user REXX library

1. Using the same allocation attributes *as* the *hlq*.*CCL*XEXEC or *hlq*.CCLXSAMP libraries, c*rea*te a hlq.USER.REXX data set.
2. Copy the sample application programs and user specific component progra*ms* from hlq.CCLXSAMP library to your *hlq*.USER.REXX library when needed*.*

This *hlq*.USER.REXX library will also contain all REXX and OPS/REXX programs needed to support user created automated applications.

Names for Rule Sets

Data set naming conventions are also important for rule sets. Rule sets are partitioned data sets, which store the OPS/REXX programs (called rules) that the CA OPS/MVS Automated Operations Facility (AOF) uses to automate system operations. The names are in the following form:

```
ruleprefix.rulesetname.rulesuffix.
```

At CA OPS/MVS startup, the AOF looks in the catalog for its rule sets.

Setting a Prefix and Suffix for Rule Sets

The CA OPS/MVS RULEPREFIX parameter specifies the prefix of the data set names for your rule sets and has a default value of SYS1.OPS. The RULESUFFIX parameter specifies the suffix of the data set names for your rule sets and has a default value of RULES.

For example, a rule set name using the default prefix and suffix might be SYS1.OPS.SYS1IEA.RULES.

The RULEPREFIX value can have as many as 10 levels and be as long as 26 characters. Use a multilevel RULEPREFIX, especially if the leading qualifier is SYS1, to speed processing as the AOF scans the catalog looking for its rule sets. The RULESUFFIX value and the rule set name, however, must have only a single level. The rule set name identifies the rule set in OPSVIEW displays.

Important! We strongly recommend that you use a unique high-level qualifier for CA OPS/MVS rule data sets. Failure to heed this warning may result in failures during CA OPS/MVS initialization, degraded performance, or both.

Valid Rule Set Names

If you use the defaults for ruleprefix and rulesuffix shown in the previous section, then the following are valid rule set names:

- SYS1.OPS.SEC.RULES
- SYS1.OPS.TOD.RULES
- SYS1.OPS.JES.RULES
- SYS1.OPS.SUPP.RULES

The rule set name SYS1.RULES.HASP.RULES is invalid because it has a different second-level qualifier.

Alternative Naming Conventions

If the above rule set naming conventions do not meet your needs, use one of the following two alternative naming conventions created for sites that must use different high-level qualifiers for different groups of rule sets. You cannot use both alternative naming conventions.

Note: CA OPS/MVS supports a maximum of 70 rule sets.

Use the RULEALTFIX Parameter

The RULEALTFIX parameter lets you use different high-level qualifiers for different groups of rule sets.

To use this alternative naming method, specify a list of alternate highest-level qualifiers using the RULEALTFIX parameter as follows:

```
var = OPSPRM("SET","RULEPREFIX","SYS1.OPS")
var = OPSPRM("SET","RULEALTFIX","SYS2,SYS3,SYSX")
var = OPSPRM("SET","RULESUFFIX","RULES")
```

The following rule sets will be used:

```
SYS1.OPS.*.RULES
SYS2.OPS.*.RULES
SYS3.OPS.*.RULES
SYSX.OPS.*.RULES
```

Take these facts into consideration:

- While the highest-level qualifiers may be different, all subsequent qualifiers must be the same.
- All of the highest-level qualifiers must be the same length.
- If you use this support, the rule set names must all begin with a high-level qualifier so that you do not accidentally define two rule sets with the same name.

  The following are examples of good rule set names:

  ```
  SYS3.OPS.SYS3MSG.RULES
  SYS1.OPS.SYS1MSG.RULES
  ```

  The following is an example of a bad rule set name:

  ```
  SYS1.OPS.MESSAGE.RULES
  ```

- The quotation marks in the example are required.

Use the RULEPREFIX2 Parameter

To use this alternative naming method, specify a single alternate prefix with the OPSPRM function of OPS/REXX as follows:

```
var = OPSPRM("SET","RULEPREFIX","SYS1.OPS")
var = OPSPRM("SET","RULEPREFIX2","SYS2.OPS2")
var = OPSPRM("SET","RULESUFFIX","RULES")
```

The following rule sets will be used:

```
SYS1.OPS.*.RULES
SYS2.OPS2.*.RULES
```

Consider the following:

- Parameter RULEPREFIX2 is ignored if parameter RULEALTFIX is specified.
- RULEPREFIX2 requires more overhead than RULEPREFIX alone, or RULEPREFIX used with RULEALTFIX.
- The high-level qualifier specified by RULEPREFIX2 is totally independent of the high-level qualifier specified by RULEPREFIX and can be up to 10 levels and a maximum o 26 characters in length.
- A duplicate rule set found using RULEPREFIX2 is ignored if a rule set with the same name is found using RULEPREFIX.

  For example, if the following data sets existed:

  ```
  SYS1.OPS.MESSAGE.RULES
  SYS2.OPS2.MESSAGE.RULES
  ```

  The SYS2.OPS2.MESSAGE.RULES data set would be ignored.
- The quotation marks in the example are required.

Create Security Product Logon IDs

CA OPS/MVS uses a number of address spaces. If you are running CA ACF2, CA Top Secret, or another security product, you may have to do the following:

- Define user IDs for the OPSMAIN, OPSECF, OPSOSF, and OPSUSS address spaces.
- Set up access rules so that these address spaces can use the data sets they need.
- Ensure that the user ID for the OPSUSS server has sufficient USS segment authority to perform the kinds of USS commands that will be requested.

Data Set Access Requirements

The following table summarizes the access requirements for CA OPS/MVS. When developing applications that update your own databases, then these databases also need access. After you have started to use the product and written your own applications, you will need to provide access to your own REXX, CLIST, OPSEXEC, and possibly user ISPF data sets.

Data Set Name	Access	User IDs
OPS.*xxx*.RULES	Read, write	OPSMAIN and authorized TSO users
OPS.CCLXLOAD	Execute	OPSMAIN, OPSOSF, OPSECF, and all TSO users
OPS.CCLXEXEC	Read	OPSMAIN, OPSOSF, OPSECF, and all TSO users
OPS.CCLXCLS0	Read	OPSMAIN, OPSOSF, OPSECF, and all TSO users

Data Set Name	Access	User IDs
OPS.OPSLOG	Read, write	OPSMAIN
OPS.SYSCHK1	Read, write	OPSMAIN
Logical Parmlib Concatenation	Read	OPSMAIN
OPS.CCLXHENU	Read	All authorized TSO users
OPS.CCLXPENU	Read	All authorized TSO users
OPS.CCLXMENU	Read	All authorized TSO users
OPS.CCLXSENU	Read	All authorized TSO users
OPS.CCLXTENU	Read	All authorized TSO users
OPS.CCLXOPEX	Read	All authorized TSO users and possibly OPSOSF

Note: If you are using SSMGA, you must also allocate the OPS.CCLXOPEX data set to the OPSMAIN procedure.

Important! Running CA OPS/MVS without giving its various address spaces enough authorization to access their data sets is the most common installation problem.

Provide TSO OPER Authority

TSO OPER authority needs to be provided through your security package to all user IDs, including the OPSMAIN and OPSOSF user IDs, that issue ADDRESS OPER commands, enter commands from the OPSVIEW 6, or opslog OPSVIEW 1 panels.

To provide TSO OPER authority, run the OPS/REXX program OPSIVP.

Similarly, this must be done for user IDs requiring the use of TSO submit, status, and cancel commands.

Provide a CA ACF2 Command Limiting List

If your site uses CA ACF2 and you use a command-limiting list, you need to add some entries to allow OPSVIEW and other product components to function.

If you are not running CA ACF2 or do not use a command-limiting list, then skip this section.

CA ACF2 Considerations

CA OPS/MVS uses SAF (RACROUTE) for most security interface calls. Specifically, this affects CA ACF2 sites that use the OPSECURE('R',....) function to perform generic resource checking. You may need to translate the SAF resource classes by creating one or more CA ACF2 CLASMAP records. CLASMAP records translate eight-character SAF resources into three-byte CA ACF2 resource-type codes.

For more information, see the *CA ACF2 Administrator Guide*.

CA OPS/MVS Command Processors

Member OPA2CMLS of OPS.CCLXASM provides a sample command-limiting list. It includes the commands shown in the following list, which should be integrated into your list. For more information, see the *CA ACF2 Systems Programmer Guide. T*he following table summarizes the access requirements for CA OPS/MVS. If you develop applications that update your own databases, they also need access.

OPAAMAIN

CA OPS/MVS Automation Analyzer.

Alias: None

OPADDRUL

Dynamic Automate-format rule (ADDRULE).

Alias: None

OPBIND

EPI session enqueue (BIND).

Alias: None

OPBOMD

Internal interface used by OPSBRW/OB to invoke OPSLOG Browse. This command should never be directly used by an end-user.

Alias: None

OPDELRUL

Delete a dynamic Automate-format rule (DELRULE).

Alias: None

OPGETSCR

EPI screen image fetch (GETSCRN).

Alias: None

OPPARSE

CLIST/REXX parse command (PARSE).

Alias: None

OPRXCMAP

Compiled REXX manager.

Alias: None

OPSBRW

Browse the CA OPS/MVS message log.

Alias: OB

OPSCMD

Issue z/OS/IMS/VM/JES operator commands (OSCMD, CPCMD).

Aliases: OC, OPSOSCMD

OPSDELV

Delete global variables (DELVAR).

Alias: None

OPSDOM

Delete retained console messages (DOM).

Alias: None

OPSESS

EPI screen entry (SESSCMD).

Alias: None

OPSEXEC

Explicit OPS/REXX interpreter.

Aliases: OX, OXDB, OXSCAN

OPSGETV

Obtain global variable value (GETVAR).

Alias: None

OPSGETVL

Obtain global variable name list (GETVARL).

Alias: None

OPSHFI

Shared file I/O command (READVAR, WRITEVAR).

Alias: None

OPSIMEX

Implicit OPS/REXX interpreter.

Aliases: OI, OIDB

OPSMODE

Provide alias entry points for former Automate users that still use the Automate command processors. Required to use the External Product Interface (EPI) Record and Playback feature.

Aliases: ADDRULE, BIND, CLIST, CPCMD, DELRULE, DELVAR, DOM, GETSCRN, GETVAR, GETVARL, MLWTO, OSCMD, PARSE, READVAR, REPLY, REXX, SESSCMD, SETVAR, SQL, STATETBL, TSOCMD, UNBIND, WAIT, WRITEVAR, WTL, WTO, WTOH, WTOR

OPSPARM

Display/modify CA OPS/MVS parameters.

Alias: OP

OPSQL

Issue SQL commands from TSO (SQL).

Alias: None

OPSREPLY

Issue reply to WTOR (REPLY).

Alias: None

OPSREQ

Issue an end-user operation request.

Alias: None

OPSRMT

Issue remote TSO commands (REXX TSOCMD, CLIST).

Alias: OR

OPSSETV

Update global variable value (SETVAR).

Alias: None

OPSSMTBL

Maintain System State Manager resource directory table (STATETBL).

Alias: None

OPSWAIT

Wait for a specified time (WAIT).

Alias: OW

OPSWTO

Issue WTO or WTOR messages (WTL, WTO, and so on).

Alias: None

OPUNBIND

EPI session dequeue (UNBIND).

Alias: None

OP310000

OPSVIEW Address Space Resource Facility.

Alias: None

O332TBLD

OPSVIEW Printer Resource Facility

Alias: None

OPSVIEW

Invoke CA OPS/MVS Interactive Services.

Alias: OPSV

Add the Subsystem ID to the Logical Parmlib Concatenation

CA OPS/MVS runs as a z/OS subsystem. Such subsystems are defined at IPL time through statements in the appropriate IEFSSN*xx* member of the Logical Parmlib Concatenation. However, in reality, most subsystems are added dynamically.

If CA OPS/MVS is not defined in your IEFSSN*xx* member, then CA OPS/MVS uses the standard z/OS interface to add its SSCT dynamically. So, whether you update your IEFSSN*xx* member depends on the policy of your site.

Member IEFSSNOP of the SYS1.OPS.CCLXCNTL data set contains the following example statements that you can insert in your production IEFSSNxx me*mb*er of the Logical Parmlib Concatenation to define the standard production and test subsystem names for CA OPS/MVS:

```
OPSB        OPSLOG BROWSE ONLY OPS/MVS
OPST        TEST OPS/MVS
OPSS        PRODUCTION OPS/MVS
```

Your production CA OPS/MVS should be listed last, as in the example above. However, if you have another subsystem that must be last, your production CA OPS/MVS should be listed just before that last system.

The parameters SSICMD and SSIMSG affect when CA OPS/MVS processes messages and commands relative to other subsystems.

Note: Some subsections below will have been partially or fully completed already if you began the configuration process using CA CSM (see earlier subsection How to Begin Configuration With CA CSM (see page 59)). Notes within such subsections will identify steps already completed by CA CSM.

Optional Configuration Tasks for the Base Components

The following sections describe tasks that you can optionally perform when configuring the base component of CA OPS/MVS.

Place Load Modules in the Link Pack Area

Running CA OPS/MVS out of the link pack area (LPA) can significantly reduce your ECSA requirement. To attain this reduction, add the name of the load library to an LPALST*xx* member of the Logical Parmlib Concatenation.

Important! You should *not* copy the following CA OPS/MVS load library modules into the LPALST concatenation libraries because they may cause errors during a z/OS IPL.

ASOEDIT, ASOEDPAR, and ASOEDSYS

These are used by the Automate rules editor. These modules are not reentrant.

OPARSX35

This is the SORT exit used when archived OPSLOGs are merged. This module is not reentrant.

CAIXNYI@

This is a data-only module that gets dynamically updated to provide information for CA Examine.

If you run CA OPS/MVS out of your LPALIB, note that most CA OPS/MVS modules are not used from the LPA (that is, if you follow the recommendation in the following paragraph). In fact, less than 10 KB of CA OPS/MVS are actually loaded into LPA. Most load modules are loaded into ELPA.

We strongly recommend that you copy the following modules into a linklist or STEPLIB instead of the LPA because they are RMODE 24:

- OPAME010 module, used by the AME reporter
- OPSQTETB module, used by the RDF table editor

Provide TSO Command Authorization

You do not need to authorize any TSO command processors, because CA OPS/MVS provides authorization service while it is running. The only exception to this rule is when any CA OPS/MVS command executes in the address space of a TSO user while CA OPS/MVS is down.

The following table lists the TSO command processors that can be authorized. Both their primary names and their aliases must go into IKJTSOxx.

Command	Alias	Description
OPSCMD	OC	Issue z/OS, JES2 or JES3, VM, and IMS operator commands
OPSDOM	None	Delete a highlighted message
OPSREPLY	None	Reply to WTOR
OPSWTO	None	Issue WTO or WTOR messages

The method used to authorize these TSO commands varies with the release of TSO/E you have installed.

You can set TSO command authorization from the appropriate IKJTSO*xx* member of the Logical Parmlib Concatenation. For an example, see member IKJTSO00 of SYS1.OPS.CCLXCNTL. You need to restart the system to make these changes effective unless you have CA SYSVIEW. If you have CA SYSVIEW, then you can use it to dynamically add these names to the TSO command tables without restarting the system.

Authorized programs or commands that are directly invoked from a REXX program can access variables created by REXX only if the variable names begin with SYSAUTH. This TSO/E restriction is only applicable if you use the TEXTVAR parameter of the OPSWTO command processor. For more information, see the IBM documentation.

Provide Access to the Load Modules

All address spaces that access any CA OPS/MVS facilities must have access to all CA OPS/MVS load modules. You can place the library where the load modules reside in your LNKLST or LPALIB, or you can add STEPLIBs to the started task JCL for OPSMAIN, OPSECF, OPSOSF, and any TSO users that will use CA OPS/MVS facilities such as OPSVIEW.

Important! Do not include SYS1.OPS.CCLXLOAD in the ISPLLIB concatenation for ISPF users because CA OPS/MVS command processors that run authorized cannot be loaded from ISPLLIB.

More information:

Place Load Modules in the Link Pack Area (see page 92)

Install OPSMODE Command Processor

If you are a former Automate user that still uses the Automate command processer, you must install the OPSMODE command processor.

If you are going to use the External Product Interface (EPI) Record and Playback feature, you must install the OPSMODE command processor.

To install OPSMODE, you can use the provided SMP/E USERMOD in member USERMODS of the OPS.CCLXCNTL data set. This member is a sample for installing OPSMODE under SMP/E. CA OPS/MVS Technical Support recommends that the USERMOD be received and applied but not accepted.

Install UNIX System Services Interface to Event Management Component of CCS for z/OS

The UNIX System Services (USS) component of CA OPS/MVS provides a class of OSF servers that execute UNIX shell commands and direct API calls to the Event Management component of CCS for z/OS.

For CA OPS/MVS USS to interface with z/OS Event Management, the Event Management component of CCS must be installed on the z/OS system, in addition to several other z/OS CCS components.

More information:

Install the UNIX System Services (see page 149)

Note: For a complete list of the CCS for z/OS components, by FMID, see the appendix "CCS for z/OS Component Requirements."

USS Interface to Event Management

The CA OPS/MVS USS interface to z/OS Event Management does the following:

- Lets Event Management console messages be available in OPSLOG
- Lets USS rules take action on the Event Management console messages
- Lets CA OPS/MVS send commands and messages to z/OS Event Management or any other CCS Event Management connected platform

Tailor and Run INSTUSEX

The OPS/MVS USS interface to z/OS Event Management requires that a message exit be copied into the z/OS Event Management HFS or zFS directory in compatibility mode. The INSTUSEX job is provided in the OPS.CCLXCNTL file for this task.

To tailor and run the INSTUSEX job

1. Change the job statement to meet installation standards.
2. Set the DISKPFX parameter to the data set name prefix of your deployed runtime libraries.
3. The data set is allocated after the job runs.
4. Set the USRPATH parameter. You need to know the directory where the CCS for z/OS module TNEMEVT2 is stored. The default directory is /cai/tngfw/lib. You can verify this by issuing the OMVS or ISHELL command from TSO. If the Event Management component of CA NSM has not been installed, then run this job when it is available.
5. The INSTUSEX job is tailored and ready to run.
6. Run the INSTUSEX job.
7. The message exit is copied into the z/OS Event Management HFS or zFS in compatibility mode directory. Verify the return codes are 0.

   Stop and restart the caiopr process of CCS for z/OS.

   The copied version of the message exit becomes active.

Summary of System Preparation Tasks

The following list summarizes the system preparation tasks for CA OPS/MVS. Review it to see the impact of each item on CA OPS/MVS operation.

- CA LMP Key Certificate

  Contains the information that you need to initialize CA OPS/MVS.

- Compatible software levels

  Incompatible levels could be a problem. Contact CA Support.

- z/OS subsystem consoles

  Limits concurrent z/OS commands. If you have no subsystem extended consoles, OPSCMD, OPSRMT, and OPSVIEW will not function.

  Default: 2

- Extended consoles

  Number of extended consoles without MIGIDs.

  Default: 8

  R**ecommended:** 8

- Enough ECSA available

  Could be a problem if you do not have enough available ECSA. Circumvent or reduce by putting SYS1.OPS.CCLXLOAD into LPA/ELPA.

  R**ecommended:** 500 KB

- DASD space for program libraries, OPSLOG Browse messages, and global variables

  For information to help you calculate how much DASD space you will need to install and run CA OPS/MVS*, see the Administration Guide.*

- Data set naming standards

  You do not need to catalog data sets in your z/OS master catalog.

- Access to CA OPS/MVS load modules

  Has to be STEPLIB, LINKLST, or LPALIB. Installation usually goes faster using the STEPLIB method.

- APF authorize load library

  Can circumvent need for IPL by copying to LINKLST library, or by dynamically authorizing SYS1.OPS.CCLXLOAD.

- TSO command authorization

  Required only to execute CA OPS/MVS TSO commands (for example, OPSCMD) when the product is down.

- Security user IDs for OPSMAIN, OPSECF, OPSOSF, and OPSUSS, and data set access if you have a security system, you need them.

- Provide TSO OPER authority to user IDs.

- This authority must be provided to all user IDs that issue z/OS commands from ADDRESS OPER, OVEVIEW 6, or OPSVIEW 1 panels, run the OPSIVP OPS/REXX program, and so on.

- CA ACF2 command limiting list

- If you run CA ACF2 and you use a command limiting list, OPSVIEW will not work unless its subsidiary commands are included.

- Subsystem ID

  The subsystem ID (that is, OPSS) is inserted dynamically if not in the parmlib library.

- VTAM definitions for the MSF optional component

  Required for the MSF, although the rest of CA OPS/MVS does not need them. The definitions can be added without recycling VTAM if you use separate members.

- VTAM definitions for the EPI optional component

  Same as above.

- IMS AOI exit use of the UEHURSVD field

  Could be a problem. Contact CA Support.

- CCS for z/OS installed

  Required.

Post-Installation Considerations

Now that you have successfully installed and started your CA OPS/MVS started tasks, you should consider the following points:

- It is important that you tune CA OPS/MVS to ensure it is optimized to handle your unique workloads and processing requirements.

Note: For more information, see the *CA OPS/MVS Event Management and Automation Administrator Guide* and *CA OPS/MVS Event Management and Automation Parameter Reference Guide*.

- After the web and server applications are installed and configured, you can access the OPSLOG WebView GUI from your web browser by initiating an OPSLOG WebView session with a URL of this form (see Resource 5):

http://hostname.domain:port/applname

hostname and domain

Hostname and domain are IP addresses. If you know the numeric IP address, then you can use it instead.

port

The IP port number that you defined for HTTP (browser) access. If you define the default port of 80, then it can be omitted from the URL.

applname

The applname is defined in the PASS statement, as described in Configure the Web Application in this chapter.

(new related group 1)

Configure the Web Application (see page 138)

Customize Parameter Library Members

If you used CA CSM to configure CA OPS/MVS, the REXX file OPSSXP00 (found in your CCLXCNTL data set) can be used to manually specify additional parameters or override existing ones. Do not modify file REXX OPSSSC00 (also found in your CCLXCNTL data set as this file is maintained automatically by CA CSM.

OPSSSC00

Contains parameters as specified in CA CSM. DO NOT MODIFY BY HAND!

OPSSXP00

Contains skeleton code to allow specification of additional parameters when CA CSM was used to configure the product

If you did not use CA CSM to configure CA OPS/MVS, the member OPSSPA00 in your CNTL data set will contain all the customization parameters.

OPSSPA00

Contains all parameters for configurations of CA OPS/MVS that did NOT utilize CA CSM for their configuration.

For more information about customizing the CA OPS/MVS parmlib members, see the *CA OPS/MVS Parameter Guide.*

Make OPSVIEW Facilities Available Under TSO

You must make OPSVIEW available to at least the people who are responsible for maintaining and administering CA OPS/MVS. Also, you will probably want to make it available to everyone who currently has access to a console.

To make the OPSVIEW data sets available to TSO users

1. Concatenate the libraries with (or copy into) the standard distribution libraries for ISPF/PDF, as you do with all ISPF-based applications.
2. Provide dynamic access to OPSVIEW data sets by either of the following methods:
   - Allocate the OPSVIEW ISPF-related data sets when the OPSVIEW user logs on to TSO
   - Dynamically when the user invokes OPSVIEW

   The OPSVLBDF member in the SYS1.OPS.CCLXSAMP data set contains a customizable example to dynamically allocate the OPSVIEW ISPF-related data sets at the time OPSVIEW is invoked. This REXX EXEC may be invoked either from the TSO/E READY prompt or from within ISPF as either an OPS/REXX or TSO/E REXX EXEC. This REXX EXEC uses the ISPF LIBDEF service to allocate the following ISPF-related DDs:

DDname	Data Set Name	Description
SYSHELPENU	SYS1.OPS.CCLXHENU	TSO help members
ISPMLIB	SYS1.OPS.CCLXMENU	Message library
ISPPLIB	SYS1.OPS.CCLXPENU	Panel library
ISPTLIB	SYS1.OPS.CCLXTENU	ISPF command tables
ISPTABL	Either a unique table output data set or your ISPF profile data set name	ISPF table output data set
ISPSLIB	SYS1.OPS.CCLXSENU	ISPF file tailoring skeletons
STEPLIB	SYS1.OPS.CCLXLOAD (see note.)	Program load library
SYSEXEC	SYS1.OPS.CCLXEXEC HLQ.USER.REXX	Base REXX programs User modifiable OPS/MVS REXX programs
SYSPROC	SYS1.OPS.CCLXCLS0	CLIST library
OPSEXEC	SYS1.OPS.CCLXOPEX	Compiled OPS/REXX programs (not required; however significantly improves OPSVIEW performance)

Important! For OPSVIEW to be fully functional, the OPS/MVS load library must be in a STEPLIB or in the LINKLIST.

