

zSecure CARLa-Driven Components

*Installation and Deployment Guide*



**Note**

Before using this information and the product it supports, read the information in [“Notices” on page 229](#).

**February 2023**

This edition applies to the following IBM® Security zSecure products, version 2, release 5, modification 0, and to all subsequent versions of these products until otherwise indicated in new editions:

- IBM Security zSecure Admin (product number 5655-N16)
- IBM Security zSecure Audit (product number 5655-N17)
- IBM Security zSecure Alert (product number 5655-N21)
- IBM Security zSecure Visual (product number 5655-N20)
- IBM Security zSecure Adapters for SIEM (product number 5655-AD8)

© **Copyright International Business Machines Corporation 1988, 2023.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>About this publication.....</b>	<b>ix</b>
IBM Security zSecure Suite library.....	ix
Related documentation for zSecure Suite (z/OS).....	xi
Accessibility.....	xiii
Technical training.....	xiii
Support information.....	xiii
Statement of Good Security Practices.....	xiii
<b>Chapter 1. Installation road map.....</b>	<b>1</b>
<b>Chapter 2. Overview of installation, configuration, and deployment.....</b>	<b>5</b>
CKRINST library.....	5
Configuration data sets.....	5
<b>Chapter 3. Preparation tasks for installation.....</b>	<b>7</b>
Verification of the release.....	7
Naming and securing the zSecure data sets.....	7
Default and site-specific data set naming conventions.....	7
Setup of security for the zSecure installation data sets.....	7
Catalog zSecure data sets in a user catalog (optional).....	8
Space planning.....	8
<b>Chapter 4. Installation of the software.....</b>	<b>9</b>
Installation with the fast method.....	9
Installing from a single installation media.....	9
Installing from multiple installation media.....	10
Installing with a System Pack, Server Pack, or CBPDO.....	10
zSecure-supplied installation jobs.....	11
Customization of the installation parameters.....	11
Updating the installation parameters in the CKRZUPDI member.....	12
Specifying the location of ISPF components in C2RIISPF.....	14
Running CKRZUPDZ to update the CKRINST library members.....	15
<b>Chapter 5. Activation of the product and customization of the configuration data sets.....</b>	<b>17</b>
Distribution of zSecure data sets to additional z/OS images.....	17
Enablement of license features.....	18
APF authorization of the software.....	18
TSO and ISPF command tables for zSecure Admin.....	18
Making the software available to TSO/ISPF users.....	20
Making the software available for batch processes.....	21
<b>Chapter 6. Deployment of the software.....</b>	<b>23</b>
About zSecure configuration data sets.....	23
Creating zSecure configuration data sets.....	25
Customization of zSecure configuration data sets.....	26
Customize jobcards to be used when reports are created as a batch job.....	26
Maintenance of existing zSecure configuration data sets.....	26
Assignment of configurations.....	27

Assignment of configurations to TSO/ISPF users.....	27
Assignment of configurations to batch jobs and started tasks.....	27
<b>Chapter 7. Verification of the installation.....</b>	<b>29</b>
Base ISPF interface functions and menu configuration.....	29
Checking the zSecure Collect function and the base batch operation of zSecure.....	29
Functions to display reports.....	29
CKGRACF command to verify security resources.....	29
Verification of ACF2 reporting.....	29
<b>Chapter 8. Setup for production.....</b>	<b>31</b>
SCKRSAMP and SCKRJOBS data sets.....	31
Capacity planning information.....	31
Introduction.....	31
zSecure Admin.....	36
zSecure Audit.....	38
zSecure Alert.....	40
Daylight saving time considerations.....	41
Use of a fresh CKFREEZE and UNLOAD each day.....	42
Requirements and rationale for running the daily CKGRACF job.....	42
Setup of zSecure Admin Command Logger.....	42
TCP/IP domain name resolution.....	48
SMTP server considerations.....	48
<b>Chapter 9. Setup for remote data access and command routing.....</b>	<b>49</b>
Installation and configuration of the zSecure Server.....	49
Installed software and multi-system support.....	49
JCL procedures and parameters.....	49
Security definitions for the started task.....	50
Configuration statements.....	51
Operator commands for the zSecure Server.....	54
START.....	55
MODIFY.....	55
STOP.....	56
Setup for secure communication using AT-TLS.....	56
Additional security measures.....	57
Setup to disable server security.....	58
Summary of Secure Server Communication.....	58
Use of the zSecure Server to limit the need for access to the security database.....	59
<b>Chapter 10. Setup of zSecure Admin Access Monitor.....</b>	<b>61</b>
Considerations when upgrading from a previous release of Access Monitor.....	61
Installation and post-installation requirements.....	62
Configuration of Access Monitor.....	62
Preparing the JCL.....	63
Definition of security resources and permissions.....	63
Required Access Monitor data sets.....	64
Customization of data collection and consolidation parameters.....	64
Required preparation for collecting UNIX event data.....	68
Optional customization for analytics preprocessing.....	69
Operation of the Access Monitor.....	70
Starting the Access Monitor STC.....	70
MODIFY command to monitor or modify the Access Monitor started task.....	71
Stopping the Access Monitor STC.....	71
Configuration of the Access Monitor function using parm.lib.....	71
Estimating space requirements for collecting UNIX file and directory event data.....	72
Memory or data storage problems when processing Access Monitor data.....	72

zEnterprise data compression (zEDC) for zSecure.....	72
Management of RACF exits installed by Access Monitor.....	74
Change of RACF EXIT calling modes.....	75
Managing the UNIX syscall exits that are installed by Security zSecure Access Business Monitor.....	75
Access Monitor function command reference.....	75
Operator commands.....	76
Configuration commands.....	78

## **Chapter 11. Setup of the z/OS Compliance Integration Manager.....89**

Installation of the z/OS Compliance Integration Manager.....	89
Configuration of the z/OS Compliance Integration Manager.....	89
Operation of the z/OS Compliance Integration Manager.....	92
z/OS Compliance Integration Manager command reference.....	94

## **Chapter 12. Setup of RACF-Offline..... 99**

Default installation data set names.....	99
Installing and activating RACF-Offline .....	99
Building the default options module (B8ROPT).....	99
Updating parmlib members for the APF library.....	101
Updating parmlib members for TSO Authorized Commands (Optional).....	101
Verifying parmlib member for SMF exits.....	102
RACF authorizations for minimal testing.....	102
Commands for creating, testing, and troubleshooting a RACF-Offline database.....	102
Check for RACF-Offline enablement.....	103

## **Chapter 13. Setup of zSecure Alert.....105**

Verification of the product and release.....	105
Considerations when upgrading from a previous release of zSecure Alert.....	105
Prerequisites for configuring and using zSecure Alert.....	105
zSecure Alert address space overview.....	106
Infrastructure.....	106
Supported ddnames for the zSecure Alert started task.....	107
Configuration.....	109
Control.....	109
Post-installation tasks.....	109
Setup of started tasks.....	109
Security resources.....	110
Required data sets.....	111
SMF requirements.....	112
Specifying data set parameters for extended monitoring.....	113
Setup of the alert configuration data set.....	114
Startup of the zSecure Alert address space.....	114
The preamble member C2PXDEF1.....	114
Starting, stopping, and modifying the zSecure Alert started task.....	114
zSecure Alert START parameters.....	115
zSecure Alert operator commands.....	116
Cleanup and deactivation of SMF exits.....	117
Configuration guidelines and performance implications.....	118
WLM service class.....	118
Filters.....	118
Intervals.....	118
Buffers.....	119
Other commands.....	121
DEBUG command.....	121
DIAGNOSE command.....	124
OPTION command.....	125
REPORT command.....	127

FILTER command.....	129
SIMULATE command.....	131
Coexistence considerations.....	131
Upgrade of zSecure Alert.....	131
Backout of an upgrade.....	132
<b>Chapter 14. Setup and use of the zSecure Visual Server.....</b>	<b>133</b>
Setup of the Visual Server.....	133
Installation requirements.....	133
Required system authorizations.....	134
Owners, directories, and file systems preparation.....	134
zSecure configuration for zSecure Visual.....	135
zSecure Visual Server software.....	136
Setup of a new zSecure Visual Server.....	136
Upgrading an existing Server to zSecure Visual 2.5.0.....	139
Compatibility of IBM Security zSecure Visual and zSecure components.....	139
Making clients known to the server.....	140
Visual server access through ISPF.....	140
Configuring the Visual Client.....	140
Canceling a password.....	142
Creating Visual Clients in bulk.....	142
Configuration of client authorities.....	143
Profiles for assigning interface levels to users.....	143
Required access for generated commands.....	143
Profiles for schedule name selection lists.....	145
Authorities required to duplicate a user.....	145
Profiles to allow the Define Alias action.....	145
Resource for RACF scoping.....	145
Password change policy for zSecure Visual users.....	146
Segment editing for users.....	146
Authority to manage client definitions.....	146
Profile for viewing system-wide RACF options.....	146
Implementing site-specific functions.....	147
Site-specific user data.....	147
Site-defined REXX scripts.....	151
zSecure Visual Server operations.....	152
Starting the Visual Server.....	152
Visual Server logs to verify initialization.....	153
Stopping the Visual Server .....	153
Problem determination.....	153
Resources to resolve system problems.....	153
Command to collect diagnostic information.....	155
Server setup (job C2RZWINI) problems.....	155
Server startup problems.....	155
Server response problems.....	156
zSecure Admin termination problems.....	157
SE.W communication problems.....	157
<b>Chapter 15. Data preparation for SIEM.....</b>	<b>161</b>
Prerequisites.....	161
SMF records for the data collection process.....	162
Generating the SMF records.....	162
Make SMF records available to SIEM.....	163
Setup of the collection process for QRadar.....	164
Updating the configuration files for LEEF creation.....	166
Setup of the collection process for Micro Focus ArcSight.....	167
Updating the configuration file for CEF data creation.....	168

Optional customization of the SIEM data.....	170
Updating the configuration file for the near real-time process.....	172
Updating the configuration file for the file polling process for LEEF data.....	172
Check the contents of the general zSecure configuration member.....	172
Assigning a userid and preparing a directory to store the LEEF data.....	173
Assigning a userid and setting up the CKQRADAR or CKQCEF started task.....	173
Assigning a userid and setting up the CKQEXSMF server started task.....	174
Operation of the CKQRADAR and CKQCEF started task.....	175
Operation of the CKQEXSMF started task.....	176
CKQEXSMF configuration statements.....	177
CKQEXSMF operator commands.....	182
QRadar log source properties.....	184
<b>Chapter 16. Data preparation for Guardium Vulnerability Assessment.....</b>	<b>187</b>
<b>Appendix A. Site module.....</b>	<b>191</b>
<b>Appendix B. Security setup for zSecure.....</b>	<b>193</b>
ISPF interface controls.....	193
Resources that configure which options are shown.....	193
Resources that configure which line commands are allowed.....	194
Other ISPF interface controls.....	195
Access to the security database.....	196
Authorization and userid mapping when using the zSecure Server.....	196
Userid mapping.....	198
Other security resources.....	199
Resources that specify which data can be seen or updated.....	199
Security checks related to zSecure Collect.....	199
<b>Appendix C. Restricted mode.....</b>	<b>201</b>
Conditions for restricted mode.....	201
Effects of restricted mode: the user's scope.....	202
Setting up Program Control and PADS access.....	202
<b>Appendix D. Configuration parameters and members.....</b>	<b>205</b>
<b>Appendix E. Configuring the ISPF interface.....</b>	<b>211</b>
Setup of default options for user groups (Setup menu).....	211
Setup (default) National Language Support (SE.D.N).....	211
Setup (default) Installation defined names (SE.D.I).....	220
Setup (default) Command files (SE.D.8).....	221
Retaining your Setup default data when upgrading zSecure.....	221
Configuring zSecure Admin to create new userids in the RACF database.....	222
Locally defined functions.....	222
Command generation.....	223
<b>Appendix F. Interfacing with IBM Security Verify Adapter for RACF with optional zSecure support.....</b>	<b>225</b>
<b>Appendix G. Troubleshooting: Large size of CKFREEZE data sets.....</b>	<b>227</b>
<b>Notices.....</b>	<b>229</b>
Trademarks.....	230
<b>Index.....</b>	<b>233</b>





## About this publication

---

The *IBM Security zSecure CARLa-Driven Components Installation and Deployment Guide* describes the installation and configuration processes for the following IBM Security zSecure components:

- IBM Security zSecure Admin
- IBM Security zSecure Audit for RACF®, ACF2, and Top Secret
- IBM Security zSecure Alert for RACF and ACF2
- IBM Security zSecure Visual for RACF
- IBM Security zSecure Adapters for SIEM for RACF, ACF2, and Top Secret

The manual includes information about the following types of installations:

- Distribution-oriented installations with multiple z/OS® images that have different configurations.
- A single installation that can run multiple configurations from the same z/OS image.

The publication is intended for people responsible for installing and maintaining zSecure products and for deploying the components of zSecure to their user communities.

Readers must be familiar with the IBM Security zSecure products to be installed and the operating systems where the products are being installed.

For error messages, explanations, and workarounds where applicable, see *IBM Security zSecure: Messages Guide*.

## IBM Security zSecure Suite library

---

This topic lists the IBM Security zSecure suite user information.

The zSecure Suite library is available at IBM Documentation for [IBM Security zSecure Suite \(z/OS\)](#). As of the Service Stream Enhancement (SSE) of February 2023, this also includes the former licensed documentation; that is, the *zSecure (Admin and) Audit User Reference Manuals* for [RACF](#), [ACF2](#), and [Top Secret](#) and the *zSecure CARLa Command Reference*.

The zSecure Suite library consists of the following topics:

- *About This Release*  
Provides information about new features and enhancements, incompatibility warnings, and documentation update information for this product version.
- *Documentation*  
Lists and briefly describes the zSecure Suite library and related documentation.
- *Program Directories*  
Are also provided with the product tapes. For the list, see [Program Directories](#).
- *zSecure CARLa-Driven Components Installation and Deployment Guide*  
Provides information about installing and configuring the following IBM Security zSecure components:
  - IBM Security zSecure Admin
  - IBM Security zSecure Audit for RACF, CA-ACF2, and CA-Top Secret
  - IBM Security zSecure Alert for RACF and CA-ACF2
  - IBM Security zSecure Visual
  - IBM Security zSecure Adapters for SIEM for RACF, CA-ACF2, and CA-Top Secret
- *zSecure Messages Guide*  
Provides a message reference for all IBM Security zSecure products and components. This guide describes the message types associated with each product or feature, and lists all IBM Security zSecure

product messages and errors along with their severity levels sorted by message type. This guide also provides an explanation and any additional support information for each message.

- *zSecure Admin and Audit for RACF Getting Started*  
Provides a hands-on guide introducing IBM Security zSecure Admin and IBM Security zSecure Audit product features and user instructions for performing standard tasks and procedures. This manual is intended to help new users develop both a working knowledge of the basic IBM Security zSecure Admin and Audit for RACF system functionality and the ability to explore the other product features that are available.
- *zSecure Admin and Audit for RACF User Reference Manual*  
Describes the product features for IBM Security zSecure Admin and IBM Security zSecure Audit. Includes user instructions to run the admin and audit features from ISPF panels. This manual also provides troubleshooting resources and instructions for installing the zSecure Collect for z/OS component.
- *IBM Security zSecure Admin and Audit for RACF Line Commands and Primary Commands Summary*  
Lists the line commands and primary (ISPF) commands with very brief explanations.
- *zSecure Audit for ACF2 Getting Started*  
Describes the zSecure Audit for CA-ACF2 product features and provides user instructions for performing standard tasks and procedures such as analyzing Logon IDs, Rules, Global System Options, and running reports. The manual also includes a list of common terms for those not familiar with ACF2 terminology.
- *zSecure Audit for ACF2 User Reference Manual*  
Explains how to use zSecure Audit for CA-ACF2 for mainframe security and monitoring. For new users, the guide provides an overview and conceptual information about using CA-ACF2 and accessing functionality from the ISPF panels. For advanced users, the manual provides detailed reference information, troubleshooting tips, information about using zSecure Collect for z/OS, and details about user interface setup.
- *zSecure Audit for Top Secret User Reference Manual*  
Describes the zSecure Audit for CA-Top Secret product features and provides user instructions for performing standard tasks and procedures.
- *zSecure CARLa Command Reference*  
Provides both general and advanced user reference information about the CARLa Auditing and Reporting Language (CARLa). CARLa is a programming language that is used to create security administrative and audit reports with zSecure. The *zSecure CARLa Command Reference* also provides detailed information about the NEWLIST types and fields for selecting data and creating zSecure reports.
- *zSecure Alert User Reference Manual*  
Explains how to configure, use, and troubleshoot IBM Security zSecure Alert, a real-time monitor for z/OS systems protected with the Security Server (RACF) or CA-ACF2.
- *zSecure Command Verifier User Guide*  
Explains how to install and use IBM Security zSecure Command Verifier to protect RACF mainframe security by enforcing RACF policies as RACF commands are entered.
- *zSecure CICS Toolkit User Guide*  
Explains how to install and use IBM Security zSecure CICS® Toolkit to provide RACF administration capabilities from the CICS environment.
- *zSecure Visual Client Manual*  
Explains how to set up and use the IBM Security zSecure Visual Client to perform RACF administrative tasks from the Windows-based GUI.
- *Support for problem solving*  
Solutions to problems can often be found in IBM knowledge bases or a product fix might be available. If you register with IBM Software Support, you can subscribe to IBM's weekly email notification service. IBM Support provides assistance with product defects, answers frequently asked questions, and helps to resolve problems.

## Program Directories

Program directories are provided with the product tapes. You can also find the latest versions of the Program Directories on IBM Documentation for [IBM Security zSecure Suite](#).

- *Program Directory: IBM Security zSecure CARLa-Driven Components*, GI13-2277  
This program directory is intended for the systems programmer responsible for program installation and maintenance. It contains information concerning the material and procedures associated with the installation of IBM Security zSecure CARLa-Driven Components: Admin, Audit, Visual, Alert, and the IBM Security zSecure Adapters for SIEM.
- *Program Directory: IBM Security zSecure CICS Toolkit*, GI13-2282  
This program directory is intended for the systems programmer responsible for program installation and maintenance. It contains information concerning the material and procedures associated with the installation of IBM Security zSecure CICS Toolkit.
- *Program Directory: IBM Security zSecure Command Verifier*, GI13-2284  
This program directory is intended for the systems programmer responsible for program installation and maintenance. It contains information concerning the material and procedures associated with the installation of IBM Security zSecure Command Verifier.
- *Program Directory: IBM Security zSecure Admin RACF-Offline*, GI13-2278  
This program directory is intended for the systems programmer responsible for program installation and maintenance. It contains information concerning the material and procedures associated with the installation of the IBM Security zSecure Admin RACF-Offline component of IBM Security zSecure Admin.
- Program Directories for the zSecure Administration, Auditing, and Compliance solutions and IBM Z Security and Compliance Center 1.1:
  - 5655-N23: *Program Directory for IBM Security zSecure Administration*, GI13-2292
  - 5655-N24: *Program Directory for IBM Security zSecure Compliance and Auditing*, GI13-2294
  - 5655-N25: *Program Directory for IBM Security zSecure Compliance and Administration*, GI13-2296
  - 5655-N34: *Program Directory for IBM Z Compliance and Auditing*, GI13-5255
  - 5655-N35: *Program Directory for IBM Z Compliance, Auditing and Administration*, GI13-5257

## Related documentation for zSecure Suite (z/OS)

This section includes titles and links for information related to zSecure Suite.

See:	For:
<a href="#">IBM Security zSecure Suite</a>	All zSecure unlicensed documentation. For information about what is specific for a release, system requirements, incompatibilities and so on, select the version of your choice and <i>About This Release</i> ; see "What's new" and "Release notes". To obtain the zSecure licensed documentation, see <a href="#">Obtain licensed documentation</a> .
<a href="#">IBM Security Verify Governance adapters</a>	Information about the IBM Security Verify Governance Adapters.
<a href="#">IBM Documentation for z/OS</a>	Information about z/OS. <a href="#">Table 1 on page xii</a> lists some of the most useful publications for use with zSecure.
<a href="#">IBM Z® Multi-Factor Authentication documentation</a>	Information about IBM Z Multi-Factor Authentication (MFA) documentation.

<b>See:</b>	<b>For:</b>
<a href="#">z/OS Security Server RACF publications</a>	Information about z/OS Security Server Information about z/OS Security Server Resource Access Control Facility (RACF). For information about the RACF commands, and the implications of the various keywords, see the <i>z/OS Security Server RACF Command Language Reference</i> and the <i>z/OS Security Server RACF Security Administrator's Guide</i> . You can find information about the various types of events that are recorded by RACF in the <i>z/OS Security Server RACF Auditor's Guide</i> .
<a href="#">QRadar® DSM Configuration Guide</a>	For more information about QRadar, see the <a href="#">IBM QRadar Security Intelligence Platform</a> on IBM Documentation.
<a href="#">IBM CICS Transaction Server for z/OS documentation</a>	Information about CICS Transaction Server for z/OS.
<a href="#">IBM MQ</a>	Information about IBM MQ.
<a href="#">IBM Z NetView®</a>	Information about IBM Z NetView.
CA-ACF2 documentation	Information about ACF2 and the types of events that can be reported using zSecure Audit for ACF2.
CA-Top Secret for z/OS documentation	Information about Top Secret and the types of events that can be reported using zSecure Audit for Top Secret.

<i>Table 1. Some of the most useful z/OS publications for use with zSecure</i>	
<b>Manual Title</b>	<b>Order Number</b>
<i>z/OS Communications Server: IP Configuration Guide</i>	SC27-3650
<i>z/OS Communications Server: IP Configuration Reference</i>	SC27-3651
<i>z/OS Cryptographic Services ICSF Administrator's Guide</i>	SC14-7506
<i>z/OS Cryptographic Services ICSF System Programmer's Guide</i>	SC14-7507
<i>z/OS Integrated Security Services Enterprise Identity Mapping (EIM) Guide and Reference</i>	SA23-2297
<i>z/OS ISPF Dialog Developer's Guide and Reference</i>	SC19-3619
<i>z/OS MVS Initialization and Tuning Reference</i>	SA23-1380
<i>z/OS MVS Programming: Assembler Services Reference, Volume 1 (ABE-HSP)</i>	SA23-1369
<i>z/OS MVS Programming: Assembler Services Reference, Volume 2 (IAR-XCT)</i>	SA23-1370
<i>z/OS MVS™ Programming: Authorized Assembler Services Reference, Volume 1 (ALE-DYN)</i>	SA23-1372
<i>z/OS MVS Programming: Callable Services for High Level Languages</i>	SA23-1377
<i>z/OS MVS System Codes</i>	SA38-0665
<i>z/OS MVS System Commands</i>	SA38-0666
<i>z/OS MVS System Management Facilities (SMF)</i>	SA38-0667
<i>z/OS Security Server RACF Security Administrator's Guide</i>	SA23-2289
<i>z/OS Security Server RACF Auditor's Guide</i>	SA23-2290

Table 1. Some of the most useful z/OS publications for use with zSecure (continued)

Manual Title	Order Number
<i>z/OS Security Server RACF Command Language Reference</i>	SA23-2292
<i>z/OS Security Server RACF Macros and Interfaces</i>	SA23-2288
<i>z/OS Security Server RACF Messages and Codes</i>	SA23-2291
<i>z/OS Security Server RACF System Programmer's Guide</i>	SA23-2287
<i>z/OS Unicode Services User's Guide and Reference</i>	SA38-0680
<i>z/OS UNIX System Services Messages and Codes</i>	SA23-2284
<i>z/OS UNIX System Services Planning</i>	GA32-0884
<i>z/Architecture<sup>®</sup> Principles of Operation</i>	SA22-7832

For information about z/VM<sup>®</sup>, see [IBM z/VM documentation](#) or see [www.vm.ibm.com/library](http://www.vm.ibm.com/library).

## Accessibility

---

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

## Technical training

---

For technical training information, see the IBM Training and Skills website at [IBM Training](#).

For a list of formal customer education for IBM Security zSecure, see the [zSecure Course Offerings](#). This PDF file is part of the [zSecure - Learning](#) information, which also includes CARLa self studies and sample applications.

## Support information

---

IBM Support provides assistance with code-related problems and routine, short duration installation or usage questions. You can directly access the IBM Software Support site at [www.ibm.com/mysupport](http://www.ibm.com/mysupport).

## Statement of Good Security Practices

---

IT system security involves protecting systems and information through intrusion prevention, detection, and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, or misappropriated or can result in misuse of your systems to attack others. Without a comprehensive approach to security, no IT system or product should be considered completely secure and no single product or security measure can be completely effective in preventing improper access. IBM systems and products are designed to be part of a regulatory compliant, comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products, or services to be most effective.

**Important:** IBM does not warrant that any systems, products, or services are immune from, or will make your enterprise immune from, the malicious or illegal conduct of any party.



---

# Chapter 1. Installation road map

The steps in this topic describe the framework to install, configure, and deploy a new installation of IBM Security zSecure.

## Procedure

1. Learn about key concepts and resources related to installing, configuring, and deploying the product.
  - a) Review the concepts of single installation and distribution-oriented installation.  
See [Chapter 2, “Overview of installation, configuration, and deployment,” on page 5.](#)
  - b) Learn about the CKRINST library, which contains sample jobs that you can customize for installation and post-installation activities.  
See [“CKRINST library” on page 5.](#)
  - c) Learn about zSecure configuration data sets, which you can use to support distribution-oriented installation  
See [“Configuration data sets” on page 5.](#)
2. Prepare for installation. See [Chapter 3, “Preparation tasks for installation,” on page 7.](#)
  - a) Verify the release.  
See [Verify the release.](#)
  - b) Name and secure the zSecure data sets.  
See [“Naming and securing the zSecure data sets” on page 7.](#)
  - c) Evaluate space requirements.  
See [“Space planning” on page 8.](#)
3. Install the software. Use one of the following installation methods:
  - Formal installation.  
See *Program Directory: IBM Security zSecure CARLa-Driven Components.*
  - Fast installation.  
See [“Installation with the fast method” on page 9.](#)
  - Installation as part of System Pack, Server Pack, or CBPDO.  
See [“Installing with a System Pack, Server Pack, or CBPDO” on page 10.](#)
4. Make the software available so that users can start it.  
See [Chapter 5, “Activation of the product and customization of the configuration data sets,” on page 17](#) for overview information.
  - a. If you want to run zSecure from data sets other than the data set where you installed it, distribute the zSecure data sets.  
See [“Distribution of zSecure data sets to additional z/OS images” on page 17.](#)
  - b. Enable the license.  
See [“Enablement of license features” on page 18.](#)
  - c. APF authorize the software.  
See [“APF authorization of the software” on page 18.](#)
  - d. Make the zSecure software available to TSO/ISPF users.  
See [“Making the software available to TSO/ISPF users” on page 20.](#)

- e. Make the software available to run in batch or as a started task.  
See [“Making the software available for batch processes” on page 21.](#)
5. Deploy the software using configuration files. See [Chapter 6, “Deployment of the software,” on page 23](#) for overview information.
  - a) Learn more about zSecure configuration data sets.  
See [“About zSecure configuration data sets” on page 23.](#)
  - b) Create the zSecure configuration data sets.  
See [“Creating zSecure configuration data sets” on page 25.](#)
  - c) Customize the zSecure configuration data sets.  
See [“Customization of zSecure configuration data sets” on page 26.](#)
  - d) (Optional) If you are upgrading, see [“Maintenance of existing zSecure configuration data sets” on page 26.](#)
  - e) Make the zSecure configuration data sets available and establish security for each configuration:
    - i) Assign the configurations to the appropriate TSO/ISPF users.  
See [“Assignment of configurations to TSO/ISPF users” on page 27.](#)
    - ii) Assign the configurations to the appropriate batch jobs and started tasks.  
See [“Assignment of configurations to batch jobs and started tasks” on page 27.](#)
    - iii) Establish security for each configuration to control access to product functions and data.  
See [Appendix B, “Security setup for zSecure,” on page 193.](#)
6. Verify the installation.
  - a) Check the base ISPF interface functions and menu configuration.  
See [“Base ISPF interface functions and menu configuration” on page 29.](#)
  - b) Check the zSecure Collect function and the base batch operation of zSecure.  
See [“Checking the zSecure Collect function and the base batch operation of zSecure” on page 29.](#)
  - c) Display reports.  
See [“Functions to display reports” on page 29.](#)
  - d) Check CKGRACF. (If you do not use zSecure Admin or zSecure Visual, this step is optional.)  
See [“CKGRACF command to verify security resources” on page 29.](#)
  - e) Verify ACF2 reporting. (If you did not install zSecure Audit for ACF2, this step is optional.)  
See [“Verification of ACF2 reporting” on page 29.](#)
7. Set up the following items as necessary for production:
  - a) Review the capacity planning information to help you determine the system resources required.  
See [“Capacity planning information” on page 31.](#)
  - b) Specify the default input set.  
See [Chapter 8, “Setup for production,” on page 31.](#)
  - c) Customize your installation for daylight saving time.  
See [“Daylight saving time considerations” on page 41.](#)
  - d) Refresh the CKFREEZE file.  
See [“Use of a fresh CKFREEZE and UNLOAD each day” on page 42.](#)
  - e) Set up the zSecure Admin Command Logger.  
See [“Setup of zSecure Admin Command Logger” on page 42.](#)
  - f) Ensure that TCP/IP domain names can be resolved.



- See [“TCP/IP domain name resolution”](#) on page 48.
- g) Check the settings for the SMTP server.
- See [“SMTP server considerations”](#) on page 48.
8. Set up multi-system support if you want to administer and audit profiles, resources, and settings from multiple systems.
- See [Chapter 9, “Setup for remote data access and command routing,”](#) on page 49.
- a. Install, configure, and activate the zSecure Server.
- See [“Installation and configuration of the zSecure Server”](#) on page 49.
- b. Specify the remote data sets for use in CKRCARLA or the ISPF User Interface.
- See [“Operator commands for the zSecure Server”](#) on page 54.
- c. Perform setup for routing RACF and selected non-RACF commands to other systems.
- See [“Setup for secure communication using AT-TLS”](#) on page 56.
9. Set up components such as:
- zSecure Admin Access Monitor.  
See [Chapter 10, “Setup of zSecure Admin Access Monitor,”](#) on page 61.
  - RACF-Offline.  
See [Chapter 12, “Setup of RACF-Offline,”](#) on page 99.
  - zSecure Alert.  
See [Chapter 13, “Setup of zSecure Alert,”](#) on page 105.
  - zSecure Visual Server.  
See [Chapter 14, “Setup and use of the zSecure Visual Server,”](#) on page 133.
  - Data preparation for QRadar SIEM.  
See [Chapter 15, “Data preparation for SIEM,”](#) on page 161.
  - Data preparation for Guardium® Vulnerability Assessment.  
See [Chapter 16, “Data preparation for Guardium Vulnerability Assessment,”](#) on page 187.



---

## Chapter 2. Overview of installation, configuration, and deployment

zSecure provides sample jobs for installing the software and setting up the product. You need access to these jobs during the installation, configuration, and deployment process. For more information about these jobs, see [“CKRINST library”](#) on page 5.

### Distribution-oriented installation

The zSecure installation, configuration, and deployment process supports both single installation and distribution-oriented installation.

- In a single installation, you install zSecure separately on each z/OS image.
- In a distribution-oriented installation, you install the product on one z/OS image, and then run it on multiple images with different configuration files. You can also run multiple configurations in a single z/OS image. For example, you might want the following configurations:
  - A *full function* configuration for central administrators
  - A *streamlined* configuration for people who run only common user administration tasks and require access only to the Quick Administration menu (option RA.Q)

For more information about the configuration files, see [“Configuration data sets”](#) on page 5.

**Note:** If you intend to perform RACF administration and auditing tasks on both z/VM and z/OS systems, you must install zSecure separately for each operating system. For information about installing the z/VM version of the product, see *IBM Security zSecure Manager for RACF z/VM: Manager for RACF z/VM Installation and Deployment Guide*.

## CKRINST library

---

You can use these sample jobs to customize installation parameters like the naming convention for your data sets, JOB statement requirements, and other items. These parameters are also used in post-installation and other activities performed after the product has been installed. Customizing these parameters is optional. However, if you customize them before installing the software or performing post-installation activities, it saves time because you update the parameter values in the sample jobs at one time rather than editing individual jobs before submitting them.

The CKRINST library is created by copying the sample jobs from the product tape or by copying the SCKRSAMP library installed with the product. Instructions for creating the CKRINST library are provided in the installation instructions. See [Chapter 4, “Installation of the software,”](#) on page 9.

## Configuration data sets

---

The data sets represent the zSecure configuration for an image and determine how the software operates on the image. For example, zSecure configuration data sets can specify what zSecure features are available as well as the data set names for the input data sources.

zSecure configuration data sets are the only data sets that are different between images. Using these data sets, you can create configurations for the following purposes:

- Special-purpose configurations for processes such as zSecure Alert, Access Monitor, and Visual Server.
- Separate configurations to deploy zSecure on different z/OS images.
- Configurations for user groups that require different input data or that restrict access to specific zSecure components.

zSecure configuration data sets are stored in partitioned data sets that are not part of the installed software. As a result, these data sets are not updated when the software is upgraded or reinstalled. Therefore, you can maintain custom configuration settings across upgrades.

zSecure provides a default configuration data set that can be copied and updated for your environment. For additional information and instructions for creating and customizing configuration data sets, see [Chapter 6, “Deployment of the software,” on page 23](#).

---

## Chapter 3. Preparation tasks for installation

- [“Verification of the release” on page 7](#)
- [“Naming and securing the zSecure data sets” on page 7](#)
- [“Space planning” on page 8](#)

---

### Verification of the release

Before you install the software:

- Verify that the product and release you are about to install is the current and supported release.
- Verify that the product is supported on the platform where you intend to use it.

See [www.ibm.com/support/home/pages/lifecycle/](http://www.ibm.com/support/home/pages/lifecycle/). (Search for zSecure.)

Also, review the [Release notes](#) document that is available at IBM Documentation for [IBM Security zSecure Suite](#) for information about the latest product updates and any incompatibility warnings.

---

### Naming and securing the zSecure data sets

1. Decide on data set name conventions.
2. Plan security for the product data sets.
3. (Optional) Specify a user catalog for the data sets.

### Default and site-specific data set naming conventions

zSecure provides a default naming convention for the data sets where zSecure is installed.

The default naming scheme uses two qualifiers in the data set name CKR.SCKRLOAD, for example.

#### **CKR**

Prefix common to all zSecure product data sets. You can replace this prefix by one or more qualifiers of your own choice.

#### **dddef**

The last qualifier is equal to the DD name definition used by SMP/E (SMP/E DDDEF) and is the low-level qualifier in the names for the target and distribution libraries. Each DDDEF starts either with S (for target libraries) or with A (for distribution libraries) followed by the product prefix. See the *Program Directory: IBM Security zSecure CARLa-Driven Components* for a list of the target and distribution library DDNAMEs.

During installation, you might want to specify your own naming convention to override these defaults. You can change the prefix CKR to a different qualifier or more than one qualifier. For example, if you install zSecure on a dead system for subsequent distribution, you can replace the prefix CKR with DEADSYS.CKR or DEADSYS.CKR.CKRv1m. In this way, you can create data sets such as DEADSYS.CKR.SCKRLOAD or DEADSYS.CKR.CKRv1m.SCKRLOAD. When you use different names during installation, you can use the default names to distribute to active system images. Using different names during installation can also help avoid naming conflicts when installing or testing a new release. For instructions for customizing the data set naming convention, see [“Customization of the installation parameters” on page 11](#).

### Setup of security for the zSecure installation data sets

If you install for distribution, protect the zSecure data sets so that only the people who install or maintain the product have access.

- Give UPDATE access to users who are responsible for system maintenance.

- Give ALTER access to users who are responsible for installation or space management.

If zSecure runs directly from the installed data sets, give READ access to zSecure users. Do not give access to other users.

## Catalog zSecure data sets in a user catalog (optional)

For easy access to the zSecure software from multiple z/OS images, catalog zSecure data sets in a user catalog. You can use an existing user catalog or create a new one. In this way, you can easily access the software from multiple z/OS images by adding an alias to that user catalog. The TSO command to create an alias to that user catalog is:

```
DEFINE ALIAS (NAME('your-high-level-qualifier') REL('your-user-catalog'))
```

Without an alias defined, the installation jobs can run without errors. However, all data sets are cataloged in the master catalog of the z/OS image where you run the installation jobs.

## Space planning

---

For programming and space requirements, see the following zSecure program directories:

- The zSecure Admin RACF-Offline component has its own program directory: *Program Directory: IBM Security zSecure Admin RACF-Offline*.
- All other CARLa-driven components of zSecure have a common program directory: *Program Directory: IBM Security zSecure CARLa-Driven Components*.

These program directories are available with the product and online at IBM Documentation for [IBM Security zSecure Suite](#).

---

## Chapter 4. Installation of the software

You can install zSecure software using one of the following methods:

- **Formal installation.**

When you use formal installation:

- You have full control over the SMP/E RECEIVE, APPLY, and ACCEPT jobs.
- You can install zSecure in existing global and product zones.

For formal installation, follow the instructions provided in the *Program Directory: IBM Security zSecure CARLa-Driven Components*.

- **Fast installation.**

This method runs most of the installation process in a single job, CKRZINST, rather than running the SMP/E RECEIVE, APPLY, and ACCEPT jobs separately. Fast installation installs the product in **new** global and product zones.

**Note:** The fast installation method does not support installing zSecure Admin RACF-Offline. If you intend to use the RACF-Offline function, use the formal installation method. For information about installing RACF-Offline, see [Chapter 12, “Setup of RACF-Offline,” on page 99](#).

- **Installation as part of System Pack, Server Pack, or CBPDO.**

If you install zSecure using a System Pack, Server Pack, or CBPDO, follow the instructions provided with the package. Do **not** use the instructions in the *Program Directory: IBM Security zSecure CARLa-Driven Components*. After installing zSecure, you must have a copy of the installation library to perform post-installation activities.

For installation instructions, see the following information:

- For formal installation, see the *Program Directory: IBM Security zSecure CARLa-Driven Components*.
- For fast installation, see [“Installation with the fast method” on page 9](#).
- For installation as part of System Pack, Server Pack, or CBPDO, see [“Installing with a System Pack, Server Pack, or CBPDO” on page 10](#).

---

### Installation with the fast method

The fast installation method installs zSecure in new global and product zones. It runs the RECEIVE, ACCEPT, and APPLY steps from a single job. You can install from a single installation media or from several types of installation media.

This method uses the zSecure-supplied installation jobs in the CKRINST library to customize the installation parameters and install the software. These jobs are also used to perform post-installation tasks. Instructions for obtaining these jobs are included in the following procedures.

- [“Installing from a single installation media” on page 9](#)
- [“Installing from multiple installation media” on page 10](#)

### Installing from a single installation media

#### Procedure

1. Create a copy of the CKRINST installation library. See [“zSecure-supplied installation jobs” on page 11](#).
2. Customize the installation parameters in the CKRINST library. See [“Customization of the installation parameters” on page 11](#).

3. Run the CKRZINST job found in the CKRINST installation library created in step “1” on page 9.

## Installing from multiple installation media

### About this task

If you received multiple installation source media, each containing one product, combine the source media so that you can install the products in shared libraries rather than installing each source media into a separate library.

### Procedure

1. Using any of the tapes (or download files), create and customize the CKRINST library. See “zSecure-supplied installation jobs” on page 11.
2. Customize the installation parameters in the CKRINST library. See “Customization of the installation parameters” on page 11.
3. Run job CKRZINST from the CKRINST library up to and including the RECEIVE job step.
4. RECEIVE all the other tapes or receive the download files into the SMP/E zone that job CKRZINST created in step “3” on page 10.

If the system issues any already received messages, you can ignore them.

5. After everything is received, run the remainder of job CKRZINST: specify RESTART=ALLOCT on the JOB statement and resubmit the job.

## Installing with a System Pack, Server Pack, or CBPDO

---

### About this task

If you install zSecure using a System Pack, Server Pack, or CBPDO, follow the instructions provided with the package you selected. Do not use the instructions in the *Program Directory: IBM Security zSecure CARLa-Driven Components*.

You also need a copy of the zSecure-supplied sample jobs for post-installation activities. These jobs are found in the SCKRSAMP library installed with the product. See the instructions for copying these jobs in the following procedure.

To install zSecure from a System Pack, Server Pack, or CBPDO:

### Procedure

1. Follow the instructions provided with the package to install the software.

If you are installing from DASD, adjust the SMPPTFIN DD in the RECEIVE step. For instructions, see the *Supplemental Installation Instructions for Performing an SMP/E Installation from DASD*, which you can obtain from IBM Software Support.

**Note:** If you received multiple zSecure components in separate packages (for example, zSecure Admin and zSecure Visual, each in a separate CBPDO), complete the following steps to combine the source and install the product.

- a. Run the installation job up to and including the RECEIVE job step.
- b. RECEIVE all the other tapes or download the files into the SMP/E zone created by the installation job.

If the system issues any already received messages, you can ignore them.

- c. After everything is received, run the remainder of the installation job by specifying RESTART=ALLOCT on the JOB statement and resubmitting the job.



2. Obtain a copy of the zSecure-supplied sample jobs by copying the SCKRSAMP library installed with the product to a new data set. For the data set name, use the default low-level qualifier CKRINST.
3. Customize the installation parameters used for post-installation activities. See [“Customization of the installation parameters”](#) on page 11.

## zSecure-supplied installation jobs

Both the formal and fast installation methods use the zSecure-supplied installation jobs. These jobs are provided to help you install and set up the software. You can customize these jobs to specify the naming convention for your data sets, JOB statement requirements, and other items. You can obtain the sample installation jobs in either of the following ways:

- Directly from the tape.
- By performing an SMP/E RECEIVE and then copying the jobs from IBM.HCKR250.F1 to a work data set for editing and submission.

The following job provides the JCL for either method. You can download a sample of this job at IBM Documentation for [IBM Security zSecure Suite](#).

Figure 1. Sample JCL to obtain the zSecure-supplied installation jobs

```
//STEP1 EXEC PGM=GIMUNZIP,REGION=0M,PARM='HASH=NO'
//SYSUT3 DD UNIT=SYSALLDA,SPACE=(CYL,(10,10))
//SYSUT4 DD UNIT=SYSALLDA,SPACE=(CYL,(15,5))
//SMPJHOME DD PATH='/usr/lpp/java/J5.0/' <===NOTE 1
//SMPCPATH DD PATH='/usr/lpp/smp/classes/' <===NOTE 1
//SMPOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SMPDIR DD PATHDISP=KEEP,
// PATH='/<ntsdire>/<orderid>/SMPRELF/' <===NOTE 2
//SYSIN DD *
<GIMUNZIP>
<ARCHDEF
name="CBCACHE.IBM.HCKR250.F1.pax.Z"
volume="<volser>" <===NOTE 3
newname="<your-prefix>.CKRINST" > <===NOTE 4
</ARCHDEF>
</GIMUNZIP>
/*
```

Before submitting this job, make the following updates based on your installation requirements:

- Add a job card that is specific to your installation requirements.
- Note 1: Change these directories to your installations java and smp classes directories.
- Note 2: Change `ntsdire` to the directory that holds your Shopz orders. Change `orderid` to your order ID, for example 2008567304\_000010\_PROD.
- Note 3: Change `volser` to a volser that you want the output dataset to reside on.
- Note 4: Change `your-prefix` to the high-level qualifier(s) for the output dataset.