3. If you use the TSO/E calling sequence for OPSVLBDF, you can also perform the following functions in the EXEC:
   - Allocate and free the CA OPS/MVS compiled REXX data set to the OPSEXEC ddname.
   - If you use the CA OPS/MVS REXXDDNAME parameter to provide a ddname for OPS/REXX other than SYSEXEC, you can also allocate and free the OPS/REXX source program data sets.
4. Copy member OP6UEXIT from *hlq*.CCLXSAMP to your *hlq*.*USE*R.REXX.
5. (Optional) Add OPSVIEW as a selection on one of your existing ISPF menus. Add the following line to the &ZSEL translation section of your ISPF panel to invoke OPSVIEW:

```
&ZSEL = TRANS( TRUNC ( &ZCMD,'.')
          S,'CMD(OPSV)'
```

Your OPSVIEW facilities are made available under TSO.

OPSVIEW Data Sets Usage Notes

The following list provides usage information for the OPSVIEW data sets:

- If you concatenate libraries, then the RECFM of all the libraries for a given ddname must match.
- The SYSEXEC ddname is searched to invoke REXX programs implicitly. You may want to concatenate installation or user REXX libraries to this ddname. If you have set the REXXDDNAME product parameter to a value other than SYSEXEC, use that value instead of SYSEXEC.
- The STEPLIB ddname is not required if you include SYS1.OPS.CCLXLOAD in the system linklist.
- If you move the CA OPS/MVS CCLXLOAD library into LPA and remove it from STEPLIB or LINKLIST, update the ISPTCM table to include the CA OPS/MVS TSO command processor names with the variable pool flag turned on.
- The OPSVIEW CLIST OPPRIMOP contains the name of the default allocation device SYSDA. You may need to change this device.
- The table output data set, allocated to the ISPTABL ddname, must also be included in the ISPTLIB concatenation.

Start the Product

After you check the above items, start CA OPS/MVS using the following command:

```
START OPSMAIN,SUB=MSTR
```

Things to Check after Starting the Product

After CA OPS/MVS becomes active, do the following checks:

- Make sure that CA OPS/MVS started at least one OPSOSF address space by issuing a DISPLAY ACTIVE command:

  ```
  D A,OPSOSF
  ```

- Test the OSF by entering the following command from a console. This command assumes that you are using the default value for the OSFCHAR parameter, which is an exclamation point (!):

  ```
  !OI OPSIVP
  ```

  - The user ID running the OPSIVP requires TSO OPER authority.
  - The OPSIVP OPS/REXX program is located in the REXX library. If CA OPS/MVS returns a Program Not Found message, the REXX library is not concatenated under the SYSEXEC ddname of the OPSOSF STC.
  - The OPSIVP program tests the CA OPS/MVS WTO and WTOR capabilities, so be sure to monitor the console for any messages or prompts. A reply to the WTOR is not required. The OPSIVP program also tests the ability of CA OPS/MVS to issue console commands using the OPSCMD TSO command. Any errors in the OPSCMD command will trigger debugging messages that indicate the OPSOSF security authority.
  - If the !OI OPSIVP command returns no output to your console, the Operator Server Facility (OSF) is not working. This situation usually results when OPSOSF address spaces cannot be started because of insufficient security authorization; that is, they do not have the authority to access their own data sets. Remember to scan console messages to check for additional problems.

- Verify your security authority by executing the OPSIVP program from your TSO user ID. To do so, issue the following command from the TSO READY screen:

  ```
  OI OPSIVP
  ```

- Test some of the CA OPS/MVS OPSVIEW capabilities using one of the following options:
  - Enter this command from your TSO session at the TSO READY prompt:

    ```
    0PSV
    ```

  - Enter this command from your TSO session under ISPF:

    ```
    TS0  0PSV
    ```

    Important! The above command with the TSO prefix can only be issued under ISPF. Entering the command with the TSO prefix from the TSO READY prompt will result in the error message COMMAND TSO NOT FOUND.
- From the OPSVIEW Primary Options Menu, select option 1 to view the OPSLOG. You should see a display log similar to your SDSF LOG or SYSLOG. If your display is blank, the CA OPS/MVS main STC may be inactive or error messages may be displayed on the console. You should also access OPSVIEW option 4.1.1, which displays all of the CA OPS/MVS parameters.
- Access OPSVIEW through the procedures implemented during Make OPSVIEW Facilities Available Under TSO (see page 100) in this chapter.

Disable Rules in the Sample Rule Set

We ship sample rules with the auto-enable flag set to OFF so that the AOF does not enable them automatically at CA OPS/MVS startup.

Important: Because they are examples, the rules in the sample rule set will not necessarily work on every system. Many of the rules must be customized before they are enabled. Before you start CA OPS/MVS, make sure that you have not enabled any sample rules that may conflict with your system setup.

Enable the Sample OPSAOF Command Rule (Optional)

Member OPSAOF of SYS1.OPS.SAMPLE.RULES contains a sample command rule that you may find useful.

The OPSAOF command gives an operator at a console control of the AOF when TSO is not up. OPSVIEW option 4.5 usually controls AOF operation. For example, to list all active rule sets, enter the following command at an MCS console:

OPSAOF LIST

If you use the CA default rule set naming conventions, one of your rule sets will be: SYS1.OPS.SAMPLE.RULES. Otherwise, copy OPSAOF to one of your own production rule sets. The OPSAOF rule is shipped with the auto-enable flag off so that the AOF does not enable it at CA OPS/MVS startup. Therefore, use the OPSVIEW option 4.5 to turn this flag on to complete the installation of this rule.

Note: For more information on enabling rules, see the *User Guide*.

Chapter 6: Configuring and Installing Optional Components

This section contains the following topics:

How You Install Separately-Licensed Components

These articles describe the installation procedures for separately-licensed CA OPS/MVS components that are not base components of CA OPS/MVS. Each article in this section is required only if you are licensed to use the particular component.

- Configure the Multi-System Facility (MSF) (see page 107)
- Install the IMS Operations Facility (see page 114)
- Install the XTDOUT CICS Operations Facility (COF) Interface for CICS/TS (see page 120)
- Customize the CA NSM SSM CA OPS/MVS Option (see page 121)
- Install the Expert Systems Interface (ESI) (see page 128)

How You Install Optional Base Components

These steps discuss the installation of optional base components of CA OPS/MVS, which you may choose not to use. Each step in this section is required only if you want to use the optional feature of CA OPS/MVS affected by the particular step.

- Install and Configure OPSLOG WebView (see page 129)
- Install JES2 environmental functions (JES2 only) (see page 146)
- Enable library sharing among CPUs with JES2OFFSETSUFFIX (JES2 only) (see page 147)
- Set up the JES3 interface (see page 147)

Tasks for Separately Licensed Components

This section discusses the tasks that you need to complete to install separately licensed components of CA OPS/MVS.

Configure the Multi-System Facility (MSF)

The CA OPS/MVS optional Multi-System Facility (MSF) feature provides communication between multiple CA OPS/MVS copies running on different z/OS machines. It also provides communication between copies of CA OPS/MVS and CA Automation Point.

If you have licensed the MSF, you must set the INITMSF parameter to YES in the OPS/REXX startup EXEC.

If you used CA CSM to configure MSF, some of these parameters will have already be generated for you in the parameter file OPSSSC00 in the CNTL data set. If you need to customize MSF parameters further, put them in file OPSSXP00 in the CNTL data set.

CA CSM configuration does not generate the system defines (MSF DEFINE) automatically.

If you did *not* use CA CSM to configure CA OPS/MVS, the parameters are kept in file OPSSPA00 in the CNTL data set.

Sample MSF Parameters

INITMSF

Initializes the MSF interface.

INITCCI

Determines whether the CAICCI interface is to be activated when a remote MSF is defined as a CCI type.

MSFLOGMODE

Specifies the default VTAM LOGMODE name for all MSF APPC sessions.

MSFRESTARTREXX

Specifies the name of an OPS/REXX program that you have written to set up your MSF environment after the MSF has been restarted.

MSFSYSWAIT

Specifies a default wait time for CA OPS/MVS components that use the MSF.

SYSID

Defines the name of the local system in a MSF or MSF network.

Set up Session Protocols

There are two types of session protocols to use for communication between copies of CA OPS/MVS:

- The logical unit (LU) 6.2 set of session protocols (APPC)

  CA OPS/MVS native MSF uses only the LU 6.2 set of session protocols.

- The communications services that CAICCI provides. CAICCI provides session protocols for LU2, XES, XCF, and TCP/IP.

The following sections discuss setting up MSF to use LU 6.2 and to use CAICCI.

Important! We do not recommend that you use the LU2 protocol of CAICCI.

Setting up MSF to Use LU 6.2 (APPC) Session Protocols

To install the MSF, you complete the following steps:

1. Define CA OPS/MVS to VTAM on each z/OS system by adding an application definition (APPL) statement to SYS1.VTAMLST.
2. Define the VTAM cross-domain resource environment on each so these applications can conduct communication sessions with each other.

VTAM APPL Statement—Define CA OPS/MVS to VTAM

On each system, CA OPS/MVS needs only one APPL statement that uses the following format:

```
netname APPL  APPC=YES,
               AUTH=ACQ,
               AUTOSES=1,
               DSESLIM=3,
               DMINWNL=1,
               DMINWNR=1,
               MAXPVT=512K,
               MODETAB=modetab,
               PARSESS=YES,
               PRTCT=vtampswd
```

Note: For information about valid values for the parameters in the above APPL statement, see the VTAM documentation.

The following example APPL definitions are provided in member OPSAPPL of the SYS1.OPS.CCLXCNTL data set. They assume that you have a network with z/OS systems (A, B, and C) that can all support cross-domain sessions to each other.

Example 1: This APPL statement appears only in the SYS1.VTAMLST of system A.

```
OPSMAINA APPL  APPC=YES,
               AUTH=ACQ,
               AUTOSES=1
               DSESLIM=3
               DMINWNL=1,
               DMINWNR=1,
               MODETAB=MTLU62,
               PRTCT=OPSMVS,
               MAXPVT=512K,
               PARSESS=YES
```

Example 2: This APPL statement appears only in the SYS1.VTAMLST of system B

```
OPSMAINB APPL  APPC=YES,
               AUTH=ACQ,
               AUTOSES=1
               DSESLIM=3
               DMINWNL=1,
               DMINWNR=1,
               MODETAB=MTLU62,
               PRTCT=OPSMVS,
               MAXPVT=512K,
               PARSESS=YES
```

Example 3: This APPL statement appears only in the SYS1.VTAMLST of system C

```
OPSMAINC APPL  APPC=YES,
               AUTH=ACQ,
               AUTOSES=1
               DSESLIM=3
               DMINWNL=1,
               DMINWNR=1,
               MODETAB=MTLU62,
               PRTCT=OPSMVS,
               MAXPVT=512K,
               PARSESS=YES
```

Define the VTAM Cross-domain Environment

To use the cross-system services, you define cross-domain resources for the systems to which you are connecting.

To define a cross-domain resource

1. Find the cross-domain member name for the system to which you want to connect.
2. Create a cross-domain resource member or modify an existing one for the system to which you want to connect. In this member, use a cross-domain resource macro to specify the system where CA OPS/MVS resides.

   For example, suppose the copy OPS1 resides on SYSTEMA and another copy that is named OPS2 resides on SYSTEMB, then SYSTEMA should have the following cross-domain resource:

   ```
   OPS2  CDRSC  CDRM=SYSTEMB,ISTATUS=ACTIVE
   ```
3. Repeat the previous step for each system.

   The VTAM cross-domain environment is defined.

Define the LU 6.2 VTAM Mode Table Entry

If you are using APPC sessions for the MSF, you must create an LU 6.2 mode table entry.

To define an LU6.2 mode table entry

1. Find an existing VTAM mode table that already contains a mode table entry with LU 6.2 session parameters.
2. If you find no such entry, select an existing mode table to contain a new LU 6.2 mode table entry.
3. Add a new mode table entry to the mode table that is associated with the LU definition.
4. Assemble and link-edit the mode table and add the load module to SYS1.VTAMLIB.

   The LU 6.2 VTAM mode table entry is defined.

Example 1: The following sample mode table entry contains a set of session parameters used for an LU 6.2 session:

```
LU62MODE MODEENT LOGMODE=LU62MODE, Mode Table Entry Name
                 FMPROF=X'13'      Function Manager Profile
                 TSPROF=X'07'      Transmission Services Profile
                 PRIPROT=X'B0'     Primary Logical Unit Profile
                 SECPROT=X'B0'     Secondary Logical Unit Profile
                 COMPROT=X'50B1'   Common Logical Unit Profile
                 RUSIZES=X'8989'   Sec/Pri RU sizes 4096/4096
                 PSNDPAC=5,        Primary Send Pacing Count
                 SRCVPAC=5,        Secondary Receive Pacing Count
                 SSNDPAC=5,        Secondary Send Pacing Count
*                                  Presentation Services Profile
                 PSERVIC=X'060200000000000000000000'
```

Example 2: If you are using dynamic cross-domain resources, you may need to define and use the following table entry in place of the one shown in Example 1:

```
LU62MODE MODEENT  LOGMODE=LU62MODE, Mode Table Entry Name
                  FMPROF=X'13',     Function Manager Profile
                  TSPROF=X'07',     Transmission Services Profile
                  PRIPROT=X'B0',    Primary Logical Unit Profile
                  COMPROT=X'78A5',  Common Logical Unit Profile
                  RUSIZES=X'8989',  Sec/Pri RU sizes 4096/4096
*                                   Presentation Services Profile
                  PSERVIC=X'060200000000000000122F00'
```

Setting up MSF to Use CAICCI

CAI Common Communications Interface (CAICCI) is a communications facility that CA OPS/MVS uses to let CA solutions communicate with one another. It provides a layer that isolates application software from the specifics of the communication environment. CAICCI is one member of a group of routines that comprise CCS for z/OS.

To use the CAICCI cross-platform communications services for communication between copies of CA OPS/MVS, follow the procedures that are described in this section.

ADDRESS OPSCTL MSF DEFINE - Set Up MSF Connections Using CAICCI

When using CAICCI, there are special rules for specifying the ADDRESS OPSCTL MSF DEFINE command for remote system definition. For the value of the APPLID keyword, you must specify the CAICCI system identifier (*sysid*) of the remote system.

For example, if the *sysid* of the remote system is CCI0B on a system that has an MSF ID of OPSS0B, your ADDRESS OPSCTL MSF DEFINE command would be:

```
ADDRESS OPSCTL "MSF DEFINE MSFID(OPSS0B) APPLID(CCI0B) CCI"
```

CCI0B

Identifies the *sysid* of CAICCI on system B.

The MSF checks to make sure that the value you specify is valid.

Note: In ADDRESS OPSCTL MSF DEFINE commands, the keyword CCI can only be used for remote systems.

The following examples illustrate several scenarios for defining an MSF connection.

Example 1: Definition of CCI as a Local System

In this example the CCI keyword is not used. The APPLID keyword is used to specify the local system ID to CAICCI.

```
ADDRESS OPSCTL "MSF DEFINE MSFID(OPSS0A) APPLID(CCI0A)"
```

Example 2: Using VTAM and CCI Simultaneously

- Local definition:

  ```
  ADDRESS OPSCTL "MSF DEFINE MSFID(OPSS0A) APPLID(OPSAPLID)"
  ```

- Remote definition to an APPC connection:

  ```
  ADDRESS OPSCTL "MSF DEFINE MSFID(OPSS0BVT) APPLID(OP2APLID) APPC"
  ```

- Remote definition to a CCI connection:

  ```
  ADDRESS OPSCTL "MSF DEFINE MSFID(OPSS0B) APPLID(CCI0B) CCI"
  ```

Example 3: Using CCI Only

- Local definition:

  ```
  ADDRESS OPSCTL "MSF DEFINE MSFID(OPSS0A) APPLID(CCI0A)"
  ```

- Remote definition:

  ```
  ADDRESS OPSCTL "MSF DEFINE MSFID(OPSS0B) APPLID(CCI0B) CCI"
  ```

CAICCI Enables Communications

CAICCI employs cross-system communication, in which it enables CA solutions to communicate with other CA solutions across any system capable of supporting the CAICCI protocols. This enables CA solutions to quickly and efficiently adapt to new network platforms without requiring extensive application changes.

Through CAICCI, CA OPS/MVS becomes a subscriber of CAICCI services, enabling all communications to be handled through CAICCI. After CA OPS/MVS becomes a subscriber on the local system, all other CA OPS/MVS systems on this system and the remote systems can use the cross-system functions.

CAICCI routines are grouped under the CA z/OS service code W411. For information about installing CAICCI and for further details about its features and functions, see your CCS for z/OS documentation.

Specify CAICCI-related Parameters

To use the CAICCI cross-platform communications services, follow these guidelines for setting parameters:

- Set the value of the INITCCI parameter to YES. The default is NO.
- Set the value of the MSFDELAY parameter to at least 10.

If you want to communicate between systems through MSF CCI links before VTAM is active in the system, set the value of the MSFNONVTAMONLY parameter to YES during product initialization.

After CA OPS/MVS is initialized, you can change the MSFNONVTAMONLY parameter so you can set it to NO after VTAM starts. Then you can connect to other systems using APPC (in addition to the CCI connections that were established earlier).

If you use CCI protocol before VTAM is active (with the intention of using VTAM later), you must specify a valid VTAM APPLID in the MSF DEFINE statement for the local system. The APPLID is not active at the time CA OPS/MVS is started, and communication is established through CCI. However, after VTAM is active, other systems can connect to the specified system through APPC connections, so in these cases you need the VTAM APPLID.

Note: For more information about these and other CAICCI parameters, see the *Parameter Reference*.

Install the IMS Operations Facility

The optional IMS Operations Facility (IOF) obtains IMS commands and unsolicited messages from the IMS AOI exit points. CA OPS/MVS dynamically inserts AOI exits into each IMS that you start. There are sample AOI exits in IMS that you may need to install. The CA OPS/MVS AOI exit does not preclude you from using your own AOI exit.

The CA OPS/MVS parameter IMS*n*INSTALLEXITS can be set to YES (the default value) for each IMS system to install the IOF exits at IMS initialization. Setting this parameter to NO before starting IMS bypasses the installation of the IOF exits.

Note: If the IOF exits were installed during the last initialization of IMS, setting this parameter to NO does not uninstall the exits. A recycle of IMS is required to remove the exits. If IMS was previously initialized without the IOF exits installed, setting this parameter to YES while IMS is up automatically installs the IOF exits on the next message event from that IMS system.

Note: The use of the IOF batch message processing (BMP) for issuing IMS commands without using the IMS WTOR is not affected by whether the IOF AOI exits are installed. There are two AOI exits. The original IMS AOI exit, DFSAOUE0, which is now designated as a TYPE 1 exit, was invoked only on a DB/DC or DCCTL-only system. A second AOI exit type, DFSAOE00, which is designated as a TYPE 2 exit, is invoked in all IMS system types (DB/TM, TM-only, and DBCTL-only). Currently, DBCTL-only IMS systems are *not* fully supported by the IOF. It is recommended that the OPS/MVS parameter, IMS*n*CHAR, for that IMS, be set to a character that is *not* equal to the CRC (command recognition character) of the DBCTL region.

If you do not have your own AOI exits (either a TYPE 1 or TYPE 2 exit) installed, then you must install the sample exits provided in the OPS.CCLXASM library. Browse your IMS RESLIB to check whether any AOI exits have been installed. If the exits are not in the RESLIB, you must install the corresponding sample exits. In the OPS.CCLXASM library, the sample TYPE 1 exit member name is OPSAOUE0, and the sample TYPE 2 name is OPSAOE00. These sample members must be assembled and link-edited into the RESLIB. The proper link-edit statements are provided at the end of each member. The OPSAOUE0 and OPSAOE00 members in the OPS.CCLXCNTL library contain JCL and related usage instructions.

In an IMS Transaction Manager (TM) environment, the possibility of multiple exit types adds a level of coexistence complexity of which a CA OPS/MVS user must be aware.

Rules to Address Complications

The IOF uses the following rules to address the complications:

Exit Combinations

Because IMS can have a combination of TYPE 1 and TYPE 2 exits, use the following guidelines. Remember, the TYPE 2 exit has complete control over whether the TYPE 1 exit is ever invoked.

- You have a user or OEM TYPE 1 exit but no TYPE 2 exit. The CA OPS/MVS supplied TYPE 2 exit must be installed.

  If your user or OEM TYPE 1 exit is called even when OPSMAIN is not running, the supplied OPSAOE00 assembler source code requires a minor modification. The instructions for making this modification are included in the source code. Make this modification before assembling and link editing the program into your IMS RESLIB.

- You have a user or OEM TYPE 2 exit but no TYPE 1 exit. In this case, the IOF TYPE 2 exit dynamically inserts itself into the IMS system you want the IOF to automate. It does not preclude you from using your own AOI exit, and your existing exit is still called.

- You have a user or OEM TYPE 2 exit, and you also have a user or OEM TYPE 1 exit. In this case, the IOF TYPE 2 exit monitors the reply code of your TYPE 2 exit. The reply code determines whether your TYPE 1 exit obtains control. This allows you to control your own TYPE 1 invocation of the exit. For a multisegment message, the IOF does not see the secondary segments if your TYPE 2 exit relinquishes control to your TYPE 1 segment.

To summarize, if your TM environment has neither a TYPE 1 or a TYPE 2 exit installed, the IOF requires the installation of its own TYPE 2 exit. The CA OPS/MVS supplied TYPE 2 exit, OPSAOE00 in OPS.CCLXASM, must be assembled, linked, and installed. If you have either one of the exits, IOF does not preclude you from using your own or OEM AOI exit (see the following restrictions).

Exit Restrictions

The IOF TYPE 1 exit has one restriction. The IOF uses the last 4 bytes of the UEHURSVD field in the IMS UEHB (User Exit Header Block) control block. You must, therefore, make sure that the AOI exit of your site does not also use this area. The sample AOI exit that ships with IMS does not use this area.

The IOF TYPE 2 exit has a restriction that is related to the IMS supplied AOE0WRKA AOI exit work area. The IOF uses the first 72 bytes and the last 20 bytes of this 256-byte work area. If you modify these areas in your own TYPE 2 EXIT, IOF overlays it and may cause your exit to fail. The IOF TYPE 2 exit always takes control first before any of your own or OEM TYPE 2 exit. For more information on the AOE0WRKA work area, see the *IMS Customization Guide*.

The CA OPS/MVS hooks for the IOF do not permit the user/OEM TYPE 1 or user/OEM TYPE 2 exits to suppress or delete IMS messages. Suppression, deletion, or both of IMS messages should be done in CA OPS/MVS message rules.

If you have your own TYPE 1 exit and need to install the CA OPS/MVS supplied TYPE 2 exit, OPSAOE00, do *not* install it ahead of time in your IMS RESLIB, unless CA OPS/MVS is operational and the INITIMS parameter is also turned on. Otherwise, it causes the user TYPE 1 exit to not be called.

If you have a conflict with our restrictions, contact Technical Support at http://ca.com/support.

Note: For more information, see the *IMS Customization Guide* and the *IMS Application Programming: Transaction Manage*r.

INITIMS Parameter Settings

The INITIMS parameter controls the activation of the IOF. The default of the INITIMS parameter is NO. If your data center is an IMS/DB-only shop, leave INITIMS set to NO to prevent IMS SVC recognition problems at startup.

Only those customers who have licensed the IOF can set the INITIMS parameter to YES, and they can do so only at product initialization. If you are a customer who has licensed the IOF but you have z/OS images on which IMS is never used, you can gain a CPU and storage performance advantage by setting the parameter to NO on those systems.