## Customization of the installation parameters

The global update process updates the values of the parameters used in CKRINST members such as the CKRZINST fast installation job and formal installation jobs like CKRZREC, CKRZAPP, and CKRZACC. The installation parameters are also used by the post-installation job CKRZPOST and later jobs. Customizing the parameters before installing the software or before performing post-installation activities saves time because you can update the parameter values across the CKRINST members rather than editing individual members within the library. [Table 2 on page 12](#) lists the CKRINST members required to customize and update the installation parameters.

Table 2. CKRINST library members: Installation jobs for customizing and updating the installation parameters

CKRINST member	Description
CKRZUPDI	This member specifies values for the installation parameters used in the formal and fast installation jobs and the post-installation job CKRZPOST, including parameters that determine which zSecure components are installed, as well as the data set naming conventions for zSecure software, and configuration data sets. Edit this job to customize these parameter values for your installation.
C2RIISPF	This member specifies the location of ISPF components required by zSecure for tasks such as updating the CKRINST library. Edit this job before running the global update job CKRZUPDZ.
CKRZUPDZ	This update job performs a global update of the CKRINST library. Run this job to apply the changes made in the CKRZUPDI member.

Use the following procedures to customize the installation parameters and update the installation jobs in CKRINST:

- [“Updating the installation parameters in the CKRZUPDI member” on page 12](#)
- [“Specifying the location of ISPF components in C2RIISPF” on page 14](#)
- [“Running CKRZUPDZ to update the CKRINST library members” on page 15](#)

## Updating the installation parameters in the CKRZUPDI member

### About this task

You can update the job card, installation parameters, and JCL for zSecure jobs from the CKRZUPDI member provided in the CKRINST library.

```
//***** Jobcard updates *****
Jobcard1>//JOBNAME  JOB ACCT,ZSECURE,MSGCLASS=A,TIME=60,USER=,
Jobcard2>//          NOTIFY=&&SYSUID
Jobcard3>//*JOB3
//***** JCL updates *****
Orderid          = 2008567304_000010_PROD
NTSdirectory     = /u/smpe/smpnts/
PrefixForTargetLibraries = CKR
VolumeForTargetLibraries =
PrefixForDistributionLibraries = CKR.DLIB
VolumeForDistributionLibraries =
JclLib           = Yes
SmpeTargetZone  = CKR250T
SmpeDistributionZone = CKR250D
PrefixForSmpeGlobalZone = CKR.SMPE.G
PrefixForSmpeOtherData = CKR.SMPE
SmpeCsiAndSmpplibVolume = SMS001
//***** Products/features to install *****
AdminRACF       = No
AuditRACF       = No
AuditACF2       = No
AuditTopSecret  = No
AlertRACF       = No
AlertACF2       = No
VisualRACF      = No
QRadarAdaptersRACF = No
QRadarAdaptersACF2 = No
QRadarAdaptersTopSecret = No
zOSComplianceIntegrationManager = No
```

Figure 2. CKRZUPDI member and JCL updates

Complete the following steps to modify the installation parameters with the values required for your installation. None of the parameters are case-sensitive.

## Procedure

1. Modify the job parameters to be added to all installation jobs

```
Jobcard1=//JOBNAME JOB ACCT,ZSECURE,MSGCLASS=A,TIME=10,USER=,  
Jobcard2=// NOTIFY=&&SYSUID  
Jobcard3=//*JOB3
```

2. Specify the Shopz order id containing the zSecure package and the directory that holds the Shopz orders.

```
Orderid = 2008567304_000010_PROD  
NTSdirectory = /u/smpe/smpnts/
```

3. Specify the high-level qualifiers for the data sets where IBM Security zSecure will be installed. If you want to distribute, make this prefix different from the other prefixes so that you can easily use `PrefixForTargetDatasets` to select the data sets that qualify for Distribution.

```
PrefixForTargetLibraries = CKR
```

4. Set the volume serial for the IBM Security zSecure target libraries. If you leave this parameter blank, the system selects the volumes.

```
VolumeForTargetLibraries
```

5. Specify the high-level qualifiers for the IBM Security zSecure distribution libraries.

```
PrefixForDistributionLibraries = CKR.DLIB
```

6. Set the volume serial for the IBM Security zSecure distribution libraries. If you leave this parameter blank, the system selects the volumes.

```
VolumeForDistributionLibraries =
```

7. Enable the JCLLIB statement parameter.

```
JclLib = Yes
```

If this parameter is set to Yes, a JCLLIB statement for the SCKRPROC data set is imbedded in the jobs in the CKRJOBS data set. The CKRJOBS data set is created during the customization job CKRZPOST.

If you intend to include the SCKRPROC data set and the data set that contains configuration members in the procedure library concatenation of your JES, the JCLLIB is not needed and you can specify No.

8. If necessary, specify the names of the SMP/E Target and Dlib zone. Normally, you do not need to change the default values.

```
SmpeTargetZone = CKR250T  
SmpeDistributionZone = CKR250D
```

9. If necessary, change the default values for the high-level qualifiers for SMP/E related data sets. Normally, you do not need to change these values.

```
PrefixForSmpeGlobalZone = CKR.SMPE.G  
PrefixForSmpeOtherData = CKR.SMPE
```

10. Specify the volume serial for SMP/E CSI data sets and SMPTLIB.

This parameter is ignored if you install into an existing SMP/E zone. If you use new zones, the parameter is required because SMP/E and IDCAMS require it. Depending on your SMS configuration, the value that you specify might or might not be used.

```
SmpeCsiAndSmpplibVolume = SMS001
```

11. Select the products or features to be installed. To install a product or feature, change the value for the corresponding parameter to Yes.

AdminRACF	= No
AuditRACF	= No
AuditACF2	= No
AuditTopSecret	= No
AlertRACF	= No
AlertACF2	= No
VisualRACF	= No
QRadarAdaptersRACF	= No
QRadarAdaptersACF2	= No
QRadarAdaptersTopSecret	= No
zOSComplianceIntegrationManager	= No

## Specifying the location of ISPF components in C2RIISPF

### About this task

The process to update the installation parameters in the CKRINST library requires ISPF services such as tables and messages. In zSecure, the location of the ISPF components is defined in the installation member C2RIISPF. The following default ISPF data set names are included in C2RIISPF:

```
ISPMLIB  ISP.SISPMENU
ISPSLIB  ISP.SISPSENU
ISPPLIB  ISP.SISPPENU
ISPTLIB  ISP.SISPTENU
```

Edit these default values to specify the ISPF data set names used in your data center. The following guidelines describe common variations of these data set names:

- The high-level qualifier of the data sets can be ISP instead of SYS1.
- Some installations use a middle qualifier that identifies the level of their ISPF product; for example, V5R2M0.
- The low-level qualifier of some ISPF data sets often reflects the national language. For example, the panel library can have low-level qualifier SISPPENU for (American) English.

### Procedure

To update the ISPF data set names for your installation:

1. Review the ISPF components currently in use on the system, run the following command from the ISPF command line:

```
TSO ISRDDN
```

2. Identify the data sets that contain the base ISPF product (as opposed to other products that exploit ISPF, such as SDSF, or local software that might be allocated to your TSO session).

You can recognize these data sets through the presence of the following members:

- In ISPTLIB: ISPCMDS, ISPPROF, ISPSPROF, and ISRKEYS.
  - In ISPMLIB: ISPP00, ISPP02, ISPP10, ISPP20, ISPP32, ISPV01, ISRD23, ISRE00, ISRE64, ISRE65, ISRE70, and ISRLS12.
3. Edit the installation member C2RIISPF to update the ISPF data set names used at your data center. Only the base ISPF product is required.  
If your ISPLOAD and ISPLPA data sets reside in the link list and LPA list, respectively, you do not need to include them. Otherwise, uncomment and adapt the STEPLIB and ISPLLIB DD statements in C2RIISPF.

To avoid enqueue conflicts, do not specify any other data sets in C2RIISPF. In particular, do not allocate a permanent ISPF profile data set.

# Running CKRZUPDZ to update the CKRINST library members

## About this task

After updating the installation parameters in member CKRZUPDZI and the ISPF data set names in C2RIISPF, review and run job CKRZUPDZ to perform a global update of the installation members.

## Procedure

1. Run CKRZUPDZ in check mode.

If any of the following types of errors occur, correct them:

- C2R8xxxx messages. These messages are described in the *zSecure Messages Guide*.
- RC=990; ISPP100 Panel 'C2RPUPDP' error -/-Panel not found

This error is caused by deleting the IBM Security zSecure installation library from the ISPLIB concatenation. The update process uses the IBM Security zSecure-supplied panel C2RPUPDP. The panel is never displayed, but its presence is required for job C2RZUPDZ.

- The following errors:
  - Abend 04C; message ISPI021 Unrecoverable error in initialization of command tables
  - RC=990; ISPV010 Profile not loaded -/-Profile table 'ISPPROF' not read. Table service RC=8
  - RC=990; ISRxxxx -/-ISRxxxx message not found in 'ISPLIB' library.

These errors are caused by not specifying the correct ISPF data sets in member C2RIISPF. For information about specifying the correct data sets, see [“Specifying the location of ISPF components in C2RIISPF” on page 14](#).

2. After running CKRZUPDZ successfully in check mode without any errors, run the job again in update mode.

**Important:** Running CKRZUPDZ multiple times in update mode is not supported and can result in corrupting the JCL. If you must update the installation members again after the first update, save both the CKRZUPDI and C2RIISPF members. Recreate the CKRINST data set. Then, run the CKRZUPDZ job again.



---

## Chapter 5. Activation of the product and customization of the configuration data sets

After installing zSecure, the zSecure target libraries and distribution data sets are available in the SMP/E-managed data sets created during installation. For example:

- If you used the default data set naming convention, the target libraries are in data sets that start with the high-level qualifier CKR, and the distribution libraries are in data sets that start with the high-level qualifier CKR.DLIB.
- If you specified your own data set naming conventions, the libraries are available in data sets that start with the high-level qualifier you specified.

You can find a complete list of the DDNAMES for the target and distribution libraries in the *Program Directory: IBM Security zSecure CARLa-Driven Components*.

After you install zSecure, perform the following tasks to activate the product and customize the configuration data sets for your installation:

- [“Distribution of zSecure data sets to additional z/OS images” on page 17](#)
- [“Enablement of license features” on page 18](#)
- [“APF authorization of the software” on page 18](#)
- [“Making the software available to TSO/ISPF users” on page 20](#)
- [Chapter 6, “Deployment of the software,” on page 23](#)

---

### Distribution of zSecure data sets to additional z/OS images

After installing the zSecure software on one image, you can distribute the associated data sets to other z/OS images where you want to run it. If you want to run zSecure from the data sets where you installed it, you can skip this procedure.

You can distribute and run the zSecure software on multiple z/OS images, if these systems have access to the volume where you installed the product. Verify that the following configuration is available.

- The DASD volumes where the software is installed must be online for the z/OS images targeted for the software distribution. The volume that contains the catalog where the zSecure data sets are cataloged must also be accessible from these images.
- The user catalog for the zSecure data sets must be connected to the master catalogs of your other z/OS images, with a correctly defined alias.

**Note:** Before distributing, read [Appendix A, “Site module,” on page 191](#). If you decide to perform the optional step to customize the Site module, you can do so in either of the following ways:

- Customize the Site module before distributing the zSecure configuration data sets so that the same customization is copied to all images.
- Customize the Site module for each image separately after distribution.

For distribution, the actual data set names that you run the software from are usually different from the ones you installed into. For some sites, the data sets to run from also have different names on each image. If you change data set names, use the new names in your configurations and set up data set security. You can use any tool that fits in your storage management policy to copy the data sets. Only the target data sets are distributed.

Do not distribute the configuration data sets themselves because they might contain image-dependent data. For more information about configurations, see [“About zSecure configuration data sets” on page 23](#).

## Enablement of license features

---

### About this task

For each z/OS image where you plan to install zSecure, but where you want to disallow use of certain features, update the parmlib member IFAPRDxx to disable such licensed features.

### Procedure

1. Copy the required PRODUCT statements for zSecure enablement from SCKRSAMP library member CKRZPROD.
2. Paste the statements in the IFAPRDxx member of the active data set for each z/OS image.
3. Update the STATE parameter for each product to reflect the enablement policy for the z/OS image.

If the IFAPRDxx member does not explicitly specify the STATE for a zSecure licensed feature, the feature is enabled.

## APF authorization of the software

---

For most purposes, the program object library containing zSecure must be APF authorized. APF authorization affects zSecure components in the following ways:

- zSecure Collect can access relevant information
- The following components work only if they are run with APF authorization.
  - The CKGRACF program, a component of zSecure Admin and zSecure Visual
  - zSecure Alert
  - The zSecure Command Execution Utility CKX
  - Access Monitor, an optional function of zSecure Admin
  - The RACF Exit Activator used by zSecure Audit, Alert, and Access Monitor for zSecure Admin
  - The zSecure Server
- The CKRCARLA program and the zSecure Audit functions can run without APF authorization. However, with this configuration:
  - You cannot directly issue commands.
  - Using a CKFREEZE data set that was created by a non-authorized zSecure Collect program produces incomplete results.
- The data set that contains the ERBSMFI program (by default, SYS1.SERBLINK) must be APF-authorized. The ERBSMFI program itself does not need APF, but zSecure Collect invokes this program, which is not allowed when ERBSMFI resides in a non-APF library. In many installations, SYS1.SERBLINK is part of the linklist and as such APF-authorized, but when the safer LNKAUTH=APFTAB setting is in effect, you must explicitly include SYS1.SERBLINK, or its local equivalent, in the APF-list.

## TSO and ISPF command tables for zSecure Admin

To run CKGRACF, a component that is used by zSecure Admin and zSecure Visual, update the TSO and ISPF command tables as described in the following sections:

- [“Updating the TSO command table” on page 19](#)
- [“Updating the ISPF TSO Command Table” on page 19](#)

If you wish to run CKGRACF from outside the zSecure ISPF environment, the program must be available from either LINKLIST, STEPLIB, or JOBLIB (just like any other authorized program). Using STEPLIB or JOBLIB has a performance penalty and this method works only when all data sets in your STEPLIB or JOBLIB concatenation are APF-authorized.



## Updating the TSO command table

### Procedure

1. Add the CKGRACF program name to the authorized AUTHCMD and AUTHPGM tables in the SYS1.PARMLIB member IKJTSOxx.

Optionally, you can also add the CKGRACF program name to the AUTHTSF table. Failing to include CKGRACF in the TSO Authorized command table can result in messages CKG905I, CKR962F, or CKX962F.

Figure 3 on page 19 shows a sample of the AUTHCMD NAMES table with the CKGRACF update.

```
/* IKJTSO00: TSO command tables                */
/*                                             */
AUTHCMD NAMES(                               /* Authorized commands: */ +
          CKGRACF                            /* zSecure Admin        */ +
          RECEIVE                             /* TSO base              */ +
          XMIT      TRANSMIT                   /*                      */ +
          LISTB     LISTBC                      /*                      */ +
          LISTD     LISTDS                      /*                      */ +
          SE        SEND                       /*                      */ +
          RACONVRT                          /*                      */ +
          IRRDPI00                          /*                      */ +
          CONSOLE   CONSPROF                  /*                      */ +
          SYNC                                             /*                      */ +
          TESTAUTH TESTA                       /*                      */ +
          PARMLIB)                                     /*                      */ +
```

Figure 3. Sample SYS1.PARMLIB member IKJTSOxx

2. After updating the table, apply the updated version of IKJTSOxx using the TSO command PARMLIB UPDATE((xx)). An IPL is not required.

Alternatively, you can add the CKGRACF program name using the CSECTs IKJEFTE2, IKJEFTE8, IKJEFTAP, and IKJEFTNS. For more information, see the *TSO/E Customization* documentation (SA32-0976).

## Updating the ISPF TSO Command Table

### About this task

If CKGRACF is run from ISPF, its use is logged in the SPFLOGx.LIST by default. The log might include passwords. To prevent this data from being logged, update the ISPF TSO Command Table to include an ISPTCM entry. After adding the entry, you must reassemble the ISPTCM table to apply the changes.

### Procedure

1. In the ISPF TSO command table, add the ISPTCM entry and specify the value for the FLAG:
  - Set bit 2 to indicate authorized command.
  - Set bit 3 to disable logging.
  - Set bit 6 for the command processor.

Bits are numbered from left to right, with the leftmost bit zero. The bits mentioned add up to 50 (decimal) or X'32' (hexadecimal).

Figure 4 on page 20 shows a sample ISPTCM entry.

```

* HEADER
*
*     ISPTCM HEADER
*
* ONE ENTRY TYPE CALL FOR EACH COMMAND IN THE TCM.
* IT IS NOT REQUIRED THAT THE ENTRY NAMES BE IN ALPHABETIC ORDER
*
*
*
*
* OWN ENTRIES
*     ISPTCM FLAG=32,ENTNAME=CKGRACF TSO COMMAND, AUTH, NOLOG
* END CARD. STATEMENTS AFTER THIS CARD WILL BE IGNORED
*     ISPTCM END

```

Figure 4. Sample ISPTCM table

2. Activate the new ISPTCM entry using one of the following methods.

- If the ISPTCM is located in a STEPLIB, exit and reenter ISPF to apply the changes.
- If the ISPTCM is located in the link list, issue the operator command F LLA , REFRESH to apply the changes.
- If the ISPTCM is located in the LPA, IPL the system using the CLPA parameter to refresh the ISPTCM data.

3. After you apply the changes, test the new ISPTCM by including it in a STEPLIB or ISPLLIB.

For more information about ISPTCM, see the *ISPF and ISPF/PDF Planning and Customizing* manual (GC19-3623).

## Making the software available to TSO/ISPF users

If another version of zSecure is already installed on your z/OS images, you can continue using any copies of the zSecure REXX CKR program specified in the default SYSEXEC or the SYSPROC concatenation of your TSO/ISPF users. (CKR was previously called C2R.) By reusing existing copies of the CKR program, you can retain any custom logic and references to your site-specific zSecure configuration files. For that reason, the copy in your CKRPARM data sets is not upgraded automatically.

To ensure compatibility with the current zSecure release, examine the current version of the shipped CKR available in the SCKRSAMP library to determine whether you must copy any new logic into your existing CKR copies.

Change data set references to point to the new zSecure data sets.

- If you have used release-dependent data set names as the target for the zSecure distribution, change the value of the CPREFIX parameter in all copies of CKR and in all zSecure configurations to point to the data sets where the new zSecure resides.
- If you have set up release-independent aliases, redefine the aliases to point to the new zSecure data sets.
- If you have never used zSecure before, copy and adapt member CKR into a data set that is in the standard SYSEXEC or SYSPROC concatenation of your TSO/ISPF users.

A customized version of the CKR REXX that uses the data sets you specified is available in the CKRPARM data set after you run job CKRZPOST. For information, see [“Creating zSecure configuration data sets”](#) on page 25. The required modification is described under [“Assignment of configurations to TSO/ISPF users”](#) on page 27.

Use the ISPF/PDF editor to copy the CKR REXX. In the ISPF/PDF editor, the copy is saved in the same format (fixed-blocked or variable-blocked) as your SYSEXEC or SYSPROC data set. See [Appendix E, “Configuring the ISPF interface,”](#) on page 211.

If the zSecure configuration does not use LIBDEF, you must make the ISPF components available in a different way. For example, you might include them in the TSO logon procedure.

## Making the software available for batch processes

---

To run programs in batch or as a started task, use the zSecure-supplied JCL procedures described in the *User Reference Manual*. These procedures allocate the data sets where the software is installed. Allocation is done using the CPREFIX parameter in either the configuration member C2R\$PARM or a custom copy of that member.

You can run programs in batch using any of the following methods.

- Run the procedures directly from the zSecure-shipped SCKRPROC data set.
- Imbed the SCKRPROC data set in your system proclib concatenation.
- Copy the procedures to your system proclib concatenation.

The system proclib provides an advantage because you must update only one place to apply changes for all JCL. However, the disadvantage is that only one version of a procedure can be effective at a time. For example, when using a shared proclib, you cannot upgrade your images one at a time.

For batch jobs, make the zSecure-supplied procedures available through the JCLLIB statement. Typically, your JCLLIB statement first specifies the data set that contains your configurations followed by the zSecure-supplied SCKRPROC data set. See [“Assignment of configurations to batch jobs and started tasks” on page 27](#).

However, for started procedures (unlike started jobs), z/OS does not support JCLLIB. As a result, you must copy some members from SCKRPROC to a data set that is part of your JES proclib concatenation.

- Do **not** include most procedures, and especially the ones that use C2RC, in your proclib concatenation:
  - C2RC requires the following members. These members might be customized and are dependent on the parameters you specify:
    - C2RIOCMD
    - C2RIOIOC
    - C2RIOSMF
    - C2RIOUNL
    - C2RI1CMD
    - C2RI1IOC
    - C2RI1SMF
    - C2RI1UNL

Normally these customized members are included from your configuration data set, rather than from SCKRPROC.

- You might have multiple zSecure configuration data sets, each with its own versions of these customized members, while the standard JES proclib concatenation can have only one version effective.
- zSecure Alert and the Access Monitor for zSecure Admin must run as started tasks. If you use either of these components, copy the following procedures:
  - C2POLICE and C2PCOLL for Alert
  - C2PACMON for Access Monitor

Procedure C2PRECI is also part of zSecure Alert, but this procedure is normally run as batch job. Do not run this procedure as a started task because it internally uses procedure C2RC.

- The zSecure Server is usually operated as a started task, although it is not required. If you use this component, copy procedure CKNSERVE.
- zSecure Visual is usually operated as a started task, although it is not required. If you use this component, copy procedures C2RSERVE, C2RSLOG, and C2RSTOP.

When copying procedures to your system proclib, you can also modify the procedures if required. For example, you might want to change the CONFIG=C2R\$PARM value, which zSecure ships as a default,

to the value that represents your own configuration member. In particular, when using a shared proclib among z/OS images, consider using a system symbol as the configuration member name or part of the configuration member name. You can then share the procedures and still support having a different configuration for each image.

In addition, the zSecure configurations that are to be used by started procedures or that you want to make available without JCLLIB must reside in a data set that is part of your JES procedure concatenation. See [“Assignment of configurations to batch jobs and started tasks”](#) on page 27.

**Note:** Copying members from SCKRPROC implies that you need to review and possibly update your copies when upgrading zSecure.

---

## Chapter 6. Deployment of the software

In most installations, several z/OS images exist. These images might, for example, separate workloads or isolate development from production. In such environments, it is often desirable to perform the actual software installation process only once and then deploy that software on several images.

To support this method of installing the software once and deploying it on multiple images, in zSecure you can create configuration data sets. These data sets can be used to specify all configuration options for a specific instance of the software in a separate data set. You can also use configuration data sets to specify a new data set name convention for the data sets where the target and distribution libraries are copied during distribution.

You can install a full set of zSecure products and features. After installing, you can use parmlib member IFAPRDxx to specify which products and features are not available on each z/OS image. For example, you might want to disable zSecure Admin and zSecure Audit for RACF on a z/OS image that uses ACF2 as the security manager.

Distribution-oriented installation is supported between images with unlike licenses, or unlike security managers. For example, if zSecure software has already been installed on a z/OS image for RACF, you can use the same software on another z/OS image even if that image uses a different security manager such as ACF2 or Top Secret. In such cases:

1. Perform a distribution to the new image.
2. Create one or more zSecure configuration files to specify the options required to operate zSecure in that environment.

**Note:** Distribution-oriented installation is not supported between z/OS and z/VM. If you want to use zSecure on both z/OS and z/VM platforms, you must install the software separately for each platform.

---

### About zSecure configuration data sets

Configuration data sets represent the zSecure configuration for an image and determine how the software operates on the image. For example, zSecure configuration data sets can specify what zSecure features are available as well as the data set names for the input data sources.

zSecure configuration data sets are the only data sets that are different between images. Using these data sets, you can create configurations for the following purposes:

- Special-purpose configurations for processes such as zSecure Alert, Access Monitor, Compliance reporting, and Visual Server.
- Separate configurations to deploy zSecure on different z/OS images.
- Configurations for user groups that require different input data or that restrict access to specific zSecure components.

zSecure configuration data sets are stored in partitioned data sets that are not part of the installed software. As a result, these data sets are not updated when the software is upgraded or reinstalled; therefore, you can maintain custom configuration settings across upgrades.

Table 3 on page 24 lists the configuration data sets that can be customized for each deployment.

Table 3. zSecure configuration data sets

Data set name	Description
<i>your.prefix</i> .CKRPARM	This data set contains the main configuration member C2R\$PARM. You can create other configuration members as well.  The CKRPARM data set also contains the REXX CKR (adapted to your naming convention). Copy and adapt this member to a SYSPROC or SYSEXEC data set. See <a href="#">“Making the software available to TSO/ISPF users”</a> on page 20.
<i>your.prefix</i> .CKACUST	This data set contains the 'compliant authorized ID population' members used for option AU.R - Rule-based compliance evaluation. This option is only available for zSecure Audit. This dataset must be a fixed blocked 80 formatted PDS or PDSE. C2R\$PARM must define which data set is to be used as the CKACUST data set, or it must be set up manually through zSecure option Setup Command files (SE.8).
<i>your.prefix</i> .CKACUSV	This data set contains the 'compliant authorized ID population' members used for option AU.R - Rule-based compliance evaluation. This option is only available for zSecure Audit. This dataset must be a variable blocked formatted PDS or PDSE. The advice is to use a logical record length of 32752. C2R\$PARM must define which data set is to be used as the CKACUSV data set, or it must be set up manually through zSecure option Setup Command files (SE.8).
<i>your.prefix</i> .CKRPROF	ISPF tables. If you intend to use the ISPF transactions under SE.D, specify this data set or a copy of it as the PROFDSN parameter in the C2R\$PARM member. See <a href="#">“Setup of default options for user groups (Setup menu)”</a> on page 211).
<i>your.prefix</i> .CKRJOBS	IBM-supplied jobs, adapted according to your data set naming convention.

These configuration data sets are usually created using the CKRZPOST job available in the CKRINST library or the SCKRSAMP library. This job allocates, fills, and updates the CKRPARM and CKRJOBS data sets. It also allocates an empty CKRPROF data set that can be used to customize the ISPF interface.

In the documentation, the configuration data sets are usually referenced by the low-level qualifier (for example, CKRJOBS). You can choose any data set names you like, but retaining the default low-level qualifier reduces your need to adapt configurations or override JCL. Each deployment has its own value for *your.prefix*. Use one of the following ways to supply your own value for the prefix:

- Edit the zSecure installation job CKRZPOST.
- Specify values in the C2\$PARM member of your installation library.

Member C2R\$PARM is the default starting point of a configuration. To be usable both in the batch and the ISPF interface, this member uses JCL SET statements, which the ISPF interface interprets. See the following example:

```
// SET CPREFIX='CKR'
// SET VOLSER=
// SET PROFDSN='CKR.IP01.CKRPROF'
// SET SYS=IP01
```

See Appendix D, “Configuration parameters and members,” on page 205 for the full syntax of the configuration member.

Usually, you create new zSecure configurations only when you are performing the following tasks.

- Installing zSecure for the first time.
- Setting up zSecure on a new z/OS image.
- Setting up a special-purpose configuration for processes such as Access Monitor.
- Setting up zSecure for a new user community.

zSecure provides a sample configuration that you can use in SCKRSAMP(C2R\$PARM). However, this sample is intended for only the most basic installation scenario: installing a single z/OS image without distributing the software or customizing the configuration.

To configure zSecure for different z/OS images or different user groups, create configurations for these z/OS images or user groups. For information, see [“Creating zSecure configuration data sets” on page 25](#). If you are upgrading to a new version or installing zSecure again, see [“Maintenance of existing zSecure configuration data sets” on page 26](#).

The following sections provide information about how to create configurations for different z/OS images or different user groups such as RACF administrators and RACF auditors. See the following information:

- [“Creating zSecure configuration data sets” on page 25](#)
- [“Customization of zSecure configuration data sets” on page 26](#)
- [“Maintenance of existing zSecure configuration data sets” on page 26](#)
- [“Assignment of configurations” on page 27](#)

## Creating zSecure configuration data sets

---

### About this task

You can create zSecure configurations for different z/OS images or different user groups such as RACF administrators and RACF auditors as described in this procedure.

**Note:** You can also create special-purpose configurations for zSecure components such as zSecure Alert, zSecure Admin, Access Monitor, and Visual Server. For information about creating configurations for these components, see the setup documentation for each component.

### Procedure

1. Review [“About zSecure configuration data sets” on page 23](#) to learn about the zSecure configuration data sets and the zSecure-supplied jobs used to manage them.
2. (Optional) Set up the global update members CKRZUPDI and C2RIISPF.

The values used to create the zSecure configuration data sets using the CKRZPOST post-installation job are based on the values specified in the global update member CKRZUPDI and C2RIISPF. If you did not run global update during the installation process, do so before running job CKRZPOST. See [“Customization of the installation parameters” on page 11](#).

3. Select the high-level qualifier for the zSecure configuration data sets.

Remember that the naming convention that you establish for the actual software might not be the best choice for the configuration data sets. Instead, consider using a high-level qualifier that indicates the z/OS image and group of users for which the configuration data sets are intended. Because your configuration data sets are supposed to persist across zSecure upgrades, do not imbed a qualifier in the data set name that represents a version or release.

4. Follow the instructions in the CKRZPOST job to customize the job for your installation.

Make sure to update the following parameters:

#### **INSTLIB**

Specify the high-level qualifier for the zSecure installation library data sets where the zSecure software runs.

## YOURPFX

Update the parameter with the high-level qualifier you want to use for the zSecure configuration data sets created using CKRZPOST.

If you do not change the default value for YOURPFX, then the configuration data sets created using the CKRZPOST job use *your.prefix* as the high-level qualifier. CKRPROF does not supply the prefixes because you can create multiple configurations for a single copy of the installed software.

Comment out the DD-statements for the configuration data sets that do not require customization.

### 5. Run CKRZPOST to create the zSecure configuration members.

If you have run job CKRZUPDZ earlier during installation, job CKRZPOST might end with a return code of 4 because some data set updates were already completed during the CKRZUPDZ run. You can ignore this return code.

### 6. Create configurations for individual user communities or z/OS images. See [“Customization of zSecure configuration data sets”](#) on page 26.

## Customization of zSecure configuration data sets

---

After you have created the zSecure configuration data sets, you can customize the members and create copies to be used for different user communities or different z/OS images. For example, to configure zSecure to be used by different groups of users such as RACF administrators and RACF auditors on the same image, create copies of member C2R\$PARM. Then, configure each member separately. When copying and configuring the members, the following rules and guidelines apply:

- Do not use member names that begin with C2R or CKR.
- For zSecure Admin users, all configuration members within the same data set share the C2RSMUMA, C2RSMUMH, and C2RSMUMP members that specify zSecure Admin settings used to create new RACF userids. See [“Configuring zSecure Admin to create new userids in the RACF database”](#) on page 222. To specify different values for these members:
  1. Copy the entire CKRPARM data set to the system from which you want to run zSecure Admin.
  2. Update the CKRPARM members as required.
- You can create multiple copies of CKRPROF to customize the ISPF interface for different z/OS images or different user communities.
- The CKRJOBS data set is intended to be further customized. For example, you might specify different configuration members depending on the environment where each job is to run. For this reason, consider creating multiple copies of the CKRJOBS data set.
- For zSecure Audit users that use option AU.R - Rule-based compliance evaluation: Remove the comment from the SET CKACUST and SET CKACUSV parameters and update the data set name.

## Customize jobcards to be used when reports are created as a batch job

---

When selecting to run a report as a batch job, zSecure creates a jobcard if no jobcard statements were provided.

zSecure installation can define default job cards by copying member C2RSDFLT from the SCKRSLIB library to the zSecure configuration data set and customize it.

## Maintenance of existing zSecure configuration data sets

---

The zSecure configuration data sets are customer assets. They are stored in partitioned data sets that are not part of the installed software. For example, when you run Setup Default (SE.D), any customized interface settings are written to *your.prefix.SCKRPROF*. Because the zSecure installation process does not automatically update these data sets, you can maintain the configuration settings across upgrades.



If you are upgrading to a new version, start with your existing configuration data sets or copy the configuration data sets from a previous release. Then, manually compare these configurations against the C2R\$PARM member in the SCKRSAMP library to decide whether any new parameters are applicable. The same applies to PROFDSN data sets, as explained in [“Setup of default options for user groups \(Setup menu\)”](#) on page 211.

If the sample configuration data sets exist, they have probably already been customized for your installation environment. To ensure that the zSecure configuration data sets are up to date, manually compare your configuration against the SCKRSAMP C2R\$PARM member to determine whether you must update your existing zSecure configuration data.

## Assignment of configurations

---

After you have created the zSecure configurations required for your installation, assign the configurations to TSO/ISPF users, batch jobs, and started tasks so that they are available for users and system processing. You must also establish security for each configuration to control access to product functions and data. For instructions, see the following sections.

- [“Assignment of configurations to TSO/ISPF users”](#) on page 27
- [“Assignment of configurations to batch jobs and started tasks”](#) on page 27
- [Appendix B, “Security setup for zSecure,”](#) on page 193

### Assignment of configurations to TSO/ISPF users

Configurations are assigned by the copy of the REXX exec CKR that is used to start the ISPF interface. The CKR exec starts C2REMAIN using the configuration data set name and member name as parameters. You can create a different copy of CKR for each z/OS image and for each group of users. Alternatively, you can adapt your copy of the CKR REXX to dynamically select the configuration and to pass parameters that override the configuration member.

For example, you can create a CKGHELP REXX exec, intended for simple administrative tasks, that adds only the overriding parameter STARTTRX(MENU(RA.H)). This exec would correspond to the single-panel HelpDesk options. Similarly, you can create a CKRQ REXX exec, intended for simple administrative tasks, that adds only the overriding parameter STARTTRX(MENU(RA.Q)). This configuration provides users with access to the Quick User Administration option described in the *zSecure Admin and Audit for RACF User Reference Manual*.

### Assignment of configurations to batch jobs and started tasks

- Jobs are submitted by zSecure ISPF transactions inherit the software location (SCKRLOAD and SCKRSAMP) from the TSO session. When applicable, the transaction can generate NJE routing and system affinity. The user might be prompted to specify this information.
- For batch jobs that are not submitted from zSecure panels, specify the configuration data set in the JCLLIB statement and INCLUDE the required member. Often, you must supply NJE routing and system affinity. See [Chapter 8, “Setup for production,”](#) on page 31.
- You can also create a configuration member, such as C2R\$PARM, in a data set that is part of the procedure library for the Job Entry Subsystem. If like-named configuration members differ across z/OS images, you need system affinity to ensure that the JCL is converted on the same z/OS image where the job is to run. The *MVS JCL Reference* documents how to specify system affinity for JES2 and JES3.
- For started procedures, JCLLIB is unavailable. Therefore, whenever you set up a started procedure, you must copy the configuration member it uses to your procedure library.
- Additional customization is described in the *IBM Security zSecure Admin and Audit for RACF: User Reference Manual*.



---

# Chapter 7. Verification of the installation

## Base ISPF interface functions and menu configuration

---

Under the ISPF/PDF command option, invoke the REXX you created under [“Making the software available to TSO/ISPF users”](#) on page 20 to display the IBM Security zSecure primary menu. The primary menu is configured based on your licenses and authorization, so the menu you see might be different from the one shown in the *User Reference Manual*.

## Checking the zSecure Collect function and the base batch operation of zSecure

---

### Procedure

1. From the IBM Security zSecure primary menu, type SE.1 (Setup files).
2. On the Setup files panel, remove the selection from all selected input sets.
3. Enter SE.2 (Setup - New files) to create new CKFREEZE and UNLOAD data sets.
4. You are prompted for allocation parameters. As a raw approximation, specify:
  - 2 MB per online DASD-volume for CKFREEZE data sets.
  - The same size as your security database for UNLOAD data sets.
5. Use the REFRESH command to submit a batch job that fills these data sets. Make sure that the job runs under a user id that has sufficient authorization. After the job finishes, examine the output for error messages.

## Functions to display reports

---

After successfully submitting the job to add data to the CKFREEZE and UNLOAD data sets, you can use reporting functions like AU . S, AU . V, and RA . U (according to your license). See the *User Reference Manual* for information about using the reporting functions.

## CKGRACF command to verify security resources

---

**Note:** This procedure is required only if you use IBM Security zSecure Admin.

From TSO line mode, run the following commands:

```
alloc reuse file(system) dataset(*)
ckgracf show myaccess
```

The command results provide a list of security resources, showing your current access level and the profile on which each access level is based.

## Verification of ACF2 reporting

---

**Note:** This step is required only if you use the zSecure Audit for ACF2 component.

To verify that zSecure Audit for ACF2 and ACF2 are in total agreement on the contents of the ACF2 database, run the C2ZAJIVP job.



---

# Chapter 8. Setup for production

## SCKRSAMP and SCKRJOBS data sets

---

As described in “Assignment of configurations to batch jobs and started tasks” on page 27, copy the jobs you need from the SCKRSAMP and SCKRJOBS data sets to a data set of your own, such as a data set that your job scheduling system uses. Do not directly edit the jobs in SCKRSAMP and SCKRJOBS because these data sets are maintained by SMP/E. If you use more than one z/OS image, editing these data sets also violates distribution-oriented installation.

A customized copy of the SCKRSAMP library, the CKRINST library, is created during the software installation process as described in “zSecure-supplied installation jobs” on page 11.

## Capacity planning information

---

The IBM Security zSecure product suite consists of multiple products and components. This topic uses information that was previously available in various manuals and application notes to help you determine the system resources required for running these applications.

### Introduction

This topic is divided into multiple sections for the different products. Most component sections are split into several subsections for the different types of system resources that are used for a product. The following types of system resources are discussed:

- DASD storage required for storing the data used or created by the program
- Virtual storage required for running the program
- CPU time used by the program
- Network data transport

### DASD storage

Data storage on DASD (disk) as used by zSecure is mainly for the following types of data:

#### **CKFREEZE**

This type of data set contains information about the system and resources. This includes many system control blocks, names of data sets and UNIX files, and also (part of) the contents of some data sets.

The CKFREEZE data set is created by the zSecure CKFCOLL program. The information is used by many zSecure products.

#### **RACF Data**

zSecure Admin and Audit need information from a RACF source. This source can be an existing RACF database or an UNLOAD data set. UNLOAD data sets created by the zSecure CKRCARLA program contain a proprietary format snapshot of the RACF database. They are similar in size to the used portion of the RACF database. The RACF-Offline component of zSecure Admin uses copies of the system RACF database. The size of the Offline RACF database is dependent on your usage.

All sensitive fields are always replaced by asterisks (masked) in an UNLOAD file and, therefore, are not reproduced during copy or recreate operations.

#### **ACF2 Data**

zSecure Audit for ACF2 needs information from the ACF2 databases. This can be an existing set of BACKUP databases or an UNLOAD data set. UNLOAD data sets created by the zSecure CKRCARLA program contain a proprietary format snapshot of the ACF2 databases. They are similar in size to the used portion of the ACF2 databases.

## SMF Data

This data is not unique to zSecure. SMF records are created to provide information about the environment and events in the system. They contain information useful for performance, planning, and auditing purposes. Data is collected about many different types of events. zSecure Audit can use SMF data as input to report about historic events in the system.

## Access Data

These data sets contain information about recorded access. They are similar in content to certain types of SMF records. The Access Monitor data sets are created by the zSecure C2PACMON program. The data can be analyzed by zSecure Admin.

The types of data unique to zSecure are the CKFREEZE data sets, security database UNLOAD data sets, and the Access Monitor data sets.

## Different types of CKFREEZE data sets

The CKFREEZE data sets are created by the CKFCOLL program and used for various purposes. Not all programs that use CKFREEZE data sets need the same amount and same type of information. For this reason, several types of CKFREEZE data sets, shown in the following list, are distinguished. In order to collect all required information, the CKFCOLL program needs to run APF authorized. Limited support is provided for running the program in non-APF mode. For more information about running APF and non-APF authorized, see the chapter about zSecure Collect for z/OS in the *User Reference Manual* for your product.

### Full-size

This type of CKFREEZE data set is created when you do not specify any parameters or selection criteria and if you have a license for zSecure Admin or zSecure Audit. If you have a license only for zSecure Admin, the collection program automatically excludes certain auditing-specific information. The Full-size CKFREEZE data set contains information from all system and user catalogs, backup, migration and tape catalogs, all VVDSs, and all VTOCs. It also contains the directory information from all APF, linklist, lpalib, parmlib, and proclib data sets. This type of CKFREEZE data set also has information about all files in UNIX HFS or ZFS data sets. All program and transaction information for all CICS and IMS systems is collected. Db2 subsystem information like tables, packages, and other information is collected.

Other information included in a Full-size CKFREEZE data set is the data needed for detailed system auditing.

Because this CKFREEZE data set has much information, it can be used for all supported auditing and reporting functions. However, because it contains much information, it also has as a disadvantage that it might take a long time to collect and process all data. Therefore, the Full-size CKFREEZE data set is primarily used only for auditing and full system analysis.

This type of CKFREEZE data set is the default. Creating it requires an Admin or Audit entitlement. No parameters need to be specified.

### Full-size without shared DASD

This CKFREEZE data set has all information as described for the Full-size CKFREEZE file with the exception of catalog, VVDS and VTOC information for volumes that are defined as shared between multiple systems.

Because this type of CKFREEZE data set lacks certain information, use it only in combination with a Full-size CKFREEZE data set. Processing this type of CKFREEZE data set might also take considerable time. It is primarily used for auditing and full system analysis of shared data (sysplex) environments.

Creating this type of CKFREEZE data set requires an Admin or Audit entitlement. The following parameter must be specified:

```
SHARED=NO
```

### Regular Admin

This CKFREEZE data set has all information that is needed for regular RACF administration purposes. Most catalog, VVDS, and VTOC information is not included. Information from the MASTER catalog is included, but information about UNIX files and system library contents is absent.

Because this type of CKFREEZE data set has a reasonable size, it can be processed quickly. However, because it lacks information from the user catalogs, it is unsuited for deleting TSO and batch userids. It can be used for copying existing userids, and defining alias entries in the master catalog. The absence of detailed information makes this type of CKFREEZE data set also not suited for detailed auditing and system analysis. Simple audit reports that do not require detailed data set or UNIX file information can be created.