When the value of INITIMS is NO, the OPSPARM/OPSPRM SHOW(ALL) command processor does not include IMS parameters in its output. Setting INITIMS to NO also inhibits the display of IMS parameters in OPSVIEW option 4.1.1. This characteristic is by design and was implemented to reduce storage and improve performance. If INITIMS is set to YES, IMS parameters appear in the OPSPRM OPS/REXX function output, the OPSPARM command processor output, and the OPSVIEW option 4.1.1 displays.

By accessing the CA OPS/MVS *Identify IMS* function (OPSVIEW option 7.4), you can create the parameter cards necessary for the IMS Operation Facility. A batch version of this function resides in member BATCHPRM of the SAMPLES library.

Note: For information about OPSVIEW, see the *OPSVIEW User Guide*.

IOF Installation Parameters

CA OPS/MVS parameters that pertain to IMS control regions, IMS1ID or IMS1DUPLICATE for example, may need to be set during CA OPS/MVS installation.

Note: For information about these parameters, see the *Parameter Reference*.

IOF Operations

After IOF is installed, it is only apparent as a set of extensions to the other facilities of CA OPS/MVS.

IMS Commands Issued from a BMP Region

The IOF can use a BMP region to issue IMS commands and retrieve command responses. This ability provides an alternative to requiring that CA OPS/MVS use the IMS WTOR method whenever it needs to issue an IMS command.

The IOF use of a BMP region for IMS commands has these advantages:

- The IOF can issue most IMS commands without waiting for the IMS WTOR.
- Command responses are more reliable and more efficient, because command output is neither automatically routed to the consoles (as it would be if the commands were issued through the IMS WTOR) nor routed through the subsystem interface (SSI).

Notes:

- The OPSCMD and ADDRESS OPER keyword BMPCMDOUT can be used to echo the current command output optionally. Possible values are OPSLOG, WTO, or NONE.
- The OPSCMD and ADDRESS OPER keyword IMSREPLY forces the current command to bypass the BMP and to issue the command through the IMS WTOR.

Note: For details about these keywords, see the *Command and Function Reference*.

Set up a BMP Region

For CA OPS/MVS to take advantage of the ability to use a BMP region to issue IMS commands, you set up a BMP region.

To set up a BMP region

1. Set the IMS parameter AOIS to a value other than N, which is the default value. For a list of possible values, see the IMS installation guide.
2. Define the BMP TRAN to IMS.

   You run a PSBGEN to define the CA OPS/MVS BMP transaction and application to IMS. Sample control statements are provided in member OPSINBMP in the OPS.CCLXCNTL data set.
3. Authorize the BMP TRAN to issue all commands.

   Authorize this transaction through your security package, as required by IMS. If you are still using the IMS Security Maintenance Utility (SMU), then you run SMU to authorize the BMP transaction to have authority to issue all commands. Sample control statements are provided in the OPS.CCLXCNTL member OPSINBMP.

4. Create a batch BMP started task JCL.

   Use the IMS PROCLIB member IMSBATCH, and make sure the RESLIB that it is using matches the RESLIB of the control region that you want to target. Add the CA OPS/MVS load module library to the STEPLIB concatenation.

   Set the CA OPS/MVS IMS*n*BMPSTC parameter to the member name of the BMP started task JCL.

5. Specify CA OPS/MVS parameters.

   To control the activation or deactivation of the BMP region that the IOF uses to issue commands, you set these CA OPS/MVS parameters:

   - IMS*n*BMPSTC
   - IMS*n*INITBMP
   - IMS*n*PSBNAME
   - IMS*n*TRANNAME

   For more information about these parameters, see the *Parameter Reference*.

BMP Versus WTOR Output Displays

There are two minor differences in the way the IMS presents the output of IMS commands when using the BMP instead of the WTOR:

- When a command is issued from the WTOR, its output has the IMS ID appended to the end of each line. Using the BMP, the IMS presents the output in the same manner, except the IMS ID is not appended to the end of each line; rather, a period, which is used as a placeholder, is appended.
- The IMS BMP handles the following command output in a different way:

  `"DFS058I hh:mm:ss cmd COMMAND {COMPLETED|IN PROGRESS EXCEPT....}"`

  When a command results in a DFS058I message with no exceptions (for example, COMMAND COMPLETED or IN PROGRESS), the IMS presents a blank line to the BMP; the blank line forces the BMP to return the message DFS058I COMMAND IN PROGRESS.

  When there is an exception (for example, START COMMAND COMPLETED EXCEPT PROGRAM XYZ), the output message is identical, except it does not have the IMS ID appended to the end of each line.

IMS Type 2 Message Considerations

IMS Type 2 message protocol is used for communicating from an OPS/MVS system to any IMS system that is a member of an IMSPLEX. The communication can be local, that is, from OPS/MVS to an IMS system on the same LPAR with the ability to contact an active IMSPLEX. It may also be cross-system, where OPS/MVS can send the IMS command using an MSF connection to another OPS/MVS system, and then to the IMPLEX from there. The issuing OPS/MVS does not communicate with IMS directly, and the target OPS/MVS is the one with the requirement to contact an IMSPLEX manager.

If this facility is used, two IBM-supplied modules, CSLSDR00 and CSLSRG00, must be available to the OPS/MVS system that is in contact with the IMSPLEX. IBM provides the modules in the IMS RESLIB, and various choices are available:

- Make them LNKLST resident
- Copy into the OPS Loadlib
- Leave in the original IMS RESLIB and concatenate to a STEPLIB chain
- Isolated into a separate Loadlib, then concatenate

The modules are downward compatible down to IMS 9. The modules from the highest IMS release should be used, and are able to service a site with a mixture of IMS systems at different release levels.

CA OPS/MVS has no requirements for any specific or unique IMSPLEX configuration or startup options. The IMSPLEX itself is tailored according to site standards, and the name is provided to the CA OPS/MVS commands at execution.

IMSPLEX security considerations for Type2 messages and commands are described in IBM manual *IMS Vnn IMSplex Admin. Guide*. The specific area of interest is the CA OPS/MVS interface with the components SCI and OM of the Common Service Layer (CSL).

Install the XTDOUT COF Interface for CICS/TS

The following list pertains to the CICS/TS interface:

- It uses the CICS global exit (XTDOUT) to intercept all transient data write requests. CA OPS/MVS matches a transient data queue name against a list of designated queue names for AOF processing.
- Messages sent to the matched queue names are forwarded to the AOF for rules processing, which also allows for message suppression and rewording. Messages sent to unmatched queue names are ignored by the exit.
- You build and maintain the designated queue name list with the ADDRESS OPSCTL COF command.
- No changes to the standard CICS DCT are required to intercept transient data messages and the selection of specific destinations can be dynamically altered.
- You can build a distinct queue name list for each CICS region, and a general default list for undefined CICS regions.

To install the XTDOUT COF interface

1. Copy load module OPCITDCN from SYS1.OPS.CCLXLOAD to a library in the CICS DFHRPL concatenation.

   The module is linked AMODE=31 and RMODE=ANY.

2. Define the transaction and program to CICS using the CICS RDO facility:

```
DEFINE GROUP(OPXTDOUT) PROGRAM(OPCITDCN)
   DATALOCATION(ANY) EXECKEY(CICS)
   LANGUAGE(ASSEMBLER) RESIDENT(YES)
   DESCRIPTION(OPS/MVS XTDOUT GLOBAL EXIT)
DEFINE GROUP(OPXTDOUT) TRANSID(OPTD) PROGRAM(OPCITDCN)
   TASKDATAKEY(CICS) TASKDATALOC(ANY)
   DESCRIPTION(OPS/MVS XTDOUT EXIT CONTROL)
INSTALL GROUP(OPXTDOUT)
ADD GROUP(OPXTDOUT) LIST(DFHLIST)
```

   The XTDOUT exit code is contained in the OPCITDCN program, and it is enabled as an entry point address in this module using the name OPCITDEX. The exit program does not need to be defined to CICS.

3. Enable the XTDOUT exit by invoking OPTD from a CICS terminal or with a MODIFY command from a z/OS console. OPCITDCN may be added to the CICS PLTPI stage 3 for automatic exit enablement at CICS initialization when desired.

4. Activate the AOF processing of CICS messages by setting the INITCOF and CICSAOF parameters to YES and define, at the least, the default transient data queue name list.

```
X = OPSPRM('SET','INITCOF','YES')
X = OPSPRM('SET','CICSAOF','YES')
ADDRESS OPSCTL "COF DEFINE JOBNAME(DEFAULT)",
     "DESTIDS(CSMT,CSSL,CADL,...)"
```

   The XTDOUT COF interface for CICS/TS is installed.

For information on permitting the suppression of transient data queue messages by AOF rules, see the description of the CICSDELETE parameter in the *Parameter Reference*.

The OPTD transaction may be used to disable and re-enable the exit at any time by invoking OPTD with a single character command code as follows:

- OPTD E-Enable the XTDOUT exit (default command)
- OPTD D-Disable the XTDOUT exit
- OPTD S-Display the status of the XTDOUT exit
- OPTD T-Issue a test message to the transient data queue
- OPTD H-Issue the periodic CICS status message, OPS3420O

Customize the CA NSM SSM CA OPS/MVS Option

This section contains information on customizing CA OPS/MVS for the CA NSM SSM CA OPS/MVS Option product. We recommend that you install and customize CA OPS/MVS before installing the CA NSM SSM CA OPS/MVS Option product. You may use the Agent Technology agent in any supported release of CA OPS/MVS as the communication method between CA OPS/MVS and CA NSM SSM CA OPS/MVS Option.

To use the Agent Technology agent

1. After you install CA OPS/MVS, customize the following parameters:

   - CAUNICONFIGSET - Set this parameter to the Agent Technology configuration name that was specified during the installation of the Agent Technology agent interface for CA OPS/MVS and CA NSM SSM CA OPS/MVS Option. The configuration data is most likely to be found in member CFGSSMO of the *prefix.*CCLXCNTL data set for the CA OPS/MVS product. The name after the colon in the statement #CONFIGSET stateman:OPSCNFG is the required name for this parameter. There is no default value.

     The configuration statements also contain the SNMP community names that will be used and the IP address or host names of the CA OPS/MVS and CA NSM SSM CA OPS/MVS Option workstations that will receive SNMP traps from the agent.

     ```
     #CONFIGSET stateman:OPSCNFG
     ```

     Then the CAUINCONFIGSET parameter is set in the CA OPS/MVS initialization REXX EXEC (typically OPSSPA00), as follows:

     ```
     OPSPRM('SET', 'CAUNICONFIGSET',"OPSCNFG")
     ```

   - CAUNICONNECTWAIT - Determines how many minutes the CA OPS/MVS subtask running in the CA OPS/MVS address space waits between retry attempts to connect to Agent Technology running on the same z/OS image. This product parameter is in your CA OPS/MVS start up OPS/REXX EXEC (usually called OPSSPA00). You can set this parameter to any numeric value between 0 and 120. This parameter can be modified at any time.

     Default Value: 0 (no attempt to connect is made)

     Recommended Value: 2

   - INITAWS - Set this parameter to YES to initialize CA OPS/MVS and CA NSM SSM CA OPS/MVS Option.

     Default Value: NO

     Recommended Value: YES

- CAUNIAGENT - Determines what SNMP agent will be used for communications with CA NSM workstations that are running the CA OPS/MVS and CA NSM SSM CA OPS/MVS Option product.

  Default Value: AWS

  Recommended Value: AWS

  Important! This parameter should not be set to anything other than the default value unless instructed to do so by CA OPS/MVS Technical Support.

- CAUNIALLOWSET - Determines whether SNMP set requests from the workstation are permitted. If CAUNIALLOWSET is set to a value of NO, set requests will be prohibited; if it is set to a value of YES, requests to modify System State Manager table and resource values that are modifiable on the workstation will be performed, regardless of the origin of the request.

  Note: You can use CA NSM security or Windows security to further filter the use of set requests by user.

- CAUNIUSERCURRENT and CAUNIUSERDESIRED - Defines a current, desired, or current-desired state combination that is assigned the user status value, which is displayed as the black icon on the CA NSM 2D map.

  Default: No user status definition.

2. Start CA OPS/MVS.

   Note: The CA OPS/MVS main address space must have read access to the TCP/IP data set (*hlq*.TCPIP.DATA) to determine the correct TCP/IP started task with which it will communicate. The data set must be allocated to ddname SYSTCPD automatically by the system, explicitly in the OPSSPA00 REXX program that is run at CA OPS/MVS initialization, or through a JCL statement in the OPSMAIN started procedure. Consult your systems programmer responsible for installing Agent Technology for the correct data set name to use for the TCP/IP that Agent Technology is communicating with. For example:

   ```
   ADDRESS TSO "ALLOCATE FI(SYSTCPD) DSN('hlq.TCPIP.DATA') SHR"
   ```

   For more information about TCP/IP client data set requirements, see the IBM documentation.

   For more detailed information, see the *CA Network and Systems Management Systems Status Manager* CA OPS/MVS *Option User Guide*.

3. (Optional) Complete this step only if your System State Manager tables do not contain the TNGNOTIFY, TNGELIGIBLE, and RESOURCE_TEXT columns.

   Insert these new columns into your System State Manager directory table and all resource tables. The OPTNGCOL member in the SAMPLES library in CA OPS/MVS contains an OPS/REXX program to assist you with this operation.

   The arguments in the OPTNGCOL REXX EXEC are:

   SUBSYS

   The name of the CA OPS/MVS subsystem (usually OPSS).

   ACTIVATE

   A list of currently managed System State Manager tables for which the value in the TNGNOTIFY column is set to ALWAYS. If a currently managed System State Manager table is not listed in the ACTIVATE argument, then when the TNGNOTIFY column is added, it will have a value of NEVER.

   Note: The RESOURCE_TEXT column is added when the TNGNOTIFY column is added.

   RESTABLE

   A list of table names to add the TNGNOTIFY and RESOURCE_TEXT columns and set the TNGNOTIFY value to NEVER. If the ACTIVATE argument is NULL, then this argument defaults to ALL. RESTABLE(ALL) means that the TNGNOTIFY and RESOURCE_TEXT columns will be added to all currently monitored System State Manager tables.

   The following is an example of the program:

   ```
   OX 'OPS.CCLXSAMP(OPTNGCOL)' SUBSYS(OPSS) ACTIVATE(SSMQA1)
   ```

4. For every System State Manager resource whose status you want reported to CA OPS/MVS and CA NSM SSM CA OPS/MVS Option, set the TNGNOTIFY column in its resource table to ALWAYS.

5. For every table that contains a System State Manager resource whose TNGNOTIFY column you set to ALWAYS, set the TNGELIGIBLE column in your directory table to YES (the default directory table is SSM_Managed_TBLS).

6. Decide whether to establish the daily warm start trap. Use the OPSMTRAP OPS/REXX function, which allows the CA OPS/MVS agent to generate warm or cold start SNMP traps on demand. You can also use any variety of AOF rule, such as a daily midnight AOF TOD rule. If you have multiple systems communicating with CA NSM SSM CA OPS/MVS Option, then we recommend that each system send the daily warm start trap at a different time to avoid overloading the network.

Define the CA OPS/MVS Option to Agent Services

Take the following steps to ensure that the CA NSM SSM CA OPS/MVS Option is properly defined to Agent Services. Those responsible for installing and customizing CCS for z/OS should be familiar with the utilities discussed in this section.

To define CA OPS/MVS Option to Agent Services

1. Make sure that CA NSM Agent Services is installed and running.
2. Load the STATEMAN MIB into Object Store using the LDMIB utility provided in the OPS CCLXCNTL(LDMIB) member. Be sure to customize the environment variables to point to the appropriate Agent and Agent Services data sets and to the STATEMAN MIB. The STATEMAN MIB is distributed with CCS for z/OS and can be found in the CCLXMIB directory. A copy can also be found in the OPS.CCLXMIB(STATEMAN) member.
3. Create the Agent Configuration file. A sample has been provided in the CA OPS/MVS CCLXCNTL(CFGSSMO) member. The configuration file is used to:
   - Override the default community definitions or trap destinations for a specific agent.

     For example, the following specifications, taken from the agents sample configuration file in CCLXCNTL(CFGSSMO), sets the destination address, community name, and port for traps issued by the agent. The #SNMPTRAP host and port information must correspond to the IP address and port of the CA OPS/MVS and CA NSM SSM CA OPS/MVS Option workstation. The port number specified should be validated by TCP/IP configuration personnel to avoid conflicts with other SNMP agents that may already be using the same port number.

     In addition, CA OPS/MVS and CA NSM SSM CA OPS/MVS Option must use the community name of public to retrieve information from the STATEMAN MIB and the community name of admin to modify fields defined in the MIB.

```
#SNMPTRAP
host          141.202.42.253
community     public
port          162
#SNMPCOMMUNITY
access        read
community     public
host          0.0.0.0
#SNMPCOMMUNITY
access       write
community    admin
host         0.0.0.0
```

- Set initial startup values for an agent.

  Only the community name and trap destination specifications of the CA OPS/MVS and CA NSM SSM CA OPS/MVS Option workstation should be customized to conform to the requirements of your site. All other settings should be left as distributed in this sample.

- If your installation is running multiple CA agents that also require #SNMPTRAP and #SNMPCOMMUNITY specifications, then these specified values must be coordinated. Assigning these values in the OPSCNFG configuration set overrides the corresponding values assigned in the CA NSM Agent Technologies aws_sadmin.cfg default configuration set. The values for these two parameters can be specified in either configuration set, but the OPSCNFG configuration set values take precedence after it is successfully loaded. At the very least, the OPSCNFG configuration set must contain all the supplied #SNMPGROUP specifications.

4. After the Configuration settings for Community name and Trap destinations have been customized, load the Configuration file into object store using the LDCONFIG utility provided in OPS CCLXCNTL(LDCFG). If Agent Services and the CA OPS/MVS and CA NSM SSM CA OPS/MVS Option are running, then be sure to stop CA OPS/MVS and CA NSM SSM CA OPS/MVS Option before the running the utility. If Agent Services is running, aws_sadmin must be recycled before the new configuration file takes effect. For additional information, see the *CA NSM Working with Agents* guide distributed on the CA NSM CD. Be sure to customize the environment variables in the LDCONFIG JCL to point to the appropriate Agent and Agent Services data sets and the modified Configuration file. In addition, make sure the community name specification corresponds to what the UCA OPS/MVS and CA NSM SSM CA OPS/MVS Option uses and that traps are routed to the CA OPS/MVS and CA NSM SSM CA OPS/MVS Option workstation.

   For additional information, see the *CA NSM Working with Agents* guide distributed on the CA NSM CD. For information on the CCS for z/OS component required to install the CA OPS/MVS and CA NSM SSM CA OPS/MVS Option, see the appendix "CCS for z/OS Component Requirements." (see page 195)

Required Data Sets

Each agent requires a separate log file to write informational or diagnostic messages during execution. The number and type of messages written depends on the value specified on the CAUNIDEBUG and CAUNITRACE parameters, covered later in this section. Since there are two cooperating agents, two additional DD statements must be added to the CA OPS/MVS started task procedure for the log files. The ddnames are as follows:

DDname	Description
ENVFILE	Points to the CCS for z/OS ENVFILE, typically found in the CCS for z/OS SCRLIB data set

DDname	Description
OPSALOG	The log file of the main agent
OPSBLOG	The log file of the secondary agent
SYSTCPD	Points to the TCP/IP profile data set

The following are the ddnames associated with the STDERR and STDOUT files for the two agents. These files are redirected at agent initialization because the two agents may otherwise overwrite the STDERR and STDOUT files of each other:

DDname	Description
OPSAERR	Redirected STDERR DD of the primary agent
OPSBERR	Redirected STDERR DD of the secondary agent
OPSAOUT	Redirected STDOUT DD of the primary agent
OPSBOUT	Redirected STDOUT DD of the secondary agent.
CEEDUMP	Dump data set for the C language Environment.

Note: If CA OPS/MVS is started with SUB=MSTR specified, then the data sets described above must be pre-allocated as permanent data sets using the CA OPS/MVS CCLXCNTL(ALLOCAWS) JCL. In addition, dynamically allocate the data sets using the CA OPS/MVS initialization REXX EXEC (usually OPSSPA00). If SUB=MSTR is not specified, then the DD can specify SYSOUT instead of a DSN specification.

In addition to the files listed in the preceding tables, the CA OPS/MVS started task procedure must include the following APF authorized data sets, concatenated to the STEPLIB DD, unless they are defined as LINKLST data sets:

- The CA OPS/MVS PDSE load library containing the OPSAGENT and OPSSTRAP modules.
- The Agent Services load library installed as part of the CCS for z/OS.
- The C runtime library, typically named CEE.SCEERUN.

Parameters for Debugging

The parameter CAUNIDEBUG=YES must be specified to activate the message logging service of the agent. This should typically be set to NO, except for trouble shooting scenarios.

The parameter CAUNITRACE=Fn controls the generation of trace messages by the SNMP DPI AWS subagent for SNMP trap requests issued by the workstation task in the CA OPS/MVS address space.

There are eight severity levels, listed in decreasing severity, even though the level numbers increase:

Security Level	Description
F0 - FATAL	F0 causes messages of severity FATAL to be logged.
F1 - CRITICAL	F1 causes messages of severity F0 thru F1 to be logged.
F2 - WARNING	F2 causes messages of severity F0 thru F2 to be logged.
F3 - INFO	F3 causes messages of severity F0 thru F3 to be logged.
F4 - DEBUG	F4 causes messages of severity F0 thru F4 to be logged.
F5 - DEBUG1	F5 causes messages of severity F0 thru F5 to be logged.
F6 - DEBUG2	F6 causes messages of severity F0 thru F6 to be logged.
F7 - DEBUG3	F7 causes messages of severity F0 thru F7 to be logged.

For additional information, see the *CA NSM* documentation distributed on the CA NSM CD.

Configure the Expert Systems Interface (ESI)

The Expert Systems Interface (ESI) allows access to some CA OPS/MVS facilities from an application written in a high-level language or in assembler language.

You must set the INITESI parameter to YES if you are licensed for and are using the ESI.

For detailed information on using the ESI, see the chapter "Expert Systems Interface" in the *User Guide*.

Configuration Tasks for Optional Base Components

This section discusses the installation tasks for optional base components.

How to Install and Configure OPSLOG WebView

The OPSLOG WebView client/server application lets you view OPSLOG messages from a PC workstation. The client side runs as a Java application launched from Microsoft Internet Explorer. This application downloads and installs automatically.

You install two components on your z/OS system to run the server side of OPSLOG WebView:

- A web application on a z/OS HTTP server (such as the IBM WebSphere product)
- A server application on the same z/OS system where CA OPS/MVS is running

Note: You install the server application on the production system where CA OPS/MVS is running. However, you can install the web application on a system that is not running CA OPS/MVS.

Install a server on at least one system in your complex, and you can use MSF to access the other systems. We recommend that you install a server on two or more systems in your installation, so that a single system outage does not close all WebView access. Running a server does not preclude MSF access. You can run a server on every system and still access systems using MSF. Access through MSF is slightly slower than direct access through a server, but it is easier to access through MSF for a short time, than it is to log off one server and connect to another server.

Follow these steps:

1. Follow the resource checklist (see page 130).
2. Define the configuration options (see page 131).
3. Define the SSL communications (see page 133).
4. Configure the web application (see page 138).
5. Configure the server application (see page 140).
6. Complete the post-installation tasks:
    a. Access the OPSLOG WebView interface (see page 144).
    b. Start OPSLOG WebView for the first time (see page 144).
    c. Operate the OPSLOG WebView server (see page 145).

Follow the Resource Checklist

You require the following resources before you install the OPSLOG WebView application:

- **Resource 1:** You *must* know the high-level qualifier (HLQ) assigned to the following data set that was installed from the CA OPS/MVS distribution media, and you must have update access to it:

  [*HLQ*].OPS.CCLXCNTL

- **Resource 2:** You *must* know the HLQ assigned to the following two data sets that were installed from the CA OPS/MVS distribution media, and you must have update access to them:

  - hlq.OPS.CCLXLOAD
  - hlq.OPS.CCLXPLD

- **Resource 3:** Determine the USS destination path (HFS or zFS in compatibility mode directory) where you want to install the OPSLOG WebView server files. This path is an example of a typical path, but you can designate any path that you want:

  ```
  /sys/opsmvs
  ```

  Note: The installation of OPSLOG WebView does not create this HFS or zFS in compatibility mode directory. This directory must exist on permanently allocated storage or a mounted file system before you install OPSLOG WebView. About 5 MB is sufficient.