Creating this type of CKFREEZE data set requires an Admin or Audit entitlement. It can be created by specifying the parameters

```
CAT=MCAT , VVDS=NO , VTOC=NO , UNIX=NO , BCD=NO , MCD=NO ,  
RMM=NO , TMC=NO , IMS=NO , CICS=NO , DB2=NO , MQ=NO
```

### Regular Audit

This CKFREEZE data set has the information that is needed for most detailed auditing and system analysis. Most catalog, VVDS, and VTOC information is not included. Information from the MASTER catalog as well as information about UNIX files is included.

The Regular Audit CKFREEZE data set has more information than the Regular Admin CKFREEZE data set. Processing takes more time, but is still faster than processing Full-size CKFREEZE data sets. Most detailed auditing reports are available, except those that require resource information like data sets, IMS, and CICS resources. Because for most auditing situations, the time needed to create the report is less important, Full-size CKFREEZE data sets are often preferred.

Creating this type of CKFREEZE data set requires an Admin or Audit entitlement. It can be created by specifying the parameters

```
CAT=MCAT , VVDS=NO , VTOC=NO , BCD=NO , MCD=NO , RMM=NO , TMC=NO , IMS=NO , CICS=NO , DB2=NO , MQ=NO
```

### Library Analysis

This is a special purpose CKFREEZE data set. It has checksum information for the specified data sets. Because calculating checksum information is a time consuming process, you create this type of CKFREEZE data set only when you want to do library analysis. The library analysis process uses several CKFREEZE data sets from multiple points in time. You probably want to exclude other information that is normally included in Full-size CKFREEZE data sets (catalog, UNIX files).

Creating this type of CKFREEZE data set requires an Audit entitlement. It can be created by specifying the parameters CHECK=YES possibly in combination with parameters and keywords to suppress data that is not needed:

```
CAT=MCAT , DASD=NO , TAPE=NO , SWCH=NO , RMM=NO , TMC=NO  
UNIX=NO , PATH=NO , SMS=NO , IMS=NO , CICS=NO , DB2=NO , MQ=NO
```

### Mini

This is the smallest size of CKFREEZE data set that still contains sufficient information for most RACF queries. It has only information that is available in-storage. Example information that it contains is information from RACF control blocks (CDT, dynamic parse, system options), SMF options, and lists of important system data sets. Its size is usually approximately 1 MB.

This type of CKFREEZE data set is primarily intended to provide information for remote queries.

Creating this type of CKFREEZE data set requires an Admin or Audit entitlement. It can be created by specifying the parameters

```
IO=NO , SMS=NO , TCP/IP=NO , MOD=NO , CICS=NO , IMS=NO , NJE=NO , S=V=NONE , NOXMEM , DB2=NO , MQ=NO
```

### Special purpose

There are several special purpose CKFREEZE data sets that are used by specific components; for example, the zSecure Alert product. Such special purpose CKFREEZE data sets have a dedicated content, and cannot be shared with other applications.

Creating this type of CKFREEZE data set usually requires an Alert, Admin, or Audit entitlement. The parameters are dependent on the specific information needed.

## **Space requirements for CKFREEZE data sets**

The required space for a CKFREEZE data set is dependent on the options used during data collection. The following list summarizes the amount of space required for each type of information. As a rule of thumb, you can use these amounts:

- 1 MB base size
- Size of the system and user catalogs
- Size of the DFHSM MCDS, BCDS and OCDS, or of the DMS catalog
- Size of the DFRMM control data sets or the TMC catalog
- 2 MB per online DASD volume
- 2 MB per gigabyte HFS/ZFS space
- 1 MB per 5000 IMS or CICS transactions or programs

In a formula:

$$\text{Size(MB)} = 1 + C + H + T + 2*D + 2*U + O/5000$$

where:

C = size of the system and user catalogs

H = size of DFHSM or DMS catalogs

T = size of tape catalogs

D = number of online disks

U = gigabytes of HFS/ZFS space

O = number of IMS and CICS transactions and programs

To manage the size of your CKFREEZE data sets, see [Appendix G, “Troubleshooting: Large size of CKFREEZE data sets,”](#) on page 227.

## **Types of security data (RACF or ACF2)**

zSecure Admin and zSecure Audit can both use information from the active RACF or backup ACF2 database. No extracts or copies are required. You can also use a private backup copy of the RACF or ACF2 database. If you use a private backup copy, plan for DASD space equal to the size of your current database.

You can also use a proprietary format UNLOAD copy of the RACF or ACF2 database. Such an unload copy can be used as frozen input to generate Admin and Audit reports. The size of an UNLOAD copy is approximately the same as the used part of your database. The UNLOAD database has as advantage that all sensitive fields (like passwords) have been removed from the UNLOAD copy.

The RACF-Offline component of zSecure Admin uses copies of the system RACF database. The size of the RACF database is dependent on your usage. Often, the RACF database used for offline usage has the same size as the active system RACF database.

## **Access Monitor data**

The zSecure Access Monitor is part of zSecure Admin. It is available only for RACF systems. The Access Monitor collects information about most access events and some profile management events. Information is saved in so-called Access Monitor data sets. The collected information is kept in several data sets:

### **Daily collection data sets**

These data sets are created by the Access Monitor started task and usually exist only during the time that Access Monitor is running. They contain multiple blocks of records collected during a measurement period. The measurement period is the same as the SMF interval, which can be specified in SMFPRMxx and which has a default of 30 minutes.



### Daily consolidated data sets

These data sets are also created by the Access Monitor started task. They contain a single block of consolidated records of a single day. Consolidation is done automatically once a day.

### Site specific consolidated data sets

It is possible to consolidate multiple daily consolidated data sets into a single data set comprising the information for one week, one month or even one year. Ideally, this consolidation process trends towards a consolidation efficiency of 100%. This means that adding an additional period does not increase the required space for the consolidated data. In practical environments, the consolidation process trends towards an efficiency of 90% or even lower. This means that adding an additional period increases the space required for the consolidated data by 10% or more of the size of the additional data.

The consolidation process retains the count and last occurrence for all different types of events. The space required for Access Monitor data is very dependent on your environment. For example, a single user accessing the same data set 1000 times a day consolidates into a single event record, while 1000 users each accessing the same data set only once, consolidate into 1000 event records.

Without actually implementing Access Monitor, there is no simple method to estimate the amount of DASD space required for the Access Monitor data sets. Running SAFTRACE for RACROUTE could give an estimate for the number of events, but there is no report that summarizes across the different resources, users, jobnames, and request options. The best approximation is to use the same size as the current size of the SMF offload data sets. During an initial startup period, the data set sizes can be monitored and adapted to better match the required space.

### Use of zEDC to reduce DASD requirements

You can specify an appropriate DATACLAS for zSecure data sets to reduce DASD space requirements. This is effective for CKFREEZE and ACCESS data sets. Specifying the DATACLAS can be done in applicable ALLOC commands, JCL, or automatically by using a DATACLAS ACS routine. For background information about zEDC, see [“zEnterprise data compression \(zEDC\) for zSecure”](#) on page 72.

### Virtual storage

Virtual storage is needed while running the program. The amount of virtual storage needed is dependent on the amount of data processed during the report or analysis. For most types of reports, the virtual storage needed is of the same order as the size of the output report. For example, if you are generating a report of four million SMF records with detail information, the program needs sufficient space to retain all unique field values that occur in these records.

Special processes, like consolidation of Access Monitor data sets, are described in their own sections.

The CKRCARLA program is the main program that is used in most zSecure products. This program is a router program that calls a release and hardware-dependent module, currently either CKR4Z196 or CKR8Z12:

- The CKR4\* program runs in 31-bit mode using a 4-byte storage model.
- The CKR8\* program runs in 64-bit mode with an 8-byte storage pointer model.

Instead of storage below the 2GB boundary, a CKR8\* program can be expected to use up to twice as much storage above the 2GB boundary compared to the storage used by a CKR4\* below the 2GB boundary.

The maximum amount of storage above the 2GB boundary is set by the MEMLIMIT parameter, and not by the REGION parameter (except that REGION=0 requests MEMLIMIT=NOLIMIT). The CKR0039 message shows what the actual limit is and what facility actually restricted it from what was requested (for example, site exit IEFUSI or a PARMLIB member). Because TSO users cannot dynamically specify the MEMLIMIT parameter, you might need to change exits, PARMLIB settings, or logon procedure keywords if TSO users require more storage above the 2GB boundary. For example, create a TSO logon procedure TSOLARGE with a modified EXEC statement and use this for running zSecure:

```
//TSOLARGE EXEC PGM=IKJEFT01,DYNAMNBR=512,MEMLIMIT=8G
```

## CPU time

The amount of CPU time needed for running the zSecure programs can be expressed in two different ways. One way is used for those situations where a single instance of the program is used once to generate a single report or perform a single analysis. The other is used for long running tasks that collect and process information in real time. Details are described in the sections about the individual components.

## Network load

zSecure Admin and Audit do not generate any significant network load. However, both applications also provide the option to do remote reporting and analysis using the zSecure server (program CKNSERVE). If you use the zSecure server to access remote data, the network load is dependent on the data needed for the report. Each call of CKRCARLA results in the transfer of all data required for the selected reports. For typical audit reports, this often involves the entire CKFREEZE data set and the entire RACF database. If you do not specify a CKFREEZE data set, a mini-CKFREEZE is used. The size of a mini-CKFREEZE is about 1 MB. This data set is always transferred in full.

For RACF reports, the zSecure Server transfers only the number of profiles needed for the report, or transfers the entire security database. This is dependent on the selection criteria used in the query and the information that is to be included in the report. For ACF2 and SMF reports, all data is transferred, and selection is done in the client application.

The zSecure Server uses point to point connections using TCPIP. It uses a single listening port that can be specified during configuration, and one ephemeral port for each configured partner server. The server must be active on all systems from which you want to retrieve information, or to which you want to send commands.

## zSecure Admin

The zSecure Admin product has multiple subcomponents; each subcomponent has its own storage characteristics. zSecure Admin and Audit can exploit services provided by the zSecure server. The resulting network load is described in [“Network load”](#) on page 36. See the paragraphs that follow for information about possible DASD and virtual storage and required CPU time.

If you want to collect profile usage information, the zSecure Access Monitor started task must run on all systems where those profiles can be used.

## DASD storage

DASD usage for zSecure Admin and Audit falls in the categories described in the following sections.

### ***CKFREEZE data sets***

For daily zSecure Admin usage, a CKFREEZE data set of type "Regular Admin" can be used. You need one for each system that you are managing. If you are using shared DASD, only one of the CKFREEZE data sets needs to be created with the SHARED=YES parameter.

### ***Storage in the RACF database***

Additional space is required for queued and timed commands. In most situations, the additional space in the RACF database is negligible.

### ***Unload of RACF database***

An UNLOAD copy of the RACF database can be used as frozen input to generate Admin and Audit reports. The size of the unload is approximately the same as the used part of your RACF or ACF2 database. You can retain as many unload data sets as required.

All sensitive fields are always replaced by asterisks (masked) in an UNLOAD file and, therefore, are not reproduced during copy or recreate operations.

## ***Access Monitor data sets***

The size of Access monitor data sets can vary significantly depending on the environment. If you want to retain information for a long period, a significant amount of data can be accumulated and retained. Some of this data can reside on tape, but most data is accessed in parallel, which requires the Access Monitor data sets to be on multiple tapes and to be mounted concurrently. The consolidation process can reduce the amount of data, but tailoring the configuration process to the organization's needs is required.

## ***Copy of RACF database***

Copies of the RACF database are required for using the Offline component. The size and number are dependent on your usage. Often, the Offline RACF database is a copy of the System RACF database. A copy of the RACF database can also be used as frozen input to generate Admin and Audit reports.

## **Virtual storage**

For regular RACF or ACF2 based queries, the amount of virtual storage needed is of the same order as the size of the output report. Use of applicable SELECT statements can reduce most reports to a manageable size.

## ***Resource consumption for processing of Access Monitor data***

Reporting about Access Monitor events can be done in two types of reports: Reports based on the profiles in the RACF database, or reports based on the recorded events. The reports based on the profiles in the RACF database are limited in size and require a limited amount of virtual storage, similar to other RACF reports. The reports based on the recorded events can become very large, depending on the different types of events, and the level of consolidation of the input data. Consequently, the amount of virtual storage required for the reports can also become very large. Reports that require 1 GB virtual storage are not uncommon. You can significantly reduced virtual storage requirements by careful selection of the type of events, users, profiles or resources to be included in the report.

Consolidation of Access Monitor data sets can also require a large amount of virtual storage. To overcome size limitations, the internal format of Access Monitor data sets was recently changed. The two formats are referenced by the release number when they were introduced, as the 1.11 format and the 1.13 format. Consolidating data in 1.11 format requires processing and retaining all input data in virtual storage. Use of the 1.13 format enables writing records to the output data set directly from the start of the consolidation process, without the need to retain the record in storage for the entire duration of the program. With the 1.13 format, the consolidation process can run in region sizes of 32 MB or less.

The 1.13 format for Access Monitor data is used for all new Access Monitor data sets. Existing 1.11 format Access Monitor data and new 1.13 format Access Monitor data can be mixed for reporting and analysis purposes. When you want to consolidate 1.11 format with 1.13 format data, you first need to convert the existing 1.11 format data sets to the 1.13 format. The conversion process has similar storage and CPU requirements as the previous consolidation process. After conversion, data sets can be consolidated using the newer, more efficient consolidation process.

## ***Resource consumption for collection of Access Monitor data***

The started task that collects Access Monitor data needs sufficient buffer space to retain information about all events that occur during a measurement interval. The buffer space is located in the private area of the Access Monitor started task. The default measurement interval is 1 minute. The amount of storage required is dependent on the number of events per minute. For example, if 1000 RACF access events occur per second, buffer storage space can be calculated as  $1000 \text{ events/second} * 60 \text{ seconds} * 100 \text{ bytes} = 6 \text{ MB}$ . Usually, the Access Monitor can run within a region size of 32 MB or less.

## **CPU time**

The amount of CPU time needed for running the zSecure programs needs to be presented in either of two different ways. The first is used for those situations where a single instance of the program is used once to generate a single report or perform a single analysis. This applies to the interactive and batch use

of zSecure Admin and zSecure Audit. The second is used for long running tasks that collect and process information real time, like the zSecure Admin Access Monitor.

## **zSecure Admin**

The CPU time needed for creating interactive or batch reports using zSecure Admin depends on the size of the data to be analyzed and the size of the resulting report. Typical reports take a few seconds to create. Large reports might take several minutes. Due to the amount of data, consolidating 1.11 (old) format Access Monitor data might take a significant amount of CPU time (tens of minutes). Similarly, reporting about large amounts of collected Access Monitor events can also require a similar amount of CPU time.

### **Access Monitor Started Task**

The long running process to collect Access Monitor events also requires CPU time. This CPU time is hard to express in absolute terms, due to the wide range in processor speeds and the number of events. A CPU-independent measure of the time needed to collect and record the information is the number of CPU service units (SU). Collecting and recording a single event requires approximately 1 standard CPU Service Unit (SU). In this context, a standard SU uses only the hardware-specific CPU factor, and not the installation Service Definition Coefficient (SDC).

You can correlate CPU service units to CPU time using the SU/SEC constant as defined for your processor and the CPU service definition coefficient as specified in your IPS or WLM configuration. The number of SUs as reported for your address spaces and system are multiplied by the Service Definition Coefficients (SDC). The default value for the SDC is 10. See the following example:

- Assume that your installation specified the default value for the CPU SDC and zero for the IO and MSO SDCs.
- Your application runs on an IBM zEnterprise 114 model Z01 (2818-Z01). The CPU model factor for this processor is 40100.2506.
- Your application currently causes 20 RACF events.
- Your application currently uses 2 seconds CPU time.
- The total number of service units reported for your application will be  $10 \text{ (SDC)} * 40100 \text{ (CPU factor)} * 2 \text{ (sec)} = 802000$  service units.
- After starting Access Monitor, the amount of CPU time needed to collect and record the information for these events is  $20 \text{ (events)} * 1 \text{ (SU)} / 40100 \text{ (CPU factor)} = 0.0005$  seconds.
- The total number of service units reported for your application will be  $802000 \text{ (base)} + 20 \text{ (events)} * 1 \text{ (SU)} * 10 \text{ (SDC)} = 804000$  service units.

If you are collecting information about UNIX Access events, additional CPU time is spent in the C2PACMON address space. This time is needed to convert the application-specified path names into absolute path names. The total time that is needed for this conversion depends on the number of different path names that are collected per Access Monitor interval. Each conversion takes about 2 standard Service Units CPU time. Using the same example as before:

- Assume that your application generates 1000 different UNIX events.
- When collecting UNIX events, C2PACMON uses additional CPU time. This amounts to  $1000 \text{ (events)} * 2 \text{ (SU)} / 40100 \text{ (CPU factor)} = 0.05$  seconds CPU time.

## **zSecure Audit**

The zSecure Audit product provides functions for reporting about information in the security database (RACF or ACF2), reporting about events that occurred in the system (SMF), and detailed analysis of the security environment. zSecure Audit can exploit services provided by the zSecure Server. The resulting network load is described in the previous section. Data set sizes, virtual storage needs, and CPU time requirements are discussed in the following paragraphs.

## DASD storage

Data storage on DASD is mainly for CKFREEZE data sets. Additional storage might be needed if you are generating reports from multiple non-shared systems, and you are not using the zSecure Server. In that case, you need DASD space for copies of the RACF or ACF2 data bases from the other systems.

### **CKFREEZE**

This information includes many system control blocks, names of data sets and UNIX files, and also (part of) the contents of some data sets. The information is used by many zSecure products.

If you want to perform library change analysis, you also need CKFREEZE data sets with checksum information. Several of these data sets might be needed.

The typical size of a CKFREEZE database is dependent on the number of online DASD volumes and the number of UNIX files.

#### *Full-size CKFREEZE*

For a shared DASD environment, you need a CKFREEZE data set for each system. You can reduce the total amount of DASD required by using some of the parameters described in the following sections.

#### *SHARED=YES/NO*

If you are using shared DASD, you can reduce the required space by specifying the SHARED parameter when creating CKFREEZE data sets. Use the parameter SHARED=YES on exactly one of the systems that share DASD. For all other systems, specify SHARED=NO. This ensures that data from shared disks is collected only once, while data from non-shared disks is also collected.

#### *BCD=NO*

The Backup Control Data set data is currently used only for determining if discrete data set profiles need to be removed. These discrete data set profiles are processed in various **AU.V** - Verify functions and are reported in the RACF profiles report (**RA.3.1**). If you do not use any discrete data set profiles or run the RACF profiles report often, you can disable this function.

This information is used for the VERIFY ONVOLUME statement (available interactively in option **AU.V**) and in the REPORT\_PROFILE NEWLIST (available interactively in option **RA.3.1**).

#### *UNIX=NO*

UNIX data requires substantial amounts of disk space (all directories, owner data, and file permissions are stored). UNIX data is used for TRUSTED reports, SENSITIVE data sets reports (HFS data set sensitivity), and auditing UNIX filesystems. SMF records pertaining to UNIX files often do not contain the path to the file, and the CKFREEZE information can be used to show the appropriate path. Without the UNIX data, most reports still give usable, though incomplete, results. When you are not auditing the HFS or zFS data sets, turn off this feature. Depending on the size of your zFS and HFS data sets, collecting UNIX information can take a long time to process.

This information is used in the following NEWLIST types:

- UNIX (RE.U)
- TRUSTED (AU.S)
- REPORT\_SENSITIVE (AU.S)
- DSN (AU.S)
- SENSDSN (AU.S)
- SMF (EV)

#### *TCP/IP=NO, IMS=NO, CICS=NO, DB2®=NO*

IMS, CICS, and Db2 data usually do not require substantial amounts of disk space. The same is true for information about the TCP/IP stacks on your system. However, if you are not reporting on TCP/IP, IMS,

Db2, or CICS, you also do not need to collect associated information. When you are not auditing the TCP/IP, IMS, Db2, or CICS environment, turn off these features.

This information is mainly used in the following NEWLIST types:

- IP\_\* (RE.I)
- IMS\_\* (RE.M)
- CICS\_\* (RE.C)
- DB2\_\* (RE.D)
- TRUSTED (AU.S/RACF user/TRUSTED and AU.S/RACF resource/Sensitive trust)
- REPORT\_SENSITIVE (AU.S/RACF resource/Sensitive profiles)

### **SMF data**

SMF data is not specific for use by zSecure Audit. Of course, if you want to report about events for a certain period, the SMF data for that period must be available. The required data can be on tape or on DASD. For reporting about SMF events, processing of the SMF data sets is sequential. This means that multiple data sets with SMF records can be on the same tape.

### **Virtual storage**

For SMF event reporting, virtual storage requirements can be significant. For most types of reports, the virtual storage needed is of the same order as the size of the output report. For example, if you are generating a report of four million SMF records with detail information, the program needs sufficient space to retain all unique field values that occur in these records. Typical reports require up to 250 MB of virtual memory. Virtual storage requirements can be significantly reduced by careful selection of the type of events, users, or resources to be included in the report.

### **CPU time**

The CPU time needed for creating interactive or batch reports using zSecure Audit depends on the size of the data to be analyzed and the size of the resulting report. Typical reports take a few seconds to create. Large reports might take several minutes. Due to the amount of data, generating large SMF summary reports might take a significant amount of CPU time.

- zSecure Admin
- zSecure Server
- Access Monitor
- RACF-Offline

## **zSecure Alert**

zSecure Alert uses several resources. It requires at least one CKFREEZE data set, but can also use multiple dedicated CKFREEZE data sets for extended monitoring. The zSecure Alert started task must run on all systems on which alerts are to be generated.

### **DASD storage**

zSecure Alert requires at least one CKFREEZE data set. This is a dedicated CKFREEZE data set comparable in size to a regular Audit CKFREEZE data set. If extended monitoring has been activated, zSecure Alert also creates and deletes multiple temporary CKFREEZE data sets. These extended monitoring CKFREEZE data sets are comparable in size to mini CKFREEZE data sets. The number of these extended monitoring CKFREEZE data sets can be configured. At a minimum, two of these data sets are required.

## Virtual storage

The started task that intercepts Alert events needs sufficient buffer space to retain information about all selected events that occur during a measurement interval. The buffer space is located in the private area of the zSecure Alert started task. The default measurement interval is 1 minute. Normally, SMF events are pre-filtered by record type based on the active alerts. WTO records are similarly pre-filtered by messageid. This pre-filtering is done automatically based on the alert specification in the ISPF user interface. The amount of storage required is dependent on the number of events per interval. For example, if 500 SMF records pass the pre-filtering per second, buffer storage space can be calculated as  $500 \text{ events/second} * 60 \text{ seconds} * 1000 \text{ bytes} = 30 \text{ MB}$ .

The amount of required storage is also influenced by the long-term alerts. These alerts are not based on a single event, but count the number of events over a certain interval (for instance, 20 logons in a 5 minute period). These type of alerts require data for a longer time period to be available, and thus increase the amount of buffer space needed.

Usually, the zSecure Alert address space can run within a region size of 256 MB or less.

## CPU time zSecure Alert data collection

The long running process to collect Alert events also requires CPU time. This CPU time is hard to express in absolute terms, due to the wide range in processor speeds and the number of events. A CPU independent measure of the time needed to collect and record the information is the number of CPU service units (SU). Collecting a single SMF or WTO event requires approximately 1 CPU SU.

You can correlate reported CPU service units to CPU time using the SU/SEC constant as defined for your processor and the CPU service definition coefficient as specified in your IPS or WLM configuration. The number of SUs as reported for your address spaces and system are multiplied by the Service Definition Coefficients (SDC). The default value for the SDC is 10. Review the following example:

- Assume that your installation specified the default value for the CPU SDC and zero for the IO and MSO SDCs.
- Your application runs on an IBM zEnterprise 114 model Z01 (2818-Z01). The CPU model factor for this processor is 40100.2506.
- Your application currently causes 50 SMF events.
- Your application currently uses 2 seconds CPU time.
- The total number of service units reported for your application will be  $10 \text{ (SDC)} * 40100 \text{ (CPU factor)} * 2 \text{ (sec)} = 802000 \text{ service units}$
- After starting zSecure Alert, the amount of CPU time needed to collect and record the information for these events is  $50 \text{ (events)} * 1 \text{ (SU)} / 40100 \text{ (CPU factor)} = 0.002 \text{ seconds}$ .
- The total number of service units reported for your application will be  $802000 \text{ (base)} + 50 \text{ (events)} * 1 \text{ (SU)} * 10 \text{ (SDC)} = 807000 \text{ service units}$ .

## CPU time for zSecure Alert alert generation

The alert issuing phase of zSecure Alert also requires resources like virtual storage and CPU time. The amount of virtual storage is usually negligible, unless a really large number of alerts are issued at the same time. CPU time for the alert issuing phase depends on the number of event records (SMF and WTO) that passed pre-filtering. Usually, processing the collected records and generating alerts requires less than a second for each alert interval (default 1 minute). The CPU time required is only marginally dependent on the alert types that have been selected.

## Daylight saving time considerations

---

zSecure Collect retrieves the time zone information from z/OS, and zSecure Audit uses this information in reports that include time zones. So after a time zone change such as changing to daylight saving time (DST), refresh your CKFREEZE data set. For the zSecure products, there are no further daylight saving time considerations.

## Use of a fresh CKFREEZE and UNLOAD each day

---

For all functions of zSecure Audit, and for many functions of zSecure Admin, a CKFREEZE data set is required. For several functions, an UNLOAD is also a good idea. To make fresh copies available, imbed job C2RJPREP in your production process.

Do not schedule job CKRJPREP to run concurrently with DFSMSHsm (or DFHSM) data set migration. Doing so might result in an incomplete CKFREEZE.

When you have multiple images, you must create a CKFREEZE data set from each system. For a shared security database, create the UNLOAD data set from the system with the highest level of z/OS. You might have to specify NJE routing, system affinity, or both to ensure that each job runs on the intended system.

You might want to use a CKFREEZE data set as secondary input to a process that handles SMF records. An example of such a process is the generation of QRadar LEEF data. If your installation writes Db2 audit records to SMF, the SMF records can be enhanced with information from the CKFREEZE data set. To allow resolution of Db2 Object IDs (OBID) to table and database names, ensure that the CKFCOLL program uses option DB2CAT=YES. This option can be explicitly specified as an input parameter to the CKFCOLL program, or it can be defaulted.

## Requirements and rationale for running the daily CKGRACF job

---

The daily CKGRACF job C2RJXRFR applies only to the zSecure Admin component. It is required when you use the Queued command or Multiple authority functions of zSecure Admin or when you use zSecure Visual. When multiple images share a RACF database, run the daily CKGRACF job on the system with the highest level of z/OS. You might have to specify NJE routing, system affinity, or both to ensure that the job runs on the intended system.

Run the C2RJXRFR job to perform the following actions:

- Revoke or resume users according to their revoke and resume schedules.
- Execute queued commands that are due.
- Expire or delete queued commands when that is due.

If the job does not run every day, there are two potential risks:

- Users might not be revoked or resumed in time.
- Queued commands might not get executed in time.

The job can be scheduled to run only once every day because the granularity of revoke and resume schedules, and of queued command times, is one day.

The C2RJXRFR job relates to the zSecure Visual product because part of its functionality is to maintain revoke and resume schedules. The job effectuates revoke and resume settings that Visual users have scheduled.

## Setup of zSecure Admin Command Logger

---

zSecure Admin provides an option to log all RACF commands that are issued through the zSecure interface in a central repository. This section provides an overview of the functions and describes the required installation and activation steps.

### About this task

The command log records are kept in a z/OS log stream. Using zSecure functions, it is possible to select and report about the recorded RACF commands. The z/OS log stream must be defined in the LOGR policy data set. The actual recording of the commands in the log stream is done through a started task (CKXLOG) that runs on each system (LPAR) where RACF commands are run. The record of the command can be annotated with a ticket identification and a ticket description. The zSecure Admin ISPF interface and zSecure Command Verifier send the commands to the zSecure Admin Command Logger.



When zSecure Admin is installed and it is verified that it is working properly, the following steps are required for the setup of the zSecure Admin Command Logger function:

- Step “1” on page 44: Define a log stream (or log stream model) for use by the Command Logger
- Step “2” on page 44: Update C2R\$PARM to specify a value for CKXCUST.
- Step “3” on page 44: Update the CKXCUST member CKXPARM for installation parameters.
- Step “4” on page 45: Define the STC user ID and STARTED profile for the CKXLOG started task.
- Step “5” on page 45: Copy the STC procedure to your STC proclib.
- Step “6” on page 45: Authorize the STC user ID to allocate and write the log stream.
- Step “7” on page 45: Start the CKXLOG started task.
- Step “8” on page 46: Define the Command Log policies for components and users.
- Step “9” on page 47: Configure the zSecure Admin ISPF user interface for command logging.
- Step “10” on page 47: Configure zSecure Command Verifier to ensure logging of commands that are issued in a batch job or from outside the zSecure Admin ISPF interface.
- Step “11” on page 47: Optionally, enable calling CKXLOGID from APF environment.

In the above steps, you define new profiles to activate and control various parts of the zSecure Admin Command Logger. The following picture shows a global overview of the flow of a log record, indicating where these different profiles are used:

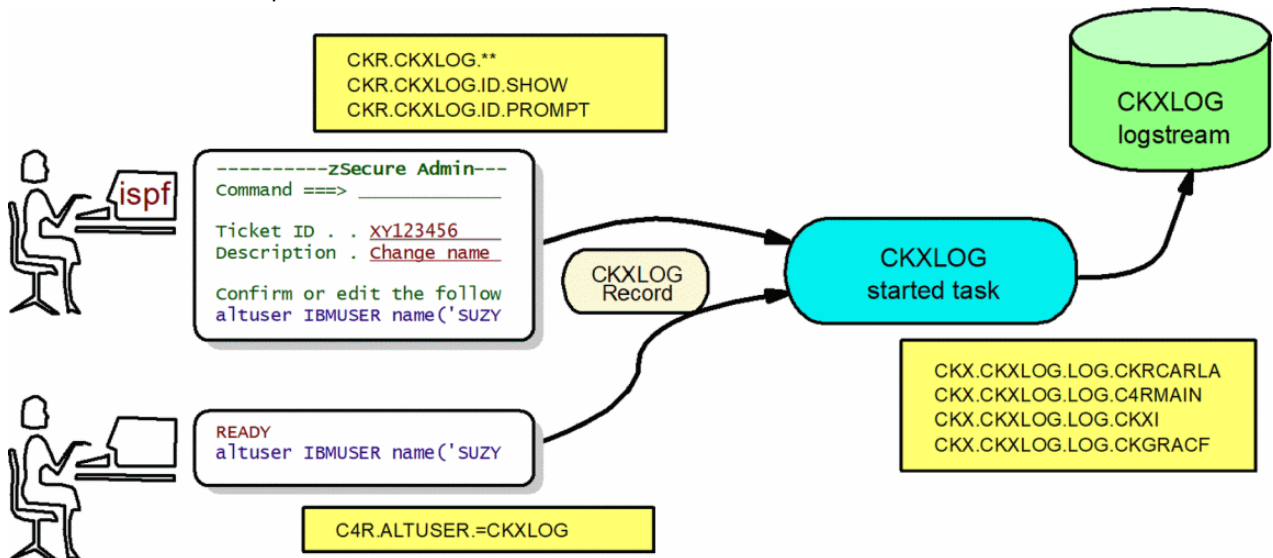


Figure 5. Global overview of the flow of a log record

The CKXLOG started task uses resources that start with CKX, as shown in the rectangular box. Access to these resources is used to determine if a log record is written to the logstream or not. Step “8” on page 46 describes these resources.

The first administrator uses the zSecure Admin ISPF interface to enter changes. The ISPF interface uses the CKR.CKXLOG.\*\* profile and resources that start with CKR as shown in the associated rectangular box. The presence and access of these profiles and resources determine if the zSecure Admin ISPF interface passes command records to the CKXLOG started task. Step “9” on page 47 describes these resources. The second administrator enters a command directly without using the zSecure Admin ISPF interface. These commands might be intercepted by zSecure Command Verifier. In that case, logging is controlled through resources that start with C4R, of which a sample is shown in the associated rectangular box. Step “10” on page 47 describes these resources.

Note that commands that are executed through the zSecure Admin ISPF interface might also pass through zSecure Command Verifier, and might thus be passed to the CKXLOG started task twice: once by zSecure Admin, and once by zSecure Command Verifier. It is the CKXLOG started task that determines whether or not the passed CKXLOG records are actually recorded in the CKXLOG log stream.

Depending on your environment, you might need profiles for one or more of these types of resources, using generic profiles where possible.

## Procedure

1. Define a log stream (or log stream model) for use by the Command Logger.

The zSecure Admin Command Logger writes the command records to a system-defined log stream. The CKXPARM member defines the name of the log stream. Before a log stream can be used, it must be defined in the system LOGR policy data set using the IXCMIAPU utility. The zSecure Command Logger supports two methods for defining the log stream:

- The log stream is defined by the systems programmer using the exact name that is specified in CKXPARM.
- The CKXLOG program itself defines the log stream, based on a predefined log stream model.

The second method is preferred. The log stream must be defined once and can then be used on all systems that share that LOGR policy data set. The CKXLOG started task tries to open the log stream as it is defined in the CKXPARM. If the log stream does not exist, CKXLOG defines the log stream using the specified (or default) log stream model.

The log stream can be defined as a DASDONLY log stream; the log stream is dedicated to a single system. It is recommended to use &SYSNAME as its first qualifier. The log stream name is specified in the CKXPARM that you create or update in step 3. Member CKXZLGST in SCKRSAMP is provided as an example for defining a DASDONLY model log stream.

It is also possible to define a COUPLING FACILITY log stream; the log stream is shared between multiple systems. Member CKXZLGCF in SCKRSAMP is provided as an example for defining a shared model log stream.

When defining the log stream model, you must consider the retention period and the size of the stream. These are largely dependent on your installation policy and the number of commands issued.

2. Update C2R\$PARAM to specify the value for CKXCUST.

The CKXLOG procedure refers to include member C2R\$PARAM. C2R\$PARAM is a sample member that contains JCL SET statements and it is usually referred to as the zSecure configuration. In the zSecure configuration file, the variable CKXCUST is used to name the configuration file for the CKXLOG procedure

```
// SET CKXCUST='your.prefix.CKRPARM'
```

You normally substitute *your.prefix.CKRPARM* with the installation CKRPARM data set, but you can specify any partitioned data set with record format FB, and a logical line length of 80. Position 73 to 80 in the records are ignored. This data set must contain the member indicated by the XPARAM parameter in the JCL procedure.

3. Update the CKXCUST member CKXPARM for installation parameters.

The CKXPARM member in the CKXCUST data set can be used to override the default values for certain parameters. The parameters are set using the SETUP statement. The following SETUP statements are supported:

### **SETUP LSNAME(log-stream-name)**

The log-stream-name specifies the name of the log stream that the CKXLOG started task is to use. The qualifier CKXLOG is used to identify the log stream as related to the Command Logger application. It is suggested that the first qualifier indicates the system or sysplex. The log-stream-name can contain system symbols, like &SYSNAME or &SYSPLEX. The default value is &SYSPLEX..CKXLOG

### **SETUP LSMODEL(model-log-stream-name)**

The model-log-stream-name specifies the name of the model that the CKXLOG started task is to use if the log stream that the LSNAME specifies does not exist. The qualifier CKXLOG is used to identify the model log stream as related to the Command Logger application. The suggested

value for the first qualifier is MODEL. The log-stream-name can contain system symbols, like &SYSNAME or &SYSPLEX. The default value is MODEL.CKXLOG.

**SETUP TICKETEXPIRE(unused-time)**

The CKXLOGID command can be used to set a ticket ID and description. The CKXLOGID command can also be used to clear such a setting. If a ticket ID has not been referenced, it can be automatically removed to free up associated resources. The unused-time specifies the period (HHMM) after which an unreferenced ticket ID is removed from the CKXLOG started task. The default value is 0100 (1 hour).

Update the CKXPARM member to specify values that are appropriate for your environment.

4. Define the STC user ID and STARTED profile for the CKXLOG started task.

If you want to use a dedicated user ID for the CKXLOG started task, you must define the user ID. The user ID and group ID can be assigned to the started task through the STDATA segment of the applicable STARTED profile. The user ID does not need to have any non-standard attributes.

5. Copy the STC procedure to your STC proclib.

The example CKXLOG procedure must be available in one of the system STC proclibs. Copy and adapt the CKXLOG procedure from the SCKRPROC library.

6. Authorize the STC user ID to allocate and write the log stream.

The CKXLOG started task writes record to the specified log stream. If the log stream does not exist, it defines the log stream. For these actions, the STC user ID needs sufficient authorization. Use the following table for the required authorization:

<i>Table 4. Authorization of the STC user ID</i>				
<b>Action</b>	<b>Type of log stream</b>	<b>Required authorization</b>	<b>Resource class</b>	<b>Resource name</b>
DEFINE	DASDONLY	ALTER	LOGSTRM	<i>log_stream_name</i>
	COUPLING FACILITY	ALTER	LOGSTRM	<i>log_stream_name</i>
		UPDATE	FACILITY	<i>IXLSTR.model_structure_name</i>
CONNECT		UPDATE	LOGSTRM	<i>log_stream_name</i>

7. Start the CKXLOG started task.

The zSecure Admin Command Logger task is started from the operator console by a START command. The command executes the procedure from the applicable system proclib. It is possible to specify startup parameters. These parameters can be given at the START command itself. An example of such a START command is:

```
S CKXLOG,, ,DEBUG
```

The following START parameters are supported:

**DEBUG**

Specifies that diagnostic messages must be issued during startup and running of the CKXLOG task. This setting is in effect until a subsequent NODEBUG command is issued from the operator console.

**FORCE**

Specifies that, regardless of a previous execution, initialization must continue. Only use the FORCE option if you cannot start the zSecure Command Logger normally. The FORCE option might enable starting the zSecure Command Logger without the need to IPL the system. During normal operation, the FORCE option is never required.

**DEBUG-FORCE**

Specifies that both the DEBUG and FORCE options must be active at startup.

To terminate the started task, the console operator can issue the STOP command. z/OS supports the use of the P command as an alias of the STOP command; for example:

```
P CKXLOG
```

The STOP command can also be issued as the parameter on the MODIFY command. z/OS supports use of the F command as an alias of the MODIFY command. An example of such a command is:

```
F CKXLOG,STOP
```

While the CKXLOG started task is active, the console operator can also issue commands to modify the zSecure Command Logger. These commands can be issued through the MODIFY console command. For example:

```
MODIFY CKXLOG,DEBUG
```

The following list describes the zSecure Command Logger operator commands that can be used in the MODIFY console command:

#### **ABND**

This command causes an immediate abend of the CKXLOG started task, with recovery. It is intended for IBM internal testing only.

#### **CRSH**

This command causes an immediate abend of the CKXLOG started task, without any recovery. It is intended for IBM internal testing only.

#### **DEBUG**

This command controls the diagnostic and monitoring messages that the program can generate; the zSecure Messages Guide describes all messages. The command is effective immediately. The DEBUG command does not support any additional keywords.

#### **NODEBUG**

Deactivates issuing of diagnostic and monitoring messages. The NODEBUG command does not support any additional keywords.

#### **SIPL**

This command is used during the migration to a new release. It results in freeing all in-memory data structures, loss of a system level linkage index (LX), and marking the address space as non-reusable. System level LX's are a limited resource and cannot be recovered without an IPL of the system. When upgrading from one release of the zSecure Command Logger to another, the installation instructions require that you shut down the previous version using this SIPL command. The SIPL command does not support any additional keywords.

#### **STOP**

Stop execution of the zSecure Command Logger started task. This results in an orderly shutdown of the task. Some memory remains reserved after termination of the task, to enable reuse of some critical system resources during a subsequent restart of the started task. The effect of the STOP MODIFY command is identical to that of the z/OS STOP command. The STOP command does not support any additional keywords.

### 8. Define the Command Log policies for zSecure components.

The zSecure Admin Command Logger checks if commands that are issued through supported zSecure components must be logged. In the absence of any profile, commands are not logged. To activate command logging, you must define one or more profiles. These switch profiles have the following format: CKX . CKXLOG . LOG . *component-name*, where *component-name* can be one of the following options:

#### **CKRCARLA**

The zSecure common program.

#### **CKXI**

zSecure Admin Command Execution module CKX. This is used for queued commands and for commands that are issued through the zSecure Server.

## CKGRACF

The zSecure Admin RACF special functions program.

## C4RMAIN

zSecure Command Verifier.

## CKXLOGID

Special logging program when testing the LOG function of the Command Logger.

You can use a generic profile to match these various component names.

When commands are issued using the zSecure Admin interface, the same command can be logged multiple times, because it passes multiple components. By defining these profiles, you can control for which components the commands must be logged and for which components logging the commands must be suppressed. In the absence of any profile, commands are not logged. To activate logging, the user that issues the command must have at least READ access to the resource, for example:

```
CKX.CKXLOG.LOG.* ID(*) ACCESS(READ)
```

### 9. Configure the zSecure Admin ISPF user interface for command logging.

Command logging from the zSecure Admin ISPF panels is enabled by defining a profile `CKR.CKXLOG.**`. Individual functions related to command logging can be controlled by also creating more specific profiles as described in subsequent paragraphs. Creating this `CKR.CKXLOG.**` profile with `UACC(READ)`, and not defining more specific profiles, changes the command confirmation level (as specified in SE.4) for all zSecure Admin users, and enforces that a ticket ID and description are entered for every generated command.

To allow users to specify ticket information, the following resource is used: `CKR.CKXLOG.ID.SHOW`. Users with at least READ access to the matching XFACILIT profile can use the CKXLOGID primary ISPF command and can specify or confirm ticket information during command confirmation. Command confirmation is active according to the user specified setting in `SETUP CONFIRM` (SE.4). The matching profile for this resource can be generic, for example the `CKR.CKXLOG.**` profile.

To require that a user fills in the ticket ID and description, the following resource is used: `CKR.CKXLOG.ID.PROMPT`. Users with at least READ access to the matching XFACILIT profile always see a command confirmation panel when commands are generated, irrespective of the value that is specified in `SETUP CONFIRM` (SE.4 or SE.D.4). They must supply ticket information for the generated commands, optionally through the CKXLOGID primary ISPF command. The matching profile for this resource can also be generic, for example the `CKR.CKXLOG.**` profile. It is expected that most organizations will define a discrete profile with `UACC(NONE)` and will not require all users to specify or confirm the ticket ID and description for every generated command.

**Note:** Additional logging of RACF commands can be provided by zSecure Command Verifier, independent of the command logging functions of zSecure Admin. See the step “10” on page 47 for setup instructions. Installations can decide to log through the zSecure Admin interface, through zSecure Command Verifier, or through both logging mechanisms. Logging is finally controlled by the user's access to `CKX.CKXLOG.component` profiles, as described in step “8” on page 46.

### 10. Configure zSecure Command Verifier to ensure logging of commands that are issued in a batch job or from outside the zSecure Admin ISPF interface.