- **Resource 4:** You must have write access to the httpd.conf configuration file on your web server. An example of where the file resides follows; however, the location depends on your web server configuration:

  ```
  /sys/http/etc/httpd.conf
  ```

  This file is an HFS or zFS in compatibility mode file; therefore, superuser permissions can be used to provide write access to it.

- **Resource 5:** Determine the URL that users need to access to open an OPSLOG WebView session.

  The URL has the following general format:

  `http://hostname.domain/applname`

  hostname

  Specifies the host name and is defined in the Logical Parmlib Concatenation and is usually the same as the JES2 or JES3 node name.

  domain

  Specifies the IP domain of your company. For example, the CA domain is ca.com.

  applname

  Specifies the name users enter to access OPSLOG WebView and is specified during the installation.

  Note: Hostname and domain are not case-sensitive, but *applname* is case-sensitive.

Define Configuration Options

You can use two methods to pass configuration information to the OPSLOG WebView server when the server is initiated:

- JCL EXEC PARM=*value* field

  This method has a 100-character limit imposed by z/OS, which could be inadequate to support all the required parameters.

- SYSIN ddname statement, which is in the OPSLOGSV started task

  This method is optional and it can do the following:

  - Accommodate an unlimited number of characters
  - Be used in addition to the PARM=value field, or in place of it
  - Name a PDS member or flat file containing fixed or variable length records

Every parameter that can be assigned in PARM can also be assigned using the SYSIN DD file. Any parameter defined in both places will be set from the SYSIN file. One difference between the two methods: The PARM field has all options in a single concatenated character string, whereas each parameter must be stored in a separate record in the SYSIN file. SYSIN must be a DASD-resident file.

There are three types of SYSIN records, which are distinguished by the character in column 1.

- - (minus sign) or / (slash, virgule, diagonal, or solidus)

  Indicates a parameter line. Both the / and - perform exactly the same function; they mark the lines to be parsed for parameter settings.

- * (asterisk)

  Signifies a comment line. Use asterisk comment lines to keep detail information about settings, or any other general information. Asterisk comments are not sent to the SYSPRINT file.

- ; (semicolon)

  Signifies a comment line that will be printed in the SYSPRINT data stream.

The following is a sample procedure:

```
//OPSWEBVW EXEC PGM=OPSLOGSV,TIME=NOLIMIT,REGION=0M
//SYSPRINT DD SYSOUT=*           <=== For server console messages
//STDOUT   DD SYSOUT=*           <=== For error/debug messages
//STDERR   DD SYSOUT=*           <=== For error/debug messages
//CEEDUMP  DD SYSOUT=*           <=== For z/OS LE reports
//SYSUDUMP DD SYSOUT=*           <=== Dump dataset
//SYSIN    DD DISP=SHR,DSN=MY.PDS(SYSIN)
```

Note: The following statement is not acceptable because the JES Spool system is not available to a started task:

```
//SYSIN    DD *
```

Define SSL Communications

Use the startup option, S, to specify that you want to use secure socket layer (SSL) communications for WebView communications. You must specify the S option three times, as shown in this excerpt from a SYSIN configuration file:

1. Set SSL mode on and specify the path to the SSL keyring.

   ```
   -S PATH /sys/usr/lpp/opsmvsc/cpp/skeys.kdb
   ```

2. Use this password to open the keyring.

   ```
   -S PASS password
   ```

3. SSL searches the keyring for a key having the following label.

   ```
   -S LABEL label
   ```

SSL uses these settings to configure access to a digital certificate and then instruct the WebView server to use SSL.

Important! Java Run Time Environment 5.0 minimum is required to run SSL in the WebView clients.

The server runs in a USS environment as an MVS started task. It uses IBM SSL, and therefore uses IBM implementations of the keyring, keyring password, and key label. The IBM key management program is *gskkyman*, located in */usr/lpp/gsksl/bin* in the HFS or zFS in compatibility mode.

Digital Certificate Protocol

SSL encryption uses public key technology to establish an encrypted link between a client/server pair. It is possible, using a very simple protocol, to establish an encrypted link between a client and server without first having to make any agreement between the communicating partners. Unfortunately, a link made under such conditions is vulnerable to a third-party attack, in which a person maliciously intercepts communications intended for the host, and either gains access to passwords, or even to all the communications for the duration of a session.

To thwart the man in the middle attack, CA chose to use a digital certificate protocol, in which a digital certificate containing the public key of the intended server is delivered by conventional mail or any other reasonably secure method to the client systems that uses SSL encryption. A related digital certificate is available to the host. This technique guarantees that the client can only link up with the intended host.

The WebView client is written in Java, and therefore uses Java SSL. The digital certificate:

- Is sent to the client machine from a Certificate Authority
- Should be installed in a file named .keystore (dot-keystore) under the Documents and Settings\userid directory in a Windows XP system
- For multiple users of the PC, can have a separate .keystore file that is installed for each user

Use Java's keytool program to install the certificate on each machine.

It is possible to install the same certificate for all users, any group of users, or only one user. For security reasons, We recommend using a different certificate for each user. For example, if an employee quits, you can revoke his certificate independently of all others, causing minimum disruption.

The management of digital certificates includes at the least creating them and distributing them. It may also entail a relationship with a Certificate Authority, which is a commercial enterprise that specializes in creating certificates. You can create your own Certificate Authority, or can use a commercial one. There are advantages to either approach.

Note: The details of certificate management are beyond the scope of this manual. For more detailed information, contact your Certificate Authority (CA) Support team.

How Security Affects OPSLOG WebView

Review the following security information.

Web Server

The OPSLOG WebView server retrieves a security environment from the System Authorization Facility (such as RACF or CA Top Secret) for each client. The security environment controls, for each user, whether the user can perform certain actions, such as issuing host commands. To obtain the security profile for clients, the server must have UPDATE access to the BPX.SERVER. If the server runs z/OS 1.7 or later, the server must also have access to BPX.CONSOLE. Otherwise, the server must run as UID=0.

In addition, check if your Web Server (HTTPD or WebSphere) is secured with the program controlled attribute of your security package. If so, then the OPSLOGV program, which resides in OPS.CCLXPLD, and the OPMFSB program, which resides in OPS.CCLXLOAD, needs to be program-controlled. See the documentation for your security package for information on setting this attribute.

Data Authentication and Encryption

Secure the OPSLOG WebView data in transit using the SSL protocol for authentication and encryption.

Note: When you specify the S option, all data communications between the OPSLOG WebView server and the OPSLOG WebView client takes place using a secure protocol such as TLS1, TLS1.1 or TLS1.2. Since the OPSLOG WebView client code is initiated via a web server, we recommend that you disable SSLv3 in your Internet browser and on your web server before launching the client.

Enable SSL for the OPSLOG WebView server by specifying the **S** option for the OPSLOGSV started task.

Transport all data unencrypted, except password data, by not specifying SSL and specifying the U option for the OPSLOGSV started task.

If neither the **S** option nor the **U** option for the OPSLOGSV started task is specified, OPSLOG WebView server, by default, bypasses client and server authentication and encrypts data in transit using the XOR encryption scheme.

System

To enable OPSLOG WebView users to issue commands on a target system when OSFSECURITY is set to CHECKUSERID, a security rule must be enabled on the target system.

If no security rule is enabled on the local system, then default permissions apply on both local and remote systems.

If OSFSECURITY is set to CHECKUSERID and a security rule is written on both the local and the remote system, the OPSLOG WebView user must have permission to issue commands on the local system before the security is checked on the remote system. In other words, the OPSLOG WebView user must have permission on both the local and the remote system to issue commands on the remote system.

Note: If you fail to supply a security rule on the target systems, then commands sent there are not executed and you cannot see any indication of an error unless the DEBUGOSF parameter is set to a value of ON.

Security Rules

You can control access to CA OPS/MVS facilities from OPSLOG WebView through security rules by specifying which users can:

- View OPSLOG messages
- Issue host commands

If there are no pre-existing security rules for controlling access to the OPSLOG, then default security permissions apply: from OPSLOG WebView, all users are permitted to view the OPSLOG but are not permitted to issue host commands.

To override default permissions to display OPSLOG messages, create a security rule to permit a user or list of users to view the OPSLOG. This authorization also enables the host command area, while enabling authorized users to enter host commands from OPSLOG WebView.

The following sample rule allows only users in the allow_users list to view OPSLOG messages:

```
)SEC OPSBRW
)PROC
allow_users = “TSOUSER1 TSOUSER2 TSOUSER3”
user = sec.opauusid
if  WORDPOS(user,allow_users) = 0 then return “reject”
else return “accept”
```

To override default security restrictions to issue host commands, create a security rule to permit a user or list of users to issue host commands.

The following sample rule allows users that are listed in the allow_users list to issue host commands:

```
)SEC OPSCMD
)PROC
allow_users = “TSOUSER1 TSOUSER2 TSOUSER3”
user = sec.opauusid
if  WORDPOS(user,allow_users) = 0 then return “reject”
else return “accept”
```

Members SECWEBV1, SECWEBV2, and SECWEBV3 of the distributed sample rules library provide examples of providing this security. For more information and a list of the steps that are required for granting security access, see the samples.

If you are currently using CA OPS/MVS security rules to secure these operational functions, then view these samples to determine the logic changes that you need to incorporate into your existing rules.

- Login IDs

  Before establishing an OPSLOG WebView session, OPSLOG WebView prompts each user to log in using a valid user ID and password for the target z/OS system. You can choose to use existing TSO user IDs or define new user IDs for this purpose. The only requirement is that the user ID be authorized to log on to the target system.

  Note: You *must* define an OMVS segment to the user IDs that require OPSLOG WebView access. These user IDs need to log on to the system where the OPSLOGSV STC is running.

SYSIN Statement Parameters

You can find the following SYSIN statement in the OPSLOGSV started task procedure's SYSIN file:

```
//SYSIN     DD DISP=SHR,DSN=MY.PDS(SYSIN)
```

Set the following parameters in the SYSIN file:

-B buffer

Sets the TCP send buffer size. This parameter can be used for TCP segmentation offload.

Limits: 256-256K

-C path

Sets the HFS or zFS in compatibility mode node where you want to store user configurations.

Example: /sys/usr/caops/cfg. There is no default.

-G 0|DISABLE or 1|ENABLE

Allows a system administrator to deny the ability of users to activate GO mode in WebView.

Note: O and DISABLE are equivalent to each other and 1 and ENABLE are equivalent to each other.

0|DISABLE

Disables GO mode for all users. The GO option in the OPTION menu of WebView is “grayed out” which makes it unavailable to the users.

1|ENABLE

Enables GO mode for all users. When GO mode is enabled, the GO option is enabled, and not grayed out. GO mode is only active if it is activated by the user. Specifying enable is optional since this condition is the default condition.

Default: 1|ENABLE

-I minutes

Specifies the number of minutes that a client can be idle before the server forces the client to disconnect. The default is 120. 0 is a special case that prevents forced logout.

-M maxcons

Specifies how many clients the server permits to log on at one time. The default and maximum are 50.

-P port#

Specifies which TCP/IP port the server and client use to communicate with each other.

-S PATH name

Selects SSL as the encryption manager, and specifies where the keyring can be found. PATH must be specified verbatim, as shown. If name begins with a slash, the keyring is assumed to be stored in the HFS or zFS in compatibility mode. If name begins with any other character than slash, the keyring is assumed to be stored by MVS in a SAF-protected stash.

-S PASS password

Specifies passwords that are used for keyrings be stored in the HFS or zFS in compatibility mode. Each keyring has its own password. Omit this line if you store your passwords in a SAF-managed stash.

-S LABEL label

Specifies the label, which is a property related to a specific key in a keyring.

-T level

Trace level (formerly debug level) specifies the level of detail that is sent to SYSPRINT.

Important! The -T level replaces the former -d (debug) level. The JCL fails if you specify -d.

-U Unencrypted

Specifies that all communication between host server and client are to be unencrypted, except for passwords, which are always encrypted. SSL overrides this form, if both SSL and Unencrypted are specified.

Configure the Web Application

A z/OS HTTP web server environment, such as the IBM WebSphere product, must be installed on your system before you configure the OPSLOG WebView web application. The detailed setup instructions that follow discuss how to add definitions to an existing server.

To configure the Web Application

1. Use a text editor to modify the httpd.conf configuration file in your web server (see Resource 4).

   Include a PASS statement similar to the following example:

   ```
   PASS /applname/* /uss target path/*
   ```

   applname

   Provides the application name of the OPSLOG WebView server that users will specify on the URL used to open an OPSLOG WebView client session (see Resource 5).

   uss_target_path

   Provides the path name where you installed the OPSLOG WebView server files (see Resource 3).

   Ensure the following AddType statements are included to direct the web server to recognize particular file types used by OPSLOG WebView:

   ```
   AddType .jnlp application/x-java-jnlp-file ebcdic 1.0 #JNLP - Java Web Start
   AddType .css  text/css                     ebcdic 1.0 #Cascade Style
   AddType .js   text/javascript              ebcdic 1.0 #Javascript
   ```

2. Use a text editor to customize the opslog.jnlp file, which is located in the USS/HFS or zFS in compatibility mode destination path:

   ```
   codebase="http://hostname.domain/applname">
   ```

   hostname.domain/applname

   Provides the hostname and domain where your web application is installed.

   Note: See Resource 3 in the Resource Checklist (see page 130).

3. (Optional) Set up the TCP receive buffer size to support the TCP segmentation offload. Use the following statement:

   ```
   <property name="jnlp.SocketReceiveBuffer" value="n" />
   ```

   n

   Specifies the size of the buffer in bytes.

   Limits: 256-256K

Examples: Customized opslog.jnlp File

- If your web server were running on host USILXXX and domain ca.com, the file would look like the following:

  ```
  codebase="http://USILXXX.ca.com/opslog">
  ```

- If your web server uses an alternate port, then it must be specified as follows:

  ```
  codebase="http://USILXXX.ca.com:4080/opslog">
  ```

- If your web server uses an alternate applname, say webview, then it must be specified as follows:

  ```
  codebase="http://USILXXX.ca.com/webview">
  ```

More information:

OPSLOG WebView Post Installation (see page 144)

Configure the Server Application

Perform these steps to configure the OPSLOG WebView server application, including security considerations:

1. A sample started task procedure, OPSLOGSV, is provided in *hlq.OPS.CCLXCNTL*. Copy this procedure to a system PROCLIB, and then tailor it with the appropriate HFS or zFS in compatibility mode path name (see Resource 3) for user configuration data storage and CA OPS/MVS data set names (see Resource 1 and Resource 2).

   You can eliminate the STEPLIB DD if these data sets are already in the link or LPA lists.

2. (Optional) By default, the OPSLOG WebView server communicates with the client program through TCP/IP sockets using port 6001.

   If this port conflicts with another port on your mainframe system or on a target client machine, then you can change the default port as follows:

   a. Using the text editor (see the following section), open the opslog.jnlp file, which is located in the USS/HFS or zFS in compatibility mode destination path (see Resource 3), in your web server and replace 6001 in the line <property name="connPort" value="6001"/> with the new port number.

   b. To apply this change, stop your web server, and then restart it.

   c. Modify the startup PROC OPSLOGSV by replacing 6001 in the PRT=6001 statement with the number of your new port.

   d. To apply this change, stop OPSLOGSV, and then restart it.

   Note: Unlike OSF TSO and USS servers, which CA OPS/MVS manages automatically, external automation, scheduling, or system facilities must be used to manage the OPSLOG server. The CA OPS/MVS System State Manager component can be used to control the starting and stopping of the server.

Specify security access permissions or restrictions to the OPSBRW and OPSCMD command processors.

3. If the OPSLOG WebView server is enabled for SSL, use the text editor to customize the opslog.jnlp file by specifying the location where the OPSLOG WebView client can expect to find the private key store. On the OPSLOG WebView client workstation, store the imported keys, certificates, or both, at the location that the opslog.jnlp file specifies.

   Java defines at least two default key stores for storing the keys and certificates. The store that holds commercial trusted certificates defaults to a file named *cacerts* and is stored in the file node $java_home\lib\security\. $java_home represents the value of environment variable java_home. The fully qualified name of the file might be similar to the following example:

   ```
   \Program Files\Java\JRE1.6.0_03\lib\security\cacerts
   ```

   You can use the keytool utility that is provided by the Java Runtime Environment to import your own certificate to this file and make it available to all users who log on the PC.

   There is also a private key store for each PC user. This store default location is at *$user.home**keystore* and is usually called .keystore. The fully qualified name might be similar to the following example:

   ```
   \Documents and Settings\username\keystore
   ```

   There can be as many such files as there are users of the PC system.

   Your opslog.jnlp file is located in the /sys/opsmvs directory and uses a properties setting to tell SSL where the client should look for the trusted certificate. The sample.jnlp file, property name, javax.net.ssl.trustStore tells the client where to find the private key store.

   The following are some valid settings for the property javax.net.ssl.trustStore:

   *USER stands for PC file node, $user_home\username\ and automatically supplies a file name of .keystore, unless another name is given. For example,

   ```
   value="*USER"           ==> \Documents and Settings\username\.keystore
   value="*USER\mykeys.kdb ==> \Documents and Settings\username\mykeys.kdb
   ```

   *SYSTEM stands for the PC path $java_home"\lib\security\" and automatically supplies a file name of cacerts, unless another name is given. For example:

   ```
   value="*SYSTEM"       ==> \Program Files\Java\JRE1.6.0_03\lib\security\cacerts
   value="*SYSTEM\OPScert" ==> \Program
   Files\Java\JRE1.6.0_03\lib\security\OPScert
   ```

   A complete path can be supplied:

   ```
   value="C:\Program Files\Java\JRE1.6.0_03\lib\security\cacerts"
   ```

   Default: *SYSTEM

4. (Optional) Change the minimum and default refresh interval. These settings apply to all clients.

   Default refresh interval: 30 seconds

   Default minimum refresh interval: 10 seconds. The Options/Settings dialog does not honor any setting less than the minimum interval.

   To set new minimum and default values, the system administrator must add the following lines to the opslog.jnlp file, which is in the HFS or zFS in compatibility mode, either before or after similar “property” lines already in the sample .jnlp file.

   These sample lines set both minimum and default to 15 seconds:

   ```
   <property name="GoModeRefresh" value="15"/>
   ```

   ```
   <property name="GoModeDflt" value="15"/>
   ```

   Note: The minimum cannot be set to less than 10, and the default cannot be set to less than the minimum.

5. To start the server, issue the z/OS start command S OPSLOGSV.
6. To stop the server, issue the z/OS stop command P OPSLOGSV.

More information:

How Security Affects OPSLOG WebView (see page 134)

Use the Text Editor

To use the text editor

1. From the TSO Ready prompt, enter ISH.

   This command starts the I Shell user interface to USS files.

2. Near the center of the screen, ISH displays an input area where you enter the USS path name for the HFS or zFS in compatibility mode directory containing the file that you want to edit. For example:

   ```
   /sys/usr/lpp/opsmvs
   ```

3. Leave the command line as blank and press Enter.

   The directory containing the desired file displays.

4. Scroll as needed using the PF7 and PF8 keys to find the file to edit
5. Enter a question mark (?) beside the file name and press Enter.

   The pop-up box displays

6. Enter 5 in the pop-up box.

   The editor initiates, which behaves like ISPF edit, and you can edit your file.

ASCII/EBCDIC Conflicts

OPSLOG WebView stores several types of files in your HFS or zFS in compatibility mode. Some of these files are distributed in binary format, and some are distributed in EBCDIC. Some examples are:

File type	Code
.html	EBCDIC
.jpg	binary
.png	binary
.gif	binary
.js	EBCDIC
.css	EBCDIC

By default, HTTPD treats files as ASCII. However, by using the Addtype configuration statement in the HTTPD configuration file, you can override the ASCII default. Many sites configure .html files to be EBCDIC coded. You can explicitly associate a file type to ASCII too, and although it is not necessary to do so, the explicit assignment serves to inform other maintainers of the configuration file that the file type is already in use.

It is possible that the character sets we have used to define WebView-related files could differ from the file formats used by other software that is already installed at your site. The .js (Java Scrip) and .css (Style Sheet) are the most vulnerable. If any of the OPSLOG WebView help files are in a different format than that already in use by other software, the most convenient fix is to translate the new OPSLOG WebView help files from EBCDIC to ASCII. You can do this by using FTP to transfer the file to a PC with conversion from EBCDIC to ASCII, then resending the file to the host as a binary format file. This establishes an ASCII instance of the file on your mainframe system. We recommend renaming and saving the EBCDIC version of the file for easy reference to its contents. You may have other ways to convert files to ASCII.

Files that are binary in nature, such as .png files, are neither ASCII nor EBCDIC. They are binary, and should be transmitted as such.

OPSLOG WebView Post Installation

After you install and configure the web and server applications, you can access the OPSLOG WebView GUI from your web browser by initiating an OPSLOG WebView session with a URL of this form (see Resource 5):

`http://hostname.domain:port/applname`

where *hostname* and *domain* are IP addresses. If you know the numeric IP address, then you can use it instead.

port is the IP port number that you defined for HTTP (browser) access. If you define the default port of 80, then it can be omitted from the URL.

applname is defined in the PASS statement.

More information:

Configure the Web Application (see page 138)

Start OPSLOG WebView for the First Time

We recommend that you use the control options of your web browser to delete temporary internet files and offline content on your workstation to ensure that updated files from the current release of OPSLOG WebView are downloaded.

The Java runtime environment must be installed on your workstation to start OPSLOG WebView. If it is already installed. We recommend that you use the Java Control Panel to remove any previously existing OPSLOG WebView application from the Java cache to ensure that the current release of OPSLOG WebView client is installed.

If the Java runtime environment is not installed on your workstation or if it is outdated, the first time you start OPSLOG WebView, the Security Warning screen automatically appears directing you to install a current release of the Java runtime environment. Follow the instructions for a typical installation.

After the Java runtime environment is installed on your workstation, Java Web Start will download the OPSLOG WebView client program to your workstation.

A screen appears indicating that the client program is being downloaded to your machine.

If the Security Warning screen appears, click Start.

Operate the OPSLOG WebView Server

The commands described in this section allow you to control various aspects of an OPSLOG WebView Server session. These commands can be issued by using the MODIFY command. The following example shows the OPSLOG WebView server command syntax:

```
F OPSLOGSV,APPL=TRACE 5
```

Note: APPL= is always necessary when issuing commands to the server.

The response from these commands is JESMSGLG and SYSPRINT for the OPSLOG started task; these responses can also be found in the system log or in OPSLOG.

In the following list of commands, uppercase letters indicate the minimum number of letters that are required when typing the command.

CANcel SOCKET *nnn*

Immediately terminates the connection to the client using socket *nnn*. Use the USERS command to determine which session you wish to terminate.

GOMODE=0, GOMODE 0, GOMODE=DISABLE, GOMODE DISABLE|

GOMODE=1, GOMODE 1, GOMODE=ENABLE, GOMODE ENABLE

Changes the GO mode option status “on the fly.” The four variants in the first row are all equivalent to each other and disable the Go mode option. The four variants in the second row also have identical effects and enable the Go mode option. These commands take effect immediately for new client connections, but an existing client retains his status, either disabled or enabled until he logs off the server and reconnects.

Example Syntax: F OPSLOGSV,APPL=GOMODE=ENABLE

STATistics | STATS

This command reports a statistical profile of server activity. Along with various settings, reported information includes:

- Number of current logged on clients
- Peak number of clients
- Number of logon failures (usually caused by a password failure)

TIMEout *nnnn*

Sets the maximum idle time, in minutes, that a user can remain idle before the server terminates their session. Valid settings range from one minute to 1440 minutes (24 hours). The time-out defaults to 99 minutes. The -i parameter in the startup procedure can also be used to set this value.

TRACE YES|NO|ON|OFF|*n*

Controls whether the server generates trace output messages, and also controls the level of detail of trace information. YES enables trace output, but does not change the numeric level of detail. NO suspends trace messages. ON and OFF are equivalent to YES and NO, respectively.

The numeric argument *n* sets the level of detail to be reported in the trace output, with 1 being the least detail, and 9 being the most. A numeric argument of 1 through 9 enables trace output as well as setting the level. A numeric argument of 0 suspends trace messages without changing the level, and is equivalent to TRACE NO.