You can use zSecure Command Verifier to record most RACF commands. To activate this function, you must define one or more profiles in the XFACILITY class. The profiles have the following format: `C4R.command.=CKXLOG`, where *command* is the unabbreviated RACF command. Generics can be used, except for the `=CKXLOG` qualifier, which must be present exactly as shown. zSecure Command Verifier logs all RACF commands through the zSecure Admin Command Logger for users that have at least READ access to the applicable profile. To suppress warning messages due to missing ticket information, or an inactive CKXLOG server, users must have at least UPDATE access to the `C4R.command.=CKXLOG` resource.

For more information about zSecure Command Verifier, see *zSecure Command Verifier User Guide*.

### 11. Optional: Enable calling CKXLOGID from APF environment.

The CKXLOGID command is a regular TSO command that does not need APF authorization. However, if it is invoked from inside an APF-authorized program, additional definitions are required. If you plan to create your own APF-authorized program that calls CKXLOGID through the TSO service route IKJEFTSR, then you must add CKXLOGID to the list of APF-authorized commands (AUTHCMD) in the active TSO parmlib member (IKJTSONn). If you do not plan to write such a program, no further action is required, and you can skip this step.

## TCP/IP domain name resolution

---

zSecure can report in various formats, including Simple Network Management Protocol (SNMP) and Simple Mail Transport Protocol (SMTP); that is, e-mail. In this respect, zSecure acts as a user of TCP/IP services. As a result, the environment where zSecure runs might need domain name resolution. The environment can be a TSO or CMS user, a batch job, or the zSecure Alert or zSecure Visual started tasks. Depending on the level of your IP stack, you might need to set up a *userid*. TCPIP . DATA, or a SYSTCPD DD-statement, or some other method that points to the TCP stack that provides the DNS function. For information, go to [IBM Documentation for z/OS](#), select the release that you are using and find [z/OS Communications Server: IP Configuration Reference](#). Also, make sure that the processes that need domain name resolution have READ access to all relevant files, such as TCPIP . DATA, /etc/resolv.conf, and /etc/hosts.

## SMTP server considerations

---

Reports, particularly in XML format, can become large. The size can be a concern when transmitting reports by e-mail. Reports that are too large can be rejected or truncated by the SMTP server. To prevent problems with transmitting files that are too large, verify, and perhaps change, the MAXMAILBYTES and CHECKSPOOLSIZE settings of your SMTP server.

---

## Chapter 9. Setup for remote data access and command routing

You can use zSecure for the administration and auditing of profiles, resources, and settings from multiple systems. You can configure the input data sources for systems of interest so that zSecure can collect the information directly from each system. The data sets can then be used through the ISPF interface or in a CARLa program. This functionality is called multi-system support because it enables reporting and managing multiple systems from a single session.

In addition to multi-system reporting, the product also supports routing commands to be run on a remote system using zSecure services or using existing RACF RRSF services. Including the support for remote system routing, zSecure provides the following command routing options: route to the local system or route to a remote system using NJE batch jobs, RACF RSSF services, or zSecure services.

To provide these functions, zSecure Admin and Audit use TCP/IP services. The zSecure Server manages the connections and handles data transport. You must install and activate the zSecure Server for access to remote data and command routing using zSecure services. The following setup tasks provide support for remote systems:

- Installation, configuration, and activation of the zSecure Server. See [“Installation and configuration of the zSecure Server”](#) on page 49.
- Specification of the remote data sets for use in CKRCARLA or the ISPF User Interface. See [“Operator commands for the zSecure Server”](#) on page 54.
- Setup for routing RACF and selected non-RACF commands to other systems. See [“Setup for secure communication using AT-TLS”](#) on page 56.

---

### Installation and configuration of the zSecure Server

Use the following sections to install and configure the zSecure Server.

#### Installed software and multi-system support

The zSecure CARLa-based installation programs install the code and panels required for the zSecure multi-system support. After the SMP/E apply, all required software is available in the standard libraries. No new libraries are added for the zSecure multi-system support. The SCKRLOAD library must be APF-authorized.

#### JCL procedures and parameters

The code for the zSecure Server is provided in the CKNSERVE load module. This program runs as a started task, but it can also run as a batch job. Running as a started task is preferred because it allows running the program in a reusable address space. A batch job cannot be run in a reusable address space. The procedure for running the program as a started task is provided in sample CKNSERVE.

The CKNSERVE procedure refers to include member C2R\$PARM. C2R\$PARM is the sample member that contains JCL SET statements and it is usually referred to as the *zSecure configuration*. Member C2R\$PARM (or any member you choose to substitute) is located in the CKRPARM data set. See [“About zSecure configuration data sets”](#) on page 23.

The zSecure configuration is the main configuration file used by the organization to specify data set names, members, and other options. The zSecure configuration file must be available for INCLUDE in the started task JCL. Usually, the zSecure configuration file must be a member in the started task procedure



library. Besides the symbols used for all zSecure programs, the zSecure configuration file must set the CKNSVPRM symbol, which is specific for CKNSERVE:

```
// SET CKNSVPRM=<installation-specified-parmlib>
```

For <installation-specified-parmlib> you normally substitute the CKRPARM data set you prepared for the zSecure Server, but you can specify any partitioned data set with record format FB, and a logical line length of 80. Position 73 to 80 in the records are ignored. This data set must contain the two members indicated by the PPARM and PCOMMON parameters in the JCL procedure. These two members together constitute the *zSecure-Server configuration* file.

**Note:** Do not confuse the zSecure configuration (commonly referred to as C2R\$PARM) with the zSecure Server configuration (consisting of the two members identified by PPARM and PCOMMON).

The CKNSERVE procedure also refers to member CKNCKFAI in the CKNSVPRM data set. The CKNCKFAI member contains the zSecure Collect parameters controlling the creation of a mini-CKFREEZE. You normally do not need to change any of the keywords and parameters contained in this member. You can copy it from the SCKRCARL library.

The member specified by PPARM is intended to contain the configuration parameters that are specific to a particular instance of the zSecure Server. In contrast, the member specified by PCOMMON is intended to contain those parameters that are common between all zSecure Servers. An example of the server-specific PPARM member is:

```
OPTION  Ownsys(PRODSYS2) servertoken(MyToken)
```

Usually, the only statement present in the PPARM member is an OPTION statement to identify the SERVERTOKEN. The OPTION statement has multiple additional keywords that are normally not required. For example, the OWNSYS keyword is needed only if there is more than one ZSECSYS entry that matches the current TCPIP domain name or the TCPIP hostname for the local system. For more information about the OPTION statement see [“Configuration file OPTION statement” on page 51](#).

The following example shows the shared PCOMMON member.

```
ZSECNODE NAME(ZSNODE1)
ZSECSYS  NAME(ZSSYST1) ZSECNODE(ZSNODE1) IPADDR(MyNode) IPPORT(7173)
ZSECNODE NAME(TSTNODE1)
ZSECSYS  NAME(TSTSYS1) ZSECNODE(TSTNODE1) IPADDR(MyTest) IPPORT(7173)
ZSECNODE NAME(PRODNODE)
ZSECSYS  NAME(PRODSYS1) ZSECNODE(PRODNODE),
         ipaddr(prodsys1.mydomain.com),
         IPPORT(7174)
ZSECSYS  NAME(PRODSYS2) ZSECNODE(PRODNODE),
         ipaddr(prodsys2.mydomain.com),
         IPPORT(7173)
```

As illustrated in the example, statements in both files can be split over multiple lines by using the comma as a line continuation character.

**Note:** Lines can be split only between keywords and not inside a keyword or parameter.

For more information about the configuration statements, see [“Configuration statements” on page 51](#).

## Security definitions for the started task

Be sure that the `userid` assigned to the task has sufficient authorizations. These authorizations are:

- Authorization to read all data sets referenced in the JCL procedure
- Authorization to read the TCPDATA data set and other TCP/IP control data sets (like TCPXLBIN)
- Access to UNIX functions, using either an OMVS segment or a default OMVS UID. The `userid` can have any UID. It does not require any specific UNIX authorization, file access, or even a home directory.
- READ access to the SERVAUTH resource describing the current TCP/IP stack. These resources have the format



```
EZB.STACKACCESS.<sysname>.<stackname>
```

- READ access to the IRR.DIGTCERT.LISTRING resource in the FACILITY class.

On all systems where you intend to deploy the zSecure Server, use job CKNRACF1 to set up these regular authorizations.

In addition, the `userid` must also be assigned a certificate to validate and encrypt communication with other zSecure Servers. See [“Setup for secure communication using AT-TLS” on page 56](#) for the requirements for securing the network connection.

## Configuration statements

The configuration statements for the zSecure Server are provided in the zSecure-Server configuration file. This file is a logical file that can be split over multiple concatenated members or data sets, as shown in the sample STC procedure. The configuration file uses two mandatory statements and one optional statement.

- The mandatory statements are ZSECNODE and ZSECSYS.
- The optional statement is OPTION.

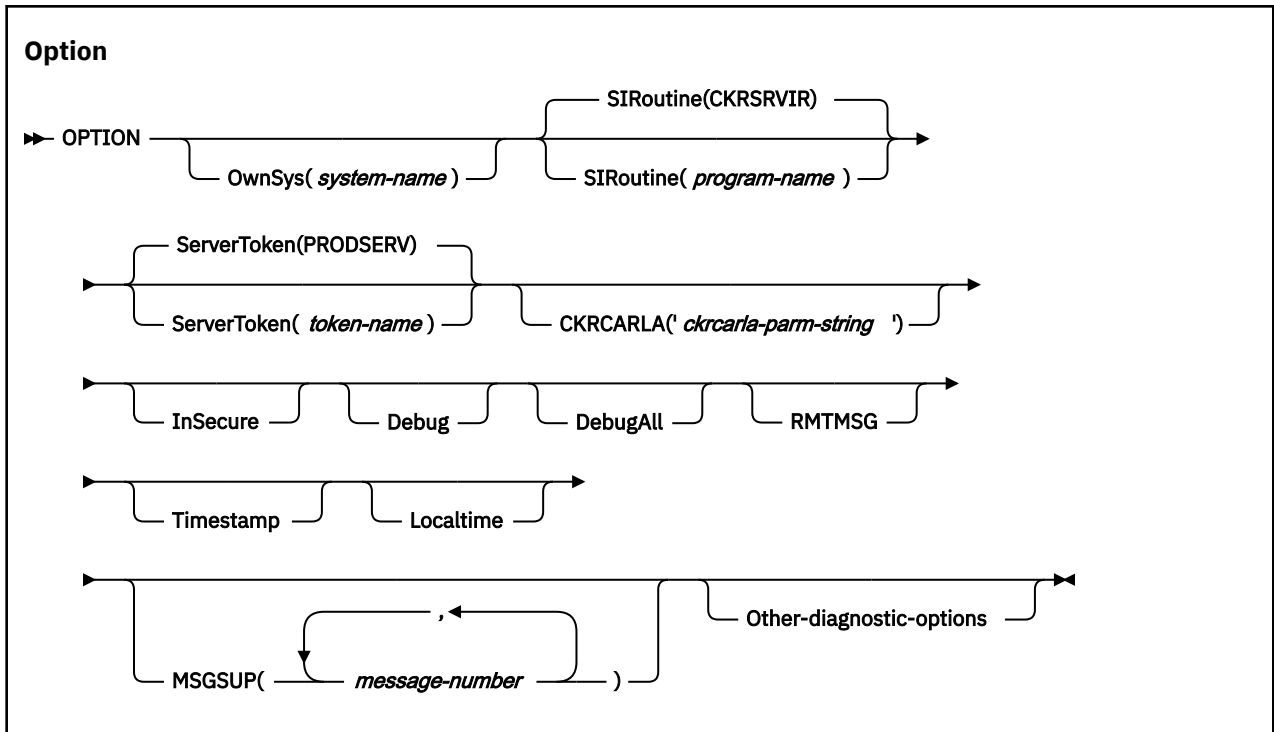
The ZSECNODE statement defines the set of systems that share a RACF database. The ZSECSYS statement defines the individual systems where a zSecure Server address space can be running. There can be as many ZSECNODE and ZSECSYS statements as needed to describe your environment. In most cases, users specify (or default) the ZSECNODE in their zSecure UI setup, while some users might use the ZSECSYS to reference specific data sets that are present only on a specific system.

The sequence of statements in the zSecure-Server configuration file is important: The ZSECNODE statement must precede the ZSECSYS statements that refer to that ZSECNODE.

The statements are organized such that in most installations, the ZSECNODE and ZSECSYS statements can be shared between all zSecure Servers. Only the OPTION statement is specific for a particular zSecure Server. If only a single server is active on the system, the OPTION statement can also be shared between zSecure Servers. The zSecure Server compares the IP-address and the name on the ZSECSYS definitions against the TCPIP domain name and IP host name of the system. If a match is found, the ZSECSYS definition is used for the local system. If multiple zSecure Servers are defined that use the same IP address, one of these systems is chosen. Which system is used is unpredictable. In this situation, use of the OPTION statement with the OWNSYS keyword is preferred.

### Configuration file OPTION statement

In most situations, the OPTION statement is not required. However, if you are running multiple servers on the same systems, the OPTION statement is required to specify unique parameters for each server. The OPTION statement can also be used to specify diagnostic settings.



The keywords and parameters have the following meanings:

### OwnSys

This keyword specifies the system name for the current server. This value is needed only if there is more than one ZSECSYS entry that matches the host name of the current TCP/IP stack. Normally, the local system name is determined based on the IPADDRESS of the ZSECSYS definitions. If there are multiple ZSECSYS statements with the same IPADDRESS specification, and OWNSYS is not specified, one of the possible ZSECSYS entries is used for the current server. Which name is used in that case is unpredictable.

### SIRoutine

This keyword specifies the name of the Server Interface Routine. Currently this keyword and parameter are ignored.

### ServerToken

This keyword specifies the eight-character suffix for the name of the Named-Token to be used to anchor the global data area used for this server. The value specified is prefixed by the value CKNSERVE. If the same value is specified for two servers, the second started instance will fail. The default value for the token is PRODSERV. If this keyword is not specified, the default value is used. You need to specify a value for the **ServerToken** only if you are running multiple zSecure Servers on the same system.

### CKRCARLA

The CKRCARLA keyword can be used to add a CARLA statement to the CKRCARLA call parameters. This keyword is intended to be used for debug parameters that are only effective on the program call parameter. The *ckrcarla-parm-string* has a maximum length of 80 characters and must be within quotes.

### InSecure

This keyword specifies that insecure communication to other zSecure Servers is acceptable. To enable communication that is not secure between two zSecure Servers, both servers must specify the INSECURE option, and the userid of the server task must have READ access to applicable CKNADMIN profiles in the XFACILITY resource class. This option is for use only during initial setup, and is not for production usage.

### Debug

This keyword specifies that additional diagnostic messages are to be issued to the server CKNPRINT output file. Use this keyword only at the request of IBM Software Support personnel.

## DebugAll

This keyword specifies that the maximum amount of diagnostic messages must be issued to the server CKNPRINT output file. Use this keyword only at the request of IBM Software Support personnel. Specific messages can be suppressed with OPTION MSGSUP.

## RMTMSG

This keyword can be used to signal the zSecure Server to include the SYSPRINT and SYSTEM output from remote applications in the local CKNPRINT output file.

For example, when a client accesses data from a remote RACF database, the remote server uses CKRCARLA to read the RACF database. The output of the remote CKRCARLA is always available in the SYSPRINT file of the client application. Including this same output also in the CKNPRINT of the local server is optional. If you specify the DEBUG option, RMTMSG is selected as well.

## Timestamp

This keyword specifies that messages issued in the server CKNPRINT output file are prefixed with a timestamp. The timestamp information is shown in UTC and uses a fixed format. If you want the local time of the system that you run on, use keyword Localtime instead. If both are present, local time is the net effect.

## Localtime

This keyword specifies that messages issued in the server CKNPRINT output file are prefixed with a timestamp. The timestamp information is in local time of the system that the server runs on and uses a fixed format. If you want UTC, use keyword Timestamp without Localtime instead.

## MSGSUP

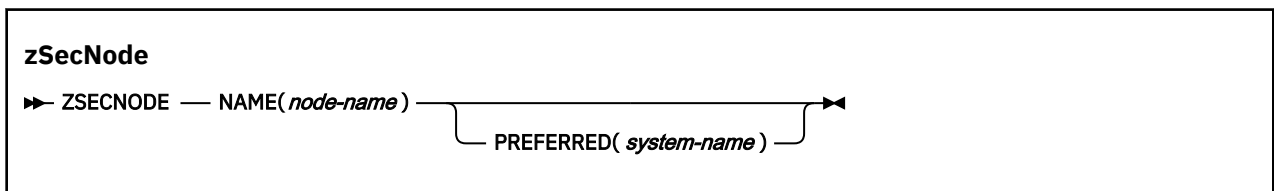
This keyword specifies a list of message numbers that are suppressed in the server CKNPRINT output file. It can be used in combination with the DEBUG command. Use this keyword only at the request of IBM Software Support personnel.

## Other-diagnostic-options

Several additional diagnostic options are available when specified in the parameter string to the CKNSERVE program. Use these options only at the request of IBM Software Support personnel; they are not intended for customer use. The currently implemented options are NOESTAE, NOCLOSE, NODUMP, NOCLEANUP, NODUMPEXIT, NOGUARD, and STORAGEEC.

## Configuration file ZSECNODE statement

A single ZSECNODE describes all the systems that share a common RACF database. A request to access data or update a profile could effectively be directed to any system belonging to the same ZSECNODE. In normal situations, the zSecure Server uses the designated preferred server. If that server is unavailable, however, the zSecure Server uses another server that is part of the same node. In that situation, the first ZSECNODE that is available is used for all node communications.



The keywords and parameters have the following meanings:

### Name

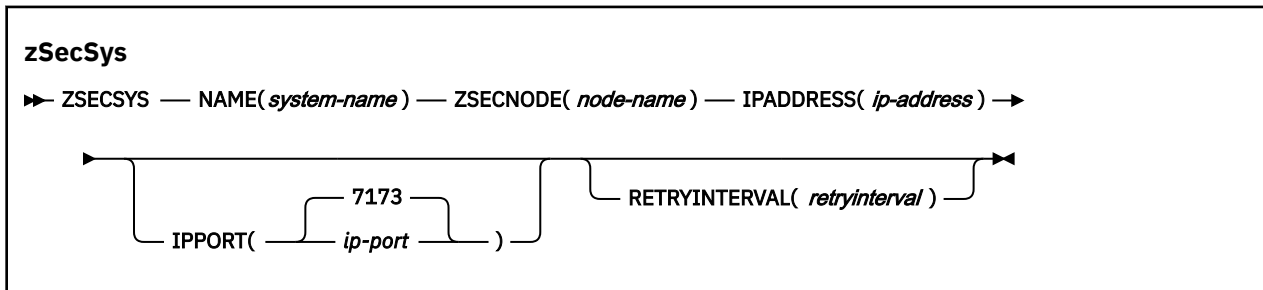
This keyword provides the name for this zSecure Node. If you have RRSF, make this name equal to the RRSF node name. Use an alternate name only in the rare case that you have multiple RRSF plexes that use the same name.

### Preferred

This keyword specifies the name of the system where the preferred zSecure Server for this node is running. This system is normally used to access the RACF database for this zSecure Node. It is an error to specify a server name that is not defined in the zSecure-Server configuration file, or to specify a server that does not refer to the current zSecure node.

## Configuration file ZSECSYS statement

The ZSECSYS statement defines the systems where zSecure Servers are running. If a preferred system for the ZSECSYS statement is not specified, or if the preferred system is unavailable, the order of the ZSECSYS statements for a specific node defines the connection preference sequence. The ZSECSYS statement can be repeated for as many system names as required. If multiple ZSECSYS statements refer to the same system name, an error message is issued and execution stops.



The keywords and parameters have the following meanings:

### Name

This keyword provides the name assigned to the zSecure System. Make this name equal to the MVS SYSNAME or the SMF-id, if they are unique. You only need to specify another name when you have more than one system with the same MVS SYSNAME.

### zSecNode

This keyword provides the nodename to which this system belongs. If you have RRSF, make this name equal to the RRSF node name. Use an alternate name only in the rare case that you have multiple RRSF plexes that use the same name.

### IPAddress

This keyword specifies the IP address that can be used to contact the zSecure Server. The IP address can be a host name or an address in either IPV4 format or IPV6 format. The preferred value is the value from the TCP/IP stack's TCPIP.DATA HOSTNAME and DOMAINORIGIN statements. This value is case-sensitive. An example of the specified value is `ourhost . company . com`.

### IPPort

This keyword specifies the portname to be used.

- For a local system, it is the portname on which a local zSecure Server is listening for incoming connections.
- For remote systems, it specifies the portname used to connect to the remote system.

The specified `IPPort` value can be the same (but need not be the same) for different zSecure Servers. The `IPPort` value specified for a particular ZSECSYS must be the same across the entire zSecure network. The Internet Assigned Numbers Authority (IANA)-assigned port number for `IPPort` is 7173.

### RETRYINTERVAL

This keyword specifies that the connection to this ZSECSYS is restarted if the connection is not active for any reason. The value for the `RETRYINTERVAL` parameter specifies the number of minutes before a connection is restarted. The value can be:

#### 0

Signals that no automatic restarts are done. This value is the default.

#### Between 1 and 1440

A value between 1 and 1440 specifies the number of minutes for which a connection can be inactive before an automatic restart of the connection is attempted. The value for this parameter is usually between 5 and 60 minutes.

## Operator commands for the zSecure Server

The following operator commands are currently supported:

- START
- MODIFY <taskname>,<action>
- STOP

## START

Because the zSecure Server CKNSERVE uses cross memory services to enable users to access the server functions, the CKNSERVE address space is associated with a system level linkage index. This system level linkage index is a resource that is retained after each server stop and reused during each subsequent start of the server. This system level linkage index is based on the `ServerToken` specified in the server control file. If you want to preserve system resources, specify the same `ServerToken` for each subsequent start of the same zSecure Server. You can use the zSecure-Server configuration file `OPTION` statement to specify the `ServerToken`.

The use of cross memory services also results in marking the address space used for the Server as `UNAVAILABLE` after use. This can be observed through the message

```
IEF352I ADDRESS SPACE UNAVAILABLE
```

in the system log after termination of the server. To avoid losing address space by repeatedly starting and stopping the zSecure Server, it is important to start the zSecure Server using the `REUSASID` keyword. An example start command of the zSecure Server then becomes:

```
START CKNSERVE,PPARM=CKNSRV00,REUSASID=YES
```

In this example:

- The zSecure Server procedure is called CKNSERVE.
- The private parameter member for this instance of the server is CKNSRV00.
- The address space to be used is to be obtained from the pool of reusable address spaces.

For more information about reusable address spaces, see the following publications:

- *z/OS MVS System Commands*
- *z/OS MVS Initialization and Tuning Reference*

## MODIFY

You can use the following actions with the `MODIFY` command:

### DEBUG

Diagnostic messages are printed in the `CKNPRINT` output file.

### DebugAll

The maximum amount of diagnostic messages are printed to the server `CKNPRINT` output file.

### NODEBUG

Diagnostic messages are no longer printed in the `CKNPRINT` output file.

### RMTMSG

Include the output from remote applications (for example, `CKRCARLA`) in the local server `CKNPRINT` output file.

The output of remote applications is always available in the `SYSPRINT` file of the client. If you specify `RMTMSG`, the same output is also available in the `CKNPRINT` of the server.

### NORMTMSG

Reverses the effect of a previous `RMTMSG` operator action, or that of an `OPTION RMTMSG` configuration statement.

### STOP

This action is equivalent to the `STOP` command.

## STOP

You can also request such a stop by using a MODIFY `<taskname>`, STOP request.

## Setup for secure communication using AT-TLS

---

The data exchanged between the zSecure Servers is often confidential in nature. For example, it can contain RACF passwords and passphrases. It is therefore important to ensure that the data communication is secure and that data cannot become exposed. Within the zSecure network, data is secured using the following methods:

- Partner verification using certificates
- Additional verification that the certificate is intended for zSecure use
- Data encryption

These methods are implemented using AT-TLS and additional verifications in the zSecure Server. The session between each zSecure Server pair must be defined using a TCP/IP Tunneled Transport Layer Security (TTLS) Policy. Within z/OS, TTLS policies are managed using the Policy Agent. You can use the *IBM Configuration Assistant for z/OS Communications Server* to create the configuration file for the Policy Agent, or use member CKNTTLS in SCKRSAMP as a starting point.

The TTLSRule statements identify the sessions you want to protect and they specify, through keyrings, where AT-TLS can find the required certificates.

You can identify the session by any of the following items:

- (RACF) userid
- jobname
- Local IP address and port
- Remote IP address and port
- A combination of the previous items

The sample CKNTTLS filters by userid, and it assumes that the userid is the same on both sides of the connection.

The certificates can be specified directly, in a TTLSConnectionAdvancedParms statement, or indirectly through a keyring. CKNTTLS uses the keyring method.

For certificates to be verified, they must be signed by a trusted root certificate, and that root certificate must be accessible on the receiving side of the connection. You can use a commercial root certificate, or you can use job CKNRACF2 to create a root certificate. If you use job CKNRACF2:

1. Run job CKNRACF2 on only one of the system images where you intend to deploy the zSecure Server.
2. Copy the export data set to the other images and run job CKNRACF3 to import it into all other RACF databases.
3. After importing, delete the data set. If instead, you retain the data set, other people might also import that root certificate and use it to sign counterfeit certificates.

You can transfer through NJE (the TSO/E commands TRANSMIT and RECEIVE), or you can use FTP. If you use FTP, make sure that you transfer in binary mode, and that the data set has record format VB, record length 84.

This export/import method ensures that the root certificate has the same keys on both sides, so that a certificate that is signed on one side can be verified on the other side.

After you create or obtain root certificates, use job CKNRACF4 to create the certificates and keyrings. Run this job on all systems where you want to deploy the zSecure server. Adapt the job to the names that apply to your installation. In particular, the certificate that is generated for the zSecure Server must be specific for use by the zSecure Server application. This is enforced by the DOMAIN name specified in the ALTNAME certificate extension. The domain name must be the value of the ZSECSYS corresponding to the OWNSYS used for the zSecure Server.

The certificates used for the other zSecure servers must be mapped to a non-revoked user on the local system. This can be done using

- One-to-one certificate to user ID association
- Certificate name filtering
- The hostIdMappings certificate extension

For more information about certificate mapping, see the chapter on RACF and digital certificates in the *RACF Security Administrator's Guide*. The one-to-one certificate mapping is the most secure method, and requires exporting the certificate on one system, and importing (adding) it on the other system. The requirement for mapping the certificate to a locally defined user is enforced by the value SAFCHECK for the ClientAuthType in the AT-TLS policy.

The requirement that the certificates must be mapped onto a local user ensures that only known certificates are used. Without this requirement, it is possible that new (unintended and unknown) certificates are accepted if they have been signed by the trusted CA (Certificate Authority) used for your zSecure Server certificates.

To use digital certificates, the server-userid must also have READ access to IRR.DIGTCERT.LISTRING in the FACILITY resource class.

## Additional security measures

If you need to control access to local data or commands based on the system, several options are available.

You can use the userid mapping rules to assign a userid with a low authorization to all users of such a system. An alternative is to assign a userid to the entire ZSECNODE using the CKNADMIN.FROMNODE.<nodename> resource described in [“Authorization and userid mapping when using the zSecure Server”](#) on page 196. If the APPLDATA of the matching profile has a value, it is used as the userid for the ZSECNODE. If the userid is present, it must have access to the individual CKNDSN resources, in addition to the regular mapped userid that represents the logged-on user. In this setup, two users must have access:

- The mapped userid representing the logged-on user
- The userid assigned to the entire ZSECNODE

If either user has insufficient access to the CKNDSN resource, access is denied.

The additional test for the user ID assigned to the entire ZSECNODE is bypassed if the *nodename* where the request originated is the same as the current zSecure nodename. So, if the source server is running on the same ZSECNODE as the target server, only the mapped user ID representing the logged-on user must have access to the CKNDSN resources.

Using these additional security measures, you can control access to input files and authority to run commands based on the system from where the requests originate. You can also retain granularity based on the logged-on user. In the following example environment:

- There are two production systems (PRD1SYS and PRD2SYS) and one external system (EXT1SYS).
- PRD1SYS is defined as part of zSecure node PRD1NODE, and PRD2SYS is defined as part of zSecure node PRD2NODE.
- The user IBMUSER is logged on to system PRD1SYS, and is accessing PRD2SYS.

The following profiles are defined on system PRD2SYS:

```
CKNDSN . RACF . PRD2NODE . PRD2SYS . ACTIVE .          READ (IBMUSER , EXT1USER)
CKNDSN . CKRCMD . PRD2NODE . PRD2SYS . CKRCMD         READ (IBMUSER)
CKNADMIN . FROMNODE . PRD1SYS                          NOAPPLDATA          READ (IBMUSER)
CKNADMIN . FROMNODE . EXT1SYS                          APPLDATA (EXT1USER) READ (IBMUSER)
CKNUMAP . * . * . *                                    APPLDATA (=USERID)
```

The last profile (CKNUMAP) is the `userid` mapping rule. It specifies identity mapping, so the ID of the logged-on user IBMUSER is also used as the ID that needs authorization on the PRD2SYS system.

The third profile (CKNADMIN.FROMNODE.PRD1SYS) describes the authority to access PRD2SYS from PRD1SYS. IBMUSER has access. The profile does not have an APPLDATA field, and thus there is no additional system-level authorization.

The first two profiles (CKNDSN.RACF.PRD2NODE.PRD2SYS.ACTIVE. and CKNDSN.CKRCMD.PRD2NODE.PRD2SYS.CKRCMD) describe the authority to access the RACF database, and to issue commands. IBMUSER has access to both profiles.

In another scenario, the user IBMUSER is logged on to the system EXT1SYS, and is again accessing PRD2SYS. The same `userid` mapping rule is used to map IBMUSER on EXT1SYS to IBMUSER on PRD2SYS.

The fourth profile (CKNADMIN.FROMNODE.EXT1SYS) specifies in its APPLDATA field that the system-level `userid` EXT1USER is to be used for access verification. Because EXT1USER does not have access to the second profile (CKNDSN.CKRCMD.PRD2NODE.PRD2SYS.CKRCMD, which describes the CKRCMD resource), no one from the EXT1SYS (including IBMUSER) is authorized to issue commands for the PRD2SYS.

If a `userid` is specified in the APPLDATA of the FROMNODE profile for a node, the `userid` must match the `userid` that is associated with the certificate for that node.

## Setup to disable server security

You can run a zSecure Server without proper security for the communication with other zSecure Servers. To run a zSecure Server in this way, specify the INSECURE keyword on the zSecure Server OPTION statement in the zSecure-Server configuration file. Both servers must specify the INSECURE keyword. Using an insecure connection for a particular connection is controlled by the CKNADMIN.INSECURE.<zsecsys-name> resource in the XFACILIT resource class. The <zsecsys-name> is the partner node, and the started task user must have at least READ access. If no matching profile is found, or if the started task user has insufficient access, the connection is rejected.

```
CKNADMIN.INSECURE.<zsecsys-name> READ(server-userid)
```

It is also possible to allow a mismatch between the hostname as present in the ALTNAME of the certificate and the zsecsys of the partner zSecure Server. This is controlled by the access of the server-userid to the profile matching resource CKNADMIN.CERTOKAY.<zsecsys-name>. If no matching profile is found, or if the started task user has insufficient access, the connection is rejected. The profile must be defined on the system that detects that its partner has a non-matching certificate.

```
CKNADMIN.CERTOKAY.<zsecsys-name> READ(server-userid)
```

## Summary of Secure Server Communication

The following table summarizes the various security-related settings:



Table 5. Security-related settings					
Area	Subarea	Field	Setting	Effects	
TTLRule	TTLGroupAction	TTLSEnabled	on	Enforces use of certificate	
	TTLKeyringParms	Keyring	value	Specifies the name of the keyring	
	TTLSEnvironmentAdvancedParms	ClientAuthType	SAFCHECK		Specifies that certificate must be mapped to a RACF user
		ApplicationControlled	OFF		Specifies that AT-TLS is always used, and is not dependent on the application code.
	TTLConnectionAdvancedParms	CertificateLabel	value	Specifies the label of the certificate	
	TTLSCipherParms	V3CipherSuites	list of values		Specifies the list of encryption methods. If omitted, simple encryption is used.
Certificate	ALTNAME(DOMAIN(zsecsys- name))			Must match zsecsys-name	
	mapping onto RACF user		certificate-userid	Enforced by SAFCHECK in TTLS policy	
RACF	IRR.DIGTCERT.LISTRING	Access list	server-userid	Allow certificate retrieval	
	CKNDADMIN.FROMNODE. <zsecnode-name>	APPLDATA	node-userid	Extra CKNDSN verification.node-userid must match certificate-userid. Only applies if source system zsecnode-name is different from target (current) system zsecnode-name	
	CKNDSN.<type>.<nodename>.<sysname>.<dsname>	Access list	node-useridclient-userid	Controls access to data source	
	CKNDSN.CKRCMD.<nodename>.<sysname>.CKRCMD	Access list	node-useridclient-userid	Controls access to execute commands	
	CKNADMIN.INSECURE.<zsecsys- name>	Access list	server-userid	Allows missing certificates. Only required if source system zsecsys-name is different from target (current) system zsecsys-name	
	CKNADMIN.CERTOKAY.<zsecsys- name>	Access list	server-userid	Allows incorrect ALTNAME(DOMAIN(zsecsys-name)). Only applies if certificate is used on the connection.	
zSecure Server	Configuration file	OPTION	INSECURE	Allow missing certificates	

## Use of the zSecure Server to limit the need for access to the security database

You can use the zSecure Server in *self-connect* mode; that is, you can have a single zSecure Server send requests to itself. This way, the original user does not need permission to read your security database. Such a permission is, in principle, a security exposure, and that exposure can be addressed by access in PADS mode, or by the zSecure Server. For more information about PADS mode, see [“Setting up Program](#)

[Control and PADS access” on page 202](#). The zSecure Server in self-connect mode is a full alternative for PADS mode.

In zSecure Server self-connect mode, the user's permission to access data is governed by profiles in the XFACILIT class (just like in multi-system mode). Actual reading of the security database and of active SMF is done under authority of the server address space, not that of the (mapped) original user. All other data access is handled using the authority of the (mapped) client ID. Only the server ID needs access to the security database. The client ID always needs access to the applicable CKNDSN resource in the XFACILIT class.

You can combine a zSecure Server to run concurrently in self-connect and multi-system mode, or you can set up a dedicated server. For example, in a single z/OS image, you would set up only a single zSecure Server, running exclusively in self-connect mode.

To set up a server in self-connect mode, use the following steps. (See the preceding sections for further instructions.)

- Set up a JCL procedure in a system proclib.
- Decide on the value for the ServerToken, the name of the ZSECNODE, and the ZSECSYS.
- For a dedicated server, define only the local server; do not define any remote connections.
- For a dedicated server, you do not need AT-TLS for the server. Nor do you need the INSECURE parameter. If your AT-TLS policy also applies to the self-connection, you must ensure that all related actions as described in the previous section have been completed.
- In **SE.D**, update the default setup files to include the name of the zsecnode/sys.
- Update the default run option to include the ServerToken.
- Optionally, define an explicit generic CKNUMAP profile; for example:

```
CKNUMAP.<zsecnode>.*.<zsecnode> with appldata('=USERID')
```

- Define the necessary CKNDSN profiles; for example, for RACF, CKFREEZE, ACCESS files and unloaded SMF files.

## Chapter 10. Setup of zSecure Admin Access Monitor

The Access Monitor is a component of zSecure Admin that you can use to collect information about actual usage of resource profiles. This data is available for reporting and analysis from the Access Monitor option provided in zSecure Admin. Administrators can use the collected information to identify and remove unused access and resource profiles from the RACF database. For more information about the Access Monitor, see the *IBM Security zSecure Admin and Audit for RACF: User Reference Manual*.

See the following sections for information about setting up and operating the Access Monitor.

- “[Installation and post-installation requirements](#)” on page 62
- “[Operation of the Access Monitor](#)” on page 70
- “[Access Monitor function command reference](#)” on page 75

### Considerations when upgrading from a previous release of Access Monitor

Use the guidelines in this topic to upgrade from a previous release of Access Monitor.

If you are upgrading from a previous release of Access Monitor, ensure that all programs from the previous release were removed from all system LPA LIST and LINK LIST data sets. Failure to do so might cause abends after a subsequent start of the zSecure Admin Access Monitor started task (C2PACMON).

If you have not started C2PACMON since the last IPL of the system, no specific upgrade steps are required. Otherwise, ensure that you are using the correct software for the current release and continue the migration:

- When stopping the previous version of C2PACMON, you must use the SIPL command; for example, you can use the following operator command:

```
MODIFY C2PACMON,SIPL
```

After stopping the C2PACMON started task, you must run a C2XACTV job, using the RECOVER keyword. This C2XACTV job must use as STEPLIB the data set containing the software level of the previous release. After completion, you can start C2PACMON using the current release of the software. An example C2XACTV job is:

```
//RECOV EXEC PGM=C2XACTV
//STEPLIB DD DISP=SHR,DSN=<loadlib.zsecure.2.5.0>
//C2XPRINT DD SYSOUT=*
//C2XIN DD *
DYNEXIT RECOVER ICHRCX02
DYNEXIT RECOVER ICHRDY02
DYNEXIT RECOVER ICHRFY04
DYNEXIT RECOVER ICHRIX02
DYNEXIT RECOVER ICHRFY02
DYNEXIT RECOVER ICHRTX00
```

If you need to return to the previous release, you must again use the SIPL command to stop C2PACMON. After a successful stop of the started task, you run the RECOVER job again, but this time using the new STEPLIB.

- After stopping the current version of the zSecure Admin Access Monitor started task, you shut down and IPL the system. After the IPL, you start the zSecure Admin Access Monitor started task using the upgraded code. In this scenario, you do not need to perform any additional steps.

If you do not follow either of these procedures, startup might fail and an ABEND of the C2PACMON started task might occur. If startup fails with messages C2P0183E and C2P0123E, you might be able to recover using the FORCE startup parameter.

If you adapted or copied any of the consolidation or conversion CARLa queries, ensure that you update these, based on the latest CARLa queries as they are provided in SCKRCARL. If you use older CARLa queries, you might lose, for example, all records for UNIX events.

## Installation and post-installation requirements

---

Use the guidelines in this topic to ensure that your zSecure installation is correctly configured to use Access Monitor.

Before you can set up the Access Monitor, you must install zSecure as described in Chapter 4, “Installation of the software,” on page 9. During the installation process, make sure that the following items are configured:

- zSecure Admin must be installed. This product is installed by enabling the installation parameter `AdminRACF` in member `CKRZUPDI` before running the fast or formal installation jobs. See “Customization of the installation parameters” on page 11.
- The `AdminRacf` license must be enabled. See “Enabling of license features” on page 18.
- The `SCKRLOAD` component must be APF-authorized.
- You must have a zSecure configuration data set that enables the use of the Access Monitor.
  - You can create this configuration by running job `CKRZPOST` as described in “Creating zSecure configuration data sets” on page 25. If you want to use a dedicated zSecure configuration for the Access Monitor, then you only need to create a `CKRPARM` data set. You can comment out the DD-statements for the other data sets in the `CKRZPOST` job.
  - If you already have a zSecure configuration that is enabled for Access Monitor, continue using that configuration.
  - zSecure Access Monitor can collect event data about files and directories in the UNIX file system. Data collection is done through the dynamic syscall exits that are available in z/OS 2.3 and higher. Although the exit routines are called using z/OS dynamic exit support, some preparation is needed. Follow the steps in “Required preparation for collecting UNIX event data” on page 68.
- To use the Access Monitor on multiple z/OS images, make sure that the access monitor data sets for different images have different names. The names can be set up using image-dependent variables with at least different `SYS` parameters (such as `&SYS` or `&SYSCLONE`) in the data set names that are specified in the `C2PAMCNT` and `C2PAMCLT` members of `CKRPARM`.
- Alternatively, you can create separate copies of members `C2R$PARM`, `C2PAMP`, or both for each instance. If you have a shared JES procedure library, you can establish separate configurations using MVS system symbols.
- If you want to use an analytics application, for example IBM Z Operations Analytics, to report about events that zSecure Access Monitor collects, you must perform additional configuration steps. These are described in section “Optional customization for analytics preprocessing” on page 69.

## Configuration of Access Monitor

---

To start the Access Monitor and begin data collection and consolidation, perform the configuration tasks in this topic.

- [Prepare the JCL.](#)
- [Define the security resources and permissions.](#)
- [Create the required Access Monitor data sets.](#)
- [Customize data collection and consolidation parameters](#)
- [Start the Access Monitor.](#)

## Preparing the JCL

### About this task

The Access Monitor must run as a started task. Therefore, you must copy the C2PACMON procedure from the SCKRPROC data set to a started task procedure library.

### Procedure

1. To run C2PACMON as a started task, copy the C2PACMON procedure from the SCKRPROC data set to a started task procedure library. You can select a different member name, provided that you update the security resources that you will create.
2. In your copy of the C2PACMON procedure, specify the member name for the zSecure configuration that you intend to use. The default is C2R\$PARM. However, if your procedure library is shared across z/OS images, use an MVS system symbol in the procedure to assign a separate configuration member to each instance of the Access Monitor.
3. If you want separate data collection and consolidation parameters for each instance, specify your member in the PPARM parameter in the JCL. If desired, you can use a system symbol in the parameter member name. For example, you can modify the default PPARM statement shown in the following example to use the system symbol:

```
// PPARM=C2PAMP,          C2PACMON parameter member
```

The following example shows the modified command:

```
// PPARM=C2PAMP&SYSCLONE, C2PACMON parameter member
```

4. Store the configuration member(s) in a started procedure library. Within the configuration member(s), uncomment the line that sets the C2PACPRM parameter.

## Definition of security resources and permissions

This section describes the security resources and permissions required by the Access Monitor function of zSecure Admin and provides instructions to set them up. Unless you are sharing the security databases, you must establish these authorizations for all systems where the Access Monitor runs.

The following security resources and permissions are required:

- The userid and group that you selected to run the C2PACMON address space must be available.
- A STARTED profile with an STDATA segment where you can assign the required userids and groups to the started tasks.
- The userid for C2PACMON must have UPDATE access to the XFACILIT resources C2X.ICHRCX02, C2X.ICHRDX02, C2X.CHRFX02, C2X.ICHRFX04, C2X.ICHRIX02, and C2X.ICHRTX00. This userid also requires READ access to the XFACILIT resource CKR.CKRCARLA.APF.
- Started task output protection.
- Data set names and the profiles covering them. This can include PROGRAM profiles when Program Access to Data Sets (PADS) access is being used.
- To support the use of the DIAGNOSE USSCAPTURE command, the user ID for C2PACMON must have READ access to the OPERCMDS resource MVS.DISPLAY.PROG, and to the FACILITY resource CSVSYNEX.LIST.