USERS

Lists the currently active user table, including the name of the user, the socket to which he is attached, the amount of time he has been logged on, and the amount of idle time.

Configure JES2 Environmental Functions (Required for JES2)

To use the JES2 environmental OPSINFO/OPSJES2 functions, assemble and link the JES2 offsets module, OPJ2CB, included in the CA OPS/MVS distribution media. The JCL to assemble and link the JES2 offsets module resides in SYS1.OPS.CCLXCNTL(JES2ASM). The source resides in SYS1.OPS.CCLXASM(OPJ2CB).

When you start CA OPS/MVS on a JES2 system and CA OPS/MVS detects that the default (null) OPJ2CB module has been loaded, it attempts to locate a default offset table for the version of JES2 running on your system. Since the default offset tables that CA Technologies supplies may not match your maintenance level of JES2, it is possible that they will not function correctly. If no default offset table is found, then the JES2-related OPSINFO() functions and the OPSJES2() function will not work. We strongly recommend that you always assemble OPJ2CB/OPJ2CB*xx*, rather than relying on the default offset tables.

The assembly of module OPJ2CB may result in a return code of 4; however, all non-zero return codes should be carefully investigated.

Important! All copies of the JES2 offsets module OPJ2CB must be reassembled before attempting to start CA OPS/MVS. Attempting to use old releases results in erroneous results, abends, or both.

Enable Library Sharing Among CPUs with JES2OFFSETSUFFIX (JES2 only)

The JES2OFFSETSUFFIX parameter lets you share a common CA OPS/MVS library among multiple CPUs that have different versions of JES2. It lets you add a two-character, alphanumeric suffix to the name of your OPJ2CB module. You can change this parameter anytime to load a new JES2 offset module.

Use the following OPSPRM REXX function to specify a suffix for the OPJ2CB module. Set this parameter as follows:

var = OPSPRM("SET","JES2OFFSETSUFFIX","*xx*")

where *xx* is the suffix desired.

To use JES2OFFSETSUFFIX, assemble and link the OPJ2CB module with its suffix. Member JES2ASM in OPS.CCLXCNTL contains assemble and link job JCL. To implement the new suffix, modify the NAME linkage editor statement in the JES2ASM member with the desired suffix:

```
MODE RMODE(ANY),AMODE(31)  SET ADDRESSING/RESIDENCE
ENTRY OPJ2CB
NAME OPJ2CBxx(R)     (<--suffix required)
```

If you want to use the JES2OFFSETSUFFIX, then the OPJ2CB load module must be the default load module that is distributed on the tape.

Note: A reassembled OPJ2CB takes precedence over an assembled OPJ2CBxx.

Set up the JES3 Interface

The JES3 command support is separate from the z/OS console support. The IATUX18 exit controls JES3 commands that are issued at JES3 RJP consoles. The subsystem interface (SSI) handles JES3 commands issued on z/OS consoles. If you have no JES3 RJP consoles, then IATUX18 provides no additional support.

CA OPS/MVS interfaces with JES3 for intercepting commands through the standard JES3 IATUX18 exit mechanism. The CA OPS/MVS exit code is a module (OPJS18PR) that you can link-edit with your existing IATUX18 load module. The created load module, named IATUX18, contains both the user exit and the CA OPS/MVS exit, with the CA OPS/MVS exit entered first during JES3 execution.

Install the IATUX18 Exit (Required for JES3)

CA OPS/MVS interfaces with JES3 for intercepting commands through the standard JES3 IATUX18 exit mechanism.

To install IATUX18 Exit, link-edit the CA OPS/MVS exit code module (OPJS18PR) with your existing IATUX18 load module. The created load module, named IATUX18, contains both the user exit and the CA OPS/MVS exit, with the CA OPS/MVS exit entered first during JES3 execution.

Install OPJS18PR

The CA OPS/MVS exit code module OPJS18PR link-edits with the existing IATUX18 load module.

To install OPJS18PR, link-edit it with the IATUX18 module at your site using member LINKUX18 of the SYS1.OPS.CCLXCNTL data set, which contains the following sample job to do this:

```
MODE        RMODE(24),AMODE(31)
INCLUDE     OPSLOAD(OPJS18PR)
INCLUDE     SYSLMOD(IATUX18)
ENTRY       OPJS18PR
NAME        IATUX18(R)
```

You can also do the link-edit with SMP/E if your installation policy requires its use.

Define the Shared File VSAM KSDS

The shared file facility allows for the permanent storage of global variables and their values on an external VSAM KSDS that can be shared (through shared DASD or VSAM RLS) between different systems. This data set is totally independent of the global variable checkpoint data set. Records are read from or written to the shared VSAM data set only through the OPSHFI command processor. One possible use for this file is the common initialization of a large group of variables that would otherwise require an extensive number of OPSVALUE() calls.

For example, the following command reads the VSAM data set and creates global temporary variables for all records that match the specified variable name prefix:

```
OPSHFI READ GLVTEMPO.DEVICE.*
```

To use the shared file with the OPSHFI command processor, examine and set the GLVSHAREDFILE, GLVSHAREDDD, GLVSHAREDRESERVE, and GLVSHAREDRLS product parameters accordingly.

Install the UNIX System Services

The base install of CA OPS/MVS contains the UNIX System Services feature. This section explains the parameter settings, JCL changes, and other customization tasks.

The base install lets you do the following:

- Use the Address USS USSCMD.
- Turn the parameters on for USS process creation and termination, which provides USS events in OPSLOG for each instance. You can then do automation using AOF USS rules.

How the UNIX System Services Feature Works

The UNIX System Services (USS) feature uses OPS/REXX programs and AOF rules to perform various functions.

The USS feature lets you perform the following tasks:

- Issue UNIX shell commands
- Receive command responses in REXX variables
- Monitor the creation and termination of USS processes

If the CCS for z/OS Event Management (EM) component is installed, you can also do the following:

- Access all messages appearing on the z/OS EM console in OPSLOG, including z/OS USS syslogd messages
- Take action on the z/OS EM messages using the AOF USS rule type
- Let commands and messages be sent from OPS/MVS to z/OS EM or any other CA EM platform that is connected

The combination of these facilities lets CA OPS/MVS expand its scope of automation to the z/OS UNIX domain and the CA NSM enterprise management network.

Note: NOOUTPUT mode is implied from AOF rule types that cannot wait, such as MSG, CMD, and so on. Command responses cannot be retrieved from the nowait rules.

Activate Process Monitoring Component

The monitoring of USS process creation and termination events requires only the installation of a dynamic system exit module at the exit points, as documented in the IBM publication: z/OS V1R4.0 Unix System Services Planning (GA22-7800-03).

To activate the process monitoring component, set both the INITUSSPROC and USSPROCRULES parameters to the value YES. AOF USS message rules may then be written to automate USS process creation and termination events.

Install the Event Management Component of CCS for z/OS

If you want to interface with z/OS EM component of CCS, then you must also install CCS for z/OS and the EM component.

To install the EM component of CCS for z/OS

1. Activate the USS-based z/OS Event Management component to provide these USS message events to AOF. A sample started task for starting the z/OS Event Management component of CCS is provided by that component.

   For a list of the CCS for z/OS components required to run the Event Management Component, see the appendix "CCS for z/OS Component Requirements *(see page 195)*."

2. Run the INSTUSEX member of SYS1.OPS.CCLXCNTL.

   The CA OPS/MVS USS message exit is copied to the z/OS EM HFS or zFS in compatibility mode directory and replaces the dummy version of the message exit installed by z/OS EM.

Important! Run this job whenever either CCS for z/OS or CA OPS/MVS maintenance changes the message exit module.

You can use the following Address USS commands that interface with z/OS EM: CMD, WTO, WTOR, PING, REPLY, and DOM.

USS Feature Parameters

The following parameters control the initialization and function of the USS features in CA OPS/MVS:

- INITUSS-Must be set to YES in the OPSSPA00 initialization REXX program
- USSSTC-Must contain the name of the started task procedure for the USS server if it is not OPSUSS
- USSSWAPPABLE-Can be set to NO to make the USS servers non-swappable
- USSALLOWRESTART-Can be set to YES to allow failed USS servers to be restarted
- USSRULES-Must be set to YES for USS message events to occur
- BROWSEUSS-Can be set to YES to display USS messages in OPSLOG
- USSACTIVE-Must be set to ON to use the ADDRESS USS REXX host command to send USS commands to the USS servers for execution
- BROWSEUSSPROC-Can be set to YES to display USS process creation and termination messages in OPSLOG
- INITUSSPROC-Must be set to YES in the OPSSPA00 initialization REXX program to activate the USS process monitoring component
- USSPROCRULES-Must be set to YES for USS process creation and termination events to occur
- USSSECURITY-Can be set to CHECKUSERID to dynamically modify the security environment to the authority of the command issuer

Security Issues

The USS servers can dynamically modify the security environment to match the authority of the command issuer. Activate this feature with the USSSECURITY parameter.

The security check runs in one of two modes, depending on the value of the USSSECURITY parameter:

USSSECURITY=CHECKUSERID

In this case, the USS servers must have super user authority and have read access to the BPX.DAEMON resource (if defined) to dynamically modify the security environment to match the authority of the command issuer. The command issuer must have sufficient authority in the USS segment of their security profiles to perform the types of USS commands that automation applications require.

USSSECURITY=NOSECURITY

In this case, the USS servers execute all commands with the current authority of the server. AOF security rules are the only way to restrict the types of USS commands that a particular user issues. The USS servers must have sufficient authority in the USS segment of their security profiles to perform the types of USS commands that automation applications require. Otherwise, you may encounter failures due to a lack of execute authority for that command or because the directory that contains the command is inaccessible. The authority that is required may range from basic user to super user.

Note: For more information about the USSSECURITY parameter, see the *Parameter Reference*. Consult your security administrator before you attempt to activate the USS feature.

USS Server Environmental Variables

To execute Event Management component of CCS for z/OS commands as shell commands or as direct API commands from the USS server, you must set several environmental variables in both the shell and the server address space. To do this, the USS server reads the file pointed to by the ENVFILE DD statement at USS server initialization. In this file, environmental variables and their values are defined. You must set the PATH variable to include the appropriate CCS for z/OS directories containing the CCS for z/OS commands. You must set the LIBPATH variable to point to the CCS for z/OS dynamic link library for CCS for z/OS direct API commands. The PATH variable may also be expanded to include the command directories for other frequently used USS-based applications. To ensure that the ENVFILE remains as simple as possible, we recommend that you use shell commands or scripts to add directories to the PATH variable.

You must also specify the CCS for z/OS variables that begin with CA in the ENVFILE. You can obtain the names and values from the PROFILE file in the base CCS for z/OS directory. Do not use $CAIGLBL0000 in the CA variables as a symbolic substitution value in the variable definitions in ENVFILE. Replace any occurrences of $CAIGLBL0000 with its actual value. The member USSENV00 in SYS1.OPS.CCLXCNTL contains a sample ENVFILE with variable definitions and comments tailored for the default CCS for z/OS directory name (/cai/tngfw).

OPSUSS JCL Changes

For the OPSUSS member:

- Change the STEPLIB data set names to the names specified during installation. If SYS1.OPSCCLXLOAD is in the linklist or LPALST, then you can remove it. The CCLXPLD load library should not be in the linklist or LPA since it is only used by the USS server and is an unauthorized PDSE. Change the name of this library to the name given at installation.
- The TCP/IP client data set must be allocated to the SYSTCPD DD if the following three conditions apply:
  - TCP/IP is installed on the system.
  - The TCP/IP client data set is not automatically allocated to every task.
  - The TCP/IP client data set name does not follow the dynamic allocation search sequence of TCP/IP.
- Otherwise, this DD statement may be eliminated.

  Note: If you are unsure of the correct action to take, then consult your TCP/IP Network Administrator.

- The ENVFILE data set must point to a sequential data set or PDS member that contains the names and values of environmental variables that will be set in the USS server address space and the UNIX shell that is attached by the server. Certain environmental variables must be set for the Event Management component of CCS for z/OS API calls and commands to function properly. Member USSENV00 of SYS1.OPS.CCLXCNTL contains a sample of these variables.

  Note: Variable values on each z/OS system may vary. Instructions for determining the correct variable values are contained in the sample member.

- The PARM field of the EXEC statement may contain an initial command to execute when the USS server starts. This command may be one of the following:

  - A USS shell command indicated by the keyword USSCMD and followed by the command text
  - An Event Management component of CCS for z/OS API command with the same syntax as the ADDRESS USS host command

  Do not alter the SERVER and SUBSYS keywords in the PARM field. The default command (USSCMD printenv) is a UNIX shell command that displays the values of the current environmental variables in the server shell after the ENVFILE has been processed.

Troubleshooting

If a USS server command receives a command not found return code, check the path designation on the command and the value of the PATH variable. You may use the USS command echo $VNAME to display the value of a specific environmental variable. You may also use the TSO ISHELL command to easily navigate the HFS or zFS in compatibility mode structure to confirm the existence of a command in the expected directory.

If all USS server direct API CCS for z/OS commands abend, check the LIBPATH environment variable value defined in ENVFILE and confirm that the file TNEMEVT2 exists somewhere in the directory paths specified by LIBPATH. Check the other CA variable values that contain directory name values and verify those as well. USS file names are always case sensitive.

If the USS servers are started using SUB=MSTR and they fail to completely initialize after the startup of the system, change the start command for EZAZSSI in the COMMNDxx member of the system parameter library to start EZAZSSI SUB=MSTR as well. Cancel or force (if cancel fails) any uninitialized USS servers.

The USS process monitoring component installs module OPUSPREX as a dynamic system exit at exit names: BPX_POSPROC_INIT, BPX_IMAGE_INIT, and BPX_PREPROC_TERM.

Use the z/OS commands D PROG,EXIT,... and SETPROG EXIT,... to display and manage exit modules at these exits. The dynamic exits are created by the OMVS kernel at initialization and may not exist at the time that CA OPS/MVS starts. The dynamic exit install module of the product waits in a subtask until OMVS defines the exits. The exit module is installed when the exits' names are detected.

Create VTAM Terminals for the EPI Component

The EPI component lets you create screen scraping automation scripts against any VTAM application that has LU2 access capabilities. Configuration of this component is determined by the automation applications that you create. Perform the following customizations to your VTAM configuration if the EPI component of CA OPS/MVS is needed.

To create VTAM terminals for the EPI component

1. Define EPI virtual terminals to VTAM by adding application definitions to SYS1.VTAMLST. The EPI requires one or more VTAM application definitions to be available to emulate real 3270 terminals.
2. Use the following statements to create a new member in SYS1.VTAMLST. Specify one APPL statement for each terminal.

```
majname   VBUILD  TYPE=APPL
xxxxnnnn  APPL    EAS=1,
                     PARSESS=NO,
                     MODETAB=modetab,
                     DLOGMOD=logmode,
                     PRTCT=vtampswd
xxxxnnnn  APPL    ...
          .
          .
          .
```

In the above statement, the DLOGMOD option specifies the name of the logmode table entry to be used when this terminal logs on to an external product. Use the name you would use for a real 3278 model 2, 3, or 4 type terminal that does not support structured fields. The logmode should not have the query bit on. EPI can override this name if you specify the LOGMODE keyword when issuing the EPI DEFINE or CHANGE command. EPI can override this name if you specify the LOGMODE keyword when issuing the EPI DEFINE or CHANGE command.

For information on valid values for other options in the above APPL statements, see the VTAM documentation.

3. Make the EPI VTAM APPLIDs active in VTAM when the EPI tries to enable virtual terminals. Usually, you activate APPLIDs at VTAM startup. If you defer activation until after VTAM becomes fully active, you can use the following VTAM operator command when you want to activate the EPI virtual terminal definitions:

   `VARY NET,ACT,ID=`*`majnode`*

   majnode

   Specifies the name of the member in SYS1.VTAMLST where the EPI APPL statements are stored.

   You can use the CA OPS/MVS AOF component to issue this command at the end of VTAM initialization, or use OPSVIEW to issue it.

Install the NetView Interface

Perform these steps to install the CA OPS/MVS NetView interface:

1. To gain access to the NetView unsolicited message stream, module OPNVEX11 must be relinked with a NetView exit alias name of DSIEX11. To add the DSIEX11 alias using SMP/E, apply usermod OPUM003 contained in library SYS1.OPS.CCLXCNTL(USEREX11). If necessary, copy exit module OPNVEX11 and alias DSIEX11 from SYS1.OPSCCLXLOAD to a library in your NetView STEPLIB concatenation. If you already have a DSIEX11 exit in NetView, then modify it to include the logic in the CA OPS/MVS-supplied exit. Copy exit DSIEX11 from SYS1.OPSCCLXLOAD to a library in your NetView STEPLIB concatenation.

   DSIEX11 resides, in source format, in SYS1.OPS.CCLXASM. The exit sends unsolicited messages to the master console, which in turn routes them through the subsystem interface where CA OPS/MVS can access and automate them.

   Important! If you decide to no longer use the DSIEX11 exit, then you can safely delete the alias name from the load library. The DSIEX11 exit is only an alias name of the OPNVEX11 module.

2. Copy SYS1.OPS.CCLXEXEC(OPSALERT) to a data set in your NetView DSICLD concatenation. This program is a NetView REXX EXEC.

3. Establish a connection between NetView and your MCS master console. The interface to issue NetView commands from CA OPS/MVS rules or REXX programs is the same interface that IBM provides to enable NetView commands to be issued from z/OS consoles.

   Use a NetView AUTOTASK command to create an association between the MCS master console and a NetView user ID. You can issue this command at a NetView terminal, or in the NetView initial CLIST member.

The command has this syntax:

```
AUTOTASK CONSOLE=consolenumber,OPID=operatorid
```

consolenumber

The MCS console number.

operatorid

The NetView operator ID to be associated with the MCS console. NetView operator IDs are defined in the DSIOPF member of the NetView parameter data set.

The CA OPS/MVS NetView interface assumes that you have established an association between a NetView operator ID and your MCS master console. Consider modifying your NetView start up CLIST to issue the AUTOTASK command for you when NetView starts.

4. Modify your NetView startup CLIST to issue the following command:

   ```
   OPSALERT NOTIFY
   ```

   This command issues a series of NPDA set recording filter (SRF) commands to give CA OPS/MVS access to NetView alert information. You can issue this command without restarting NetView after you complete step 2 of this NetView installation process.

5. Copy the message table entry in SYS1.OPS.CCLXCNTL(OPSAUTO) to your NetView automation message table. If you do not have a NetView automation message table, copy the OPSAUTO member to a data set in the NetView DSIPARM concatenation and issue the following command:

   ```
   AUTOMSG MEMBER=OPSAUTO
   ```

   To have this command invoked automatically at NetView startup, place it in the NetView start up CLIST.

   If you already have a NetView message table, copy the message table entry to the bottom of your existing message table and reactivate it with the AUTOMSG command.

6. Copy SYS1.OPS.CCLXEXEC(ALERT) to a library that CA OPS/MVS rules can access (that is a library in the SYSEXEC concatenation). Doing this enables CA OPS/MVS rules and programs to use the ALERT function.

7. CA OPS/MVS message rules can set or reset a bit in the MSG.AFLAGS variable.

   For more information, see the *AOF Rules User Guide*.

8. To use the subset of POI command processors that can run as NetView command processors, define each command processor in the DSICMD member of the NetView parmlib.

For example:

```
*-----------------------------------------------*
*     CA OPS/MVS NETVIEW CAPABLE POI COMMANDS  *
*-----------------------------------------------*
OPSGETV  CMDMDL  MOD=OPSGETV,TYPE=R,RES=Y,PARSE=N
           CMDCLASS=1,2,3
OPSGETVL CMDMDL  MOD=OPSGETVL,TYPE=R,RES=Y,PARSE=N
           CMDCLASS=1,2,3
OPSSETV  CMDMDL  MOD=OPSSETV,TYPE=R,RES=Y,PARSE=N
           CMDCLASS=1,2,3
OPSDELV  CMDMDL  MOD=OPSDELV,TYPE=R,RES=Y,PARSE=N
           CMDCLASS=1,2,3
OPSQL    CMDMDL  MOD=OPSQL,TYPE=R,RES=Y,PARSE=N
           CMDCLASS=1,2,3
*-----------------------------------------------*
*        AUTOMATE/MVS COMMAND ALIASES OF OPSMODE  *
*-----------------------------------------------*
GETVAR   CMDMDL  MOD=GETVAR,TYPE=R,RES=Y,PARSE=N
           CMDCLASS=1,2,3
GETVARL  CMDMDL  MOD=GETVARL,TYPE=R,RES=Y,PARSE=N
           CMDCLASS=1,2,3
SETVAR   CMDMDL  MOD=SETVAR,TYPE=R,RES=Y,PARSE=N
           CMDCLASS=1,2,3
DELVAR   CMDMDL  MOD=DELVAR,TYPE=R,RES=Y,PARSE=N
           CMDCLASS=1,2,3
SQL      CMDMDL  MOD=SQL,TYPE=R,RES=Y,PARSE=N
           CMDCLASS=1,2,3
OPSMODE  CMDMDL  MOD=OPSMODE,TYPE=R,RES=Y,PARSE=N
           CMDCLASS=1,2,3
```

If the load modules defined in the example above are not available in the system linklist or LPA, you must add a STEPLIB for the CA OPS/MVS load library to the NetView procedure JCL and the library must be APF-authorized.

Since multiple CA OPS/MVS subsystems may be active on one system, default routing of all command requests to a desired subsystem name can be accomplished by allocating a dummy data set with a ddname of OPS$*xxxx*, where *xxxx* is the subsystem name; you may use JCL or the NetView ALLOCATE command. OPSS is the default subsystem name. For example:

```
ALLOC FILE(OPS$OPST) DUMMY
//OPS$OPST DD DUMMY
```

The CA OPS/MVS security rules do not currently have access to the NetView user ID for security checking of global variable access. To permit global variable access to NetView command processors, you must enable a generic security rule for the NetView address space. For example:

```
)SEC OPSGLOBAL*
)PROC
  If Opsinfo('JOBNAME') = 'netview job name' Then
    Return 'ACCEPT'
  Else
    Return 'NOACTION'
)END
```

Install the NetView Operator Facility

Installing the NetView Operator Facility (NOF) requires you to make changes to both your CA OPS/MVS and NetView environments. You may want to consult the NetView systems programmer at your site for help with installing the NOF.

The NOF resides on the CA OPS/MVS distribution media. It uses the following libraries:

- The CA OPS/MVS sample rules library, OPS.CCLXRULB
- The CA OPS/MVS NetView CLIST library, OPS.CCLXCLS0

  Note: We also provide this library in variable block format.
- The CA OPS/MVS load library, OPSCCLXLOAD
- The CA OPS/MVS control library, OPS.CCLXCNTL

To install the NOF

1. Create a rule set to house the sample rules supplied with the NOF by performing one of the following steps:

   - Create a new rule set.
   - Copy all of the members from OPS.CCLXRULB into an existing rule set.

   If your site uses the CA OPS/MVS SECURITYRULESET parameter, copy OPNFSEC into your security rule set. OPNFSEC is a security rule that gives NetView access to CA OPS/MVS global variables. You can use the OPNFCPYR job in the OPS.CCLXCNTL data set to copy OPNFSEC and other NOF rules.
2. Copy the NetView REXX programs from OPS.CCLXCLS0 to a library in the DSICLD concatenation in NetView. You can use the OPNFCPYE job in the OPS.CCLXCNTL data set to do this.

   Note: If you only concatenate the CA OPS/MVS load library to the NetView STEPLIB concatenation, then the CA OPS/MVS DSIEX11 module (part of the former CA OPS/MVS NetView interface) gets control. See Installing the NetView Interface in this chapter.

3. Make the OPSCCLXLOAD library available to NetView. If your CA OPS/MVS load library is in the z/OS LNKLST, NetView already has access to it. Otherwise, you need to copy the following load modules named from OPSCCLXLOAD to your NetView STEPLIB library. You can use the OPNFCPYL job in the OPS.CCLXCNTL data set to accomplish this.

   OPNFSGLV enables NetView to set CA OPS/MVS global variables.

4. Include the entries from the OPNFATBL member of the OPS.CCLXCNTL data library in your NetView message automation table. These entries trap events that CA OPS/MVS is interested in. We recommend that you use the NetView %INCLUDE feature to include the OPNFATBL entries, because this method enables you to maintain the CA OPS/MVS table entries separately.