You can use Job C2PZRINO in the SCKRSAMP data set to help set up these security resources. Note, however, that the security resources you create must be subject to your security policy, such as choices between generic and discrete profiles. You can decide whether to run this job after reviewing the RACF commands.

In this job, some things are assumed and not customized during the zSecure configuration. Consequently, you might need to change the job based on the following information.

- Group name SYSAUDIT is assumed as the group to contain system auditors, but you might choose another one. It is assumed that you (the installer) are connected to the SYSAUDIT group. If you are not connected to this group, the allocation of data sets for the Access Monitor using job C2PZRIN1 will fail.
- The group owner is set to SYSAUTH.
- It is assumed that profiles have been set up for the SCKRLOAD data set and that a separate profile exists for the other data sets. If you have a different setup, adapt the JCL in this job to reflect the requirements for your environment.
- It is assumed that PROGRAM profile CKRCARLA exists. If you do not use PROGRAM profiles you can remove the rdefines, ralters and permits for them.
- If you are activating analytics preprocessing, ensure that the started task user ID has authorization to create and remove files from the specified analytics directory. Also, the started task user ID must have at least read access to the input files that are specified in the analytics configuration member C2PAMANC. For more information, see [“Optional customization for analytics preprocessing”](#) on page 69.

## Required Access Monitor data sets

The Access Monitor function requires the following data sets:

- Data sets with intermediate data for the C2PACMON address space. In the JCL, these data sets are identified as SYSPRST1 and SYSPRRPT. The C2PACMON address space allocates these data sets as shared so that an authorized user such as an administrator can view them during troubleshooting. However, these data sets should not be shared otherwise. By default, these data sets contain the system ID in their names.
- The parameter data set specified in the zSecure configuration file. For more information about the zSecure configuration file, see [“Creating zSecure configuration data sets”](#) on page 25.
- The data sets that are used for the daily collection and the daily consolidation processes must be large enough to hold the access monitoring data. The required size depends on your environment. To prevent data loss, monitor these data sets closely for sufficient space allocation.

Job C2PZRIN1 in the CKRINST library is supplied to help you create these data sets. Before submitting the job, customize the JCL as follows:

- Unless you updated the default zSecure configuration file, change the JCLLIB and INCLUDE statements to specify the zSecure configuration file that you prepared.
- If you intend to run the Access Monitor function of zSecure Admin on multiple z/OS images, run the job multiple times, once for each image. Preferably, run each of these jobs under the z/OS image where the corresponding Access Monitor function will run.
- If you want to share the parameter data set among z/OS images, ensure that the C2PACPRM allocation is done only once. This data set is usually allocated by job CKRZPOST.

## Customization of data collection and consolidation parameters

You can customize these parameters to control the time interval for data collection and consolidation. Before specifying the Access Monitor parameters, you must create the parameter data set as described in [Create the required Access Monitor data sets.](#)

### Parameter file for Access Monitor started task

The Access Monitor function requires a parameter file for the started task. This parameter file is included using the PPARM JCL parameter in the Access Monitor started task procedure as described in [Prepare the JCL.](#) The parameter file must always be present. If you do not want to change any of the default values for the parameters, specify a parameter file with at least one line, which can be a comment line. A sample parameter file, showing default values, can be found in member C2PAMP in the SCKRSAMP data set.

The data collection that is done by Access Monitor must be done in a timely manner. To ensure that sufficient CPU resources are available, the Access Monitor started task resets itself to the WLM service

class SYSSTC. If you must assign a lower dispatching priority, create an applicable WLM classification rule that assigns the required WLM service class. You must also specify NoSetSRVClass on the OPTION statement to prevent the program from resetting itself to the SYSSTC service class. If you are running on a single processor system, careful analysis of your workload is required and using NoSetSRVClass might be preferred.

For more information about configuration parameters, see [“Configuration commands” on page 78](#).

## Passing VERIFY event records to zSecure Alert

The zSecure Access Monitor intercept routines for RACF events collect information that might otherwise be unavailable. One of the most obvious events is a signon, or more general the RACF VERIFY process. The default for this RACF function is to skip writing an SMF record for the event. The application has to explicitly request that an SMF record is created. zSecure Access Monitor collects information about such events even if the application does not request an SMF record. To enhance the functionality of zSecure Alert, you can configure zSecure Access Monitor to pass the collected information to zSecure Alert. The only event type that is currently supported is the RACF VERIFY event.

If you specify the EventsToAlert parameter on the Access Monitor OPTION statement, all VERIFY events detected by Access Monitor are passed to zSecure Alert. If you include this parameter while zSecure Alert is not active, a message is issued to indicate that the message transfer failed. For more information about the EventsToAlert parameter, see the OPTION command in [“Configuration commands” on page 78](#).

## Definition of the users or classes for which to collect detail data

The procedure for the C2PACMON started task has three DDNAMEs that refer to three members in the data set pointed to by the C2PACPRM configuration parameter. The default value for this data set is the CKRPARM data set used for the zSecure configuration. The three members are C2PAMJOB, C2PAMRCL, and C2PAMPCL. They are used to specify for which USERIDs the JOBNAME information is collected, and for which resource class and POE class the Port Of Entry (POE) information is collected. Discuss with the users of the collected Access Monitor events for which events the detail information is needed. Depending on the use of different jobnames and ports of entry, collecting this detail information might result in a significant increase in resource usage for the collected data, and for the data consolidation process. The default configuration members specify that no jobname or POE information is collected.

- Collection of jobname information is controlled by the contents of the C2PAMJOB member. This member has a two-column layout. The member name and the ruler line are not part of the member, but are shown here for clarity only. The ruler line highlights that the second column must start in position 10 of the record. All information in this member must be specified in uppercase. The following figure shows an example.

```
C2PAMJOB
-----1-----2
  IBMUSER  YES
  C2PSUSER *
```

The first column contains a USERID or the prefix of a USERID for which jobname information can be retained. The second column can contain the value YES or any other value. Jobname information is collected only for those user IDs or user ID prefixes for which the value YES has been specified. If the same user ID or user ID prefix value occurs on multiple lines, only the first instance is used. For user IDs that are not included in the C2PAMJOB member, or that have a value other than YES specified, jobname information is not collected. Effectively, records without the value YES are ignored. If the actual user ID matches multiple lines (as a prefix, or as a user ID) in the C2PAMJOB file, any YES specification overrides other specifications.

Jobname information is always collected for jobs or started tasks that run without a RACF-defined user ID.

- Collection of Port Of Entry information is controlled by the contents of the C2PAMRCL and C2PAMPCL members. These members each have a two column layout. Examples are shown after this paragraph.

The member name and the ruler line are not part of the member, but are shown here for clarity only. The ruler lines highlight that the second column must start in position 10 of the record.

```
C2PAMRCL
  ----+-----1-----+-----2
  OPERCMDS YES
C2PAMPCL
  ----+-----1-----+-----2
  CONSOLE YES
  TERMINAL YES
```

The first column contains a resource class for which POE information is controlled. The second column can contain the value YES or any other value. The C2PAMRCL member specifies the resource class for which the access verification is done. This can be any RACF resource class, such as DATASET, FACILITY, or OPERCMDS. The C2PAMPCL member specifies the resource class (type) of the POE. The following POE classes are recognized: TERMINAL, CONSOLE, JESINPUT, APPCPORT, and SERVAUTH. POE information is collected only for those events for which the Resource class and the POE class both have the value YES specified. If either class specifies any other value, POE information is not collected for this access monitor event. Be sure that all information in these configuration members is specified in uppercase.

Updates to the three configuration members described here are effective for data collected after a restart or after the C2PACMON started task has done a consolidation run. For more information about restarting the C2PACMON started task, or the consolidation process as done by the C2PACMON started task, see [“Operator commands” on page 76](#).

## Definition of data collection and data consolidation files

zSecure Access Monitor collects data in two types of data sets: the daily collection data sets, and the daily consolidation data sets. Several times during the day, the collected Access Monitor records are saved in the daily collection data sets. Once a day, the daily collection data sets are combined and consolidated into the daily consolidation data sets.

### Daily collection data sets

As the name suggests, the daily collection data sets are used to collect data during the day. At the end of each SMF interval, the data in these data sets is saved. The default value for the SMF interval (INTVAL parameter in the SMFPRMxx member in PARMLIB) is 30 minutes, resulting in 48 intervals each day.

The daily collection data sets must be large enough to contain all collected records. Depending on the CollectionDataSet specification in the Access Monitor parmlib member, the data set must accommodate records for a single SMF interval, or records for the entire day:

- If you specified SingleInterval for the CollectionDataSet parameter, the daily collection data set is closed and a new one is created at the end of every SMF interval.
- If you specified MultiInterval, available data is saved, and the data set is prepared for additional data to be added at the end of the next SMF interval.

Using SingleInterval has the advantage that data collected during the day is available for analysis. Using MultiInterval has less processing overhead.

Actual data collection is done in a temporary data set that is renamed at the end of the SMF interval, or at ConsolidateTime.

The specification for the data set name in member C2PAMCLT must end in D&LYR2.&LMON.&LDAY. . T&LHR.&LMIN. or D&LYMMDD. . T&LHR.&LMIN. This results in a timestamp that is formatted as *Dyymmdd.Thhmm*. The date and time qualifiers are replaced by X000000.X0000 for the data set name of the temporary data set that is used for the actual data collection. After adding all required records, the data set is renamed using the date and time when it was created.

### Daily consolidation data sets

Once a day, the daily collection data sets are combined and consolidated into the daily consolidation data sets. The daily consolidation is done at the time that is specified through the ConsolidateTime



parameter in the Access Monitor configuration member. During the consolidation process, the data from the various intervals is combined into a single interval. By default, the consolidation is done a few minutes after midnight.

The daily consolidation is done at the time that is specified through the ConsolidateTime parameter in the Access Monitor configuration member. During the consolidation process, the data from the various intervals is combined into a single interval. By default, 48 intervals are collected each day, based on 24-hour activity and an SMF interval of 30 minutes. By default, the consolidation is done a few minutes after midnight.

The Access Monitor function allows flexibility in specifying the names and other allocation parameters for the data collection and consolidation data sets. You specify the allocation parameters in two parmlib members: the C2PAMCLT member for the daily collection data set and the C2PAMCNT member for the consolidation data set. These two parmlib members must contain a TSO ALLOCATE command. Sample members are available in the SCKRSAMP data set provided with zSecure. The following two examples show the contents of these sample files.

#### **Example: C2PAMCLT parmlib member to allocate the data collection file**

```
alloc reuse fi(c2pamcol) -
DA('your_prefix.C2PACMON.D&LYR2.&LMON.&LDAY..T&LHR.&LMIN.') -
mod space(1,1) cylinders release -
recfm(v b) lrecl(2123) blk(27998) storclas(your_class)
```

#### **Example: C2PAMCNT parmlib member to allocate the data consolidation file**

```
alloc reuse fi(c2pacmon) -
DA('your_prefix.DATA.C2PACMON.D&LYR2.&LMON.&LDAY.') -
mod space(1,1) cylinders release -
recfm(v b) lrecl(2123) blk(27998) storclas(your_class)
```

In the previous ALLOC commands, the following rules apply:

- Multiple input lines are allowed.
- The minus sign (-) indicates continuation lines.
- Columns 73-80 are ignored.
- The total command length must be less than 255 characters. The length includes all blanks between the last significant character on a line and the subsequent line continuation character, the minus sign (-).
- Aside from the symbol substitution, the command entered in these members must be a complete and valid TSO ALLOCATE command. Remove any keywords that are not needed (for example, the VOLUME keyword).
- The reuse and file keywords must be kept as shown in the example. The file name specified must be C2PAMCOL for the data collection file and C2PACMON for the data consolidation file.
- System symbols can be included anywhere in the command. They must be specified in uppercase. User and JCL symbols are not supported.
- The record format of the data set must be variable blocked, as indicated by the RECFM(V B) keyword.
- The data set name specification must begin with the string DA( '
- The data set name specification must end with the string ' )
- The specification for the data set name in member C2PAMCLT must end in D&LYR2.&LMON.&LDAY..T&LHR.&LMIN. or D&LYMMDD..T&LHR.&LMIN. This results in a timestamp that is formatted as *Dyymmdd.Thhmm*. The date and time qualifiers are replaced by X000000.X0000 for the data set name of the temporary data set that is used for the actual data collection. After adding all required records, the data set is renamed using the date and time when it was created.
- The specification for the data set names in member C2PAMCNT must end in D&LYR2.&LMON.&LDAY or D&LYMMDD. This results in a timestamp formatted as *Dyymmdd*.

- You can specify any leading qualifiers you want, as long as the data set name after substitution remains valid.
- Specifying different prefixes for the daily collection files in member C2PAMCLT and the daily consolidation files in member C2PAMCNT has distinct benefits. The main advantage is that the DSNPREF keyword can be used to refer to a particular type of files. The daily consolidation files can be further consolidated by using the UNLOAD statement. However, the daily collection files must first be converted by using a SUMMARY statement before fast consolidation using the UNLOAD statement is possible. (An example CARLa using such a SUMMARY statement is provided in member C2PAMCVT.) Using a different prefix allows easy separation for different processing requirements.
- You can specify additional parameters for the allocation. For example, if your installation supports specification of SMS constructs such as STORCLAS, MGMTCLAS, or a DATACLAS for zEDC compression, you can use them here. For more information about zEDC data compression, see [“zEnterprise data compression \(zEDC\) for zSecure” on page 72](#).
- Optional comment lines must be included at the end. Comments are included between the comment delimiters /\* and \*/.

Failure to follow these rules can result in error messages and failures allocating the correct data sets during the daily consolidation process.

## Required preparation for collecting UNIX event data

zSecure Access Monitor can collect event data for files and directories in the UNIX file system. To enable the use of the dynamic syscall exits, the exit points must first be activated in UNIX.

### About this task

The activation consists of the following steps.

### Procedure

1. Add or verify the SC\_EXITTABLE statement in the active BPXPRMxx member in parmlib.

The follow shows an example statement:

```
RESOLVER_PROC (RESOLVER)
SC_EXITTABLE ('/etc/sc_exittable')
VERSION ('&UNIXVER.')
```

For more information about the SC\_EXITTABLE statement, see the BPXPRMxx section in *z/OS MVS Initialization and Tuning Reference*.

2. Add or update the sc\_exittable file that is specified in the SC\_EXITTABLE statement.

If the file does not exist, you can create one with UID(0) as owner, and access mode(750). An example sc\_exittable is provided in member C2PUSCEX in SCKRSAMP. Ensure that the sc\_exittable has at least the specifications for the syscall exits as provided in the example. For more information about the contents of the sc\_exittable file, see section "Customizing the BPXPRMxx member of SYS1.PARMLIB" in *z/OS UNIX System Services Planning*.

3. To activate the updates, issue a SETOMVS or SET OMVS command.
4. To start collecting UNIX file and directory event information, start C2PACMON, by using a parameter file with the OPTION CAPTUREUSSEVENTS.

**Note:** Collecting UNIX file and directory event information significantly increases the amount of data that is retained in the ACCESS files. Before starting to collect UNIX event information, ensure that your ACCESS files can accommodate the additional space requirement. For information about estimating the required space, see [“Estimating the required space for UNIX event data” on page 72](#).

## Optional customization for analytics preprocessing

zSecure Access Monitor can provide preprocessed access records for use by an analytics product like IBM Z Operations Analytics. The records are saved in a UNIX file and can be retrieved by one of the Analytics components. To enable this process, several additional steps have to be performed.

To enable zSecure Access Monitor to provide preprocessed access records, perform the following customization steps:

- Define a UNIX directory where Access Monitor can store the files. The started task user ID must have sufficient authority to create and remove files in this directory.
- Specify analytics keywords and parameters in the Access Monitor configuration member.
- Configure your analytics application to retrieve records from the UNIX work directory.
- When analytics preprocessing is active, ensure that the started task user ID has authorization to create and remove files from the specified analytics directory. Also, the started task user ID must have at least read access to the input files that are specified in the analytics configuration member C2PAMANC.

### Prepare a directory for storing analytics data

The analytics data is saved as a CSV file. These CSV files are kept for the specified number of days and then automatically removed (default is 5 days). The directory must have sufficient space to store the summarized access data. On average, the amount of disk space required for each day is 50% of the disk space of the daily collection data sets that are specified through C2PAMCLT.

If you run Access Monitor on multiple systems and use a shared USS file system, you must specify a dedicated directory for each system. In the Access Monitor configuration file, you can use system symbols in the name of the specified USS directory. System symbols must be specified in uppercase for correct substitution.

Like any UNIX directory, the directory must have both an owning user and an owning group. It is easiest to assign the user ID and group of the C2PACMON started task as the owner of this directory. You might want to create a dedicated USS file system for this directory and use an automount policy. The following is an example UNIX command to create the work directory:

```
mkdir -m 750 /u/c2pacmon
```

To specify the owner, you can use a command that is similar to the following example:

```
chown c2psuser:sysaudit /u/c2pacmon
```

If you specify a different user or group as owner, ensure that the C2PACMON started task user ID has sufficient authority to create and remove files in the specified directory.

### Update C2PACMON configuration files

When you have created the directory for the analytics files, update the C2PACMON configuration members. There are two members that must be changed:

- Copy member C2PAMANC from SCKRCARL to your Access Monitor configuration data set (as indicated by symbol C2PACPRM in your CKRPARM member). The C2PAMANC member contains the CARLa specification of the RACF input source and the name of a daily refreshed CKFREEZE data set. For more information about the daily CKFREEZE data set, see ["Use of a fresh CKFREEZE and UNLOAD each day" on page 42](#)". During an initial install, member C2PAMANC is copied to the CKRPARM data set as part of job CKRZPOST.
- Member C2PAMP (or the one indicated by parameter PPARM in the C2PACMON procedure) must contain statements to activate and configure the analytics file creation process. Activation is through the use of the ANALYTICS keyword on the REPORT statement. Configuration can be done through the

specification of sub-keywords and parameters on the ANALYTICS keyword. For more information on the syntax of the REPORT statement, see [“Configuration commands” on page 78](#).

## Configure CDP to retrieve and forward data to IBM Z Operations Analytics

IBM Z Operations Analytics uses its Common Data Provider (CDP) component to retrieve records from the z/OS UNIX log files. The CDP component must be configured to identify the directory where the files are located, and the naming pattern for the files.

The naming pattern of the CSV files created by zSecure Access Monitor is `/u/c2pacmon/AccMon.D?????.T?????.csv`, where `/u/c2pacmon` is the directory name that is specified in the C2PAMP configuration member.

## Operation of the Access Monitor

You can manage the RACF Access Monitor function by issuing commands from the operator console at startup and while the task is running. You can also control the operating environment for the Access Monitor function by providing input parameters in the parmlib DD-statement in the startup procedure. For instructions, see the following sections:

- [“Starting the Access Monitor STC” on page 70](#)
- [“MODIFY command to monitor or modify the Access Monitor started task” on page 71](#)
- [“Stopping the Access Monitor STC” on page 71](#)
- [“Configuration of the Access Monitor function using parmlib” on page 71](#)
- [“Memory or data storage problems when processing Access Monitor data” on page 72](#)
- [“Management of RACF exits installed by Access Monitor” on page 74](#)

Some commands are primarily intended to be issued as part of the PARMLIB file. These commands and their keywords and parameters are described in [“Configuration commands” on page 78](#).

## Starting the Access Monitor STC

To start the Access Monitor function of zSecure Admin, issue a START command from the operator console as shown in the following example:

```
S C2PACMON
```

In a production environment, use Automated Operation software or PARMLIB member COMMNDxx to automatically start the Access Monitor task soon after each IPL.

This command runs the procedure from the applicable system proclib. When entering the START command, you can also specify startup parameters to run diagnostic tests or force program initialization. These parameters are described in [“Access Monitor START parameters” on page 70](#).

### Access Monitor START parameters

For normal execution of the Access Monitor, you do not need to specify any startup parameters. By default, the Access Monitor detects if it is already active and issues an appropriate error message before ending. The Access Monitor is designed to use system resources effectively. If the Access Monitor started task has been shut down previously, the newly started task reuses those critical system resources that can be obtained only once and that cannot be returned to the system.

In some error situations, the Access Monitor started task fails to initialize. In these situations, you might need to specify one of the optional START parameters.

The following example shows a START command with the DEBUG parameter specified:

```
S C2PACMON, , , DEBUG
```

## DEBUG

Issues diagnostic messages during the first part of the initialization. These diagnostic messages can also be used to determine possible problems in processing the standard PARMLIB parameters. This setting is in effect until a subsequent DEBUG command is issued either from the operator console, or using PARMLIB.

## FORCE

Forces initialization to continue regardless of a previous execution. Use the FORCE option only when the Access Monitor started task cannot be started using other methods, and IPLing the system is undesirable. During normal operation, it is not necessary to use the FORCE command to start the system. If you have to use this command, create a problem report so that the issue can be investigated.

## DEBUG-FORCE

Activates both the DEBUG and FORCE options at startup.

The started task procedure C2PACMON provided with zSecure also provides the PPARM parameter to specify the main PARMLIB member that initializes the Access Monitor started task. This parameter can be used to override the value specified in the procedure. The default value specified in the procedure for this parameter is C2PAMP.

## MODIFY command to monitor or modify the Access Monitor started task

When the Access Monitor started task is active, the console operator can monitor or modify Access Monitor operations using the MODIFY console command. You can use the F command as an alias for the MODIFY command. The following example shows a modify command that displays the current status and options for the access Monitor:

```
MODIFY C2PACMON,DISPLAY
```

Be sure that the text after the comma is one of the supported operator commands for the Access Monitor started task. For information about these commands, see [“Operator commands” on page 76](#).

## Stopping the Access Monitor STC

To stop the Access Monitor started task, run the STOP command from the console. You can use the P command as an alias of the STOP command, for example:

```
P C2PACMON
```

The STOP command can also be issued as the parameter on the MODIFY command.

```
F C2PACMON,STOP
```

For more detailed information about the Access Monitor operator commands, see [“Configuration commands” on page 78](#).

**Note:** Do not use the CANCEL command to stop the Access Monitor STC. If the CANCEL command is used, collected event data is lost. Also, existing data sets are not processed and cleared, which might lead to out-of-space ABENDs.

## Configuration of the Access Monitor function using parmlib

You can control Access Monitor functions by setting parameter values in the parmlib.

By default, the PARMLIB DD-statement refers to the C2PAMP member in the C2PACPRM data set. The following commands are examples of commands that can be specified in parmlib:

- DEBUG to diagnose problems
- OPTION for managing the in-memory data buffers
- REPORT for specifying the data capture interval, the CARLa statement members, and other items.

The input parameters can be specified in the form of commands with keywords. Use TSO conventions when specifying these commands. For details about the C2PAMP parameter file, see [“Parameter file for Access Monitor started task”](#) on page 64.

## Estimating the required space for UNIX event data

---

Collecting UNIX file and directory event information significantly increases the amount of data that is retained in the ACCESS files. Before starting to collect UNIX event information, ensure that your ACCESS files can accommodate the additional space requirement.

### About this task

An estimate of the required space can be obtained by using an existing CKFREEZE data set as input.

### Procedure

1. Use option **SE.1** to select a CKFREEZE data set that contains UNIX information.
2. On the detail panel with the data set name specification, use the L line command to list information about the CKFREEZE data set.
3. Select the SPACE report, and look for the lines for record types 61 and 62 (UNIX directory and UNIX file).
4. Add the sizes that are shown for these two record types.
5. Multiply the result by five to obtain a first order estimate for the additional size that is needed for the UNIX events.
6. Ensure that your daily collection and daily consolidation data sets have sufficient space to accommodate the UNIX information.

## Memory or data storage problems when processing Access Monitor data

---

If you have problems with memory or storing data, you might need to adjust some of the following configuration settings for the Access Monitor program:

- Data collected in the access monitor records is transferred to the CKRCARLA program to be saved to disk. The interval period is controlled by the INTVAL parameter in the SMFPRMxx in the parmlib. The default value is 30 minutes.
- Access Monitor runs as a started task and captures RACF events for all tasks in the system. In large systems with much activity, the amount of buffer space required by the program can be significant. If you find that the buffer space is not sufficient to run the Access Monitor collection, you can adjust the buffer space parameters to specify values for your installation. For details, see [“Parameter file for Access Monitor started task”](#) on page 64. The Access Monitor started task provides buffer usage statistics messages that can help you select the optimum buffer size for your installation
- The data sets used for daily collection and consolidation must be large enough to hold the required data. The required size of these data sets is largely dependent on your environment. If necessary, you can adjust the allocation and characteristics of these data sets using the Access Monitor parmlib members C2PAMCLT and C2PAMCNT. To prevent data loss, monitor these data sets closely for sufficient space allocation.

## zEnterprise data compression (zEDC) for zSecure

---

Use the guidelines in this topic to plan the implementation of zEnterprise data compression in your organization.

DFSMS (BSAM/QSAM) introduced a new type of compression for non-VSAM extended format data sets: zEnterprise data compression (zEDC). With zEDC compression, no separate dictionary needs to be

created, as zEDC compression hides the dictionary in the data stream. A new dictionary starts in each compression unit. The system can decompress the segment as is.

zEDC compression works well with consolidated zSecure Admin Access Monitor data sets. Compression ratios of more than a factor 10 are possible.

Requesting new zEDC compression for new data sets is similar to requesting the existing types of compression (generic and tailored compression). It can be selected at the data set level, system level, or both.

#### **Data set level**

In addition to the existing Tailored (T) and Generic (G) values, new zEDC Required (ZR) and zEDC Preferred (ZP) values are available on the COMPACTION option in data class. These values indicate how the system is to proceed if the zEDC function cannot be used for the data set being created, as follows:

##### **ZR**

zEDC Required. The system fails the allocation request if the zEDC function is not supported by the system or the minimum allocation amount requirement (5 MB, or 8 MB Primary if no Secondary) is not met.

##### **ZP**

zEDC Preferred. The system does not fail the allocation request, but rather create either a tailored compressed data set if the zEDC function is not supported by the system, or create a non-compressed extended format data set if the minimum allocation amount requirement (5 MB, or 8 MB Primary if no Secondary) is not met.

#### **System level**

In addition to the existing TAILORED and GENERIC values, new zEDC Required (ZEDC\_R) and zEDC Preferred (ZEDC\_P) values are available on the COMPRESS parameter found in IGDSMSxx member of SYS1.PARMLIB.

##### **zEDC\_P**

Tells the system to not fail the allocation request but rather create either a tailored compressed data set if the zEDC function is not supported by the system or create a non-compressed extended format data set if the minimum allocation amount requirement (5 MB, or 8 MB Primary if no Secondary) is not met.

##### **zEDC\_R**

Tells the system to fail the allocation request if the zEDC function is not supported by the system or the minimum allocation amount requirement (5 MB, or 8 MB Primary if no Secondary) is not met.

zEDC compression can be activated with SET SMS=xx or at IPL. Data class continues to take precedence over system level. The default continues to be GENERIC.

If you do not have zEC12 GA2/zBC12 or zEDC Express® cards with the previously mentioned Microcode Update, DFSMS can still create zEDC compressed format data sets based on user options (in Data Class or PARMLIB). In this case, BSAM/QSAM writes data non-compressed. BSAM/QSAM decompresses existing compressed data through software inflate.

For more information about zEDC compression, see *z/OS MVS Callable Services for High Level Languages*.

### **zEDC Express via QSAM/BSAM – Setup: PARMLIB**

To request the use of zEDC compression at the system level when you create new compressed format data sets (COMPACTION=Y in Data Class), new values are defined for the COMPRESS parameter found in IGDSMSxx member of SYS1.PARMLIB: COMPRESS (TAILORED | GENERIC | zEDC\_R | zEDC\_P). See [System level](#) for the explanations of zEDC\_P and zEDC\_R.

### **How to use zEDC for zSecure Access Monitor**

You can add an appropriate DATACLAS to SCKRSAMP(C2PAMCNT) to be used when you create consolidation data sets, and to the C2PECDTE REXX for monthly consolidation. In the TSO ALLOC



command that is generated, replace **MGMT** with MGMTCLAS and DATACLAS. You can do the same for the Y12MON data set.

## Management of RACF exits installed by Access Monitor

The Access Monitor started task dynamically installs additional RACF exits. Internally, the Access Monitor program (C2PACMON) calls the C2XACTV program to effectuate the required changes. The C2XACTV program can also be called as a stand-alone program. The Access Monitor RACF exits are implemented using a two level approach. The top level is an exit router module that is pointed to directly by a RACF control block. The exit router module calls up to three functional sub exits: a pre-processing, a main, and a post-processing sub exit. If a RACF exit is already active at the time that Access Monitor is started, the original exit routine is moved down to the main sub-exit. Access Monitor installs its data collection exit as the post-processing sub exit.

Normally, the Access Monitor program removes the sub exits and the exit router module during termination. However, there might be situations where the removal of the exits fails. In those situations, the sub exits are still installed and called for the related RACF events. You do not need to take any action to remove these exits, because they perform no function if the started task is not active. Of course, if present, the original installation exit is still called by the exit router module. If you want to remove the Access Monitor exits that were dynamically installed, you can run a job similar to the following:

```
//RUNIT EXEC PGM=C2XACTV
//STEPLIB DD DISP=SHR,DSN=<hlq.SCKRLOAD>
//SYSTSPRT DD SYSOUT=*
//C2XPRINT DD SYSOUT=*
//C2XIN DD *
DYNEXIT DEACTIVATE ICHRCX02
DYNEXIT DEACTIVATE ICHRFX02
DYNEXIT DEACTIVATE ICHRFX04
DYNEXIT DEACTIVATE ICHRDY02
DYNEXIT DEACTIVATE ICHRIX02
DYNEXIT DEACTIVATE ICHRTX00
```

The Access Monitor RACF exit router module uses either z/OS dynamic exit support or a direct branching method to call the functional sub exits.

If z/OS dynamic exit support is used, the sub exits are each protected using standard z/OS recovery services. If a sub exit abends, the sub exit is automatically disabled to avoid any subsequent abends. Disabling the sub exit is not done immediately, but only after the same sub exit has abended 255 times. If a sub exit becomes disabled, a message similar to the following is shown on the operator console and the system log:

```
CSV430I MODULE ICHRCX02 FOR EXIT C2X.ICHRX02 HAS BEEN MADE INACTIVE DUE TO
ABEND=0C1000 REASON=00000001
```

If this occurs, the sub exit can be reactivated using either of the following methods:

- Use the C2XACTV utility program to DEACTIVATE and ACTIVATE the affected exit. This requires JCL similar to the following:

```
//RUNIT EXEC PGM=C2XACTV
//STEPLIB DD DISP=SHR,DSN=<hlq.SCKRLOAD>
//SYSTSPRT DD SYSOUT=*
//C2XPRINT DD SYSOUT=*
//C2XIN DD *
DYNEXIT DEACT ICHRCX02
DYNEXIT ACT ICHRCX02
```

The user running this job must have UPDATE authority to C2X.ICHRX02 in the XFACILIT resource class.

- In some situations, it might also be possible to issue an operator command similar to the following:

```
SETPROG EXIT,modify,exitname=c2x.ichrcx02.pst,modname=c2prcx02,state=active
```

- It is also possible to RESTART Access Monitor processing by issuing the following operator command:



```
MODIFY C2PACMON,RESTART
```

The Access Monitor started task stops all internal subtasks and calls the C2XACTV program to remove its RACF exits. Next, it performs all required functions similar to those during a regular start of the program. The RESTART function acts as an efficient method to STOP and START the entire C2PACMON task, without side effects like the loss of a non-reusable address space.

For more information about the options for the dynamic exits, see “OPTION command” on page 80. For more information about the C2XACTV program see the relevant sections in the *IBM Security zSecure Admin and Audit for RACF: User Reference Manual*.

## Change of RACF EXIT calling modes

The Access Monitor OPTION statement is used during initialization of the started task. If you want to switch the exit calling method, you can use the C2XACTV program. The RECOVER function removes the exits from storage. Using the CSVDYNEX, DIRECT, or FASTSTORE keyword on the ACTIVATE function installs the router exit module for the desired mode. A job like the following can be used to switch to FASTSTORE mode:

```
//RUNIT      EXEC PGM=C2XACTV
//STEPLIB   DD DISP=SHR,DSN=<hlq.SCKRLOAD>
//SYSTSPRT DD SYSOUT=*
//C2XPRINT  DD SYSOUT=*
//C2XIN     DD *
DYNEXIT DEACTIVATE ICHRFX04
DYNEXIT RECOVER   ICHRFX04
DYNEXIT ACTIVATE  ICHRFX04 FASTSTORE
```

For more information about the C2XACTV program, see the relevant sections in the *IBM Security zSecure Admin and Audit for RACF: User Reference Manual*.

## Management of UNIX syscall exits installed by zSecure Access Monitor

During normal activation of zSecure Access Monitor with the CaptureUSSEvents option, the initialization part of the program adds the dynamic exit routines to the defined exit points. Three exit points are used by zSecure Access Monitor, as described in the following table:

Exit point name	Exit routine
BPX_PRE_SYSCALL	C2PUSC01
BPX_POST_SYSCALL	C2PUSC02
BPX_PREPROC_TERM	C2PUSC03

During normal termination of zSecure Access Monitor, the exit routines are removed from these exit points. You can also manually add or remove these routines by using appropriate operator SETPROG or SET PROG commands. Members C2PAMP01 and C2PAMP02 of SCKRSAMP provide two sample PROGxx members to add or remove the Access Monitor routines from these exit points.

## Access Monitor function command reference

- For information about operator commands, see “Operator commands” on page 76.
- For information about configuration commands, see “Configuration commands” on page 78.

## Operator commands

When entered as an operator command, these commands do not require additional keywords. You can also abbreviate a command to the first four characters; for example, type CONS for the CONSOLIDATE command.

### CRSH

This command causes an immediate abend of the C2PACMON started task, without any recovery. It is intended for IBM internal testing only.

### CONSOLIDATE

Runs the daily consolidation process, which includes the following steps:

1. The daily collection processing task is stopped and started.
2. The daily collection data set is closed, reallocated according to the C2PAMCLT template file, and then reopened for continued daily data collection.
3. The consolidation task is restarted.
4. The daily collection data sets from the previous day are consolidated into the summary data set from the previous day.

When the daily collection processing task restarts, the restart process activates the Access Monitor commands and parameters specified in the parmlib (default member C2PAMP). At this time, the OPTION command is not processed. Other commands, such as DEBUG and REPORT, are processed. Because the consolidation process is designed to be non-disruptive so that all relevant RACF events continue to be collected, it can take several minutes to complete the entire process.

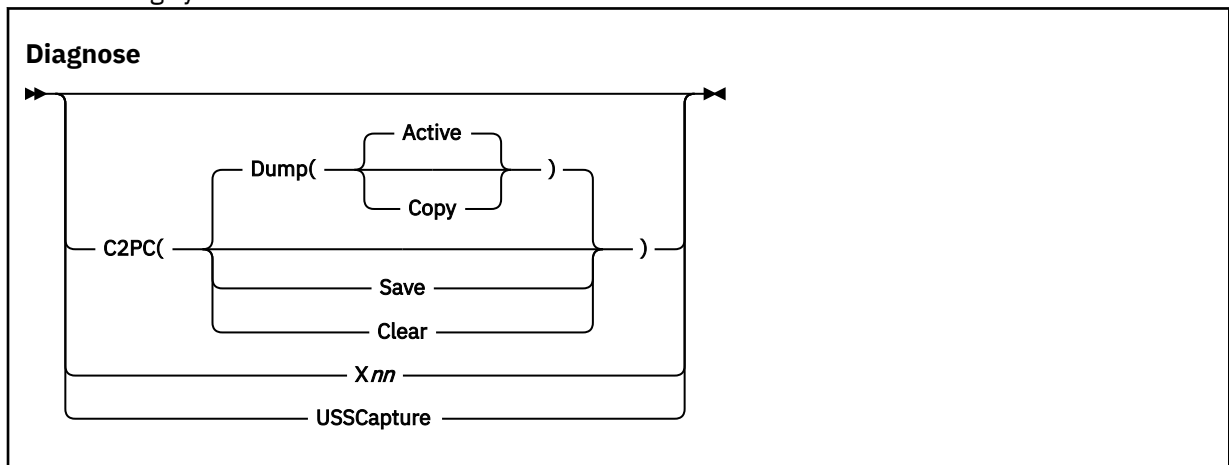
### DEBUG

Controls the diagnostic and monitoring messages that can be generated by the program. The command is effective immediately until the next daily consolidation run.

For a complete description of the keywords for the DEBUG command, see [“DEBUG command” on page 78](#).

### DIAGNOSE

The DIAGNOSE command is intended for IBM support personnel to diagnose problems. It allows dumping internal control blocks and testing certain recovery scenarios. The DIAGNOSE command has the following syntax:



The keywords and parameters have the following values:

#### C2PC

Information from the internal C2PC control block is to be saved or displayed on the operator console. This information is intended for IBM support personnel only. The following suboptions are available:

**Dump**

The Active or saved Copy of the C2PC data area is displayed on the operator console in dump format.

**Save**

The active C2PC data area is saved in the C2PC copy area.

**Clear**

The saved copy of the C2PC data area is cleared (reset to binary zeros).

**Xnn**

This keyword is intended for IBM internal use only. It causes certain error situations which are used to test recovery actions. *nn* is a numeric value.

**USSCapture**

This keyword causes the Access Monitor started task to issue several MVS display commands to show the status of the UNIX dynamic syscall exits and the required modules. It also shows the contents of the UNIX syscall exit table control block (ETAB). The contents of the ETAB are not documented. The display is only intended for diagnostic purposes by IBM support personnel.

**DISPLAY**

Displays the current status and option settings for the Access Monitor function. The display includes the current option settings, buffer space used, the number of the buffer currently in use, and the status of several error indicators if they are set.

The DISPLAY command does not support any additional keywords.

**REPORT**

Sets the values for the keywords that control the processing of the captured data. The new values are used the next time that they are needed. When the command is issued using the operator MODIFY command, that time can be almost instantaneous, or never. For example, a new value for the *Interval* parameter is used starting at the beginning of the next *Interval*. On the other hand, a new value for the *consolidatetime* is never used, because it is referenced only when the daily consolidation run is completed. At that time the value is reset to the value specified in the parmlib.

For a complete description of the keyword for the REPORT command, see [“REPORT command” on page 86](#).

**RESTART**

Gracefully shuts down the Access Monitor data collection processing, and then immediately reinitializes the task. The address space in which the Access Monitor started task is running is not stopped, and no additional console operator commands are needed to reactivate Access Monitor function processing.

The main difference between a restart and a STOP command followed by a START command for the started task is the preservation of the Address Space ID (ASID). Also, possible changes in the started task procedure are not effective during RESTART processing.

During the time required to process the RESTART command, some RACF access or define requests are not recorded.

Because the STOP/START sequence results in marking the address space as non-reusable, the RESTART command is preferred in most situations. This command prevents loss of potentially critical system resources.

The RESTART command does not support any additional keywords.

**SIPL**

Issue this command only at the request of IBM Software Support personnel, or when explicitly required during release migrations. When the command is run, all in-memory data structures are freed, a system-level linkage index (LX) is lost, and the address space is marked as non-reusable. System level LXes are a limited resource that cannot be recovered without an IPL of the system.

If you upgrade the Access Monitor program, the installation instructions might require you to shut down the previous version of the Access Monitor using this SIPL command.

The SIPL command does not support any additional keywords.

### STOP

Gracefully shuts down the Access Monitor started task. After the task ends, some memory remains reserved so that critical system resources can be used during a subsequent restart of the Access Monitor started task. The effect of the STOP modify command is identical to that of the MVS STOP command.

The STOP command does not support any additional keywords.

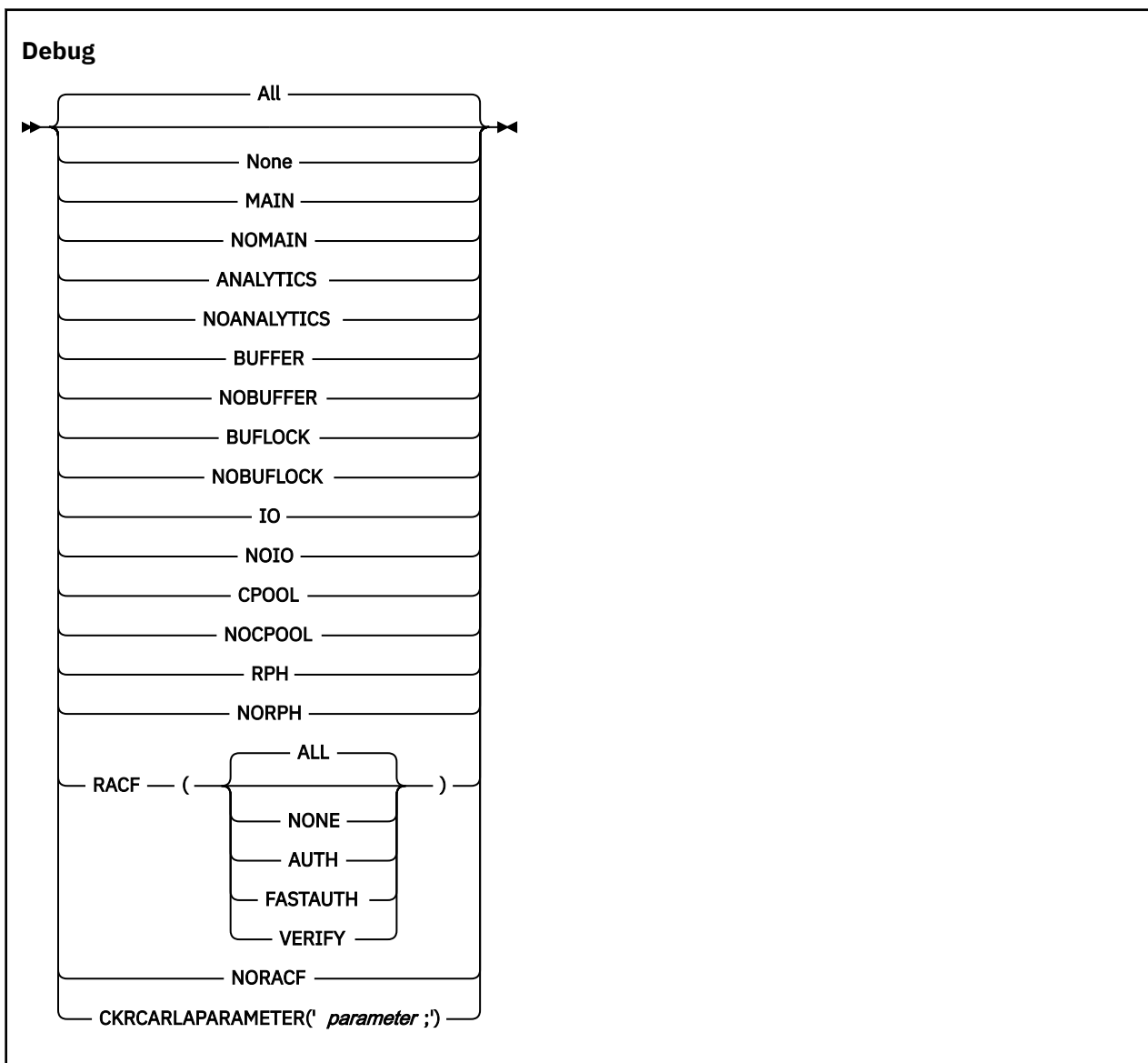
## Configuration commands

Normally, only the OPTION and REPORT commands are required. If you want to obtain diagnostic information, use the DEBUG command.

### DEBUG command

The command syntax is provided in the following diagram.

**Note:** You can specify only one option at a time. To enable multiple debug options, issue the DEBUG command multiple times. The DEBUG command can be run from the operator console or from the parmlib.



The keywords and variables have the following values:

**All**

Write all diagnostic messages to the console. ALL is the default setting for displaying messages if you do not specify a parameter on the DEBUG commands. Most of these messages are intended to assist during problem determination, and are not intended for routine customer use. The messages resulting from a DEBUG BUFFER command can be used routinely to determine the minimum size needed for the data buffers.

**None**

Deactivates creation of all diagnostic messages.

**MAIN**

Write diagnostic messages related to mainline processing to the console. This includes responding to operator commands, initialization and management of all subtasks, and major buffer management functions.

**NOMAIN**

Do not write diagnostic messages related to mainline processing to the console. This setting suppresses the following types of messages:

- Responses to operator commands
- Initialization and management of all subtasks
- Major buffer management functions.

**ANALYTICS**

Write diagnostic messages related to analytics file processing to the console.

**NOANALYTICS**

Do not write diagnostic messages related to analytics file processing to the console.

**BUFFER**

Write buffer usage statistics to the console (and to joblog and syslog) at the end of each reporting interval. These messages can help determine the number of Access Monitor records captured, and the amount of storage required. You can use these messages to track the minimum and maximum amount of buffer storage needed.

**NOBUFFER**

Do not write buffer usage statistics to the console.

**BUFLOCK**

This debug option is intended to assist in diagnosing the reason that a task could not save a record in the C2PACMON buffers. If such a situation occurs, an SVC dump is created of the address space where the event occurred. The BUFLOCK option is automatically disabled until the DEBUG command is issued again, either through an operator command or from PARMLIB.

Note that the SVC dump is not an indication that any error occurred. The dump is created only to assist in determining why the task could not save the record.

**NOBUFLOCK**

The BUFLOCK debug option is not used.

**IO**

Trace all operations processed by the Access Monitor interface routine to CKRCARLA using SYSLOG. Using this parameter might result in large numbers of writer-to-operator (WTO) messages. This function is intended to help IBM Software Support personnel diagnose internal problems in the product.

**NOIO**

Do not generate I/O diagnostic messages.

**CPOOL**

Show the number of cell pool extents at the end of each reporting interval. The messages that this generates helps to determine the number of concurrent users of these cell pools. Activate this debug option only on request of zSecure support.

**NOCPOOL**

Do not show the number cell pool extents.

**RPH**

Show diagnostic messages about use and management of the resolved path cache.

**NORPH**

Do not show diagnostic messages about use and management of the resolved path cache.

**RACF**

Specifies the RACF events for which diagnostic information about the collected data is shown on the system operator console. The subkeyword specifies the type of event. If no event is specified, diagnostic information for all events types is shown. This function is intended to help IBM Software Support personnel diagnose internal problems in the product.

**NORACF**

Messages for events related to the RACF data collection process are not issued. This function is intended to help IBM Software Support personnel diagnose internal problems in the product.

**CKRCARLAPARAMETER**

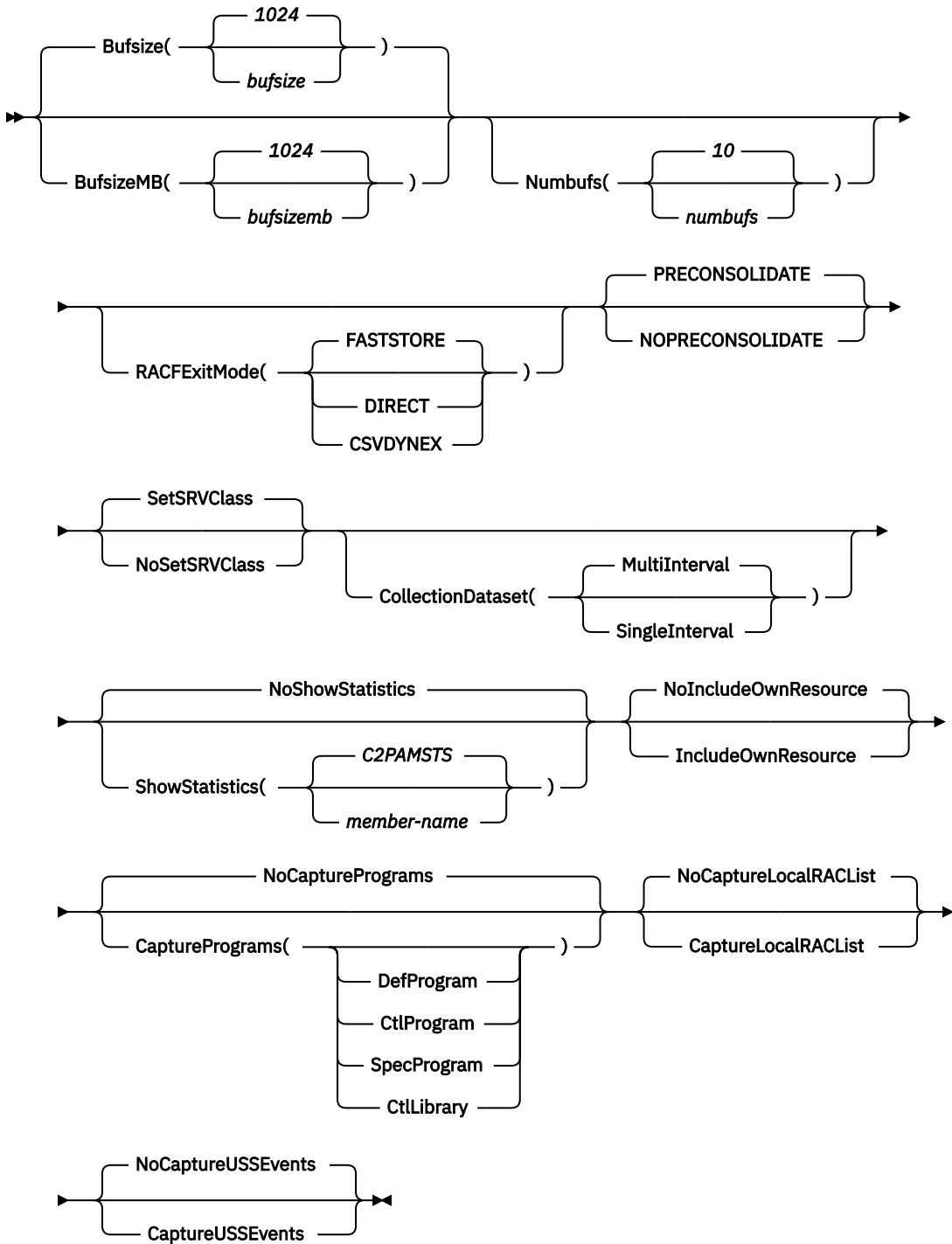
Specifies a string that is to be passed to all instances of CKRCARLA that are used within the C2PACMON started task. The string as specified must end with a semicolon, and must be enclosed in quotation marks. This parameter is intended for IBM Software Support personnel to diagnose problems. The maximum length of the string is 63 characters.

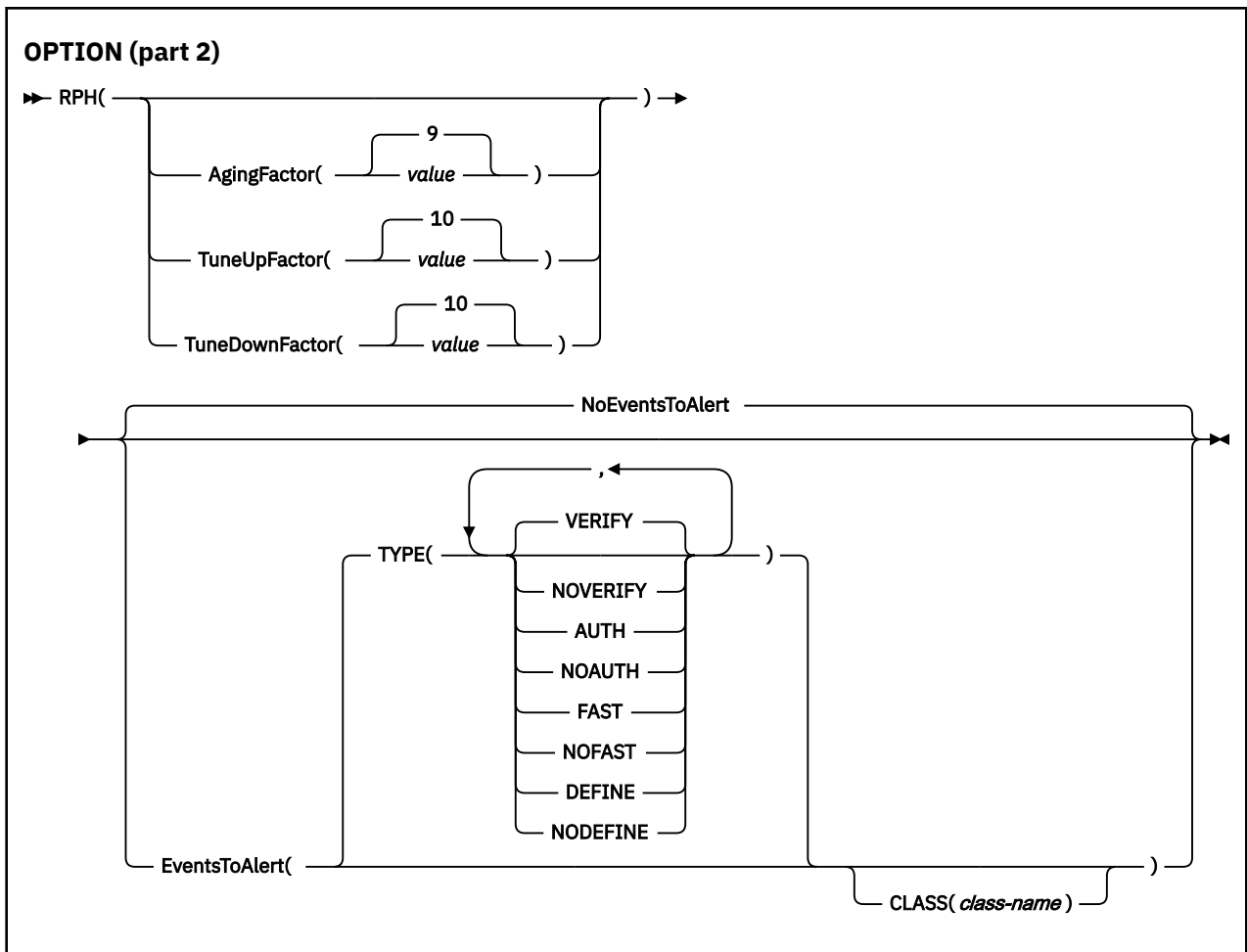
**OPTION command**

The OPTION statement is only valid from the parmlib. The main purpose of the OPTION statement is to specify the number and size of the in-memory data buffers. It can also be used to specify other processing options that are effective for the duration of the entire Access Monitor started task.

The OPTION statement has the following syntax (part 1 and part 2):

### Option (part 1)





The keywords and variables have the following values:

### Bufsize/BufsizeMB

The Bufsize/BufsizeMB keyword can be specified only when the OPTION statement is used during startup or during RESTART processing. It is ignored during CONSOLIDATE processing.

Bufsize/BufsizeMB specifies the size of the in-memory buffers used for storing the Access Monitor records during the *interval* period. Make sure that the buffer is large enough to contain all Access Monitor records collected during that period. If the buffer is too small, the Access Monitor data-capturing routines attempt to switch to an unused buffer. If no unused buffer is available, a buffer overflow message is issued, and the buffer containing the oldest data is used instead.

If you use the Bufsize keyword, specify the required buffer size in kilobytes. If you use the BufsizeMB keyword, specify the size in megabytes. Valid sizes for the buffers are between 1 kilobyte and 1 gigabyte. The size that you specify is rounded up to the nearest megabyte. If you use both keywords in an OPTION statement, the last value specified is used by the program. The buffers are allocated in 64-bit storage and count towards the specified MEMLIMIT of the started task. The use of multiple buffers during periods of high activity can significantly reduce the required buffer size. In general, it is more efficient to specify, for example, 10 buffers of 1 megabyte instead of 2 buffers of 5 megabytes.

### Numbufs

The Numbufs keyword can be specified only when the OPTION statement is used during startup or during RESTART processing. It is ignored during CONSOLIDATE processing.

Numbufs specifies the number of buffers allocated. Valid values for *numbufs* are between 2 and 32. Make sure that the total number of buffers is sufficient to hold all captured Access Monitor records during periods of high activity.

To reduce the *bufsize* required to save all data collected during high-activity periods, specify multiple buffers. If no additional buffers are available, the oldest buffer is used instead, resulting in data loss.



## **RACFEXITMODE**

The RACFEXITMODE keyword can be specified only when the OPTION statement is used during startup or during RESTART processing. It is ignored during CONSOLIDATE processing.

The **RACFEXITMODE** keywords specify whether functional sub exits are called using z/OS dynamic exit services, or are called using a direct branch instruction. Using z/OS dynamic exit services provides additional flexibility and recovery, but uses more resources. Using a direct branch instruction is more efficient, but does not provide additional flexibility or recovery above that which is provided by the called sub exit. Possible choices for the parameters are:

### **FASTSTORE**

This keyword indicates that dedicated modules are used that combine the router function and the functional routine. These modules use Cell Pool storage that is allocated in the user's address space. Cell Pool storage is not automatically returned to the system when it is no longer used. Instead, it is kept for future reuse. The Cell Pool storage that zSecure uses is allocated in subpool 249, which is private storage in the area between 16MB and 2GB. In most environments, use of FASTSTORE is more efficient than DIRECT mode.

The FASTSTORE mode exploits Cell Pool storage. Therefore, it avoids certain resource contentions that might occur in the CSVDYNEX or DIRECT mode. Using FASTSTORE mode also causes less switching to General Purpose processors for tasks that are zIIP-eligible. The FASTSTORE mode is preferred for most environments

### **DIRECT**

This keyword indicates that the RACF exit router module uses a direct branch instruction to call the functional sub exits. This option uses less system resources than the CSVDYNEX mode. The zSecure routines use the STORAGE macro to acquire and release working storage.

The ICHRF02 and ICHRTX00 exits are not supported in DIRECT mode. If you select DIRECT mode, these exits are automatically installed in FASTSTORE mode.

### **CSVDYNEX**

This keyword indicates that the RACF exit router module uses z/OS dynamic exit services for calling the functional sub exits. This option provides additional flexibility and recovery for the called sub exits.

The ICHRF02 and ICHRTX00 exits are not supported in CSVDYNEX mode. If you select CSVDYNEX mode, these exits are automatically installed in FASTSTORE mode.

If the **RACFEXITMODE** keyword is not specified, or if no value is specified, RACF exits are called using the FASTSTORE method.

## **PRECONSOLIDATE, NOPRECONSOLIDATE**

This keyword indicates if Access Monitor uses in-memory pre-consolidation. Pre-consolidation implies that the count of similar events is maintained in the records in the in-memory buffers. The date and time on the first record in the in-memory buffer is used for all pre-consolidated events. If pre-consolidation is not active, every event with its own date and time stamp is individually recorded in the in-memory buffer and passed to the CKRCARLA program for consolidation. Pre-consolidation has the advantage that fewer event records are created and processed. This can lead to a significant reduction in the required in-memory buffer size, used CPU time and virtual storage. For most situations, using pre-consolidation is preferred. The default value is PRECONSOLIDATE.

## **SETSRVCLASS, NOSERSRVCLASS**

Determines whether the started task runs within the SYSSTC service class. The SYSSTC service class is intended for high priority system tasks. The collection of events and saving the collected data should be done in a timely manner. Therefore, a sufficiently high priority setting is important. If you must assign a lower dispatching priority, create an applicable WLM classification rule that assigns the required WLM service class. You must also specify NoSetSRVClass on the OPTION statement to prevent the program from resetting itself to the SYSSTC service class.

If you are running on a single processor system, careful analysis of your workload is required and using NoSetSRVClass might be preferred.

The default value of this option is SETSRVCLASS.

## CollectionDataset

The CollectionDataset keyword can be specified only when the OPTION statement is used during startup or during RESTART processing. It is ignored during CONSOLIDATE processing.

The CollectionDataset keyword specifies whether the daily collection data set is used for data collected during a single SMF interval or during multiple SMF intervals.

### SingleInterval

At the start of the collection process, a daily collection data set is created. At the end of the SMF interval, the captured data is written to the data set, and the data set is closed and renamed. A new data set is created to contain the data from the next SMF interval. This has the advantage that data that is collected during the day is available for analysis. However, it requires more processing resources. Data from the day is split over multiple data sets. If the default SMF interval is used, there are 48 data sets per day. These data sets are consolidated into a single daily consolidation data set. The daily collection data sets are then deleted.

### MultiInterval

At the start of the collection process, a daily collection data set is created. At the end of the SMF interval, the captured data is written to the data set, and the data set is prepared for additional data from the next SMF interval. Data that is collected during the day is available only after consolidation (either through a CONSOLIDATE command, or as the result of the automatic daily consolidation). Data is normally kept in a single data set that contains all SMF intervals of the day. The multiple SMF intervals are consolidated into a daily consolidation data set. The daily collection data set is then deleted.

## SHOWSTATISTICS, NOSHOWSTATISTICS

This keyword determines if the specified CARLa member is included and run at the end of every SMF interval. At the end of the SMF interval, the data that is collected in storage is written out to the daily collection data set. The default member *C2PAMSTS* writes information about the number and type of events that are collected during this interval to the z/OS system log and to the joblog of the STC. The following figure shows example output:

```
C2P8000I Access data for period 31Aug2016 21:40:23 - 31Aug16 21:45:30
C2P8001I Totals                1365
C2P8001I      Auth              1090
C2P8001I      Fast               7
C2P8001I      Define             2
C2P8001I      Verify             266
C2P8002I Output records       100
```

Using the SHOWSTATISTICS keyword and the provided *C2PAMSTS* member requires that you use the provided *C2PAMCOL* member for daily collection. The *C2PAMCOL* member contains the DEFTYPE and DEFINE statements that are used in *C2PAMSTS* to calculate the number of output records. The *C2PAMSTS* member is included from the SC2PSAMP concatenation in the started task procedure. The default value for this keyword is NOSHOWSTATISTICS.

## INCLUDEOWNRESOURCE, NOINCLUDEOWNRESOURCE

Determines whether Access Monitor records are created for Access Monitor events logged when users request access to their own resources. These resources might be, for example, private data sets or jobs running with a user's own userid. Using the INCLUDEOWNRESOURCE option can be helpful to diagnose suspected problems with missing events. However, because this option can significantly increase the amount of data collected, use it only when required. The default for this option is NOINCLUDEOWNRESOURCE.

## CapturePrograms, NoCapturePrograms

Determines whether zSecure Access Monitor captures events for program access through the SAF Router exit ICHRTX00. When activated, the SAF Router exit is used to create ACCESS records for program access. Information is collected before any RACF processing takes place. No information is captured about the result of RACF processing. The default for this option is NOCAPTUREPROGRAMS.

If no sub-parameters are specified, Access Monitor collects information for all programs, independent of their RACF definition. It is also possible to specify that only events for selected programs are collected. The OPTION statement supports only the specification of one criterion, but you can repeat

the OPTION statement when you need multiple selections. An OPTION statement with only the (NO)CAPTUREPROGRAMS keyword, without any subkeyword, overrides any previous program capture specification. You can specify the following program selection criteria:

**DefProgram**

The program is matched by a profile in the PROGRAM class. The matching PROGRAM profile can end with an asterisk. The program can be loaded from any controlled or non-controlled library. This selection can be used to locate (potentially unprotected) copies of controlled programs.

**CtlProgram**

The program is matched by a profile in the PROGRAM class, and was loaded from the corresponding library and optional volume.

**SpecProgram**

The program is matched by a non-backstop entry in the PROGRAM class. The matching PROGRAM profile can end with an asterisk. The program must be loaded from the corresponding library and optional volume.

**CtlLibrary**

The program was loaded from a controlled library. The program does not need to match any defined PROGRAM profile. Only the library name and the optional volume are inspected. This selection can be used to detect other programs that are loaded from controlled libraries. This can be relevant when setting up program access to data sets (PADS).

**CaptureLocalRACLIST, NoCaptureLocalRACLIST**

Determines whether zSecure Access Monitor captures events for FASTAUTH access events through the RACF post-processing exit ICHRF02. When activated, this FASTAUTH exit is used to capture information about access events against LOCAL RACLISTed resources. The ICHRF04 exit is used for GLOBAL RACLISTed resource classes, and for other situations where RACF calls the ICHRF04 exit. The default for this option is NoCaptureLocalRACLIST.

**CaptureUSSEvents, NoCaptureUSSEvents**

Determines whether Access Monitor captures events for UNIX file and directory access events. When activated, the UNIX dynamic syscall exits are used to create ACCESS records for selected events. Specifying this option is effective only if the steps described in [“Required preparation for collecting UNIX event data”](#) on page 68 are followed. The default for this option is NOCAPTUREUSSEVENTS.

Before starting to collect UNIX event information, ensure that your ACCESS files can accommodate the additional space requirement for these UNIX events. For information about estimating the required space, see [“Estimating the required space for UNIX event data”](#) on page 72.

**RPH**

Specifies the value for the RPH (Real (Resolved) Pathname) aging factor (AgingFactor), LRU stack increase factor (TuneUpFactor), or LRU stack decrease factor (TuneDownFactor). This option is intended to be used only at the request of IBM Software Support personnel.

**EventsToAlert, NoEventsToAlert**

Determines if information for certain events is forwarded to zSecure Alert. The default is not to forward any event information. If you specify only the EventsToAlert keyword without any detail keywords or parameters, VERIFY events are forwarded to zSecure Alert. The current release of Access Monitor only supports TYPE(VERIFY) as detail specification. Detail keywords and parameters for other events and selections are reserved in the syntax. These keywords and parameters are currently not supported and using them results in an error message.

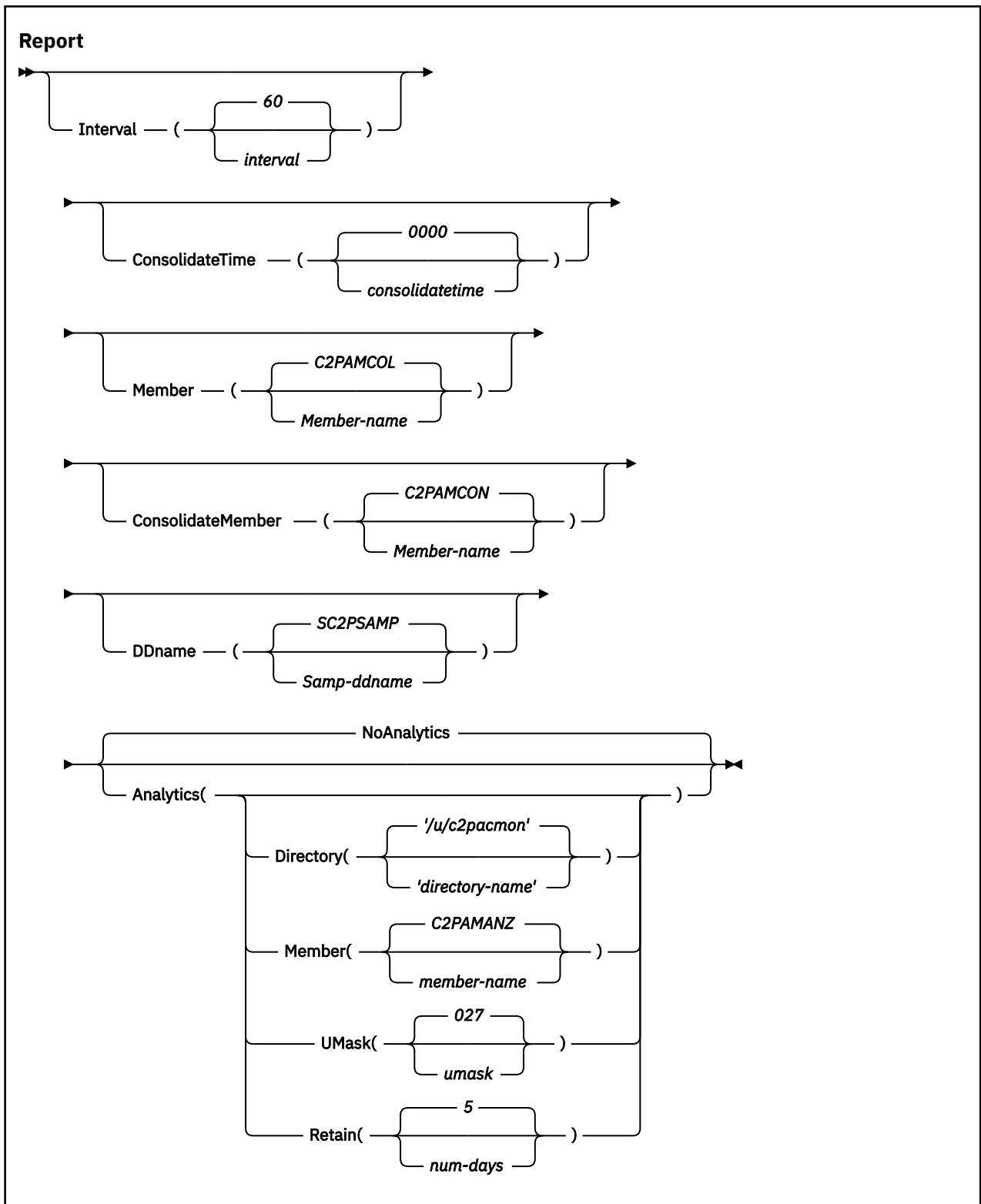
To activate forwarding of supported captured events, specify either `Option EventsToAlert` or `Option EventsToAlert(Type(Verify))`. To deactivate forwarding of supported captured events specify either `Option NoEventsToAlert` or `Option EventsToAlert(Type(NoVerify))`.

If you activate event forwarding to zSecure Alert, the Access Monitor started task performs additional processing. At the beginning of every interval as defined through the REPORT statement, the started task verifies that events can be forwarded to zSecure Alert. If the Alert started task is not active, an error message is issued. Only activate event forwarding if you are using the event data to generate alerts, for example through predefined alert 1122.

Forwarding events to zSecure Alert has no effect on recording events in the Access Monitor ACCESS files.

## **REPORT command**

This command controls the timing of the buffer management process, the timing of the daily consolidation process, and the source for the CARLa statements used for the data collection and consolidation processes. The effects of the REPORT command can be delayed due to the cyclic nature of various tasks in the Access Monitor function. For example, a modified value for the *Interval* is only used after the current interval expires. The REPORT command has the following syntax.



The keywords and variables have the following values:

**Interval**

Specifies the interval at which the Access Monitor started task transfers the collected records to the CKRCARLA task for statistical analysis. The value *interval* specifies the time interval in seconds. A time interval can be in the range of 10 and 3600 seconds. The default value is 60 seconds.

**ConsolidateTime**

Specifies the time of day at which the Access Monitor will start the daily data consolidation process. At the specified time, the current daily record collection data set is closed. The daily records of the previous day are consolidated into the data set specified using the C2PAMCNT file.

The preferred, default value for the *consolidatetime* is 0000 (midnight).

**Member**

Specifies the *member* name in the partitioned data set that is used for the daily data collection. It contains the CARLa statements that summarize the Access Monitor records during the SMF interval period. At the end of the SMF interval, the collected records are written to disk. In normal situations, you do not need to specify this keyword. The program uses the default member name, C2PAMCOL.

**ConsolidateMember**

Specifies the *member* name in the partitioned data set used by the consolidation process. The consolidation process summarizes all individual records for the SMF interval periods. This process significantly reduces the amount of space required to store the daily data. The default value for *ConsolidateMember* is C2PAMCON. Normally, this default value is overwritten in the C2PAMP member to the value C2PAMCMP. This member includes additional CARLa statements to reduce data set size. It also includes a reference to the installation specific mapping rules as defined in member C2PAMMAP.

**DDName**

Specifies the JCL DD-name pointing to the partitioned data set containing the CARLa statements that run the Access Monitor daily collection and consolidation process. *DDName* must contain at least the members indicated by *member* and *consolidateMember*.

**Analytics**

Specifies that a daily file with statistical information is created. This daily file is intended for further processing by an analytics program like IBM Z Operations Analytics. Specifying this keyword on the REPORT statements activates this support using default parameters. You can override these defaults by specifying one or more of the following keywords. You can repeat the REPORT ANALYTICS statement as often as needed to specify all four keywords.

**Directory**

Specifies the z/OS UNIX directory where the prepared files are stored. The C2PACMON started task user ID must have sufficient authority to create and remove files in this directory. The directory that you specify must be entered between quotes. It must start with a slash ("/") and not end with a slash. If you are sharing the C2PACMON configuration file between systems, you must ensure that every C2PACMON started task uses a different directory. You can use system symbols in the directory name, for example &SYSCZONE. System symbols must always be specified in uppercase.

**Note:** The configuration statements do not support continuation lines. You can abbreviate keywords using the normal TSO conventions (for example abbreviate DIRECTORY to Dir).

If you do not specify a value, the default directory name is **/u/c2pacmon**.

**Member**

Specifies the CARLa member that contains the statements to process the access records in a format that is suitable for use by an analytics program. The default member C2PAMANZ is supplied in the SCKRCARL data set. It imbeds member C2PAMANC that must be configured before use. Member C2PAMANC is contained in the data set that is indicated by symbol C2PACPRM in your CKRPARM member.

**UMask**

Specifies the UMASK that is in effect when creating new files in the specified directory. The default value 027 resets write access for the group and resets all access for other users.

**Retain**

Specifies the number of days that the analytics files are retained before being deleted automatically. The specified value must be in the range of 2 to 99 days. The default value (5) allows several days for the analytics product to pick up the files. Expiration of files is automatic and based on the date and time stamp in the file name.

---

## Chapter 11. Setup of the z/OS Compliance Integration Manager

The z/OS Compliance Integration Manager started task is the component of the Z Security and Compliance Center that runs on z/OS and provides a zSecure started task that is required to create SMF 1154 records for certain z/OS subsystems. Other subsystems write their own SMF 1154 records.

See the following sections for information about setting up and operating Z Security and Compliance Center.

- [“Installation of the z/OS Compliance Integration Manager” on page 89](#)
- [“Configuration of the z/OS Compliance Integration Manager” on page 89](#)
- [“Operation of the z/OS Compliance Integration Manager” on page 92](#)
- [“z/OS Compliance Integration Manager command reference” on page 94](#)

---

### Installation of the z/OS Compliance Integration Manager

Use the guidelines in this topic to ensure that your z/OS Compliance Integration Manager installation is correctly set up for use.

Before you can set up the z/OS Compliance Integration Manager, you must install zSecure; see [Chapter 4, “Installation of the software,” on page 9](#). During the installation process, make sure that the following items are configured:

- The zSecure base code and the z/OS Compliance Integration Manager must be installed. This component is installed by enabling the installation parameter `zOSComplianceIntegrationManager` in member `CKRZUPDI` before running the fast or formal installation jobs. See [“Customization of the installation parameters” on page 11](#).
- The ZSCC license must be enabled. See [“Enablement of license features” on page 18](#).
- The `SCKRLOAD` library must be APF-authorized.
- You must have a zSecure configuration data set that enables the use of the z/OS Compliance Integration Manager. Alternatively, you can set the `&CPREFIX` variable in the `CKCS1154` proclib member, and remove the `INCLUDE` for member `&CONFIG`. The `&CKCSPARM` variable must also be set to the name of the data set that you use for the parameter member.
- You must create a z/OS Compliance Integration Manager parameter member that specifies the options for the started task.
- You must run the z/OS Compliance Integration Manager on all systems where the reported services are active.

---

### Configuration of the z/OS Compliance Integration Manager

Perform the configuration tasks to start the z/OS Compliance Integration Manager and create SMF 1154 for the specified subsystems.

- [“Preparation of the JCL” on page 90](#)
- [“Definition of security resources and permissions” on page 90](#)
- [“Customization of the parameters” on page 91](#)
- [“Copy ZSCC configuration files for z Common Data Provider \(zCDP\)” on page 91](#)

## Preparation of the JCL

The z/OS Compliance Integration Manager must run as a started task. Therefore, you must copy the CKCS1154 procedure from the SCKRPROC data set to a started task procedure library. Perform the following steps.

1. Copy the CKCS1154 procedure from the SCKRPROC data set to a started task procedure library. Optionally, you can decide to use a different member name, provided that you update the security resources that you must then create accordingly.
2. In your copy of the CKCS1154 procedure, specify the member name for the zSecure configuration that you intend to use. The default is C2R\$PARM. Alternatively, you can remove the INCLUDE for the CONFIG member, and replace it with a direct SET of the &CPREFIX variable to the prefix of the installation data sets. For example:

```
// SET CPREFIX=IBM.ZSCC
```

3. Specify the value for the &PPARM variable. This variable points to the member name for the parameters of the z/OS Compliance Integration Manager started task.
4. Verify that the variable &CKCSPARM points to the data set that contains the z/OS Compliance Integration Manager parameter member.

## Definition of security resources and permissions

This section describes the security resources and permissions that the z/OS Compliance Integration Manager requires, as well as instructions to set them up. Unless your the security databases are shared, you must establish these authorizations for all systems where the z/OS Compliance Integration Manager started task runs.

The following security resources and permissions are required:

- The user ID and group ID or logonid that you selected to run the CKCS1154 address space must be available. The user ID must have an OMVS segment for RACF, or the logonid must have an OMVS record for ACF2. It is recommended that the started task has a unique user id that is PROTECTED.
- The started task name and security resources to assign the required user ID and group ID or logonid to the started task.
- Started task output protection.
- The user ID for CKCS1154 must have READ access to the XFACILIT resources CKF . AUDIT and CKR . CKRCARLA . APF.
- If MQ regions are active, the started task ID must have access to the following MQ resources:

Table 6. Resource access required for collection for MQ region

Resource class	Resource name	Required access
MQCONN	ssid.BATCH	READ
MQCMDS	ssid.DISPLAY.AUTHINFO	READ
MQCMDS	ssid.DISPLAY.CHANNEL	READ
MQCMDS	ssid.DISPLAY.CHINIT	READ
MQCMDS	ssid.DISPLAY.CHLAUTH	READ
MQCMDS	ssid.DISPLAY.CONNPCF	READ
MQCMDS	ssid.DISPLAY.NAMELIST	READ
MQCMDS	ssid.DISPLAY.PROCESS	READ
MQCMDS	ssid.DISPLAY.QMGR	READ



Table 6. Resource access required for collection for MQ region (continued)

Resource class	Resource name	Required access
MQCMDS	ssid.DISPLAY.QUEUE	READ
MQCMDS	ssid.DISPLAY.SECURITY	READ
MQCMDS	ssid.DISPLAY.SYSTEM	READ
MQCMDS	ssid.DISPLAY.TOPIC	READ
MQQUEUE	ssid.CKFCOLL.COMMAND.REPLY.*	UPDATE
MQQUEUE	ssid.SYSTEM.COMMAND.INPUT	UPDATE
MQQUEUE	ssid.SYSTEM.COMMAND.REPLY.MODEL	UPDATE

Because the CKFCOLL.COMMAND.REPLY queue is a unique dynamic queue, the last qualifier is replaced by a character string that IBM MQ generates. A generic profile must be used to protect this queue.

### Customization of the parameters

The z/OS Compliance Integration Manager requires a parameter data set for the started task. This parameter data set is included using the PPARM JCL parameter in the started task procedure as described in “Preparation of the JCL” on page 90. The parameter data set must always be present. If you do not want to change any of the default values for the parameters, specify a parameter data set with at least one line, which can be a comment line. A sample parameter data set, showing default values, can be found in member CKCSP in the SCKRSAMP data set.

The data collection and processing that is done by the z/OS Compliance Integration Manager must be done in a timely manner. To ensure that sufficient CPU resources are available, the CKCS1154 started task resets itself to the WLM service class SYSSTC. If you must assign a lower dispatching priority, create an applicable WLM classification rule that assigns the required WLM service class. You must also specify NoSetSRVClass on the OPTION statement to prevent the program from resetting itself to the SYSSTC service class. If you are running on a single processor system, careful analysis of your workload is required and using NoSetSRVClass might be preferred.

As part of customization, you can specify the member name and ddname for the required CARLa scripts. You can also specify the stagger time (see `StaggerTime` for the OPTION command). For more information about configuration parameters, see “z/OS Compliance Integration Manager configuration commands” on page 95.

### Copy ZSCC configuration files for z Common Data Provider (zCDP)

Z Compliance Integration Manager works with z Common Data Provider (zCDP) to send the data to the IBM Z Security and Compliance Center.

z Common Data Provider needs configuration files for ZSCC. They are provided in SCKCJSON and must be copied before configuration of zCDP by means of the CKCZJSON job in SCKRSAMP. For first time setup, all 3 JSON files are needed. Then, when the `zsc . streams . json` is updated and must be replaced, use the same job but only use the `COPYSTRE` step to prevent overwriting the `zsc . enable . json` and `zsc . protocols . json` files.

Edit the job to provide the zCDP Configuration Tool Working Directory; see “Setting up a working directory for the Configuration Tool” in [Deploying the Z Common Data Provider documentation](#).

## Operation of the z/OS Compliance Integration Manager

Perform the configuration tasks to start the z/OS Compliance Integration Manager and create SMF 1154 for the specified subsystems.

You can manage the z/OS Compliance Integration Manager function by issuing commands from the operator console at startup and while the task is running. You can also control the operating environment for the z/OS Compliance Integration Manager function by providing input parameters in the parmlib DD-statement in the startup procedure. For instructions, see the following sections:

- [“Start the z/OS Compliance Integration Manager started task” on page 92](#)
- [“z/OS Compliance Integration Manager START parameters” on page 92](#)
- [“MODIFY command to monitor or modify the z/OS Compliance Integration Manager” on page 93](#)
- [“Stop the z/OS Compliance Integration Manager started task” on page 93](#)
- [“Configuration of the z/OS Compliance Integration Manager using parmlib” on page 93](#)

Some commands are primarily intended to be issued as part of the PARMLIB file. These commands and their keywords and parameters are described in “Configuration commands” on page 79.

### Start the z/OS Compliance Integration Manager started task

To start the z/OS Compliance Integration Manager, issue a START command from the operator console as shown in the following example:

```
S CKCS1154
```

In a production environment, use Automated Operation software or PARMLIB member COMMNDxx to automatically start the z/OS Compliance Integration Manager task soon after each IPL. This command runs the procedure from the applicable system proclib. When entering the START command, you can also specify startup parameters to run diagnostic tests or force program initialization. These parameters are described in [“z/OS Compliance Integration Manager START parameters” on page 92](#).

### z/OS Compliance Integration Manager START parameters

For normal execution of the z/OS Compliance Integration Manager, you do not need to specify any startup parameters. By default, the z/OS Compliance Integration Manager detects if it is already active and issues an appropriate error message before ending. In some error situations, the z/OS Compliance Integration Manager started task fails to initialize. In these situations, you might need to specify one of the optional START parameters. The following example shows a START command with the DEBUG parameter specified:

```
S CKCS1154, , , DEBUG
```

#### DEBUG

Issues diagnostic messages during the first part of the initialization. These diagnostic messages can also be used to determine possible problems in processing the standard PARMLIB parameters. This setting is in effect until a subsequent DEBUG command is issued, either from the operator console or using PARMLIB.

#### FORCE

Forces initialization to continue, regardless of a previous execution. Use the FORCE option only when the z/OS Compliance Integration Manager started task cannot be started using other methods, and IPLing the system is undesirable. During normal operation, it is not necessary to use the FORCE command to start the system. If you must use this parameter, create a problem report so that the issue can be investigated.

#### DEBUG-FORCE

Activates both the DEBUG and FORCE options at startup.

The started task procedure CKCS1154 as it is shipped also provides the PPARM parameter to specify the main PARMLIB member that initializes the z/OS Compliance Integration Manager started task. This parameter can be used to override the value that is specified in the procedure. The default value that is specified in the procedure for the PPARM parameter is CKCSP.

## **MODIFY command to monitor or modify the z/OS Compliance Integration Manager**

When the z/OS Compliance Integration Manager started task is active, the console operator can monitor or modify operations by using the MODIFY console command (or use the F command as an alias). The following example shows a modify command that displays the current status and options:

```
MODIFY CKCS1154,DISPLAY
```

or

```
F CKCS1154,DISPLAY
```

Be sure that the text after the comma is one of the supported operator commands for the started task. For information about these commands, see [“z/OS Compliance Integration Manager operator commands”](#) on page 94.

## **Stop the z/OS Compliance Integration Manager started task**

To stop the z/OS Compliance Integration Manager started task, run the STOP command from the console (or use the P command as an alias). For example:

```
STOP CKCS1154
```

or

```
P CKCS1154
```

The STOP command can also be issued as the parameter on the MODIFY command:

```
F CKCS1154,STOP
```

For more detailed information about the z/OS Compliance Integration Manager operator commands, see [“z/OS Compliance Integration Manager configuration commands”](#) on page 95.

## **Configuration of the z/OS Compliance Integration Manager using parmlib**

By default, the DD-statement refers to the CKCSP member in the CKCSPARM data set. The following commands are examples of commands that can be specified in parmlib:

### **DEBUG**

To diagnose problems.

### **OPTION**

For managing the stagger time; that is, a delay to stagger the data collection process for multiple systems in the same sysplex. See [StaggerTime](#) for the OPTION command.

### **REPORT**

For specifying the CARLa statement members.

The input parameters can be specified in the form of commands with keywords. Use TSO conventions when specifying these commands. For details about the CKCSP parameter file, see [“Customization of the parameters”](#) on page 91.

## z/OS Compliance Integration Manager command reference

The z/OS Compliance Integration Manager command reference lists the operator commands and configuration commands.

See the following sections:

- [“z/OS Compliance Integration Manager operator commands” on page 94](#)
- [“z/OS Compliance Integration Manager configuration commands” on page 95](#)

### z/OS Compliance Integration Manager operator commands

When entered as an operator command, these commands do not require additional keywords. You can also abbreviate a command to the first four characters; for example, type DISP for the DISPLAY command.