5. Configure a user ID called OPSMAIN on NetView so that OPSMAIN is a task that starts automatically when NetView starts. You can use an existing autotask if you change the OPNFATBL member to route messages to it.

   Note: Using a new autotask is preferable, because doing so enables you to use the NetView TASKUTIL command to track NOF resource consumption. The easiest way to create the autotask is to copy the autotask definition for AUTO1, which is a standard NetView autotask.

6. (Optional) If you want to use the NetView STATMON interface, modify the DSICMN member of the NetView parameter library (typically, DSIPARM) by removing comments from the statements that begin with the text SENDMSG. Activating these statements will cause the NetView status monitor to issue CNM094I messages whenever a managed resource changes state. You can control the volume of CNM094I messages by determining which types of resources should generate these messages.

7. Make sure that the NetView subsystem address space is active. This is required to generate NetView alerts. You can use the CA OPS/MVS System State Manager feature to manage this address space.

   Note: You can use the OPSNETV function of OPS/REXX to determine the status of the NetView subsystem address space. For more information about OPSNETV, see the *User Guide*.

8. Use NetView LOADCL commands to load the NOF REXX programs into storage. This enables NetView to use the in-storage copy of the program instead of having to get it from disk for every message and alert.

9. Modify the NetView startup procedure to issue the appropriate alert filtering commands. These commands are:

   NPDA SRF (set recording filter)

   Specifies which alerts you want to keep and filters out alerts you do not want. To enable all alerts to flow to NPDA, to be displayed on the NPDA screen, and to be automated by CA OPS/MVS, you must issue the following command in your NetView startup CLIST or after NetView is active:

   ```
   NPDA SRF AREC PASS DEFAULT
   ```

   NPDA SVF (set viewing filter)

   Specifies which alerts you want to see. To enable all alerts that CA OPS/MVS generates to appear on the NPDA display, issue the following command in your NetView start up CLIST or after NetView is active:

   ```
   NPDA SVF PASS DEFAULT
   ```

After you have completed the steps listed above, the NOF is ready to operate. When you activate your new NetView message automation table, the NOF will behave like your existing DSIEX11 module (that is, if your DSIEX11 module echoes unsolicited VTAM messages to the console, so will the NOF). At this point, you may want to set up your NOF parameters using the OPNOF command.

Set Up Interfaces to Tivoli OMEGAMON XE

The CA OPS/MVS AOF component can respond to exceptions detected by any or all of the Tivoli OMEGAMON XE on z/OS products. Currently, this means that CA OPS/MVS can interact with the following products:

- Tivoli OMEGAMON XE on z/OS
- Tivoli OMEGAMON for IMS on z/OS
- Tivoli OMEGAMON XE for CICS on z/OS
- Tivoli OMEGAMON XE for DB2 Performance Monitor/Expert on z/OS.

The Exception Analysis Process

One function that all Tivoli OMEGAMON XE on z/OS products have in common is *exception analysis*. Every *n* seconds, a Tivoli OMEGAMON XE on z/OS product analyzes the system that it is monitoring to detect exceptional situations, then reports these exceptions as messages on the OMEGAMON terminal. Exception analysis commonly detects many exceptions in a system that is running with no problems-so many, in fact, that they will not all fit on the physical screen.

Interface to Exception Analysis Process

The AOF cannot directly automate OMEGAMON exception messages because they are not routed through z/OS console support. Fortunately, all Tivoli OMEGAMON XE on z/OS products support a log file onto which they write a copy of their logical exception screen at the end of each analysis interval. The size of the OMEGAMON logical screen is one of its startup parameters (LROWS), and users typically set it to a size much greater than the number of lines on the physical screen. Thus, while important exception messages may not appear on the physical screen of an OMEGAMON for lack of room, they will fit on the OMEGAMON logical screen and therefore are written to the log file.

Interface to OMEGAMON

To establish an interface between CA OPS/MVS and OMEGAMON, insert an OxREPORT DD statement in the OMEGAMON JCL procedure that uses the SUBSYS keyword to identify CA OPS/MVS as the target of that file.

```
SEND OMEGAMON MVS EXCEPTIONS TO CA OPS/MVS
//OMREPORT     DD   SUBSYS=(OPSS,OMEGAMON,MVS),
//            DCB=(RECFM=FBA,LRECL=81,BLKSIZE=81)
               SEND OMEGAMON CICS EXCEPTIONS TO CA OPS/MVS
//OCREPORT     DD   SUBSYS=(OPSS,OMEGAMON,CICS),
//            DCB=(RECFM=FBA,LRECL=81,BLKSIZE=81)
               SEND OMEGAMON CICS EXCEPTIONS TO CA OPS/MVS;
               IDENTIFY SOURCE CICS SYSTEM
//OCREPORT     DD   SUBSYS=(OPSS,OMEGAMON,CICS,CICSTEST),
//            DCB=(RECFM=FBA,LRECL=81,BLKSIZE=81)
               SEND OMEGAMON IMS EXCEPTIONS TO CA OPS/MVS
//OIREPORT     DD   SUBSYS=(OPSS,OMEGAMON,IMS),
//            DCB=(RECFM=FBA,LRECL=81,BLKSIZE=81)
               SEND OMEGAMON DB2 EXCEPTIONS TO CA OPS/MVS
//ODREPORT     DD   SUBSYS=(OPSS,OMEGAMON,DB2),
//            DCB=(RECFM=FBA,LRECL=81,BLKSIZE=81)
```

The format for the JCL examples above is as follows:

```
//ddname DD SUBSYS =(ssid,OMEGAMON,type{,reportid})
```

ddname

Specifies the ddname associated with the file.

ssid

Specifies the four-character CA OPS/MVS subsystem ID to which these messages are routed (usually OPSS).

type

Identifies the specific Tivoli OMEGAMON XE on z/OS product. The *type* must be MVS, CICS, IMS, or DB2.

reportid

(Optional) Specifies a unique report ID you can use in a rule to identify the source of the message.

Potential Concerns

If you try to start a JCL procedure that has a DD card specifying SUBSYS=*name* and the subsystem name is not active, the START command fails with a JCL error. This should not be a problem because CA OPS/MVS generally comes up early in the IPL and stays running for the life of the IPL.

You can take CA OPS/MVS down and back up without stopping the OMEGAMON tasks that are feeding exceptions to CA OPS/MVS. CA OPS/MVS will continue handling the exceptions when it comes back up.

Provide OMEGAMON Exceptions

Ensuring that the correct output is being written to the *O*xREPORT file requires some OMEGAMON customization. Customization includes choosing thresholds and options to create and define a profile. Use the OMEGAMON User Profile Facility to customize these parameters.

When using the AOF to automate OMEGAMON exceptions, note the following:

1. CA OPS/MVS must be started before OMEGAMON.
2. An OMEGAMON session must be active to feed the exception event process. If you have a dedicated mode terminal next to the console that is always left on the exception analysis screen, use that terminal to provide the exception data. If that terminal often displays other screens, then you risk missing important exceptions when the operators use it for other functions. CA OPS/MVS can monitor exceptions only while the exception analysis screen is active.

   The simplest way to configure the interface is to have a dedicated session with exception analysis always active. However, this solution has two drawbacks, the first of which is mentioned in the previous paragraph. The second drawback is that it requires a locally attached 3270 device.

   An alternative solution is to use the OMEGAMON VTAM interface with an EPI logical terminal. This solution is more complicated to configure, but it eliminates both of the problems associated with a dedicated 3270 terminal. The EPI session is hidden, so no one can walk up and change the screen. No real 3270 terminal is required, since the EPI is used as a virtual 3270.
3. Check the LROWS parameter of the OMEGAMON started task JCL to ensure that all exceptions fit on the logical screen that is written to the *O*xREPORT file. The default value for the LROWS parameter is two times the physical screen minus one; the maximum value is 999.
4. All exceptions must be unboxed, either by setting the BOX parameters to NO for all exceptions or by turning boxes off in the installation or user profile. You cannot alter the default profile. You can set some control options with the .SET command and you can set some exception thresholds using the XACB command, the XSET command, or both. Use these commands on the actual exception analysis screen for testing, but for production usage, place them in the installation or user profile so that they execute at OMEGAMON startup.

   Note: OMEGAMON installation procedures and actual commands can vary from one platform to another. The commands referenced above may be specific to OMEGAMON for MVS. Consult the appropriate installation guide for the IMS, CICS, and DB2 releases.
5. Set the page limit for the OMEGAMON *O*xREPORT file to a high number. To do so, either specify .PLM 999999999 in a screen space or preferably use the PAGELIMIT option in the user profile.

6. The OMEGAMON logging facility must be turned on. You need to issue the LOGON command to OMEGAMON to tell it to write screens to the *O*xREPORT file.

   OMEGAMON 7.1.0 and OMEGAMON II Configuration for dedicated terminals:

   Create an initial screen space and enter the following commands on separate lines following the rules for creating OMEGAMON screen spaces (commands should start in column 2):

   ```
   OUTP REPORT
   DDNM OPREPORT      (or whatever DDNAME is used in proc)
   .LOGOUT
   .LOGON
   .FGO exscrn
   ```

   exscrn

   Specifies the name of the screen space containing the exception analysis command.

   OMEGAMON II Configuration for OMVTAM:

   a. Create an initial screen space and enter the following commands on separate lines following the rules for creating OMEGAMON screen spaces (commands should start in column 2):

      ```
      OUTP REPORT
      DDNM OPREPORT       (cannot be OMREPORT)
      .LOGOUT
      .LOGON
      .FGO exscrn
      ```

   b. Logon to OMVTAM:

      ```
      LOGON APPLID(OMVTAM) DATA('FSCR=yyyy')
      ```

      yyyy

      Specifies the name of the screen space containing the commands described above. The purpose of the initial screen space (in either dedicated or VTAM mode) is to configure the logging facility when OMEGAMON starts. The .FGO command then transfers control to the exception analysis screen space, which then remains on the screen and drives the exception analysis process on a regular interval (the OMEGAMON session must be in auto-update mode).

7. Invoke exception analysis through one of these commands: LEXSY (for OM), LXIMS (for OI), or LCXSY (for OC). Place the command in column 1 and be sure to prefix it with an L. The L tells OMEGAMON to label the exception by putting its four-character name on the screen in addition to the message. These exception names are the message IDs that CA OPS/MVS uses to invoke its OMEGAMON rules.

At this point, you should see OMEGAMON messages appearing in OPSLOG, and you can enable rules to execute in response to them. Each exception generates a message each time the screen is refreshed, so you may want to review your exception thresholds and your refresh time to ensure that you do not flood OPSLOG with unimportant messages. Use the CA OPS/MVS BROWSEOMG parameter to keep OMEGAMON messages from appearing in OPSLOG. If you set the BROWSEOMG value to OFF, you can audit the occurrence of OMEGAMON messages that execute OMEGAMON rules by including a SAY statement or an ADDRESS WTO host command that reports the text of the exception message processed in the rules.

If you are licensed to use the OMEGAMON Exception Logging Facility (XLF), then you may want to consider using the XLFLOG DD as an alternative to the OMEGAMON report file. The XLFLOG has the advantage that it does not repeatedly generate exception events to CA OPS/MVS every OMEGAMON cycle. If you choose to use XLF, then you must customize the OMEGAMON exception analysis values for persist and limit.

Install the MVS/QuickRef Interface

For OPSVIEW users who want to also use the ChicagoSoft MVS/QuickRef product interface under ISPF, you must provide access to the MVS/QuickRef load modules. To do this, place the modules in the LNKLIST, LPALIB, STEPLIB, or ISPLLIB concatenation.

It is strongly recommended that you specify the MVS/QuickRef database name in the MVS/QuickRef options module (QWIKOPTS). If you do not, then specify the name of the MVS/QuickRef database through the CA OPS/MVS QUICKREFDBASE parameter. For instructions on modifying the MVS/QuickRef options table, see the MVS/QuickRef documentation.

You can use a different MVS/QuickRef database for any user by allocating the desired database in the LOGON procedure of that user. Or, you can accomplish this dynamically through the TSO ALLOCATE command. For more information, see your MVS/QuickRef documentation.

Note: OPSVIEW users must have access to the MVS/QuickRef load modules. To provide this, place the load modules in the LINKLIST, LPALIB, STEPLIB, or ISPLLIB concatenation.

The QUICKREFTYPE product parameter should be allowed to default or be set to TSOHELP so that current CA OPS/MVS message information is extracted from the CA OPS/MVS help file rather than from the MVS/QuickRef database, which most likely will not match the version of the product you are running.

Verify the Availability of the OPSQW Command

If desired, verify that the OPSQW command is in the OPBOCMDS command table on the CA OPS/MVS distribution media. To add the OPSQW command, use ISPF/PDF option 3.9 so long as you are not using OPSLOG Browse at the same time.

Note: You can assign the OPSQW command to any appropriate PF key by using the ISPF KEYS command while you are in OPSLOG Browse.

Set up the Interface with CA 7 WA

The interface between CA OPS/MVS and CA 7 WA allows CA OPS/MVS to send commands to CA 7 WA and to process messages destined for the CA 7 WA Browse log.

Send Commands to CA 7 WA

There are two methods in which CA OPS/MVS can send commands to CA 7 WA. Following is a description of these two methods:

- **Method 1**-Issuing Commands through the ADDRESS CA7 Host REXX environment

  This method allows for a two-way interface where a CA OPS/MVS OPS/REXX program can issue a command to CA 7 WA and receive the command responses. Perform the following steps to implement this type of command interface:

  - Verify that CA 7 WA is at Release 3.3 or higher.
  - Ensure that CA GSS is active. This can be installed from the CCS for z/OS tape.
  - Add the ADDRESS CA7 IMOD to the CA GSS procedure. Contact CA 7 WA Technical Support to verify the following IMOD statement:

    ```
    ADDRESS CA7 CAL2X2WR 15 DETACH TYPE 0
    ```

  After all of the above requirements have been met, you can issue and receive command responses from an ADDRESS CA7 host REXX statement coded within CA OPS/MVS OPS/REXX programs. For an example of this, see member ADDRCA7 in the *hlq*.CCLXSAMP data set that is created during CA OPS/MVS installation.

- **Method 2**-Issuing Commands Through the OPS/REXX OPSCA7 Function

  This method uses the U7SVC routine to issue commands from CA OPS/MVS to CA 7 WA. With this method, command responses are not returned to the issuing CA OPS/MVS OPS/REXX rule or OPS/REXX program. No additional installation requirements are needed to use this method.

  For more information on the OPSCA7 function, see the *Command and Function Reference*.

Access the CA 7 Browse Log

You must add a data control module (DCM) to the ENF database to access the CA 7 WA Browse log.

To access the CA 7 WA browse log

1. Install CA 7 WA Release 3.3 or higher.
2. Set the CA OPS/MVS INITCA7 parameter to YES in the OPSSPA00 member. If you want CA OPS/MVS to generate ENF-related trace messages, then you must also set the CA OPS/MVS DEBUGENF parameter to YES. Additionally, depending on the volume of browse messages that CA 7 WA produces, you may need to tailor the default values of the CAIENFMAX and CAIENFRATE parameter. For more information on these parameters, see the *Parameter Reference*.
3. Add the DCM to ENF. Verify with CA 7 WA Technical Support that their SAMPJCL contains an L232DCM1 job.

   This job installs the CA 7 WA browse event. An ENF EVENT command listing all of the DCMs that are installed should display:

   ```
   DCM module name: CAL2DCM1 Description: CA 7 BROWSE EVENT Installed
    date: 01.010 time: 11:25:20
   ```

Configure CA OPS/MVS Web Center Monitor

The CA OPS/MVS Web Center Monitor provides a web user interface into System State Manager and an Alert Monitor. The Alert Monitor shows alerts that are generated by this product's ALERTMON REXX environment. For information about installation, configuration, and operation of this component, see the *CA OPS/MVS Web Center Monitor Installation Guide*.

How to Install and Configure CA OPS/MVS RESTful Web Services

The CA OPS/MVS RESTful web services server application provides an API that lets you access Relational Data Framework (RDF) tables and start REXX programs on OSF servers from any web-enabled client application.

You require the following resources before you install CA OPS/MVS Web Services:

- **Resource 1:** You need to know the high-level qualifier (HLQ) that is assigned to the following data set. The CA OPS/MVS distribution installed this data base, and you must have update access to the data set:

  [HLQ].OPS.CCLXCNTL

- **Resource 2:** You need to know the HLQ that is assigned to the following two data sets. The CA OPS/MVS distribution media installed these data sets, and you must have update access to them:

  – hlq.OPS.CCLXLOAD

  – hlq.OPS.CCLXPLD

To have access to CA OPS/MVS Web Services API with your z/OS system, follow these steps:

1. Install an Apache Tomcat web server on the same z/OS system where CA OPS/MVS is running:

   - Using the CA Common Services Component Tomcat (see page 170).
   - Using a Self-Configured Tomcat (see page 172).

   Note: CA Common Services provides a prepackaged Tomcat server. However, you can deploy the server application to any Tomcat regions that already exist.

2. Configure the web services server application in that Tomcat region. For example, you can configure Tomcat for HTTPS (HTTP Secure) (see page 173).
3. Complete the conversion for ASCII/EBCDIC conflicts (see page 178).
4. Complete the post-installation steps (see page 179) to verify that you configured the CA OPS/MVS Web Services properly.

Important! Currently, you must install the server application (and Tomcat) on all systems that you want to target with the web services API.

Using the CA Common Services Component Tomcat

The CA OPS/MVS web services API component is delivered as a Web Archive File (WAR), and this component requires a configured Apache Tomcat server. These instructions describe how to configure the web services API using a CA Common Component (CCS) version of Tomcat. However, if you want to use your own Tomcat region, see the z/OS V1R11.0 DFSMSrmm Implementation and Customization Guide (SC26-7405-10) for information about installing a WAR file.

Follow these steps:

1. Install CCS Tomcat.

   For information about how to install the CA Common Services Tomcat, see the CA Common Services for z/OS Installation Guide Release 14.1.

   Note: CA OPS/MVS web services requires the Tomcat Version 7.0.40 Update. To update Tomcat to the appropriate version, see this support document for RI59832.

2. Configure the Started Tomcat task.

   The [hlq].OPS.CCLXCNTL(OPSWS) member contains the sample STC that start CA OPS/MVS Web Services under CCS Tomcat. You must customize this STC to include the following values:

   SET PRODDIR

   The full USS path to the CCS Tomcat installation product directory.

   SET VERSION

   Enter either **70** or **76**. 70 indicates that you want to run the 32-bit version of JVMLDMxx (JZOS startup module). 76 indicates that you want to run the 64-bit version of JVMLDMxx.

   SET LIBRARY

   (Optional) The dataset on your local host that contains the version of JVMLDMxx indicated by SET VERSION. If JVMLDMxx is loaded in LPA/Linklist, you do not need SET LIBRARY. In this case, you can comment out the SET VERSION, and you can comment out the STEPLIB statement that follows.

   Additionally, you must update the following DD statement:

   - Verify that STDENV DD points to the opwebsvc.env environment file.

3. Set the configuration parameters.

   For Common Component Tomcat users, a configuration application has been provided. The JCL job is located in [hlq].OPS.CCLXCNTL(OPWBSVCT). This JCL, which runs the configuration script OPWBSVCG, must be updated to include the following:

   - A valid job card.
   - If applicable a valid SYSAFF value.

   You must set the following values in the STDENV ddname statement:

   CACCSDIR

   The USS path to CA Common Services Component

   CCLXHFSD

   USS Path to mounted HFS for CA OPS/MVS installation of CA OPS/MVS Web Services FMID: CCLXC21

   JAVAHOME

   Explicit full USS path to the IBM JDK home directory where you want to run CCS Tomcat

   Note: JAVAHOME is expected to point to a 64-bit JDK. If you select a 32-bit JDK, you must make additional changes to the opwebsvc.env file. For more details, see the comment section at the top of the opwebsvc.env file.

   OPSUBSYS

   Specify default OPS subsystem for web services requests.

   OPREXXLIB

   Specify the OPS CCLXEXEC library associated with the default OPS subsystem.

4. When submitted, the OPWBSVCT JCL runs the configuration script OPWBSVCG. The OPWBSVCG configuration script completes the following actions:

   - Creates and populates /distrib directory under CCS Tomcat.
   - Sets the CCS install path and Java JDK path into the environment file.

Using a Self-Configured Tomcat

If you want to load the CA OPS/MVS Web Services into your own Tomcat region, which is not a CA Common Services Component, see the Apache Tomcat download and install instructions.

Additionally, you must perform the following tasks manually:

1. Create a /distrib directory under the Apache Tomcat base installation directory..
2. Copy the following files under the /distrib directory from the USS-mounted files from the CA OPS/MVS Web Services CCLXHFS.
   - From OPWBSVCF to opwebsvc.config
   - From OPWBSVEN to opwebsvc.env
   - From OPWBSVPR to opwebsvc.prop
   - From OPWBSVRX to OPWShttp.rex.
3. Copy the WAR file under the Apache Tomcat webapps directory from OPWBSVWR to opsmvs.war.
4. Set the Tomcat path and Java JDK path into the environment file
5. Specify the USS path to the environment file in the STC PROC of your Tomcat region.

Configure Tomcat for HTTPS (HTTP Secure)

You can optionally configure your Tomcat server to use HTTPS instead of HTTP for user access. Because HTTPS includes SSL encryption, this option alleviates concerns about exposing the data in clear text on the network.

Apache Tomcat is an Open-source third-party product. Apache provides documentation to help you configure SSL for Tomcat with the Apache Tomcat SSL Configuration HOW-TO. You can use the *Apache Tomcat HTTP Connector Reference* to look up the definitions of the configuration properties that the HOW-TO document utilizes. You can also find other web tutorials that describe the configuration. Locate and follow the steps that are appropriate for the version of Apache Tomcat that your CA OPS/MVS release installed. There are three methods that you can use for your keystore file.

Method 1: Use an existing trusted certificate

Follow this step:

1. Use the keytool program to create the keystores, truststores, and certificates to achieve your desired security configuration.

   Note: For more information about trusted certificates, see the Apache Tomcat 7.0 on the Web (Apache Tomcat 7.0 SSL Configuration HOW-TO).

Method 2: Create and use your own self-signed certificate

Follow these steps:

1. Select the appropriate method to create a keystore that contains a self-signed certificate:
2. Execute the sample scripts that are provided in the {CCS_installation_dir}/OPS/distrib directory.

   For example:

   - Execute *makeks* script from USS command line (TSO OMVS).

     How to use OMVS (enter the following commands under TSO or ISPF):

     ```
     OMVS
     cd {CCS_installation_dir}/OPS/distrib
     makeks
     ```

   - Execute the *listks* script to verify that you created .keystore successfully.
   - (Optional) Execute "ls -la" to view that .keystore file.
3. Copy file *.keystore* file from {*CCS_installation_dir*}/OPS/distrib to your {*CCS_installation_dir*}/OPS/tomcat/conf.
4. Save the full path to the .keystore file under {CCS_installation_dir}/OPS/tomcat/conf. You have to specify this path later on the keystoreFile keyword that server.xml specifies.

Note: The JCL job [hlq].OPS.CCLXCNTL(OPWBSVMK) lets you execute the makeks and listks scripts in a batch environment. You must customize this JCL before it is submitted at your site.