#### CRSH command

Causes an immediate abend of the CKCS1154 started task, without any recovery. It is intended for IBM internal testing only.

#### DEBUG command

Controls the diagnostic and monitoring messages that the program can generate. The DEBUG command is effective immediately. For a complete description of the keywords for the DEBUG command, see [“DEBUG command” on page 95](#).

#### DIAGNOSE command

The DIAGNOSE command is intended for IBM support personnel to diagnose problems. It allows simulation of an ENF86 event. This triggers the creation of the SMF 1154 records, similar to those that are created for a real ENF86 event. The DIAGNOSE command has the following syntax:



DIAGNOSE has one keyword:

#### ENF86

Causes the z/OS Compliance Integration Manager started task to start processing as if an ENF86 event was received. The simulated event specifies the value CKCS1154SIMEVENT as request identifier and specifies the current system's SMFID.

#### DISPLAY command

Displays the current status and option settings for the z/OS Compliance Integration Manager. The display includes the current settings and the status of the subtasks. The DISPLAY command does not support any additional keywords.

#### REPORT command

Sets the values for the keywords that control the processing of the captured data. The new values are used at the next time that they are needed. For example, a new value for the stagger time parameter is used only for a subsequent ENF86 event. For a complete description of the keywords for the REPORT command, see [“REPORT command” on page 96](#). For stagger time, see [StaggerTime](#) for the OPTION command.

#### RESTART command

Shuts down the z/OS Compliance Integration Manager, and then immediately reinitializes the task. The address space in which the z/OS Compliance Integration Manager is running is not stopped, and no additional console operator commands are needed to reactivate the z/OS Compliance Integration Manager.

#### SIPL command

Issue this command only at the request of IBM Software Support personnel, or when explicitly required during release migrations. When the command is run, all in-memory data structures are freed, and the task is terminated. When you upgrade z/OS Compliance Integration Manager, the

installation instructions might require you to shut down the previous instance using this SIPL command. The SIPL command does not support any additional keywords.

### STOP command

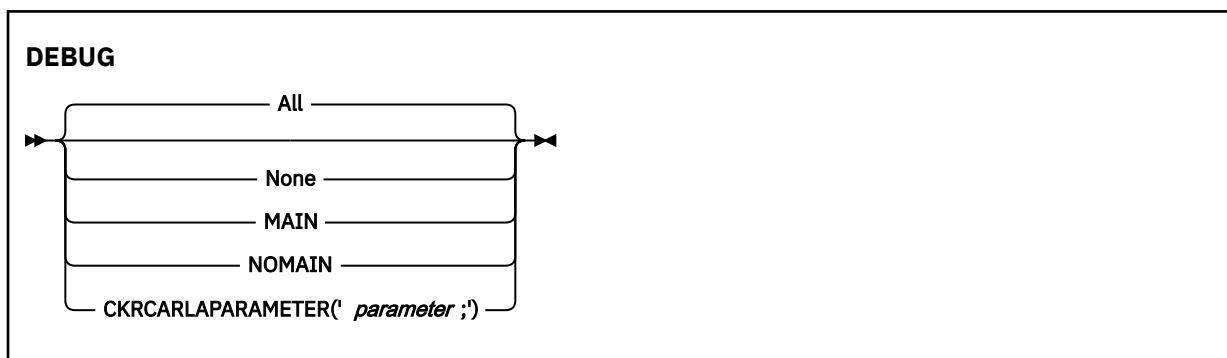
Shuts down the z/OS Compliance Integration Manager started task. After the task ends, some memory remains reserved for reuse during a subsequent restart of the z/OS Compliance Integration Manager. The effect of the STOP modify command is identical to that of the MVS STOP command. The STOP command does not support any additional keywords.

## z/OS Compliance Integration Manager configuration commands

The configuration commands are usually only specified in the parmlib member. They can also be specified through the operator modify command. After completing the processing for an ENF86 event, the contents of the parmlib member are re-processed. At that time, possible options that have been specified through operator commands are reset to those that are present in the parameter member. Normally, only the OPTION and REPORT commands are required. Use the DEBUG command if you want to obtain additional diagnostic information.

### DEBUG command

The DEBUG command has the following syntax.



**Note:** You can specify only one option at a time. To enable multiple debug options, issue the DEBUG command multiple times. The DEBUG command can be run from the operator console or from the parmlib.

The keywords and variables have the following values:

#### All

Write all diagnostic messages to the console. ALL is the default setting for displaying messages if you do not specify a parameter on the DEBUG commands. Most of these messages are intended to assist during problem determination, and are not intended for routine customer use.

#### None

Deactivates the creation of all diagnostic messages.

#### MAIN

Write diagnostic messages that are related to mainline processing to the console. This includes responding to operator commands, initialization and management of all subtasks, and major buffer management functions.

#### NOMAIN

Do not write diagnostic messages that are related to mainline processing to the console. This setting suppresses the following types of messages:

- Responses to operator commands
- Initialization and management of all subtasks

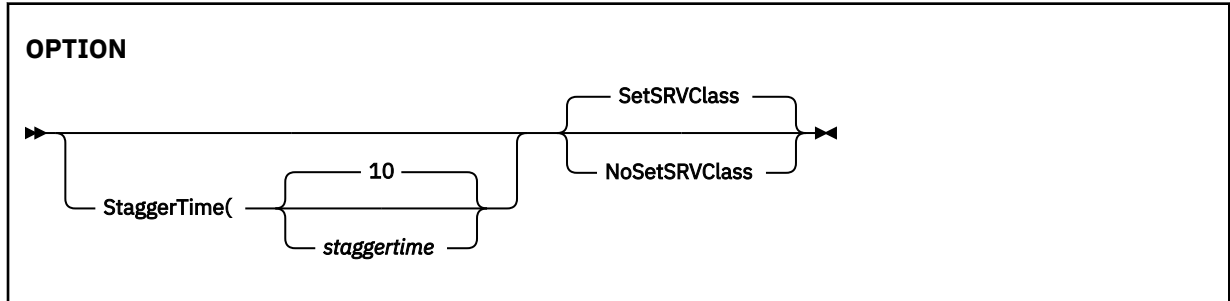
#### CKRCARLAPARAMETER

Specifies a string that is to be passed to all instances of CKRCARLA that are used within the CKCS1154 started task. The string that is specified must end with a semicolon, and must be

enclosed in quotation marks. This parameter is intended for IBM Software Support personnel to diagnose problems. The maximum length of the string is 63 characters.

### OPTION command

The main purpose of the OPTION statement is to specify the stagger time. The OPTION statement has the following syntax:



OPTION has one keyword:

#### StaggerTime

Specifies the time interval at which multiple z/OS Compliance Integration Manager tasks in the sysplex start processing an ENF86 event. The ENF86 event specifies a list of systems that should respond to the event. The index of the current system in the list determines the delay before processing starts. The index is multiplied by the specified stagger time to determine the delay. When specifying a zero (0) stagger time, all systems in the sysplex start to collect data and generate SMF 1154 records at the same time. This might cause resource contention. The specified stagger time must be between 0 (no delay) and 300 seconds. If the specified value exceeds 300 seconds, the maximum value of 300 seconds is used. The default value is 10 seconds.

#### SETSRVCLASS, NOSETSRVCLASS

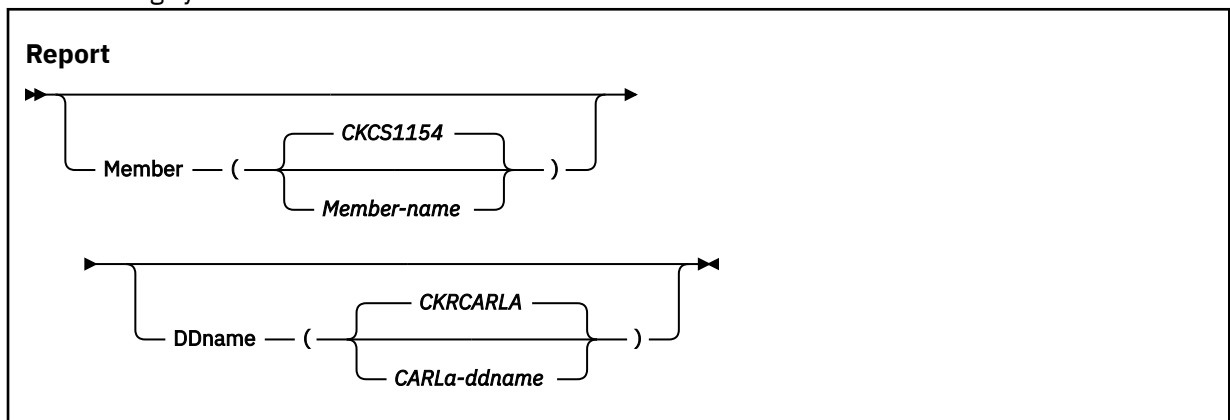
Determines whether the started task runs within the SYSSTC service class. The SYSSTC service class is intended for high priority system tasks. The collection of events and saving the collected data should be done in a timely manner. Therefore, a sufficiently high priority setting is important. If you must assign a lower dispatching priority, create an applicable WLM classification rule that assigns the required WLM service class. You must also specify NoSetSRVClass on the OPTION statement to prevent the program from resetting itself to the SYSSTC service class.

If you are running on a single processor system, careful analysis of your workload is required and using NoSetSRVClass might be preferred.

The default value of this option is SETSRVCLASS.

### REPORT command

This command controls the source for the CARLa statements that are used for the data collection and SMF record generation processes. The values that are specified on the REPORT command are effective at the next time that these parameters are used. For example, a modified value for the member is used the next time that the CARLa member is included only. The REPORT command has the following syntax.



The keywords and variables have the following values:

**Member**

Specifies the member name in the partitioned data set that is used for the generation of the SMF 1154 records. In normal situations, you do not need to specify this keyword. The program uses the default member name, CKCS1154.

**DDname**

Specifies the JCL DD-name that points to the partitioned data set that contains the CARLa statements that run the SMF 1154 record generation process. DDName must contain at least the member that *member* indicates. The program uses the default DD-name, CKRCARLA.





## Chapter 12. Setup of RACF-Offline

The RACF-Offline function is a component of zSecure Admin that allows you to execute and test RACF commands on a RACF database that is not active in the system. Using this program, you can test changes to RACF definitions without impacting any other software running on the system and without using a dedicated test system. For more information about RACF-Offline, see the *IBM Security zSecure Admin and Audit for RACF: User Reference Manual*.

Instructions for installing RACF-Offline are provided in the *Program Directory: IBM Security zSecure Admin RACF-Offline*. After you have installed the product, you must perform additional post-installation activities to activate the function. For information, see [“Installing and activating RACF-Offline”](#) on page 99.

### Default installation data set names

RACF-Offline is installed in two system libraries:

- A load library, SB8RLNK, which must be APF authorized
- A JCL sample library, SB8RSAMP

See *Program Directory: IBM Security zSecure Admin RACF-Offline* for more information.

### Installing and activating RACF-Offline

#### About this task

To perform each task, see the procedure shown in the table.

Step	Description	Job name	Status
1	Install RACF-Offline. See the <i>Program Directory: IBM Security zSecure Admin RACF-Offline</i> .		
2	<a href="#">Build the default Options module (B8ROPT)</a>	B8RJOPT	
3	<a href="#">Update Parmlib member for APF library</a>	B8RSPROG	
4	<a href="#">Update Parmlib member for TSO Authorized Commands (Optional)</a>	B8RSIKJ	
5	<a href="#">Verify Parmlib member for SMF Exits</a>		
6	<a href="#">Define Minimal RACF authorizations for testing</a>	B8RJRDF	
7	<a href="#">Test RACF Offline</a>	B8RJTST	
8	<a href="#">“Check for RACF-Offline enablement” on page 103</a>		

### Building the default options module (B8ROPT)

#### About this task

The B8ROPT options module specifies the RACF general resource class to be used for authorization verifications performed by the product. The default resource class used is the XFACILIT class. The options module can also contain additional RACF-Offline commands that specify:

- The default RACF databases.
- The default LOG data sets.

- The SMF processing options.

Specify these optional commands between the resource class specification, CLASS, and the END command.

The following example job, B8RJOPT, can be modified to specify this information. This job consists of assembly and linkedit JCL with inline source for B8ROPT. The job is available in the B8ROPT member in the data set where RACF-Offline was installed.

```

B8ROPT  CSECT
B8ROPT  AMODE 31
B8ROPT  RMODE ANY
CLASS   DC CL80'XFACILIT'           RACF RESOURCE CLASS
RACFDB  DC CL80'RACFDB '<your-offline-racfdb>' ' ' DSNAME
SMF     DC CL80'SMF ID($B8R)'       SMF OPTIONS
END     DC CL80'END'                 MANDATORY END
END

```

Before running this job, adapt the inline source with option settings applicable to your environment. If you do **not** run this job, RACF-Offline uses the default resource class, and does **not** use a default RACF database.

## Procedure

Follow this two-step process to build the default options module for your environment:

1. Edit the B8RJOPT member to set options for your system environment.
  - a) In the CLASS statement, specify the resource class name to be used for authorization verifications done in the product. The resource class used in the sample job is XFACILIT.
  - b) For the RACFDB statement, specify the data set name for the default RACF database to be used when the user does not select any other RACF database. Specify the name within two single quotation marks (') as shown in the following example:

```

RACFDB  DC CL80'RACFDB '<BCSC.ROFFLINE.TESTDB1>' ' ' SEQ(1)' Dsname
        DC CL80'RACFDB '<BCSC.ROFFLINE.TESTDB2>' ' ' SEQ(2)' Dsname

```

- c) For the SMF statement, specify the SMF processing required for your environment. The complete syntax of the SMF statement is described in the RACF-Offline chapter of the *IBM Security zSecure Admin and Audit for RACF: User Reference Manual*. A possible SMF processing option is to change the SMFID; for example:

```

SMF     DC CL80'SMF ID($B8R)'       SMF options

```

- d) Do **not** modify the END statement.

If your RACF database is physically split in multiple databases, after editing, your B8ROPT module might look like the following example:

```

B8ROPT  CSECT
B8ROPT  AMODE 31
B8ROPT  RMODE ANY
CLASS   DC CL80'$B8R'               Resource class
RACFDB  DC CL80'RACFDB '<BCSC.ROFFLINE.TESTDB1>' ' ' SEQ(1)' Dsname
        DC CL80'RACFDB '<BCSC.ROFFLINE.TESTDB2>' ' ' SEQ(2)' Dsname
SMF     DC CL80'SMF ID($B8R)'       SMF options
END     DC CL80'END'                 Mandatory
END

```

### Note:

- i) RACFDB is a RACF-Offline control command. Specifying the dsname in quotation marks is useful for standard parsing processing. Because this name is part of a literal string in the assembler source, use two single quotation marks (') around this value in the B8ROPT source.
- ii) The SMF option is a RACF-Offline control command. Without this option, the SMF records for changes in the System RACF database would be indistinguishable from the records created for changes in the offline database. Several processing options are supported. For more information

about the SMF options, see the RACF-Offline chapter of the *IBM Security zSecure Admin and Audit for RACF: User Reference Manual*

2. Submit the B8RJOPT job.

After you have adapted the settings in B8RJOPT for your environment, submit the job to apply the changes to the B8ROPT options module.

## Updating parmlib members for the APF library

### About this task

To activate the RACF-Offline function, make the program library APF-authorized and place it in the link list. Both operations can be performed dynamically using either operator commands or prepared members in PARMLIB. The installation library provides a sample PARMLIB member B8RSPROG that can be adapted for your environment.

### Procedure

1. To add the library to the active APF list:

- a) Add or modify a PROG $xxx$  member for the program library to the PARMLIB member.

```
APF ADD DSNAME(SYS1.SB8RLNK) SMS
```

- b) Issue the T PROG= $xxx$  operator command to activate the new PARMLIB member.

2. To run the B8RACF program, use either a STEPLIB statement or add the program library to the link list.

For initial testing, or incidental usage of B8RACF, use a STEPLIB.

3. To add the library to the active link list:

- a) Add a PROG $xxx$  member such as the following one to PARMLIB.

```
LNKLST DEFINE NAME(LNKLSTB8) COPYFROM(CURRENT)
LNKLST ADD NAME(LNKLSTB8) DSN(SYS1.SB8RLNK)
LNKLST ACTIVATE NAME(LNKLSTB8)
LNKLST UPDATE, JOB=*
```

- b) Issue the T PROG= $xxx$  operator command to activate the new PARMLIB member.

## Updating parmlib members for TSO Authorized Commands (Optional)

### About this task

RACF-Offline can be run as a TSO command and as the main program in a batch job. You can skip this step in the activation process if you only want to run RACF-Offline as a job-step program in the batch environment.

### Procedure

To run RACF-Offline as a TSO command, add several commands to the list of APF authorized commands that can be executed from TSO.

1. In parmlib member IKJTSO $xxx$ , add these commands to the AUTHCMD list.

```
B8RACF /* zSecure Admin RACF-Offline */ +
B8RVARY /* zSecure Admin RACF-Offline */ +
B8REPLAY /* zSecure Admin RACF-Offline */ +
B8RACFLG /* zSecure Admin RACF-Offline */ +
```

If you decide to insert these lines as the last lines of the AUTHCMD statement:

- Verify that the previous line is properly continued, using a plus sign (+).

- Verify that the line is properly terminated, using a right parenthesis ( )).

A sample with instructions is included in member B8RSIKJ.

2. Run the TSO PARMLIB UPDATE (xxx) command to activate this PARMLIB member.

## Verifying parmlib member for SMF exits

The SMF records created by the regular RACF commands issued against the system RACF database are indistinguishable from the records issued by RACF-Offline to an offline RACF database. To enable identification, the SMF records created under RACF-Offline can be modified using the dynamic SMF exits IEFU83, IEFU84, and IEFU85. These exits must be enabled on the system and specified for the entire system or for the relevant subsystems such as TSO, JES2, and JES3. The following example shows an SMFPRMxxx member that has been configured to set up the dynamic SMF exits.

```
ACTIVE                               /* ACTIVE SMF RECORDING           */
DSNAME(SYS1.MAN1,                    /*                                */
        SYS1.MAN2,                    /*                                */
        SYS1.MAN3)                    /*                                */
NOPROMPT                             /* DO NOT PROMPT OPERATOR         */
REC(PERM)                             /* TYPE 17 PERM RECORDS ONLY     */
MAXDORM(3000)                          /* WRITE IDLE BUFFER AFTER 30 MIN */
STATUS(010000)                         /* WRITE SMF STATS AFTER 1 HOUR  */
JWT(0100)                              /* 522 AFTER 1 HOUR              */
SID(IDFX)                              /*                                */
LISTDSN                               /* LIST DATA SET STATUS AT IPL   */
SYS(NOTYPE(40,42,99),EXITS(IEFU83,IEFU84,IEFU85,IEFACTRT,
                           IEFUSI,IEFUJI,IEFU29),NOINTERVAL,NODETAIL)
SUBSYS(STC,NOTYPE(40,42,99),EXITS(IEFU29,IEFU83,IEFU84,IEFU85))
```

If the SMF exits are not enabled, SMF records created for commands updating the Offline RACF database will seem to modify the System RACF database. That is, the SMF ID on records modified in the Offline RACF database will be the same as it is for records modified in the RACF database.

## RACF authorizations for minimal testing

At the testing stage, define a top generic profile with a UACC(NONE) and the userid for the test job with UPDATE on the access list. If you used the XFACILIT resource class as the resource class for the RACF-Offline profiles, you can use these commands:

```
SETR GENERIC(XFACILIT)
SETR CLASSACT(XFACILIT)
RDEF XFACILIT B8R.** UACC(NONE) OWNER(owner-of-your-choice)
PE B8R.** CLASS(XFACILIT) ACCESS(UPDATE) ID(userid-of-the-tester)
SETR GENERIC(XFACILIT) REFRESH
SETR RACLIST(XFACILIT) REFRESH
```

The example job B8RJDF contains JCL that can be used to define this minimal set of testing profiles.

## Commands for creating, testing, and troubleshooting a RACF-Offline database

You can test RACF-Offline by issuing some RACF commands in the RACF-Offline environment. Running any of the RACF-Offline functions requires RACF access to the authorization profiles. You also need access to an offline RACF database.

### Creating an Offline RACF database

The example job B8RJUT2 provides the JCL to create a copy of the RACF database.

```
//STEP1 EXEC PGM=IRRUT200
//SYSRACF DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD SYSOUT=*
//SYSRACF DD DISP=SHR,DSN=Your-System-RACF-database
//SYSUT1 DD DISP=(NEW,CATLG),DSN=Your-Offline-database,
// UNIT=3390,SPACE=(4096,space,,CONTIG,ROUND),
// DCB=(LRECL=4096,RECFM=F)
```

You can adapt this job for your environment, or use your standard installation job for creating a copy of the RACF database. When using sample job B8RJUT2, specify the correct names and sizes for the RACF databases. In this JCL sample, the *space* is specified in blocks of 4 KB. If your current RACF database is allocated on an IBM 3390 Direct Access Storage Device in cylinders, you can multiply the number of cylinders by 180 to find the number of blocks required.

### Running commands against the Offline RACF database

After you have created a copy of the RACF database, you can use that RACF database to run some RACF-Offline commands. The member B8RJTST in the RACF-Offline installation library contains the following test JCL that runs RACF-Offline commands.

```
//RUNIT    EXEC PGM=B8RACF
//STEPLIB DD DISP=SHR,DSN=Your-Product.SB8RLNK
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
LU
END
//B8RPARM DD *
RACFDB 'Your-Offline-database'
SMF ID($B8R)
//
```

Adapt this JCL for your environment, then run the job. The following example shows the output from the job.

```
B8R121I B8ROPT options module successfully processed
B8R274I RACF DB used is BCSC.ROFFLINE.TESTDB
B8R304I New SMF-ID: $B8R
B8R143I B8RPARM file processed
B8R200A Enter RACF Command or "END"
LU
USER=B8RTEST  NAME=UNKNOWN  OWNER=B8R          CREATED=03.169
...
SECURITY-LABEL=NONE SPECIFIED
B8R200A Enter RACF Command or "END"
END
```

### Troubleshooting

If you receive RACF error messages such as IRR51004I, IRR51011I, or IRR52115I, or if you encounter an ABEND 483-024, update the Offline RACF database with the current templates by running IRRMIN00 with PARM=UPDATE.

## Check for RACF-Offline enablement

At startup, the B8RACF command verifies whether RACF-Offline is enabled or disabled by checking IFAPRDxx in PARMLIB.

- If RACF-Offline is enabled, or not defined in IFAPRDxx, initialization of RACF-Offline continues normally.
- If RACF-Offline is disabled, a message (B8R106E) is issued and processing stops.

To explicitly enable RACF-Offline, add an entry such as the following one to an active IFAPRDxx member.

```
OWNER('IBM CORP')
  NAME('zSecure Admin')
  ID(5655-N16)
  VERSION(*) RELEASE(*) MOD(*)
  FEATURENAME('RACF-Offline')
  STATE(ENABLED)
```

To disable RACF-Offline, add an entry such as the previous one to IFAPRDxx. Then, replace the STATE(ENABLED) parameter with the STATE(DISABLED) parameter.

After updating IFAPRDxx, apply the updates by running the operator command SET PROD=XX.



---

## Chapter 13. Setup of zSecure Alert

zSecure Alert is a real-time monitor for z/OS systems protected with the Security Server (RACF) or CA-ACF2. zSecure Alert issues alerts for important events relevant to the security of the system at the time they occur. zSecure Alert is part of the IBM Security zSecure suite and builds on zSecure Audit. For more information about zSecure Alert, see the *IBM Security zSecure Alert: User Reference Manual*.

---

### Verification of the product and release

See [www.ibm.com/support/home/pages/lifecycle/](http://www.ibm.com/support/home/pages/lifecycle/) for the IBM Software Support Lifecycle. (Search for zSecure in all products.)

---

### Considerations when upgrading from a previous release of zSecure Alert

If you are upgrading from a previous release of zSecure Alert, you must use one of the following procedures to ensure that you are using the correct software for the current release. If you have not started the zSecure Alert started task (C2POLICE) since the last IPL of the system, no specific upgrade steps are required.

- When stopping the previous version of C2POLICE, you must use the SIPL command; for example, you can use the following operator command:

```
MODIFY C2POLICE,SIPL
```

If you need to return to the previous release, you must again use the SIPL command to stop C2POLICE. After a successful stop of the started task, you can immediately start C2POLICE by using the previous release software.

- After stopping the current version of the zSecure Alert started task, you shut down and IPL the system. After the IPL, you start the zSecure Alert started task using the upgraded code. In this scenario, you do not need to perform any additional steps.

If you do not follow either of these procedures, startup might fail and an ABEND of the C2POLICE started task might occur. If startup fails with messages C2P0183E and C2P0123E, you might be able to recover using the FORCE startup parameter.

---

### Prerequisites for configuring and using zSecure Alert

#### About this task

zSecure Alert is one of the CARLa-driven components in the zSecure product family. For all CARLa-driven components, SMP/E installation is done concurrently, as described in [Chapter 4, “Installation of the software,”](#) on page 9 and the *Program Directory: IBM Security zSecure CARLa-Driven Components*. All CARLa-driven components make use of a zSecure configuration.

#### Procedure

Before configuring or using zSecure Alert, you must complete the following procedure:

1. Complete the basic installation process documented in [Chapter 4, “Installation of the software,”](#) on page 9 and the *Program Directory: IBM Security zSecure CARLa-Driven Components*. The basic, shared part of the installation process is described in [Chapter 4, “Installation of the software,”](#) on page 9.
2. As part of the installation process, create and customize a library with the low-level CKRINST qualifier. This library is where you can find the setup jobs for zSecure Alert.

3. The SCKRLOAD component must be APF-authorized; see [“APF authorization of the software”](#) on page 18.
4. A zSecure configuration that enables zSecure Alert is required.

The default zSecure configuration is shipped in *your.prefix.CKRPARM(C2R\$PARM)*. You might have configurations of your own. If you want to use zSecure Alert on multiple z/OS images, you must have a separate zSecure configuration for each image, with at least different SYS parameters. For additional information, see [“Distribution of zSecure data sets to additional z/OS images”](#) on page 17.

The Alert-enabled zSecure configuration is required to access the ISPF panels to configure zSecure Alert; pass this configuration to the C2POLICE and C2PCOLL started tasks. For more information, see Chapter 6, [“Deployment of the software,”](#) on page 23. You can also use this configuration when allocating data sets for zSecure Alert.

In the configuration, the parameters relevant for zSecure Alert are:

```
//* Parameters only used for zSecure Alert
//* SET C2PCUST='C2R.DATA.C2POLICE.C2PCUST'
//* SET C2POLICE='C2POLICE'
//* SET SIMESM=
```

As shipped, these parameters are commented out. You must uncomment them and supply your own choices for the zSecure Alert configuration data set and for the name of the address space.

## zSecure Alert address space overview

---

zSecure Alert runs as a started task. To collect all information for alert generation, it dynamically defines SMF exits. It also installs itself as an EMCS console and periodically starts the zSecure Collect program to obtain information about the system environment. For analysis and report generation, it invokes zSecure Audit at each reporting interval.

The following sections provide information about the available commands and options, guidelines for configuring the product, and information about the associated performance implications.

### Infrastructure

zSecure Alert runs as a permanently active started task (STC). Only one zSecure Alert address space can be active in a z/OS system image. It is started from the operator console using a regular start command.

The SMF exits capture all SMF records from all tasks in the system before they are written to the SMF log, that is, the MAN $\alpha$  data sets or the system logger LOGSTREAM. The records are passed unmodified to possible other SMF exits and subsequent SMF processing. Only SMF records specified in the active SMFPRM $\alpha\alpha$  parmlib member can be captured by SMF exits.

The EMCS console captures all WTO messages directed to the hardcopy set, typically the same as the SYSLOG. It does not modify any of the messages and allow standard processing to proceed.

The captured SMF records and WTO messages are optionally filtered. The remaining records and messages are saved in an in-memory buffer allocated in the private area of the STC for further processing.

The zSecure Collect program comes with zSecure Audit. It is used to periodically gather information about system libraries, UNIX files, current parmlib options, and so on. This information is gathered into a so-called CKFREEZE file. It is used to generate alerts for certain events related to critical system data sets and resources. It does not need for the installation to explicitly specify which data sets and resources are critical.

Some alerts base their selection criteria on the contents of the security database on your system as well as on the contents of the CKFREEZE file. This information is periodically refreshed by a preprocessing task, which prepares the queries included in the actual reporting step.

During each reporting interval, the data is passed to zSecure Audit for analysis and alert generation. Alerts can be issued through:

- E-mails



- Short text messages to pagers or cell phones
- WTOs, which can, for example, be captured by an automated operations package
- SNMP traps, which can in turn be captured by a network console like Tivoli® NetView
- UNIX Syslog messages, which, for example, can be sent to QRadar or ArcSight

**Note:** When commands are directed by RRSF or CPF, SMF records and consequently, alerts, are generated on the sending and on the receiving systems.

The analysis and reporting functionality provide great flexibility in the type of record selection criteria, the use of thresholds, and the formatting of alert messages. It also allows annotating, for example, userids with parts of the associated installation data or user data from the security database. It also allows general key-based lookups in other external files. See *IBM Security zSecure: CARLa Command Reference* for detailed information about the CARLa Auditing and Reporting Language (CARLa).

zSecure Alert uses dynamic SMF exits to capture all SMF records from all tasks in the system. However, it is possible only if these EXITS have been enabled by the SMF parameter member in PARMLIB. z/OS release 2.3 provides SMF record support for more SMF record types and more granular timestamps. When running on z/OS release 2.3 or higher, zSecure Alert uses the IEFU86 exit to capture all SMF records. You must ensure that the IEFU86 exit routine is enabled for SYS and for all subsystems specified in SMFPRMxx. On z/OS release 2.2 and below, zSecure Alert uses SMF exits IEFU83, IEFU84, and IEFU85, and you must ensure that these exit routines are enabled. On z/OS release 2.3 or higher, when IEFU86 is not enabled, zSecure Alert uses the z/OS 2.2 exits for a partial fall back. The z/OS 2.2 exits do not capture SMF records that exploit the enhanced z/OS 2.3 SMF record support. For more information about specifying EXITS in SMFPRMxx, see the relevant chapter in *MVS Initialization and Tuning Reference*.

If you have changed SMF EXIT definitions and activated them dynamically using a SET SMF command, you must also reinitialize the zSecure Alert started task by a RESTART command. See [“zSecure Alert operator commands”](#) on page 116 for details about the RESTART command.

## Supported ddnames for the zSecure Alert started task

The zSecure Alert started task uses the following DD statements. Not all DD statements are required in all environments. Member C2POLICE in the SCKRPROC provides an example procedure for the zSecure Alert started task.

### C2P1OUT

The C2P1OUT DD statement identifies a work data set for the output of the pre-processing CARLa step, which is passed to the alert generating CARLa step.

### C2PDEBUG

The C2PDEBUG DD statement identifies the output stream that is used for error messages that CKRCARLA, CKFCOLL and C2POLICE issue.

### C2PEMFRB

The C2PEMFRB DD statement identifies a data set that retains information about the current base configuration that is used for extended monitoring. It is only needed if extended monitoring is active.

### C2RCMD

The C2RCMD DD statement identifies an output stream that is used for commands if you selected the **Write TSO-RACF command to C2RCMD DD** option on the Action Customization panel while activating an alert.

### C2RSMTTP

The C2RSMTTP DD statement identifies an output stream that is used for SMTP messages if you selected the **Write e-mails to C2RSMTTP DD** or **Write text messages to C2RSMTTP DD** option on the Alert Destination panel.

### C2RSNMP

The C2RSNMP DD statement identifies an output stream that is used for SNMP traps if you selected the **Write SNMP traps to C2RSNMP DD** option on the Alert Destination panel.

## **C2RSYSLG**

The C2RSYSLG DD statement identifies an output stream that is used for UNIX type SYSLOG messages from two sources:

- If you selected the **Write messages to C2RSYSLG DD** option on the Alert Destination panel, the syslog message for that alert is routed to the file.
- If a syslog message cannot be delivered to any TCP or UDP destination, it is automatically sent to the C2RSYSLG file, unless SUPPRESS SYSLOG\_FALLBACK\_FILE is specified. See also message CKR2003.

The syslog messages are in UTF8 format.

## **C2RWTO**

The C2RWTO DD statement identifies an output stream that is used for WTO messages if you selected the **Write WTOs to C2RWTO DD** option on the Alert Destination panel.

## **CKFREEZE**

The CKFREEZE DD statement identifies the CKFREEZE data set that is used for the standard alert generation process. The C2PCOLL started task updates it daily.

## **CKGPRINT**

The CKGPRINT DD statement identifies the output stream that is used for error messages that the CKGRACF program issues. You need this DD statement only if CKGRACF commands are specified as alert action commands.

## **PARMLIB**

The PARMLIB DD statement identifies the C2PCUST data set that contains the primary Alert configuration member.

## **SC2PSAMP**

The SC2PSAMP DD statement identifies the data set that contains standard and customized CARLa members for use during alert generation. This data set should also contain the secondary Alert configuration members.

## **STEPLIB**

The STEPLIB or JOBLIB DD statement identifies the library containing the C2POLICE and CKRCARLa programs.

## **SYSINCKF**

The SYSINCKF DD statement identifies a data set with CKFCOLL input statements. Use this DD statement at the request of zSecure software support only.

## **SYSPRCKF**

The SYSPRCKF DD statement identifies a work data set for the output of the CKFCOLL program that is used to collect data for the extended monitoring process. It is only needed if extended monitoring is active.

## **SYSPRRPT**

The SYSPRRPT DD statement identifies a work data set for the output of the alert generating CARLa step (also known as the Reporting phase).

## **SYSPRST1**

The SYSPRST1 DD statement identifies a work data set for the output of the pre-processing CARLa step (also known as the stage-1 phase).

## **SYSTCPD**

The SYSTCPD DD statement identifies the TCP/IP configuration data set that must be used for the zSecure Alert started task.

## **SYSTSIN**

Unused.

## **SYSTSPRT**

The SYSTSPRT DD statement identifies the output stream that is used for messages and debug output for TSO commands that are executed in the zSecure Alert environment. These are mainly ALLOCATE commands for Extended Monitoring data sets.

## Configuration

By default, zSecure Alert captures all SMF records from all tasks in the system. Analysis is done at specified intervals. In large systems with high activity, the amount of buffer space needed can be significant. To reduce the amount of storage and speed up processing, you can specify a *filterlist* as part of the startup parameters to pass only certain records to the SMF record analysis routines. Doing so can also significantly reduce the required processing time. Likewise, by default, zSecure Alert captures all WTO messages (except most C2P messages), and it is possible to filter these messages.

By default, data analysis is performed every 60 seconds. Environment data is read in every hour. The CKFREEZE file with the I/O configuration data is refreshed daily. Furthermore, there is a *time window* interval for averaging alerts that use history data.

Specifying the values for filtering, analysis interval and buffering can be done through PARMLIB and console operator commands. The various interfaces for controlling zSecure Alert processing are described in this chapter. When using the zSecure Alert ISPF interface for configuration definition, most of the PARMLIB statements are automatically set and updated as required.

The zSecure Alert ISPF interface allows configuring the Extended Monitor settings. Selecting Extended Monitoring alerts is possible only if the person responsible for installation and deployment of the software has completed the process of enabling Extended Monitor alerts.

As previously mentioned under Infrastructure, the SMF record analysis is done using zSecure Admin and Audit functionality. All the possible alert situations are defined using CARLa scripts. They are specified by the SYSIN DD statement in the startup JCL. It is possible for an installation to modify the selection criteria and thresholds for the defined alerts, and to add installation-specific alerts. See the information about predefined alerts in the *IBM Security zSecure Alert: User Reference Manual* for a complete overview of the alerts shipped with zSecure Alert.

## Control

At startup time and during the execution of the program, it is possible to control its execution using commands from the operator console. The commands for direct operator interaction to manage the started task are documented in [“Starting, stopping, and modifying the zSecure Alert started task” on page 114](#). Other commands are documented in [“Other commands” on page 121](#). They are DEBUG, OPTION, REPORT, FILTER and so on. DEBUG is for diagnostics; OPTION is to manage the in-memory data buffers and the zSecure Collect background data gathering process. REPORT is used to specify the reporting interval and the events to report on. FILTER is used to limit the SMF record types and WTO messages collected.

## Post-installation tasks

---

After installation of zSecure Alert, complete the tasks described in the following sections.

### Setup of started tasks

zSecure Alert must run as a started task. Copy the C2PCOLL and C2POLICE procedures from SCKRPROC, and your zSecure Alert-enabled zSecure configuration members, to a started task procedure library on the system where you are going to use zSecure Alert. When copying these procedures you can choose different names:

- For the zSecure Collect started task, the name that you choose is the one you specify during Alert configuration. See the section about specifying general settings during Alert configuration in the *zSecure Alert User Reference Manual*.

Please note that zSecure Collect is executed with parameter UNIX=YES, which might result in additional RACF SMF type 80 success records being generated. This depends on your RACF SETROPTS and UAUDIT settings:

- If SETROPTS AUDIT(FSOBJ) has enabled command auditing for the FSOBJ class, RACF will write SMF records for successful CHDIR events that are performed during zSecure Collect processing.

- If SETROPTS LOGOPTIONS(ALWAYS(DIRSRCH)) has set the auditing options for the DIRSRCH class to always log events, RACF will write SMF records for successful DIRSRCH events that are performed during zSecure Collect processing.

These classes (FSOBJ or DIRSRCH) need not be active to control auditing.

Depending on the size of your z/OS UNIX file system, the number of SMF records that are produced could be large.

- For the zSecure Alert address space, where the default is C2POLICE, use the name as specified in the C2POLICE parameter in the zSecure configuration.
- In both procedures, you must specify your Alert-enabled zSecure configuration.
- In the zSecure Alert JCL, you must update the PPARM parameter to reflect the name of the Alert configuration with which you intend to run. The PPARM parameter must be equal to the zSecure Alert configuration name, with a P appended. You can also use a system symbol in the name of the zSecure Alert configuration to be used. For example, you might change

```
// PPARM=C2PDFLP          C2POLICE parameter member <setname>P
```

to

```
// PPARM=&SYSNAME.P      C2POLICE parameter member <setname>P
```

If you further customize the JCL, make sure that you do not include an OUTPUT statement with DEFAULT=YES and an OUTDISP of PURGE or HOLD. Doing so prevents all email and text message alerts from being sent.

## Security resources

You must set authorizations on all systems where you are going to run zSecure Alert, unless you share the security databases. Security requirements for the zSecure Alert started tasks include:

- The userid and group or logonid that you intend to run the C2POLICE and C2PCOLL address spaces, or the names you selected for them. For C2PCOLL, the userid or logonid must have an OMVS segment for RACF, or an OMVS record for ACF2. For most alert destinations (for example, your SIEM), the C2POLICE started task also needs access to OMVS.
- Started task names and security resources to assign the required userids and group or logonid to these started tasks.
- On both RACF and ACF2 systems, READ access to the XFACILIT resource CKR.READALL and CKR.CKRCARLA.APF is required. The userid also requires READ access to several OPERCMDS resources. See job C2PZAIN0 for RACF or job C2PZAINA for ACF2.
- The userid or logonid for C2PCOLL must have READ access to the XFACILIT resources CKF.AUDIT and CKF.ALERT, unless your installation configured a different security class. See [Appendix A, “Site module,” on page 191](#).
- Started task output protection.
- Data set names and profiles covering them. This might include PROGRAM profiles when PADS access is being used.

The verification function emulates the Alert address space processing. Therefore, the user ID that performs the verification cannot be in restricted mode. This user ID also requires access to the security database and CKFREEZE that are specified in the Alert configuration. Or, if the user ID does not have access, they must specify an UNLOAD, or a different security database source, and a CKFREEZE that they do have access to. See **Use SETUP FILES input instead of zSecure Alert input data** in the Setup Alert panel (*zSecure Alert User Reference Manual*, section "Alert configuration: verify alert configuration").

**Note:** This option applies only to the verification function. The Alert address space always uses the security database as configured and the CKFREEZE data set as specified in the C2POLICE JCL member.

For the users who configure zSecure Alert, the following security requirements apply:

- READ access to the data sets where the zSecure software resides.

- Access to the zSecure-specific resources that determine which options and actions are available. See [Appendix B, “Security setup for zSecure,” on page 193](#).
- For configuring, UPDATE or WRITE for ACF2 access to the zSecure Alert configuration data set is required. With READ access, the user can examine the configuration.
- The user must have access to the following resources:

– UPDATE:

```
OPERCMDS: MVS.MODIFY.STC.C2POLICE.C2POLICE
```

– READ:

```
OPERCMDS: MVS.MCSOPER.userid* (the actual resource depends on the ISPF screen ID)
TSOAUTH: CONSOLE
```

You need this access to determine which alert configuration is currently active and to refresh an alert configuration. If TSOAUTH CONSOLE access is missing but SDSF is installed, the SDSF REXX API is used. Without access to issue operator commands, both of the following situations occur:

- The Act indicator on the "Managing alert configurations" panel remains blank. See the information about managing alert configurations using option SE.A.A in the *IBM Security zSecure Alert: User Reference Manual*.
- Updates to the alert configuration are not effective until the operator issues the F C2POLICE, REFRESH command, during an environment refresh, or when the zSecure Alert address space is restarted.

To prevent the user from issuing operator commands from the zSecure Alert ISPF panels, remove access for the user's ID to TSOAUTH CONSOLE and SDSF ISFCMD.ODSP.ULOG.JES%.

If you use zSecure Alert on multiple z/OS images, there is no cross-image communication. Therefore, the Act column is blank even if an alert configuration is in use on a different z/OS image, and there is no automatic refresh on the other images.

Job C2PZAIN0 for RACF and job C2PZAINA for ACF2 in the SCKRSAMP library are supplied to help you set up these security resources. Note, however, that the security resources you create should be subject to your security policy, such as choices between generic and discrete profiles. You can decide to run this job after reviewing the RACF or ACF2 commands.

In these jobs, some things are assumed and not customized during the zSecure configuration; you might want to change these things:

- The started tasks for zSecure Alert run under a common userid or logonid.
- On RACF systems, group name SYSAUDIT is assumed as the group to contain system auditors, but you can choose another group. Connect to the SYSAUDIT group. If you do not, the allocation of data sets for zSecure Alert (job C2PZAIN1) will fail.
- On RACF systems, the group owner is set to SYSAUTH.
- It is assumed that profiles or rules have been set up for the SCKRLOAD data set and that a separate profile or rule exists for the other data sets. If you have a different setup, adapt this in this job.
- On RACF systems, it is assumed that PROGRAM profile CKRCARLA exists. If you do not use PROGRAM profiles, you can remove the rdefines, ralters and permits for the PROGRAM profiles.