Method 3: Complete the following steps to generate a keystore containing a self-signed certificate manually

Follow these steps:

1. Enter the following command from the USS command line (OMVS):
   ```
   cd {CCS_installation_dir}/OPS/distrib
   $JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA -keystore
   .keystore
   ```
   A series of prompts appears:
2. Specify a password, press Enter, and answer the following questions:
   - Remember the password that you specify here.
   - We recommend that you use the z/OS hostname where Tomcat runs for the CN value (common name), so that if you are prompted to accept the certificate, it is clear which server sent the prompt.
   - A keystore is created in your keystore directory with one self-signed certificate inside. The actual filename is *.keystore.*

     For example:

     ```
     Enter keystore password: tomcat
     Re-enter new password: tomcat
     What is your first and last name?
       [Unknown]:  localhost
     What is the name of your organizational unit?
       [Unknown]:  CA
     What is the name of your organization?
       [Unknown]:  CA
     What is the name of your City or Locality?
       [Unknown]:  Pittsburgh
     What is the name of your State or Province?
       [Unknown]:  PA
     What is the two-letter country code for this unit?
       [Unknown]:  US
     Is CN=localhost, OU=CA, O=CA, L=Pittsburgh, ST=PA, C=US
     correct?
     ```

After you use one of the three methods, continue with the following steps and complete the Tomcat configuration:

1. (Optional) Enter the following command from the USS command line (OMVS) and verify the contents of your keystore:

   ```
   cd {CCS_installation_dir}/OPS/distrib
   ```

   ```
   $JAVA_HOME/bin/keytool -list -keystore .keystore
   ```

   The results appear like the following example:

   ```
   Enter keystore password:
   Keystore type: jks
   Keystore provider: IBMJCE
   Your keystore contains 1 entry

   Alias name: tomcat
   Creation date: Feb 3, 2014
   Entry type: keyEntry
   Certificate chain length: 1
   Certificate[1]:
   Owner: CN=tomcat, OU=CA, O=CA, L=Pittsburgh, ST=PA, C=US
   Issuer: CN=tomcat, OU=CA, O=CA, L=Pittsburgh, ST=PA, C=US
   Serial number: 114822b8
   Valid from: 2/3/14 8:15 AM until: 5/4/14 9:15 AM
   ...
   ```

2. Within the Tomcat server.xml configuration file, modify the Connector element which has *port="8443"*.

   This port is the TLS connector. Specify a keystore file and a keystore password.

   a. Update the Apache Tomcat configuration parameters in the server.xml file as follows:

      - If you use the Tomcat that CA Common Services distributed, you can locate the server.xml configuration file under **{CCS_installation_dir}/OPS/tomcat/conf**.
      - If you use your own Tomcat server, you can locate the server.xml configuration file under {tomcat_home}/conf.
      - Understand that server.xml is an ASCII file. As such, you must use ISPF 3.17 to edit server.xml.

      A typical connector element for an SSL port appears like the following example:

      ```
      <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
             maxThreads="150" scheme="https" secure="true"
             clientAuth="false" sslProtocol="TLS" />
      ```

      Important! You *must* uncomment this connector element if it is commented. Comment blocks are defined in this file by a starting token of “<!--“ and an ending token of “-->

   b. Add the keystoreFile and keystorePassword keywords as follows:

      Figure 1

      ```
      <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
          maxThreads="150" scheme="https" secure="true"
          clientAuth="false" sslProtocol="TLS"
          sslEnabledProtocols=”TLSv1.2,TLSv1.1,TLSv1″
          keystorePass="tomcat"

      keystoreFile="{CCS_installation_dir)/OPS/distrib/.keystore"
      />
      ```

      We also recommend that you disable SSLv3 in your Internet browser and on your web clients before connecting to CA OPS/MVS web services. Disabling SSLv3 on either client side or server side will mitigate the vulnerability to cyber-attack due to recent compromises with the SSLv3 protocol.

      Note: To avoid the potential for cyber-attack, we recommend that you disable SSLv3 in your Apache Tomcat web server. Specify the sslEnabledProtocols attribute (see Figure 1) with only the TLS protocols listed. This step avoids usage of the older SSL protocols. You can find documentation about the sslEnabledProtocols attribute in the JVM documentation under method SSLSocket.setEnabledProtocols(). See the Oracle JDK documentation for Java 7 or Java 8.

If you want to use client certificate authentication, follow these basic steps:

- Generate a self-signed server-certificate on the server (Tomcat host).
- Download the server-certificate to the client.
- Import the server-certificate to the Java-keystore of the client.
- Generate a self-signed client-certificate on the client.
- Upload it to the server.
- Import the client-certificate to the Java-keystore of the Tomcat server.
- Configure Tomcat to use this keystore.
- If you want to ensure that the client connections are also authorized by certificate, set clientAuth="true" in server.xml.

Note: For specific details about accomplishing these tasks, see the Apache Tomcat 7.0 SSL Configuration HOW-TO Java™ Secure Socket Extension (JSSE) Reference Guide.

3. Add the following lines before </web-app> at the end of the web.xml file that is located in {tomcat_dir}/conf:

```
<security-constraint>
    <web-resource-collection>
           <web-resource-name>Tomcat</web-resource-name>
           <url-pattern>*.html</url-pattern>
    </web-resource-collection>
    <user-data-constraint>
           <transport-guarantee>CONFIDENTIAL</transport-guarante
    e>
    </user-data-constraint>
</security-constraint>
```

4. After you restart Tomcat, you now have access to your URIs over both a Secured connection and with an unencrypted connection.

   - Test your secure connection by specifying URIs with the HTTPS network scheme and the 8443 port.
   - For example, use this format: https://*your-host*:8443/opsmvs/web/tables (substituting your own hostname for *your-host)*.

   This test assumes that you retained the default TLS port number of 8443.

5. (Optional) After you tested your secured connection successfully, you can disable (comment out) the unencrypted Connector element, which has port 8080. This procedure prevents Tomcat from serving any requests over the unencrypted connection.

Complete the Conversion for ASCII/EBCDIC Conflicts

CA OPS/MVS Web Services store several types of files in your HFS or zFS in compatibility mode. Web Services distribute some of these files in binary format, and it distributes some files in EBCDIC. The following table provides some examples:

File Type	Code
.war	binary
.zip	binary
.txt	EBCDIC
.rex	EBCDIC
.conf	EBCDIC
.sh	EBCDIC
.env	EBCDIC
.prop	EBCDIC

CA OPS/MVS Web Services distributes its configuration files in EBCDIC to ease viewing and editing in the USS environment. Apache Tomcat can recognize either format.

If you want both configure Tomcat and keep all the configuration files in ASCII, you first convert the files that CA OPS/MVS distributes from EBCDIC to ASCII. We recommend that you rename and save the EBCDIC version of the file for easy reference to its contents.

You can use the following methods to complete the conversion:

1. Enter **iconv -f IBM1047 -t ISO8859-1**.

   Note: For syntax and usage, see the USS help in *man iconv*.

2. Use FTP to transfer the file to a PC with conversion from EBCDIC to ASCII. Then, resend the file to the host as a binary format file.

   You may have other methods to convert files to ASCII. Use the best method for your environment.

Note: Files that are binary in nature, such as .zip files, are neither ASCII or EBCDIC. They are binary, and you transmit these files as binary.

Complete the Post-Installation Steps

To test and make sure that you configured CA OPS/MVS Web Services properly, go to the following location in your web browser:

`http://mvshostname:port/opsmvs`

This URL displays a splash page with links pointing to downloadable zip files containing sample client applications.

Note: For more information, see the RESTful Web Services chapter in the *Administrative Guide*.

If you plan to use the web services osfrexx resource to execute REXX commands on your OSF servers, you must make one additional change to the opwebsvc.env file:

- Modify the statement “export OPS_REXXLIB=<opshlq.CCLXEXEC>” to point to your CA OPS/MVS Deployments SYSEXEC library. Use of the osfrexx web resource requires that this library is set correctly.

Configure Hardware Services (HWS)

To activate CA OPS/MVS Hardware Services (HWS) initialization parameters must be appropriately set. Also, since HWS interfaces with the Hardware Interface Service , the Hardware Interface Service, must be available on the system where CA OPS/MVS is running.

HWS Parameters

To activate HWS, set the INITHWS parameter to YES:

```
OPSPRM('SET','INITHWS','YES')
```

Setting INITHWS to YES activates the general CA OPS/MVS Hardware Services component. This component includes both the use of address HWS host command environment, and the processing of HWS hardware events. In order to activate hardware event notification, parameter HWSRULES must also be set to YES:

```
OPSPRM('SET','HWSRULES','YES')
```

When HWSRULES is set to YES, HWS provides hardware event notifications in the form of API events. These events can be automated through)API rules.

HWS can be activated and deactivated at anytime through the initialization parameters.

For detail information on hardware event types and associated variables that are available through HWS event notification, see "Hardware Event API Rules" in the *CA OPS/MVS AOF Rules User Guide*.

Note: For the address HWS environment, INITHWS is the only required parameter. We recommend that HWSRULES is set to YES as the response to actions taken through this address environment are returned as event notifications. For more information on the address HWS environment, see "Address HWS Commands" in the *CA OPS/MVS Command and Function Reference*.

Note: If you are changing the value of INITHWS from NO to YES after CA OPS/MVS initialization, you must issue the z/OS command: MODIFY OPSS,RESTART(HWS) for the new value to take effect (where OPSS is the CA OPS/MVS subsystem name).

For more information on the INITHWS and HWSRULES parameters, see "Hardware Services (HWS)" in the *CA OPS/MVS Parameter Reference Guide.*

Since hardware events are presented as OPS)API events, the OPS API interface must also be activated to receive the hardware events. To activate the OPS API interface, set the OPS APIACTIVE parameter to YES:

```
OPSPRM('SET','APIACTIVE','YES')
```

For more information on the APIACTIVE parameter, see "Application Programming Interface Parameters" in the *CA OPS/MVS Parameter Reference Guide*.

For general information on coding API rules and specific information for coding hardware event API rules, see "Generic Event Application Program Interface" in the *CA OPS/MVS AOF Rules User Guide.*

To have CA OPS/MVS generate HWS-related trace messages, set the DEBUGHWS parameter to ON:

```
OPSPRM('SET','DEBUGHWS','ON')
```

Setting up the Hardware Interface Service

HWS utilizes the Hardware Interface Service. The Hardware Interface Service provides CA Technologies products with a common interface/API for accessing hardware functions. CA OPS/MVS interfaces with the Hardware Interface Service to implement its HWS functions. Therefore, the Hardware Interface Service must be configured and started on the system where CA OPS/MVS is running in order for HWS to provide services such as hardware event notification.

Install and Configure the Hardware Interface Service

See the *CA Common Services for Z/OS Installation Guide* for information about installation, configuration, and operation of the Hardware Interface Service.

Configure Linux Connector Interface (LXC)

The CA OPS/MVS Linux Connector Interface (LXC) provides the ability to automate unsolicited VM and Linux messages. These messages are forwarded to CA OPS/MVS through the Linux Connector Component product running on the same system. The Linux Connector Component also provides command and response processing to the connected VM and Linux systems. LXC does not directly communicate with the VM and Linux guest systems.

Linux Connector Component Set Up

See the *CA Common Services for Z/OS Installation Guide* for information about installation, configuration, and operation of the Linux Connector Component.

Note: The CA OPS/MVS main address space must have read access to the TCP/IP data set (hlq.TCPIP.DATA) to determine the correct TCP/IP started task with which it will communicate. The data set must be allocated to ddname SYSTCPD automatically by the system, explicitly in the OPSSPA00 REXX program that is run at productname> initialization, or through a JCL statement in the OPSMAIN started procedure.

LXC Parameters

To activate LXC, set the INITLXC parameter to YES:

```
OPSPRM('SET','INITLXC','YES')
```

Setting INITLXC to YES causes the LXC subtask to connect with Linux Connector Component and begin to receive unsolicited messages from connected VM and Linux systems.

The parameters LXCONMSG and LXCONCMD must match the values of the Linux Connector Component parameters MSGTOKEN and CMDTOKEN respectively. These parameter values are the name portion of z/OS Name/Token pairs that contain the IP port numbers for the unsolicited message and command processing IP servers. When the default CA OPS/MVS parameter values, do not match the Linux Connector Component values set the parameters using statements like:

```
OPSPRM('SET','LXCONMSG','CAMSGTOKENVAL:')
OPSPRM('SET','LXCONCMD','CACMDTOKENVAL:')
```

In order to activate the unsolicited message AOF API rule events, set the parameter LXCRULES to YES:

```
OPSPRM('SET','LXCRULES','YES')
```

Implement automation and tracking of VM and Linux systems by coding one or more AOF API rules with the special message ID values attached to the VM and Linux messages. For more detailed information about the Linux Connector API event types and associated variables, see the Linux Connector API Rules topic in the *CA OPS/MVS AOF Rules User Guide*.

Set the BROWSELXC parameter to YES to include the Linux Connector Component unsolicited message events that are passed to CA OPS/MVS in OPSLOG. Set the parameter as follows:

```
OPSPRM('SET','BROWSELXC','YES')
```

To have CA OPS/MVS generate LXC-related trace messages, set the DEBUGLXC parameter to ON:

```
OPSPRM('SET','DEBUGLXC','ON')
```

LXC can be activated and deactivated at any time through the initialization parameters.

Note: When changing the value of INITLXC after CA OPS/MVS initialization, issue the z/OS command: MODIFY OPSx,RESTART(LXC) for the new value to take effect. OPSx is the CA OPS/MVS subsystem name.

For more information on the INITLXC, LXCRULES and other LXC parameters see "Linux Connector Interface Related Parameters" in the *CA OPS/MVS Parameter Reference Guide.*

Direct Generic Data Set Output

You can direct output from data sets such as log files to CA OPS/MVS for processing by the AOF component. To do this, use the generic data set interface (GDI). With this interface enabled, CA OPS/MVS sees each record written to the generic data set as a message event, which executes AOF message rules.

Define a Generic Data Set

To establish a generic data set interface with CA OPS/MVS, specify JCL that identifies CA OPS/MVS as a target for the output. Specify this JCL as follows:

```
//ddname DD SUBSYS =(ssid,OPSDSN{,color}{,reportid}{,posmgid})
```

ddname

The *ddname* associated with the file.

ssid

The four-character CA OPS/MVS subsystem ID that receives generic interface messages (usually OPSS).

color

(Optional) Specifies the color in which generic data set messages appear in OPSLOG.

Valid values: GREEN, BLUE, RED, WHITE, PINK, YELLOW, or TURQ

reportid

(Optional) Specifies a unique report ID that an AOF rule can use to identify the source of the message.

posmsgid

(Optional) Specifies either of the following:

- The numeric starting position in the text of each record at which CA OPS/MVS is to begin its scan for a message ID.
- A character string that is to be used as the message ID for all the records in this data set.

When the numeric starting position is longer than any particular record in the file, the message ID scan starts at the beginning of that record.

Example 1: Send Messages to Subsystem OPSS

- Messages are blue in the OPSLOG
- MSG.COLOR is blue
- The report ID is PERFRPT
- The AOF starts scanning for the MSGID in the first column of each record.

```
//DD1      DD     SUBSYS=(OPSS,OPSDSN,BLUE,PERFRPT)
```

Example 2: Send Messages to Subsystem OPST

- Messages are pink in the OPSLOG
- MSG.COLOR is pink
- The report ID is ESPLOG
- All messages from this data set have a MSGID of ESPMSG.

Note: This message ID is not inserted into the message.

```
//DD2    DD   SUBSYS=(OPST,OPSDSN,PINK,ESPLOG,ESPMSG)
```

Example 3: Send Messages to Subsystem OPSS

- Messages are red in the OPSLOG
- MSG.COLOR is red
- The report ID is MYLOG
- The AOF starts scanning for the MSGID in column 22 of each record.

Note: This technique is useful for log files that either have a fixed length, time stamp, or both at the beginning of each record, or some other fixed length prefix followed by the message ID.

```
//DD3    DD   SUBSYS=(OPSS,OPSDSN,RED,MYLOG,22)
```

Generic Data Set Interface Guidelines

Consider the following guidelines before using the generic data set interface:

- A CA OPS/MVS security event occurs every time a subsystem data set directed to CA OPS/MVS is opened. You can write a security rule to allow or disallow the opening of the data set.
- Started tasks, batch programs, or TSO programs can use the generic data set interface.
- The CA OPS/MVS subsystem specified as *ssid* must be active when the job or started task is started.
- The application writing to the data set must use either standard QSAM or BSAM.

- There are three possible techniques for selecting message IDs from each file:
  - If you do not specify the posmsgid parameter, the application must place a message ID as the first token of each record. This token can contain from one to ten characters.
  - If you do specify the posmsgid parameter and it is a non-negative numeric value, CA OPS/MVS begins scanning the message ID at or immediately following that position in each record.
  - If neither of the above techniques is practical, you can assign a unique one to eight character non-numeric MSGID for all records from the file in the posmsgid parameter. For examples of each of these three techniques, see the previous section.
- Messages exceeding 128 characters are truncated.
- If CA OPS/MVS terminates, generic data set interface messages are no longer automated. If you restart CA OPS/MVS, the messages will again be sent to the AOF. You do not need to stop and restart the application if you use the same CA OPS/MVS subsystem ID when restarting CA OPS/MVS.

Set Up Interface to CA MIC

The interface between CA OPS/MVS and CA MIC provides the following capabilities:

- The CA OPS/MVS subsystem can issue cross-system commands through the CA MIC subsystem by using the OPSCMD command processor or the ADDRESS OPER OPS/REXX host command environment to any system in the MICplex. The solicited command response messages are returned to the command issuer and may optionally be recorded in the OPSLOG.
- The CA OPS/MVS subsystem can receive unsolicited messages from any system in the MICplex and record them in the OPSLOG.
- AOF rules can recognize and interrogate fields from solicited and unsolicited CA MIC imported messages and take action based on the message data presented.

The MICplex can consist of up to 128 systems configured in a single sysplex, non-sysplex systems, systems in multiple sysplexes, or VM systems where CA MIC for VM is running as a service machine. Messages from up to 128 systems can now be forwarded through CA MIC to any CA OPS/MVS subsystem.

When all of your systems are in a single sysplex, you can use sysplex services to perform most of these functions. However, the CA MIC message filtering criteria are superior to those provided by sysplex. If you have licensed the Multi-System Facility (MSF), you can perform these functions by using the SYSTEM keyword of OPSCMD and ADDRESS OPER and by writing AOF rules to forward messages from one system to another.

Configure the Interface

For instructions on how to configure CA MIC to do the following, see the *CA MIC Message Sharing Systems Programmer Guide*:

- Use the LINK command to enable the cross-system command and response feature
- Use the COLLECT command to have CA MIC import unsolicited messages to local CA OPS/MVS subsystem

If you only intend to use the CA MIC cross-system command interface and do not want to automate the command responses or have them displayed in OPSLOG, then no CA OPS/MVS configuration is required.

If you intend to display CA MIC imported messages in OPSLOG, you must set the BROWSEMESSAGES parameter to MVSGLOBAL. If you intend to have CA MIC imported messages automated by AOF rules, you must set the AOFMESSAGES parameter to MVSGLOBAL.

Note: Changing this parameter may have a major impact on your automation.

In most sites that run both products, CA OPS/MVS is usually started prior to CA MIC. However, if CA MIC is started before CA OPS/MVS and the CA OPS/MVS SSIMSG parameter is set to a value of YES, you will find that the CA MIC internal encrypted messages (all of which have message IDs that start with GCM/) appear in the OPSLOG. We recommend that you always start CA OPS/MVS before CA MIC. However, if that sequence does not fit into your automation scheme, use the following sample rule (which has also been included in member GCM of the OPS/MVS sample rules library) that demonstrates how to exclude all the GCM messages from the OPSLOG.

Note: You should not attempt to suppress these GCM/ messages or you will impact the functionality of CA MIC.

```
)MSG GCM/* NOOPSLOG
)PROC
return
```

Identify Messages Received from CA MIC

When writing AOF rules you need to be aware that CA MIC imported messages have the following attributes:

- The MSG.MIC environmental variable is set to 1.
- The MSG.REISSUE environmental variable is set to 1.
- The MSG.SYNA environmental variable contains the name of the system from which the message originated.
- The MSG.JOBNM environmental variable contains the job number of the task that originally issued the message. This field contains a value of NONE when the originating task was a z/OS subsystem or a VM application, which did not have a job number.
- The MSG.JOBID environmental variable contains the job number of the task that originally issued the message. This field contains the MVS subsystem name or the VM application name when the originating task was a z/OS subsystem or a VM application, which did not have a job number.
- The MSG.JOBNAME environmental variable contains the name of the task that originally issued the message.

CA MIC presents imported messages to CA OPS/MVS using the above standards, regardless of any CA MIC message editing parameter values in effect on any system. In other words, CA MIC consistently presents CA OPS/MVS with original message data regardless of the CA MIC message editing that may have taken place on a given system based on the CA MIC MIMINIT EDITMESSAGE, SYSNAME, SYSTYPE, and JOBID parameters.

When the local CA MIC subsystem is directing imported messages to the local CA OPS/MVS subsystem, it is important that AOF rules interrogate the MSG.SYNA, the MSG.REISSUE environmental variables, or both to identify the systems from which messages are originating. Otherwise, these rules may misinterpret CA MIC imported messages as being from the local system, which may result in unpredictable or incorrect actions.

The following sample AOF MSG rule allows imported CA MIC messages to be easily identified in OPSLOG. Filtering on the USER column with a value of MIC limits the display to CA MIC imported messages. The display can also be limited to those messages imported from a particular system by filtering on the COLOR column.

Note: This logic colorizes all imported messages from systems XE13, XE12, and XE07. If you only want to colorize the CA MIC imported messages, the select statement needs to be subject to the MSG.MIC = 1 condition. If you decide to implement this rule, we suggest that you merge the rule logic into any existing MSG * rules that you may have.

```
)MSG *
)PROC
if MSG.MIC = 1 then
  MSG.USER = "MIC"
select
  when MSG.SYNA = "XE13" then
    MSG.COLOR = OPSCOLOR("TURQ")
  when MSG.SYNA = "XE12" then
    MSG.COLOR = OPSCOLOR("YELLOW")
  when MSG.SYNA = "XE07" then
    MSG.COLOR = OPSCOLOR("PINK")
  otherwise
    nop
end
```

Install the Optional CA 7 Browse Log Messages Feature

The optional CA 7 Browse Log messages feature allows you to perform automation on CA 7 messages that would typically only appear in the CA 7 log.

In the CA OPS/MVS OPSLOG, CA 7 messages appear as MSG-type events and may cause MSG rules to execute. These MSG-type events have an exit type of CA 7.

How to Install ENF Services

CA OPS/MVS can monitor and automate messages from CA 7 WA that are destined for the CA 7 Browse Log data set through the Common Services portion of Event Notification Facility (ENF) services of CCS for z/OS, or CAI ENF.

You must install the ENF services to activate this feature. For details, see the *CCS for z/OS* documentation.

Note: For information on the CCS for z/OS component required to run the CA 7 Browse Log, see the appendix "CCS for z/OS Component Requirements (see page 195)."

Parameter for Use with CAI ENF

You need to set the following parameter for CAI ENF:

INITCA7

Enables CA OPS/MVS to detect CA 7 browse ENF events.

For more information about this parameter, see the *Parameter Reference*.

Multiline CA 7 WA Messages

Some CA 7 WA messages are multiline and may present problems in the message rule specification process. Because the primary line of a CA 7 WA message is the only line that has a valid message ID, CA 7 WA cannot ensure that the secondary lines of a multiline message will always follow the primary line. Therefore, CA OPS/MVS assigns the message ID CA7BRWSE to all secondary lines, ensuring that you will not receive invalid information when message lines intermix. The automation rule or program determines and validates secondary lines through the use of temporary and permanent global variables.

Adding a Browse Event DCM

CA OPS/MVS requires the addition of a DCM to CAI ENF for CA 7 Browse Log events.

For information about adding a browse event DCM to CAI ENF for CA OPS/MVS, see the *CA 7 Workload Automation Interfaces Guide*.

Set up the z/OS Automatic Restart Management Facility

z/OS systems include a feature called Automatic Restart Management (ARM), which-in the event of a system failure-provides automatic restarting of jobs and started tasks on the same system or, optionally, across any system in a sysplex. To use the ARM facility, a job or an STC must register with ARM using a sysplex unique element name and it must notify ARM when it is fully initialized and ready to perform work. If the task terminates without issuing a deregister call, ARM restarts the task using policy guidelines defined by the installation in the ARM couple data set. The policy can specify the order of the restarts for the tasks that depend on other tasks as well as the frequency, time, and system resource constraints for restarts. For a detailed description of ARM, see the IBM documentation.

The use of ARM by CA OPS/MVS is limited to restarting the product on the same system on which it was running when it unexpectedly terminated due to a severe error condition such as excess message rate. The STOP command causes CA OPS/MVS to deregister with ARM as part of the normal shutdown. If CA OPS/MVS is canceled or forced from the system, it will not restart unless the ARMRESTART operand is also specified on the z/OS CANCEL or FORCE command.

ARM rules for the AOF are available to control the restarting of other jobs or started tasks. Using the dynamic exit facility of z/OS, CA OPS/MVS installs an ARM restart event exit at the IXC_ELEM_RESTART exit point. Using the data from the parameter list that was passed to this exit (IBM macro IXCYERE), an ARM event is created and passed to the AOF. For AOF ARM rules to execute, the INITARM and ARMRULES parameters must be set to YES.

Tailor AOF ARM Rules

You can enable ARM to let CA OPS/MVS restart itself after a failure by tailoring the AOF ARM rules.

To tailor the AOF ARM rules

1. Consult with the systems programming group at your site to determine how ARM is being used. At a minimum, an ARM couple data set with at least a default policy must exist and be accessible to all sysplex systems on which ARM is to be used. The following command, which starts ARM, must be issued during system initialization:

   ```
   SETXCF START,POLICY,TYPE=ARM
   ```

   To display the status of ARM, use the following command:

   ```
   D XCF,ARMSTATUS,DETAIL
   ```

   If CA OPS/MVS is to use ARM to restart itself after a failure, you should determine a sysplex unique element name and, optionally, an element type. Tailor the ARM policy based on the restart criteria you desire.

   CA OPS/MVS will only restart on the system on which it is running since other copies of the product are already active on the other sysplex systems.

2. Set the required ARMELEMNAME parameter and optional ARMELEMTYPE and ARMELEMASSOC parameters to the desired values in the OPSSPA00 REXX program. These parameters can only be set at this time.

   Example:

   ```
   var = OPSPRM('SET','ARMELEMNAME','OPSMVSSYSA')
   ```

3. Set the INITARM and ARMRULES parameters in the OPSSPA00 REXX program.

   Examples:

   ```
   var = OPSPRM('SET','INITARM','YES')
   var = OPSPRM('SET','ARMRULES','YES')
   ```

   Note: ARMRULES can be changed at any time.

   When CA OPS/MVS starts, the message OPS0311I/OPS0312E is issued to indicate the status of each ARM call for the product. For the return codes and error condition descriptions, see the IBM documentation.

4. If the INITARM parameter was set to YES for AOF ARM rules, the message OPS0310I is displayed for the installation of the OPMVAREX dynamic exit module at the IXC_ELEM_RESTART MVS exit point. This exit remains active even after the product terminates, and, when the product restarts, it will be reclaimed by the original product subsystem that loaded it. To display the status of the exit module, issue this z/OS command:

   ```
   D PROG,EXIT,EXITNAME=IXC_ELEM_RESTART,DIAG
   ```

   You can also control the exit manually with this z/OS command:

   ```
   SETPROG EXI,ADD/MODIFY/DELETE,EXITNAME=IXC_ELEM_RESTART,MODNAME=OPMVAREX,...
   ```

   If the OPMVAREX exit module is not properly installed, AOF ARM rule events will not occur. If the exit module is modified by maintenance and a new copy must be reloaded, the following CA OPS/MVS command will deactivate the exit module, reload the new version of the module, and reactivate the exit:

   ```
   F OPSS,RELOAD(OPMVAREX)
   ```

If the exit fails to install, set the DEBUGDYN parameter to YES in the OPSSPA00 REXX program and examine the messages in OPSLOG to determine the reason for the failure and the return codes. The codes are explained in the IBM publication *Authorized Assembler Services Reference ALE-DYN* in the section about the CSVDYNEX macro.

Appendix A: System Preparation Checklist

This section contains the following topics:

Record Tasks

This checklist provides an easy way for you to record and check off tasks that you perform before installing CA OPS/MVS.

- Valid C LMP Key Certificate

  Yes_____ No _____

- TSO/E-any IBM-supported release

  Yes_____ No _____

- IMS 8.1, 9.1, or 10.1 (if installed and the IOF optional component is licensed)

  Yes_____ No _____

- CICS/TS Version 2.3, 3.1, or 3.2 (if installed and the COF optional component is licensed)

  Yes_____ No _____

- CA ACF2 or CA Top Secret (if installed)-any CA-supported release

  Yes_____ No _____

- z/OS subsystem consoles generated

  _____

- Enough ECSA available (500 KB)?

  Yes_____ No _____

- DASD space for program libraries (270 cylinders)

  Volser: _____

- OPSLOG Browse data (435*BROWSEMAX)

  Note: Calculate this number after you have installed CA OPS/MVS. For more information on the BROWSEMAX parameter, see the *Administration Guide* and *Parameter Reference*.

  Volser: _____

- Global variable checkpoint (10 cylinders)

  High-level qualifier: _____

- Data set naming standards

  High-level qualifier: _____

- Rule data set prefix

  High-level qualifier: _____

- Program access

  STEPLIB/Linklist: _____

- APF authorization

  Yes_____ No _____

- TSO command authorization

  Yes_____ No _____

- Security IDs for started tasks

  _____

- TSO OPER authority

  Yes_____ No _____

Appendix B: CCS for z/OS Component Requirements

This section describes the CA Common Services for z/OS components and their corresponding FMIDs that are required by CA OPS/MVS to perform various functions. For more complete and up-to-date information, see Installation Dependencies in the chapter "System Requirements" in the *CCS for* z/OS *Getting Started Guide*.

This section contains the following topics:

CA LMP (License Management Program)

The following CCS for z/OS components are required to validate base product licensing for CA OPS/MVS.

CAS9C00 or CAS9E00 or CAS9E10

Specifies the CAIRIM component

CAW1C00 or CAW1E00 or CAW1E10

Specifies the CAIENF component

CAW4C00 or CAW4E00 or CAW4E10

Specifies the CAICCI with SSL component

Interface to IBM Health Checker

The following CCS for z/OS component is required for the CA OPS/MVS interface to the IBM Health Checker.

CEF5E00 or CEF5E10

Specifies the CA Health Checker Common Service component

Note: For information on additional setup and configuration steps that must be completed, see the CCS for z/OS documentation.

ADDRESS CA7

The following CCS for z/OS components are required to run ADDRESS CA7 on CA OPS/MVS.

The FMIDs based on CCS for z/OS r12:

CCF3410 or CCF3E00

Specifies the CA-GREXX component

CBYS280 or CBYSE00

Specifies the CA-GSS component

ADDRESS CASCHD

The following CCS for z/OS components are required to run ADDRESS CASCHD on CA OPS/MVS.

CCF3410 or CCF3E00

Specifies the CA-GREXX component

CBYS280 or CBYSE00

Specifies the CA-GSS component

DASD Requirements for Program Libraries

CA OPS/MVS requires 60 3390 cylinders, either in your libraries or as private libraries.

ADDRESS JOBTRAC

The following CCS for z/OS components are required to run ADDRESS JOBTRAC on CA OPS/MVS.

The FMIDs based on CCS for z/OS r11 SP8:

CCF3410 or CCF3E00

Specifies the CA-GREXX component

CBYS280 or CBYSE00

Specifies the CA-GSS component

Automation Measurement Environment

The following CCS for z/OS component is required to run the Automation Measurement Environment (AME) on CA OPS/MVS.

CAF3C00 or CAF3E00

Specifies the CA-C Runtime component

Interface to CA Automation Point

The following CCS for z/OS components are required for the interface between CA OPS/MVS and CA Automation Point.

CAS9C00 or CAS9E00 or CAS9E10

Specifies the CAIRIM component

CAW1C00 or CAW1E00 or CAW1E10

Specifies the CAIENF component

CAW5C00 or CAW5E00 or CAW5E10

Specifies the CAIENF/DB2 component

CAW4C00 or CAW4E00 or CAW4E10

Specifies the CAICCI with SSL component

Note: For information on additional setup and configuration steps that must be completed, see the CCS for z/OS documentation.

CA 7 Browse Log Interface

The following CCS for z/OS components are required to run the CA 7 Browse Log interface on CA OPS/MVS.

CAS9C00 or CAS9E00 or CAS9E10

Specifies the CAIRIM component

CAW1C00 or CAW1E00 or CAW1E10

Specifies the CAIENF component

CAW5C00 or CAW5E00 or CAW5E10

Specifies the CAIENF/DB2 component

CA Service Desk Interface

The following FMIDs are required for the interface between CA OPS/MVS and CA Service Desk.

CAS9C00 or CAS9E00 or CAS9E10

Specifies the CAIRIM component

CAW1C00 or CAW1E00 or CAW1E10

Specifies the CAIENF component

CDYFC00 or CDYFE00 or CDYFE10

Specifies the CAISDI/med and CAI/soap components

CAW4C00 or CAW4E00 or CAW4E10

Specifies the CAICCI with SSL component

For information on additional setup and configuration steps that must be completed, see the CCS for z/OS documentation.

Interface to the CA Network and Systems Management System Status Manager CA OPS/MVS Option

The following CCS for z/OS component is required to run the CA Network and Systems Management System Status Manager CA OPS/MVS Option of CCS for z/OS on CA OPS/MVS.

CB6DB30

Specifies the Agent Technology component

For information on additional setup and configuration steps that must be completed, see the CCS for z/OS documentation.

CA OPS/MVS Multi-System Facility Using CAICCI

The following CCS for z/OS components are required to run the CA OPS/MVS Multi-System Facility (MSF) using CAICCI on CA OPS/MVS.

CAS9C00 or CAS9E00 or CAS9E10

Specifies the CAIRIM component

CAW1C00 or CAW1E00 or CAW1E10

Specifies the CAIENF component

CAW5C00 or CAW5E00 or CAW5E10

Specifies the CAIENF/DB2 component

CAW4C00 or CAW4E00 or CAW4E10

Specifies the CAICCI with SSL component

Note: For information on additional setup and configuration steps that must be completed, see the CCS for z/OS documentation.

OPSCAWTO OPS/REXX Function

The following CCS for z/OS components are required to run the OPSCAWTO OPS/REXX function on CA OPS/MVS.

CAS9C00 or CAS9E00 or CAS9E10

Specifies the CAIRIM component

CAW1C00 or CAW1E00 or CAW1E10

Specifies the CAIENF component

CAW5C00 or CAW5E00 or CAW5E10

Specifies the CAIENF/DB2 component

For information on additional setup and configuration steps that must be completed, see the CCS for z/OS documentation.

Interface to the CA Event Manager Component

The following CCS for z/OS components are required to run certain ADDRESS USS commands on CA OPS/MVS.

The following specific commands communicate with the z/OS Event Console:

- ADDRESS USS WTO
- ADDRESS USS WTOR
- ADDRESS USS REPLY
- ADDRESS USS DOM
- ADDRESS USS PING
- ADDRESS USS CMD

CD5IB30

Specifies the Event Management component

CAS9C00 or CAS9E00 or CAS9E10

Specifies the CAIRIM component

CAW1C00 or CAW1E00 or CAW1E10

Specifies the CAIENF component

CAW5C00 or CAW5E00 or CAW5E10

Specifies the CAIENF/DB2 component

CAW4C00 or CAW4E00 or CAW4E10

Specifies the CAICCI with SSL component

CAF3C00 or CAF3E00

Specifies the CA-C RUNTIME component

Switch Operations Facility (SOF)

The following CCS for z/OS component is required to run SOF on CA OPS/MVS.

CAW4C00 or CAW4E00 or CAW4E10

Specifies the CAICCI with SSL component of CCS for z/OS

Note: For information on additional setup and configuration steps that must be completed, see the CCS for z/OS documentation.

OPSVASRV OPS/REXX Function

The following CCS for z/OS components are required to run the OPS/REXX function OPSVASRV.

CAS9E10

Specifies the CAIRIM component.

CFA9E10

Specifies the CAVARSRV (CA Common Variable Service) component

CEI0E10

Specifies the CAMASTER (CA Master) component

Interface to Hardware Interface Services

The following CCS for z/OS component is required to utilize API HW* Events and the OPS/REXX ADDRESS HWS command.

CC2D770

Specifies the Management Services Assembler component that is utilized by CAHISRV 2.0

Interface to Linux Connector

The following CCS for z/OS components are required to utilize API LX* Events and the OPS/REXX ADDRESS LXC command.

CC2D770

Specifies the Management Services Assembler used by CA Linux Connector 2.0

CC2D771

Specifies the Management Services Other used by CA Linux Connector 2.0

CE2J200

Specifies the Operations Services used by CA Linux Connector 2.0

Appendix C: DASD Calculation Chart

This section contains the following topics:

DASD Requirements for OPSLOG Messages

Default: 568 3390 cylinders

Recommended: Messages from one week; dependent upon your console traffic

# OPSLOG Messages	Device Type	Events Per Cylinder	Required Cylinders
400000	3380	1409	284
400000	3390	1690	237
600000	3380	1409	426
600000	3390	1690	356
800000	3380	1409	568
800000	3390	1690	474
1000000	3380	1409	710
1000000	3390	1690	592
1500000	3380	1409	1065
1500000	3390	1690	888
2000000	3380	1409	1420
2000000	3390	1690	1184
3000000	3380	1409	2130
3000000	3390	1690	1776
4000000	3380	1409	2839
4000000	3390	1690	2367

DASD Requirements for Global Variable Checkpoint DIV Data Sets

For information on the DASD requirements for global variable checkpoint DIV data set, see Defining OPSLOG and Checkpoint VSAM Linear Data Sets in the chapter ”Installation” for restrictions on shared DASD.

# Global Variables	Length of Each Global Variable (in Bytes)	Device Type	Blocks (256 Bytes) Per Cylinder	# Blocks per Variable	Calculated Cylinders	Required Cylinders Including Extra 20%
5000	44	3380	2400	1	3	4
5000	44	3390	2880	1	2	3
5000	100	3380	2400	2	5	6
5000	100	3390	2880	2	4	5
5000	200	3380	2400	2	5	6
5000	200	3390	2880	2	4	5
5000	500	3380	2400	3	7	9
5000	500	3390	2880	3	6	8
5000	1000	3380	2400	5	11	14
5000	1000	3390	2880	5	9	11
5000	2500	3380	2400	11	23	28
5000	2500	3390	2880	11	20	24
10000	44	3380	2400	1	5	6
10000	44	3390	2880	1	4	5
10000	100	3380	2400	2	9	11
10000	100	3390	2880	2	7	9
10000	200	3380	2400	2	9	11
10000	200	3390	2880	2	7	9
10000	500	3380	2400	3	13	16
10000	500	3390	2880	3	11	14
10000	1000	3380	2400	5	21	26
10000	1000	3390	2880	5	18	22

# Global Variables	Length of Each Global Variable (in Bytes)	Device Type	Blocks (256 Bytes) Per Cylinder	# Blocks per Variable	Calculated Cylinders	Required Cylinders Including Extra 20%
10000	2500	3380	2400	11	46	56
10000	2500	3390	2880	11	39	47
25000	44	3380	2400	1	11	14
25000	44	3390	2880	1	9	11
25000	100	3380	2400	2	21	26
25000	100	3390	2880	2	18	22
25000	200	3380	2400	2	21	26
25000	200	3390	2880	2	18	22
25000	500	3380	2400	3	32	39
25000	500	3390	2880	3	27	33
25000	1000	3380	2400	5	53	64
25000	1000	3390	2880	5	44	53
25000	2500	3380	2400	11	115	138
25000	2500	3390	2880	11	96	116
50000	44	3380	2400	1	21	26
50000	44	3390	2880	1	18	22
50000	100	3380	2400	2	42	51
50000	100	3390	2880	2	35	42
50000	200	3380	2400	2	42	51
50000	200	3390	2880	2	35	42
50000	500	3380	2400	3	63	76
50000	500	3390	2880	3	53	64
50000	1000	3380	2400	5	105	126
50000	1000	3390	2880	5	87	105
50000	2500	3380	2400	11	230	276
50000	2500	3390	2880	11	191	230

# Global Variables	Length of Each Global Variable (in Bytes)	Device Type	Blocks (256 Bytes) Per Cylinder	# Blocks per Variable	Calculated Cylinders	Required Cylinders Including Extra 20%
100000	44	3380	2400	1	42	51
100000	44	3390	2880	1	35	42
100000	100	3380	2400	2	84	101
100000	100	3390	2880	2	70	84
100000	200	3380	2400	2	84	101
100000	200	3390	2880	2	70	84
100000	500	3380	2400	3	125	150
100000	500	3390	2880	3	105	126
100000	1000	3380	2400	5	209	251
100000	1000	3390	2880	5	174	209
100000	2500	3380	2400	11	459	551
100000	2500	3390	2880	11	382	459

DASD Requirements for a Shared VSAM Database (optional)

The OPAMSVDB member in the SYS1.OPS.CCLXCNTL data set contains the IDCAMS DEFINE commands and the JCL to create and initialize the shared VSAM KSDS. Tailor and run this JCL to create the file. Information needed to determine the DASD requirements for a shared VSAM database, such as setting the key size, the average and maximum record size for the file, and the primary and secondary record allocations, is contained in the comments of the JCL. Note that each record holds a global variable. The volume parameter must be set.

DASD Requirements for the RDF and System State Manager

To determine the DASD requirements for the RDF and System State Manager, Calculate the DASD space you need based on the number and size of the tables you have.

Providing Global Variable Database Control (Optional)

So that you can closely monitor your global variable databases, CA OPS/MVS issues warning messages as the database becomes full. CA OPS/MVS provides parameters that enable you to closely control and monitor these database indicators. They should be set when you install CA OPS/MVS.

The following CA OPS/MVS parameters control the levels and frequency of the warning messages:

GLOBALWARNTHRESH

The threshold percentage of global variables at which warning messages start to be issued.

Default: 80

GLOBALWARNINTVAL

Specifies, in minutes, how often the warning message for global variables is reissued. This parameter prevents the message from being reissued too frequently.

Default: 5

GLOBALTEMPWARNTH

The threshold percentage of temporary (life-of-CA OPS/MVS) global variables at which warning messages start to be issued.

Default: 80

GLOBALTEMPWARNIV

Specifies, in minutes, how often the warning message for temporary (life-of-CA OPS/MVS) global variables is reissued. This parameter prevents the message from being reissued too frequently.

Default: 5

Usage Warning Messages

CA OPS/MVS also issues warning messages each time database usage increases by 5 percent above the threshold (for instance, at 85 percent, 90 percent, and 95 percent of capacity), even between GLOBALWARNINTVAL intervals. The usage levels triggering the warning messages are not reset in a CA OPS/MVS life cycle unless you change the GLOBALWARNTHRESH parameter to a different value. In this case, the high-usage level is reset to the threshold value.

The warning message OPS4290O, which can apply to either the permanent global variable database or the temporary global variable database, contains the following information:

- Whether the warning is for the temporary or the permanent global variable database
- Current percentage of the database that is full
- Number of blocks currently in use
- Total number of blocks in the database (determined by the value of the GLOBALMAX or GLOBALTEMPMAX parameter)
- Name of the program or rule, once executed, that caused the threshold to be met or exceeded. This program or rule may or may not be responsible for filling the database.

Note: CA OPS/MVS checks for the threshold being exceeded only when a new global variable is allocated or an existing global variable is extended. Therefore, the interval between the messages may be greater than the defined interval.

For an explanation of DASD allocation requirements for global variables see the chapter “Preparing Your System” in the *Getting Started*.

If either the permanent or the temporary global variable database fills completely, CA OPS/MVS issues the OPS1093I message. If this occurs, your automated operations will probably cease to function properly. Because of this, you should make use of the CA OPS/MVS threshold warning message that can alert you to the imminence of such a situation *before* a failure.

Appendix D: Data Sets Created by CA CSM

This section contains the following topics:

Post SMP/E, Deployment, and Configuration Data Sets (see page 209)

Post SMP/E, Deployment, and Configuration Data Sets

Data sets are created by CA CSM after successfully completing each installation step, that is the SMP/E, deployment, and configuration procedures.

Each step in the installation creates the following data sets.

- SMP/E see the data sets in the POST SIS column in the following table. These data sets are collectively known as the CA OPS/MVS SMP/E environment.
- Deployment see the data sets in the POST SDS column in the following table. These data sets are collectively known as the CA OPS/MVS deployment environment.
- Configuration see the POST SCS column in the following table. These data sets are collectively known as the CA OPS/MVS runtime environment data sets.

POST SIS	POST SDS	POST SCS
SMPEHLQ.ACLXASM		
SMPEHLQ.ACLXCLS0		
SMPEHLQ.ACLXCNTL		
SMPEHLQ.ACLXEXEC		
SMPEHLQ.ACLXHENU		
SMPEHLQ.ACLXHFS		
SMPEHLQ.ACLXMENU		
SMPEHLQ.ACLXMIB		
SMPEHLQ.ACLXMOD0		
SMPEHLQ.ACLXOPEX		
SMPEHLQ.ACLXPENU		
SMPEHLQ.ACLXRULE		

POST SIS	POST SDS	POST SCS
SMPEHLQ.ACLXSAMP		
SMPEHLQ.ACLXSENU		
SMPEHLQ.ACLXSHSC		
SMPEHLQ.ACLXSIDE		
SMPEHLQ.ACLXTENU		
SMPEHLQ.ACLXXML		
SMPEHLQ.CCLXASM	SDSHLQ.CCLXASM	OPSPFX.CCLXASM
SMPEHLQ.CCLXCLS0	SDSHLQ.CCLXCLS0	OPSPFX.CCLXCLS0
SMPEHLQ.CCLXCNTL	SDSHLQ.CCLXCNTL	OPSPFX.CCLXCNTL
SMPEHLQ.CCLXEXEC	SDSHLQ.CCLXEXEC	OPSPFX.CCLXEXEC
SMPEHLQ.CCLXHENU	SDSHLQ.CCLXHENU	OPSPFX.CCLXHENU
SMPEHLQ.CCLXHFS.ZFS	SDSHLQ.CCLXHFS.ZFS	OPSPFX.CCLXHFS.ZFS
SMPEHLQ.CCLXHFS.DATA	SDSHLQ.CCLXHFS.DATA	OPSPFX.CCLXHFS.DATA
SMPEHLQ.CCLXLOAD	SDSHLQ.CCLXLOAD	OPSPFX.CCLXLOAD
SMPEHLQ.CCLXMENU	SDSHLQ.CCLXMENU	OPSPFX.CCLXMENU
SMPEHLQ.CCLXMIB	SDSHLQ.CCLXMIB	OPSPFX.CCLXMIB
SMPEHLQ.CCLXOPEX	SDSHLQ.CCLXOPEX	OPSPFX.CCLXOPEX
SMPEHLQ.CCLXPENU	SDSHLQ.CCLXPENU	OPSPFX.CCLXPENU
SMPEHLQ.CCLXPLD	SDSHLQ.CCLXPLD	OPSPFX.CCLXPLD
SMPEHLQ.CCLXRULB	SDSHLQ.CCLXRULB	OPSPFX.CCLXRULB
SMPEHLQ.CCLXRULM	SDSHLQ.CCLXRULM	OPSPFX.CCLXRULM
SMPEHLQ.CCLXRULS	SDSHLQ.CCLXRULS	OPSPFX.CCLXRULS
SMPEHLQ.CCLXSAMP	SDSHLQ.CCLXSAMP	OPSPFX.CCLXSAMP
SMPEHLQ.CCLXSENU	SDSHLQ.CCLXSENU	OPSPFX.CCLXSENU
SMPEHLQ.CCLXSIDE	SDSHLQ.CCLXSIDE	OPSPFX.CCLXSIDE
SMPEHLQ.CCLXTENU	SDSHLQ.CCLXTENU	OPSPFX.CCLXTENU
SMPEHLQ.CCLXXML	SDSHLQ.CCLXXML	OPSPFX.CCLXXML
SMPEHLQ.CSI		
SMPEHLQ.CSI.DATA		
SMPEHLQ.CSI.INDEX		

POST SIS	POST SDS	POST SCS
SMPEHLQ.SMPHOLD		
SMPEHLQ.SMPLOG		
SMPEHLQ.SMPLOGA		
SMPEHLQ.SMPLTS		
SMPEHLQ.SMPMTS		
SMPEHLQ.SMPPTS		
SMPEHLQ.SMPSCDS		
		RULEPREFIX.BASE.RULESUFFIX
		RULEPREFIX.SAMPLE.RULESUFFIX
		RULEPREFIX.STATEMAN.RULESUFFIX
		OPSPFX.USER.REXX
		OPSPFX.SMFID.SYSCHK1
		OPSPFX.SMFID.SYSCHK1.DATA
		OPSPFX.SMFID.OPSLOG OPSPFX.SMFID.OPSLOG.DATA OPSPFX.SMFID.OPSLOG2 OPSPFX.SMFID.OPSLOG2.DATA

Index