## Required data sets

zSecure Alert requires the following data sets:

- A dedicated CKFREEZE data set. This data set must pertain to the z/OS image where zSecure Alert is running, and so it cannot be shared with zSecure Alert on other z/OS images. Because of serialization issues, the data set cannot be shared with zSecure Admin or zSecure Audit. The C2PCOLL address space periodically refreshes the contents of this CKFREEZE data set.

- Data sets with intermediate data for the C2POLICE address space. In the JCL, these data sets are identified as SYSPRST1, SYSPRRPT, SYSPRCKF, C2P1OUT, and C2PEMFRB. The C2POLICE address space allocates them as shared so that you can view them for diagnostic purposes, but they must not be otherwise shared. By default, these data sets contain the system id in their names.
- The Alert configuration data set. This is the data set that you previously specified in the zSecure configuration. It is written into from the ISPF interface when option SE.A "Configure zSecure Alert" is used to customize zSecure Alert. The data set must be a PDS/E to prevent space abend. A PDS/E can be shared between system images. If the data set is shared between system images, then a different configuration can be used for different images, but they *can* also be the same.

Make sure that the Alert configuration data set is well protected against any attempt of intrusion into its configuration. For instance, intruders could find out the cell-phone numbers that are used for your alert messages and saturate them with messages that do not look intrusion-related before starting the actual intrusion.

When upgrading from a previous release of zSecure, do not create new configuration data sets. Instead, continue to use the data sets that contain the results of your earlier configuration effort.

Job C2PZAIN1 in the SCKRSAMP library is supplied to assist you in creating these data sets. Before submitting, customize as follows:

- Unless you updated the default zSecure configuration, change the JCLLIB and INCLUDE statements to specify the zSecure Alert-enabled zSecure configuration you prepared.
- If you install zSecure Alert on multiple z/OS images, you must run the job multiple times: once with each zSecure configuration. Preferably, run each of these jobs under the z/OS image where the corresponding zSecure Alert is to run. If this is not possible (for example, because you are installing into a z/OS image that has not yet been IPL'd), make sure that all data sets are allocated on volumes that are accessible from the intended z/OS image.
- If you want to share the zSecure Alert configuration data set among z/OS images, remove the C2PCUST allocation from all C2PZAIN1 runs except one.

## SMF requirements

zSecure Alert uses dynamically defined SMF exits to capture all SMF records from all tasks in the system, before they are written to the SMF log (that is, the MANx data sets). The records are passed unmodified to possible other SMF exits and subsequent SMF processing. SMF creates only those records that have been selected from SMFPRMxx. Also, the SMF dynamic exits are invoked only if the exits are enabled in SMFPRMxx. Before continuing, identify the SMF records that are required for your selected alerts, and enable the necessary SMF exits in your SMFPRMxx. Most alerts require one of the following:

- SMF records types 30 and 80 on RACF systems
- The type written by ACF2, which has default type 230 but might be different for your installation

For a detailed description, see the alert descriptions in the *zSecure Alert User Reference Manual*. For z/OS release 2.3 or higher, the SMF exit that must be enabled is IEFU86. For z/OS 2.2 and older, exits IEFU83, IEFU84, and IEFU85 must be enabled. Each exit point is used in a specific environment and for specific SMF records. If IEFU86 is not enabled on z/OS release 2.3 or higher, zSecure Alert uses the z/OS 2.2 exit routines as a partial fallback. However, this might cause a failure to detect certain alert conditions. Ensure that these exit points are enabled for the entire system and for all defined subsystems. For example:

```
SYS(EXIT(IEFU86))
```

or

```
SYS(EXIT(IEFU83, IEFU84, IEFU85))
```

Incorrect specifications in SMFPRMxx can lead to failure to detect some alert situations.

# Specifying data set parameters for extended monitoring

## About this task

For extended monitoring, the C2POLICE started task creates a CKFREEZE snapshot data set at each environment refresh interval. The parameters for allocating these data sets are entered in a special template format in member C2PEMFRT in the C2PCUST data set.

## Procedure

To create the C2PEMFRT member, follow these steps:

1. Copy member C2PEMFRT from the SCKRSAMP data set to the data set you specified for C2PCUST in your configuration member.
2. Edit the copied C2PEMFRT member. The sample member starts with the following lines:

```
alloc reuse fi(ckfreeze) -  
DA('unique.sysplex-wide.prefix.CKFREEZE.D&LYMMDD..T&LHR.&LMIN.') -  
new vol(your.volume) space(2,1) cylinders release -  
recfm(v b s) lrecl(X) blk(27998)
```

3. In the sample member, edit the data set name, the space parameters, and the data set placement parameters to follow the installation conventions used in your environment.
  - Multiple input lines are allowed.
  - Indicate continuation lines with the minus sign (-).
  - Columns 73-80 are ignored.
  - The total command length must be less than 255 characters. The length includes all blanks between the last significant character on a line and the subsequent line continuation character, the minus sign (-).
  - Aside from the symbol substitution, the command entered in these members must be a complete and valid TSO ALLOCATE command. Remove any keywords that are not needed, for example, the VOLUME keyword.
  - The REUSE and FILE keywords must be kept as shown in the example. The filename specified must be CKFREEZE.
  - System symbols can be included anywhere in the command. User and JCL symbols are not supported.
  - The record format of the data set must be variable blocked spanned, as indicated by the RECFM(V B S) keywords.
  - The data set name specification must begin with the string DA('
  - The data set name specification must end with the string ')
  - The data set names as specified must end in D&LYR2&LMON&LDAY.T&LHR&LMIN. Specifying the data set names in this way results in a timestamp formatted as Dyyymmdd.Thhmm.
  - You can specify any leading qualifiers you want, as long as the data set name after symbol substitution remains valid.
  - The high level qualifiers must uniquely identify the data set as reserved for zSecure Alert Extended Monitoring. Also the name must not match the name used by another zSecure Alert started task. You can use system symbols, like &SYSCONE and &SYSNAME to uniquely identify these data sets.
  - You can specify additional parameters for the allocation. For example, if your installation supports specification of SMS constructs such as STORCLAS or MGMTCLAS, you can use them here.
  - Optional comment lines must be included at the end. Comment is included between the comment delimiters /\* and \*/.

**Note:** It is important to specify the qualifiers before the final date and time qualifiers so that only the intended CKFREEZE snapshot data sets match. All data sets starting with these qualifiers are considered as temporary CKFREEZE data sets that are eventually deleted.

4. Save the C2PEMFRT member.

## Setup of the alert configuration data set

Set up alert configuration as described in the *IBM Security zSecure Alert: User Reference Manual*. Before you enter the zSecure ISPF interface, first create the zSecure Alert Configuration Data Set.

## Startup of the zSecure Alert address space

When all configuration steps have status OK, you can start the zSecure Alert started task with the MVS command `START C2POLICE`. When you use a zSecure Alert configuration other than the default C2PDFL, change the PPARM parameter in the C2POLICE started task JCL. That is, PPARM must be the configuration name suffixed with a P. Alternatively, specify the PPARM on the start command.

After the first time you start the zSecure Alert address space, issue the following MVS command:

```
F C2POLICE, COLLECT
```

Issuing this command ensures that the zSecure Alert address space uses a matching CKFREEZE file. Subsequent refreshes are done automatically.

Start zSecure Alert soon after each IPL, using your Automated Operation software or PARMLIB member `COMMNDxx`. However, do not start zSecure Alert before OMVS is fully initialized. The following message indicates that OMVS is fully initialized:

```
BPXI004I OMVS INITIALIZATION COMPLETE
```

The wait is required because zSecure Alert requires TCP/IP services.

## The preamble member C2PXDEF1

The C2PXDEF1 member is automatically created (empty) in the zSecure Alert configuration data set by the SE.A transaction when the member does not exist yet and update is allowed. This member is used as a preamble for zSecure Alert processing, both in the zSecure Alert address space and during Verify. It is intended to be used only as directed by IBM Software Support.

Errors in other software sometimes cause data that is not valid to be written. For example, badly formatted SMF records can cause error messages in zSecure Alert. However, these error messages do not make it clear that the errors are caused by invalid input, and usually IBM Software Support is contacted. If this happens, IBM can send you a set of CARLa statements that you can temporarily use in C2PXDEF1 until the problems with the OEM-vendor are solved.

## Starting, stopping, and modifying the zSecure Alert started task

zSecure Alert is started from the operator console by a `START` command. The command executes the procedure from the applicable system proclib. It is possible to specify startup parameters. These parameters can be given at the `START` command itself. An example of such a `START` command is:

```
S C2POLICE, PARM.C2POLICE=DEBUG
```

See [“zSecure Alert START parameters”](#) on page 115 for the available startup parameters.

zSecure Alert also supports parameter input from the PARMLIB DD statement in the startup procedure. The PARMLIB DD statement is used for those parameters that determine the normal operational environment. The parameters can be specified in the form of commands with keywords. TSO conventions are used for these commands. See the sections on the following pages for detailed descriptions of the supported commands, keywords, and parameters.

During execution of the started task, the console operator can also issue commands to monitor, or modify the functioning of zSecure Alert. All these commands can be issued through the MODIFY console



command. MVS supports use of the F command as an alias of the MODIFY command. An example of such a command is:

```
MODIFY C2POLICE,DISPLAY
```

The text after the comma must be one of the supported zSecure Alert operator commands.

To terminate the started task, the console operator can issue the STOP command. MVS supports use of the P command as an alias of the STOP command; for example:

```
P C2POLICE
```

The STOP command can also be issued as the parameter on the MODIFY command.

```
F C2POLICE,STOP
```

See “zSecure Alert operator commands” on page 116 for the detailed description of the operator commands available in zSecure Alert.

## zSecure Alert START parameters

zSecure Alert supports two startup parameters. Startup parameters can be used by the operator as part of the START command.

```
S C2POLICE,PARM.C2POLICE=FORCE
```

For normal execution of zSecure Alert, you do not need to specify any startup parameter. By default, zSecure Alert detects if it is already executing, and issue an appropriate error message and terminate. Also, when zSecure Alert has been shut down previously, it reuses those critical system resources that can be obtained only once. These system resources cannot be returned to the system. It ensures that no system resources are wasted.

The C2POLICE started task shares the zSecure SMF exits with the zSecure SMF Collector (CKQEXSMF):

- If zSecure Alert is not yet active, the SMF exit routines are installed as Dynamic Exit routines.
- If the zSecure SMF Collector (CKQEXSMF) is already active, the exits that CKQEXSMF installed send selected events to the buffers that C2POLICE maintains and the other way around.
- If you stop either CKQEXSMF or C2POLICE, the exits continue to be used for the other task.
- If you stop both CKQEXSMF and C2POLICE, the exits are uninstalled.

It is important that the same level of the zSecure code is used for CKQEXSMF and C2POLICE. If you are also planning to use the zSecure SMF Collector (CKQEXSMF), you cannot start the C2POLICE and the CKQEXSMF tasks at the same time; you must separate the start times by a few seconds. You must also issue STOP commands with several seconds in between.

In some error situations, initialization of zSecure Alert fails. In those situations, one of the optional START parameters can be required.

### DEBUG

Specifies that diagnostic messages must be issued during the first part of the initialization. These diagnostic messages can also be used to determine possible problems in processing the standard PARMLIB parameters. This setting is in effect until a subsequent DEBUG command is issued either from the operator console, or from PARMLIB.

### FORCE

Specifies that irrespective of a previous execution, initialization must continue. Only use the FORCE option if you cannot start zSecure Alert normally. The FORCE option might enable starting zSecure Alert without the need to IPL the system. During normal operation, the FORCE option is never required.

### DEBUG-FORCE

Specifies that both the DEBUG and FORCE options must be active at startup.

## zSecure Alert operator commands

The zSecure Alert operator commands that might be used in the MODIFY console command are described in the following list:

### CRSH

This command causes an immediate abend of the C2POLICE started task, without any recovery. It is intended for IBM internal testing only.

### STOP

Stop execution of the zSecure Alert started task. This results in an orderly shutdown of the task. Some memory remains reserved after termination of the task to enable reuse of some critical system resources during a subsequent restart of the started task. The effect of the STOP MODIFY command is identical to that of the MVS STOP command.

The STOP command does not support any additional keywords.

### RESTART

This command results in an orderly shutdown of zSecure Alert processing, followed by an immediate reinitialization. The address space in which the started task is running is not terminated. No additional console operator command is needed to reactivate zSecure Alert processing. The main difference between a restart and a STOP command followed by a START command for the started task is the preservation of the ASID. Also, possible changes in the started task procedure cannot be effected during RESTART processing. During the time required to process the RESTART command, alert situations are not recognized, and no reports are generated.

Because the STOP/START sequence results in marking the address space as *non-reusable*, the RESTART command is preferred in most situations. This command prevents loss of potentially critical system resources.

The RESTART command does not support any additional keywords.

### REFRESH

This command results in the reprocessing of the command and parameters specified in the PARMLIB and a refresh of some subtasks. Not all PARMLIB commands can be processed. The OPTION command is not supported during a REFRESH. The preprocessing subtask is started and the reporting task is ended and restarted. Because the alert generating reporting task can only be restarted after completion of the preprocessing task, it might take several minutes for the refresh process to complete. During this period, alerts are still generated according to the existing configuration.

During the time that the zSecure collect process is running, the REFRESH command is accepted, but most processing is delayed. When the collect process has finished, subtask start and restart processing as described is performed.

The REFRESH command does not support any additional keywords.

### COLLECT

This command results in an immediate, synchronized execution of the zSecure Collect started task. Processing is identical to that resulting from the normal scheduled start of the zSecure Collect task. The name of the started task is controlled by the *CollectSTCName* parameter. The regular scheduled start of the STC is not affected and remains at the time specified by *CollectTime*.

While the zSecure Collect process is running, the reporting task remains active. Alerts continue to be issued as usual. The preprocessing task might be delayed until the collect task has completed.

The COLLECT command does not support any additional keywords.

### SIPL

This command is to be used only in emergency situations. It results in freeing all in-memory data structures, loss of a system level LX, that is, linkage index, and marking the address space as *non-reusable*. System level LX's are a limited resource, and cannot be recovered without an IPL of the system. When upgrading from one release of zSecure Alert to another, the installation instructions require that you shut down the previous version using this SIPL command.

The SIPL command does not support any additional keywords.

### **DISPLAY**

This command results in a display of the status and options of zSecure Alert processing. It shows the current options, the buffer space used, the buffer number in use at the time, and the status of several error indicators if set.

The DISPLAY command does not support any additional keywords.

### **REPORT**

This command enables you to set the values for the keywords that control processing of the captured data. The new values are used the next time that the program refers to them. Values specified through an operator MODIFY command are overwritten by values from PARMLIB during the next REFRESH, which might be started by the console operator, or automatically at the end of each *Stage1Interval*.

### **FILTER**

This command enables you to set the filtering criteria for the SMF-records and the WTO-messages before they are captured in the in-memory buffers. Efficient use of these filter criteria can significantly reduce the amount of buffer space needed. The new filter criteria are effective immediately. For a complete description of all keywords, see [“FILTER command” on page 129](#).

**Note:** Values specified through an operator MODIFY command are overwritten by values from PARMLIB during the next REFRESH, which might be started by the console operator, or automatically at the end of each *Stage1Interval*.

### **DEBUG**

This command controls the diagnostic and monitoring messages that can be generated by the program. All messages are described in *IBM Security zSecure: Messages Guide*. The command is effective immediately. For a complete description of all keywords, see [“DEBUG command” on page 121](#).

### **DIAGNOSE**

This command is used to display detailed information or perform diagnostic tasks. It allows dumping some internal control blocks and tables for problem determination. The control blocks displayed are intended for IBM support personnel to diagnose certain problems. For a complete description of all keywords, see [“DIAGNOSE command” on page 124](#).

## **Cleanup and deactivation of SMF exits**

### **About this task**

The C2POLICE started task shares the zSecure SMF exits with the CKQEXSMF started task that can be used to collect SMF records for CKQRADAR. If CKQEXSMF is not yet active, the SMF exit routines are installed as Dynamic Exit routines.

If CKQEXSMF is already active, the exits that CKQEXSMF installed send selected events to the buffers that C2POLICE maintains and the other way around. If you stop either C2POLICE or CKQEXSMF, the exits continue to be used for the other task. If you stop both CKQEXSMF and C2POLICE, the exits are uninstalled. It is important that the same level of the zSecure code is used for CKQEXSMF and C2POLICE.

If both CKQEXSMF and C2POLICE are stopped, but the exits are still installed and active, some residual processing is done by the exit routines to check for the presence of either task. In this case you might want to manually deactivate the SMF exit routines. If you do this while either CKQEXSMF or C2POLICE is still active, SMF events are no longer captured; this results in missing events in CKQRADAR and possibly missing alerts from C2POLICE.

Use the following procedure to manually remove residual SMF exits.

### **Procedure**

1. Issue the following commands to determine whether the SMF exit modules are active and under what exit names:

```
d prog,exit,mod=c2psmfu8
d prog,exit,mod=c2psmf86
```

The output looks like the following example:

```
CSV462I 13.11.54 PROG,EXIT DISPLAY 494
MODULE C2PSMFU8
EXIT(S) SYS.IEFU85 SYS.IEFU84 SYS.IEFU83
EXIT(S) SYSTS0.IEFU83 SYSSTC.IEFU83 SYSASCH.IEFU83
EXIT(S) SYSJES2.IEFU83 SYSJES3.IEFU83 SYSTS0.IEFU84
EXIT(S) SYSJES3.IEFU84 SYSASCH.IEFU84 SYSJES2.IEFU84
EXIT(S) SYSSTC.IEFU84 SYSTS0.IEFU85 SYSSTC.IEFU85
EXIT(S) SYSASCH.IEFU85 SYSJES2.IEFU85 SYSJES3.IEFU85
```

2. Deactivate each exit by name:

```
setprog exit,modify,en=<exit name>,mod=c2psmfu8,state=inactive
```

## Configuration guidelines and performance implications

zSecure Alert processing consists of several parts. The parameters specified at startup influence the overall performance of zSecure Alert and its impact on other users. The most important parameters in this respect are the *intervals* and the *filters*.

### WLM service class

The data collection and processing that is done by zSecure Alert must be done in a timely manner. To ensure that sufficient CPU resources are available, the Alert started task resets itself to the WLM service class SYSSTC. If you must assign a lower dispatching priority, create an applicable WLM classification rule that assigns the required WLM service class. You must also specify NoSetSRVClass on the OPTION statement to prevent the program from resetting itself to the SYSSTC service class. If you are running on a single processor system, careful analysis of your workload is required and using NoSetSRVClass might be preferred.

### Filters

As indicated in “Configuration” on page 109, filtering is mostly a performance issue, although too narrow a filter can cause alerts to be lost. Filtering is done based on WTO message identifiers and SMF record types and subtypes only. The actual event selection must be done in the CARLa in the skeleton members for the individual alerts. If you specify no filter, all SMF records and WTO messages are captured. An exception exists for most C2P messages, which are not captured. The predefined alerts have their filter settings preset. For your own alerts, you must specify the correct filter settings from interface option SE.A.A, see the information about adding your own alerts in the *IBM Security zSecure Alert: User Reference Manual*. When you Verify an alert configuration, the correct overall filter settings are generated for activation through the Refresh action.

### Intervals

There are several relevant intervals:

- Reporting interval for performing data analysis and generating alerts
- stage 1 interval for reassessing the environment
- "average" interval for "moving window" analysis

By default, data analysis is done every 60 seconds. This interval might be increased if you do not need almost real-time alert messages. If you need a faster response, you can reduce the interval time.

**Note:** For each reporting interval, a new buffer is used so that this ties in with the buffer considerations explained in the next section.

The preprocessing subtask (also known as stage-1) obtains current information about the system environment and user attributes. This task is carried out hourly by default. If you require current information, you must process the security database and the CKFREEZE file more frequently. Processing the security database is relatively quick, but obtaining a new I/O configuration image is a costly process. zSecure Collect is typically scheduled to run once a day at a particular time to refresh the full CKFREEZE file. However, it is also possible to have zSecure Alert dispatch this task by using the operator command `MODIFY C2POLICE,COLLECT`. At the preprocessing interval, zSecure Alert can also create a small CKFREEZE snapshot of a subset of the system environment. This small CKFREEZE snapshot is taken and processed only if extended monitoring is active. The small CKFREEZE is not intended for any other process.

As part of SMF processing, the CKRCARLA program retains certain SMF data to complete other SMF records that lack this data. An example of such SMF data is the user ID for SMF record type 15. By default, the refresh of the environment information involves stopping and starting the CKRCARLA subtask. As a result, the retained information is lost, and must be re-established. This often results in the fields being reported as "missing". It is possible to retain the information for a longer period through specification of the `REFRESHMODE(INTERNAL)` option (see "[RefreshMode](#)" on page 127). The necessary SMF information will be retained until the C2POLICE started task is restarted or stopped.

Some "averaging" alerts with thresholds might use a time window larger than the reporting interval. For these alerts, SMF records are kept in history buffers for five times the reporting interval, for example. This long-term analysis interval can be adjusted as well, depending on your reporting needs.

## Buffers

Another important consideration for the configuration of zSecure Alert is the in-memory buffer usage. The buffer space used by zSecure Alert is regular pageable storage in the private area of the zSecure Alert started task address space. It is similar in all aspects to the working storage of a TSO user editing a data set. As a guideline for calculating the buffer size, you can perform the following steps.

**Note:** The numbers given in the steps are for illustration purposes only and must not be used as a starting point for your system.

1. Look at the output of your SMF dump program. Summarize the number of RACF SMF records (Record type 80) or ACF2 SMF records, and Accounting SMF records (Record type 30) written per day.

For instance, on a small system, during an average day, the MAN data sets are switched and dumped five times. The output of the IFASMFDP program shows the following numbers of RACF or ACF2 SMF records: 50,000 32,000 69,000 49,000 and 27,000. The total number of RACF or ACF2 SMF records written during that average day is 227,000. The number of SMF 30 Records were: 19000 15000 31000 23000 and 17000. The total number of SMF 30 records during the day is 105,000.

2. Assuming an alert reporting interval of 1 minute (the default), calculate the number of records per interval.

In this example, it yields  $227,000 / 1440 = 158$  RACF or ACF2 records, and  $105,000 / 1440 = 73$  SMF-30 records per minute.

3. Look at the output of your SMF dump program for the average record length of these SMF records. It must be 250 - 300 bytes for the RACF records, 600 - 700 bytes for ACF2 records, and 1000 - 1500 bytes for the SMF-30 records.

4. Multiply the average number of records by the average record length to find the average buffer size per interval.

In the example of the small system, it results in  $(158 * 274) + (73 * 1224) = 132,644$  bytes.

5. To accommodate for normal fluctuations in system workload, multiply the average found by a factor of 5, and round up to the nearest "nice" number to find the best starting point for your *bufsize* parameter.

In the example, a good setting for the *bufsize* parameter is 700 KB.

After determining the minimum buffer size, the next concern is about the number of buffers required. As mentioned, the minimum number of buffers is also related to your long-term event analysis. For instance, if you want to generate an alert whenever a user generates more than 10 RACF logon violations in 10

minutes, the amount of data kept in the buffers must represent at least 10 minutes. Because one buffer is always being filled with new events and therefore not available for the averaging process, the formula becomes:

```
Numbufs > (AverageInterval / Interval) + 1
```

As a starting point, use twice the number of buffers based on this formula. So, assuming that you use the default values for *Interval* (60 seconds) and for *AverageInterval* (300 seconds), you end up with  $2*((300/60)+1) = 12$  buffers.

Additional buffers allocated through this procedure can be used as overflow buffers for periods with high system activity. Typically, such periods do not last long. The previous example calculation allows for short periods (1 minutes or 2 minutes) where three to four times the normal amount of SMF records must be captured.

In the previous examples, it is assumed that the default values for *Interval*, and *AverageInterval* are used. The main criteria for determining these parameters are the reporting requirements. For most installations, an alert response time of about 1 minute seems appropriate. It is also well in the normal response time of people to e-mails, or other methods of alert delivery. For the *AverageInterval*, the use of a 5-minute interval is sufficiently long to avoid excessive false alarms, It is also short enough to detect most situations for which alerts are wanted.

You can use the following values as starting values for these OPTION and REPORT parameters:

#### **Bufsize**

1024 (=1 MB) or 2048 for ACF2

This is based on the average length of an RACF or ACF2 SMF-record, the following specified interval, and an average of 40 RACF or ACF2 SMF-records per second during periods of high activity.

#### **NumBufs**

12

This is based on the long-term threshold time-period (*AverageInterval*) and the *Interval* period. It also allows for an additional six overflow buffers.

#### **Interval**

60 Seconds

#### **AverageInterval**

300 Seconds

During initial execution of zSecure Alert, monitor the in-memory buffer usage, using the DEBUG BUFFER command. This results in three messages at the end of each *Interval* period. The C2P0325 and C2P0326 messages indicate how much buffer space was used for SMF-records and WTO-messages. The total amount of space for the SMF-records and WTO-records must approximately match the expected space as calculated in step 4. In step 5, the buffer size was specified at five times the average expected space required. So, the buffers are expected to be used for only about 20 percent. It leaves ample space for fluctuations in system activity.

Using the same numbers as used in the previous example calculation, you might expect these messages:

```
C2P0333I Buffer index is 09
C2P0325I Buffer stats: SMF(cnt,len) 00000214-00131928
C2P0326I Buffer stats: WTO(cnt,len) 00000000-00000000
```

The messages confirm that your expected record rate was about right, that is, 214 records versus the expected 231, and that the average size of the records was also in the right order of magnitude, that is, 131,928 versus the expected 132,644.

When activating buffer debug messages, zSecure Alert also generates a message whenever there is a need for an overflow buffer. See the following message example:

```
C2P0334I Extended buffer used
C2P0333I Buffer index is 02
C2P0325I Buffer stats: SMF(cnt,len) 00002728-01037650
```

```
C2P0326I Buffer stats: WTO(cnt,len) 00000000-00000000
C2P0333I Buffer index is 03
C2P0325I Buffer stats: SMF(cnt,len) 00000014-00307855
C2P0326I Buffer stats: WTO(cnt,len) 00000000-00000000
```

These messages are issued in addition to the regular buffer usage messages. The indicated buffer '02' is the previous buffer that was overflowing into the subsequent buffer ('03'), which is shown in the regular C2P0325 and C2P0326 messages that follow.

Using the steps previously outlined, you are able to select a minimum buffer size and number of buffers that fits your needs, without using excessive system resources. The method starts with small buffers that can be increased when needed. An alternative approach is to start with many large buffers, and monitoring the buffer statistics messages. After a few tests, you can decide by which amount the buffer size must be reduced.

When allocating buffers, you must also consider the MEMLIMIT specified in the zSecure Alert started task JCL. The default MEMLIMIT value provided in the C2POLICE member in SCKRPROC is 8GB. This value must be at least 64MB larger than the total buffer space specified through *bufsize* and *numbufs*.

## Other commands

---

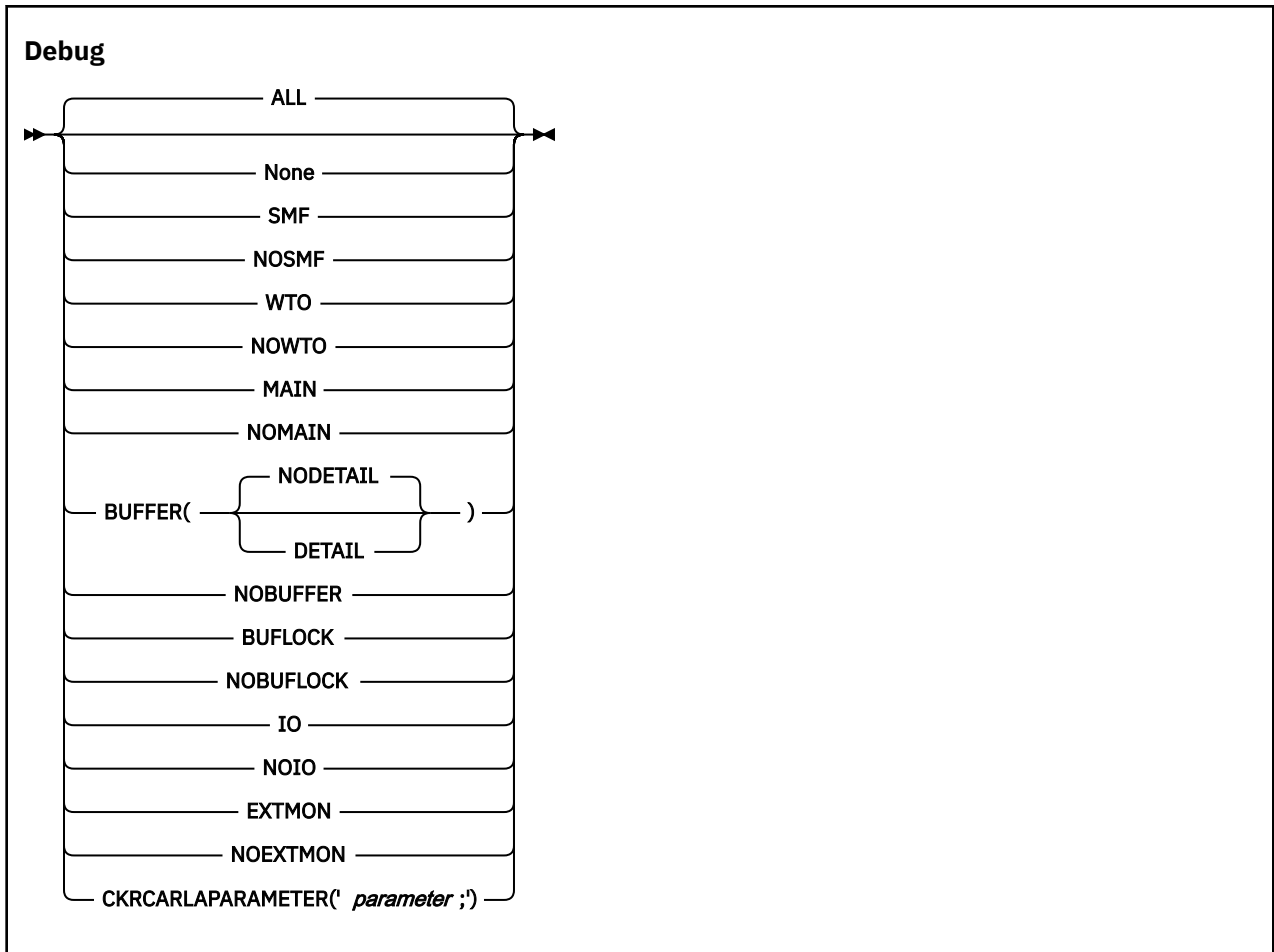
The following commands are not normally required. The DEBUG command enables you to obtain diagnostic information. You can enter these commands in the C2PXPARM member in the Alert configuration data set; see [“Required data sets” on page 111](#).

The other commands are normally generated by the interface; see the information about configuration in the *IBM Security zSecure Alert: User Reference Manual*.

## DEBUG command

The DEBUG command has the following syntax.

**Note:** Only one option can be specified. If you want to receive all messages except those messages related to WTO processing, you must issue two DEBUG commands (DEBUG ALL, followed by DEBUG NOWTO). The DEBUG command is valid both from PARMLIB and from the operator console.



The keywords and variables have the following values:

**All**

This default level specifies that all diagnostic messages must be written to the console. Most of these messages are intended to assist during problem determination, and are not intended for routine customer usage. Use the messages resulting from DEBUG BUFFER routinely to determine minimum size for the data buffers.

**None**

Deactivates creation of all diagnostic messages.

**SMF**

Diagnostic messages related to processing SMF records are written to the console.

**NOSMF**

Diagnostic messages related to processing SMF records are not written to the console.

**WTO**

Diagnostic messages related to processing WTO messages are written to the console.

**NOWTO**

Diagnostic messages related to processing WTO messages are not written to the console.

**MAIN**

Diagnostic messages related to mainline processing are written to the console. It includes responding to operator commands, initialization and management of all subtasks, and major buffer management functions.

**NOMAIN**

Diagnostic messages related to mainline processing are not written to the console. It includes responding to operator commands, initialization and management of all subtasks, and major buffer management functions.



## **BUFFER**

Buffer usage statistics are written to the console, joblog, and syslog at the end of each reporting interval. These messages can be used to determine the number of SMF records and WTO messages captured, and the amount of storage required for each. You can use these messages to track the minimum and maximum amount of buffer storage needed.

If you include the **DETAIL** keyword, counts for the SMF records and subtypes are included. The following example illustrates resulting output:

```
+-----+
| C2P0333I Buffer stats for buffer 08
| C2P0325I Buffer stats: SMF(cnt,len) 00000256-00276854
| C2P0544I Rectype Subtype Count
| C2P0544I      14          2
| C2P0544I      15          1
| C2P0544I      30 total    4
| C2P0544I          2      4
| C2P0544I      42          6
| C2P0544I      80 total    3
| C2P0544I          2      1
| C2P0544I          38      1
| C2P0544I          39      1
| C2P0544I      99        234
| C2P0544I     100         4
| C2P0544I     102         2
+-----+
```

## **NOBUFFER**

Buffer usage statistics are not written to the console.

## **BUFLOCK**

This debug option is intended to assist in diagnosing the reason that a task could not save a record in the C2POLICE buffers. If such a situation occurs, an SVC dump is created of the address space where the event occurred. The **BUFLOCK** option is automatically disabled until the **DEBUG** command is issued again, either through an operator command or from **PARMLIB**.

Note that the SVC dump is not an indication that any error occurred. The dump is created only to assist in determining why the task could not save the record.

## **NOBUFLOCK**

The **BUFLOCK** debug option is not used.

## **IO**

Specifies that all operations processed by the zSecure Alert data analysis I/O routine must be traced through **SYSLOG**. It might result in large numbers of WTO messages. This function is intended to be used by IBM support personnel to assist in diagnosing internal problems in the product.

## **NOIO**

I/O diagnostic messages are not to be generated.

## **EXTMON**

Diagnostic messages pertaining to Extended Monitoring alert processing are to be written to the operator console.

## **NOEXTMON**

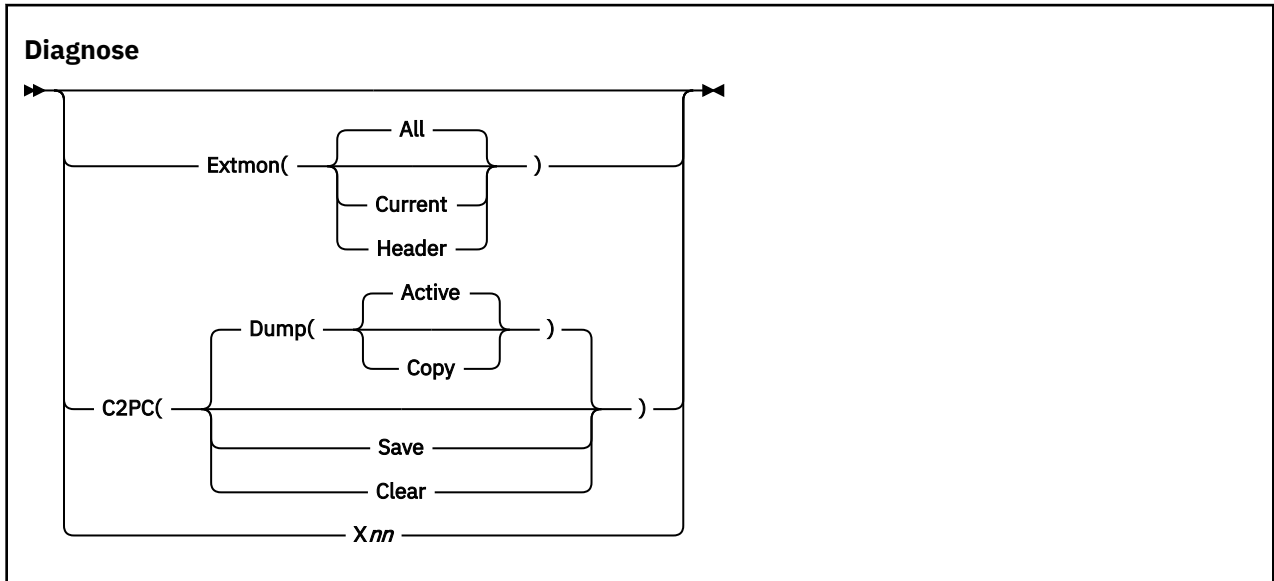
Diagnostic messages pertaining to Extended Monitoring alert processing are not to be written to the operator console.

## **CKRCARLAPARAMETER**

Specifies a string that is to be passed to all instances of **CKRCARLA** that are used within the **C2POLICE** started task. The string as specified must end with a semicolon, and must be enclosed in quotation marks. This parameter is intended for IBM Software Support personnel to diagnose problems. The maximum length of the string is 63 characters.

## DIAGNOSE command

The DIAGNOSE command causes the dumping of some internal control blocks and tables for problem determination. The control blocks displayed are intended for IBM support personnel to diagnose problems. The following diagram shows the syntax of the DIAGNOSE command.



The keywords and parameters have the following values:

### Extmon

Specifies that status information for Extended Monitoring snapshot data sets is to be displayed on the operator console. The available suboptions are:

#### All

The names and status of all CKFREEZE snapshot data sets is displayed, The status information has the following layout:

```
LCB..CED
L    The data set is listed in the system catalog
C    This is the CURRENT snapshot data set
B    This is the BASE snapshot data set
.    Reserved
.    Reserved
C    The snapshot data set is being created
E    This is an expired snapshot data set
D    This snapshot data set has been deleted
```

#### Current

The names and status of the Current and Base snapshot data sets are displayed.

#### Header

The header information from the internal CKFT control block is displayed on the operator console in dump format. This information is intended for IBM support personnel only.

### C2PC

Information from the internal C2PC control block is to be saved or displayed on the operator console. This information is intended for IBM support personnel only. The following suboptions are available:

#### Dump

The Active or saved Copy of the C2PC data area is displayed on the operator console in dump format.

#### Save

The active C2PC data area is saved in the C2PC copy area.

#### Clear

The saved copy of the C2PC data area is cleared (reset to binary zeros).



## Bufsize/BufsizeMB

The Bufsize/BufsizeMB keyword can be specified only when the OPTION statement is used during startup or during RESTART processing. It is ignored during REFRESH processing.

Bufsize/BufsizeMB specifies the size of the in-memory buffers used for storing the SMF records and WTO messages during the *interval* period. Make sure that the buffer is large enough to contain all SMF records and WTO messages collected during that period. If the buffer is too small, zSecure Alert attempts to switch to an unused buffer. If no unused buffer is available, the buffer containing the oldest history data is used instead. If this new buffer is not available, a buffer overflow message is issued. Records of the current reporting interval are lost.

If you use the Bufsize keyword, specify the required buffer size in kilobytes. If you use the BufsizeMB keyword, specify the size in megabytes. Valid sizes for the buffers are between 1 kilobyte and 1 gigabyte. The size you specify is rounded up to the nearest megabyte. If you use both keywords in an OPTION statement, the last value specified is used by the program. The buffers are allocated in 64-bit storage and count towards the specified MEMLIMIT of the started task. Use of overflow buffers, also called extended buffering, can significantly reduce the required buffer size. See [“Configuration guidelines and performance implications” on page 118](#) for guidelines for selecting an appropriate buffer size for your installation. In general, it is more efficient to specify, for example, 10 buffers of 1 megabyte instead of 2 buffers of 5 megabytes.

## Numbufs

The Numbufs keyword can be specified only when the OPTION statement is used during startup or during RESTART processing. It is ignored during REFRESH processing.

Numbufs specifies the number of buffers allocated. The value *numbufs* must be between 2 and 32 inclusive. The total number of buffers must be sufficient to hold all captured SMF-records and WTO-messages, as required for the reporting specifications.

```
Numbufs > (AverageInterval / Interval) + 1
```

Specifying more buffers than the minimum enables their use for overflow purposes. This way you can reduce the bufsize and save all data collected during high-activity periods. If no overflow buffers are available, use the oldest history buffer instead. It results in losing some data required for long-term threshold analysis. See [“Configuration guidelines and performance implications” on page 118](#) for guidelines on selecting an appropriate number of buffers for your installation.

## CollectTime

Specifies the time of day that the zSecure Collect started task (C2PCOLL) must be started. The time must be specified in 24 hour format as four consecutive digits, that is, HHMM. For example, 1 AM must be specified as 0100, while 1 PM must be specified as 1300.

Optionally, the specified time can be followed by a stagger time. The stagger time is used together with the index of the current system in the sysplex to determine extra time before starting the C2PCOLL started task. This enables automatically running the C2PCOLL tasks at different times on the members of the SYSPLEX. The same shared zSecure Alert configuration can be used while exploiting system-specific starting times of C2PCOLL. When specified, the stagger time must use the HHMM format. The stagger time must be between 0000 (none) and 0259 (an offset of almost three hours per system).

The collect time is specified between 0001, that is, 1 minute after midnight, and 2359, that is, 1 minute before midnight. The time value 0000 signifies that the zSecure Collect STC is not automatically started.

If the OPTION statement is processed as part of startup or RESTART processing, the specified CollectTime value is effective immediately. If a changed value is processed during REFRESH processing, it will not take effect until after the next zSecure Collect started task run.

## CollectSTCName

Specifies the name of the started task (STC) in the system proclib. It can be used to generate an internal START command of the form

```
START name.name
```

Before using this feature, ensure that the procedure exists, that the correct userid and group are assigned to the started task, and that the started task has sufficient authorization to execute the zSecure Collect functions.

The zSecure Alert started task needs sufficient authorization for the start command. Follow the steps described in [“Prerequisites for configuring and using zSecure Alert” on page 105](#) to define the necessary profiles.

### **RefreshMode**

Specifies whether the environment refresh involves a complete stop and start of the CKRCARLA subtask, or whether it exploits the CKRCARLA internal restart function. During an external refresh (default), the CKRCARLA subtask is stopped and the SMF job information is lost. The internal restart mode has the advantage that SMF job information, that is retained to complete other SMF records, is kept for a longer period. Information that is lacking in certain SMF records (for example, the user ID in record type 15) is saved from previous SMF records; this information is available in the applicable CARLa fields.

The saved job information is kept until the C2POLICE started task itself is restarted or stopped. The amount of retained data can be significant if the started task is kept running for long periods. You must ensure that sufficient storage above the 2GB boundary is available. The example C2POLICE started task procedure in SCKRPROC specifies MEMLIMIT=8G. One gigabyte of storage is sufficient to retain information for approximately 8 million jobs.

### **SETSRVCLASS, NOSERSVCLASS**

Determines whether the started task runs within the SYSSTC service class. The SYSSTC service class is intended for high priority system tasks. The collection of events and saving the collected data should be done in a timely manner. Therefore, a sufficiently high priority setting is important. If you must assign a lower dispatching priority, create an applicable WLM classification rule that assigns the required WLM service class. You must also specify NoSetSRVClass on the OPTION statement to prevent the program from resetting itself to the SYSSTC service class.

If you are running on a single processor system, careful analysis of your workload is required and using NoSetSRVClass might be preferred.

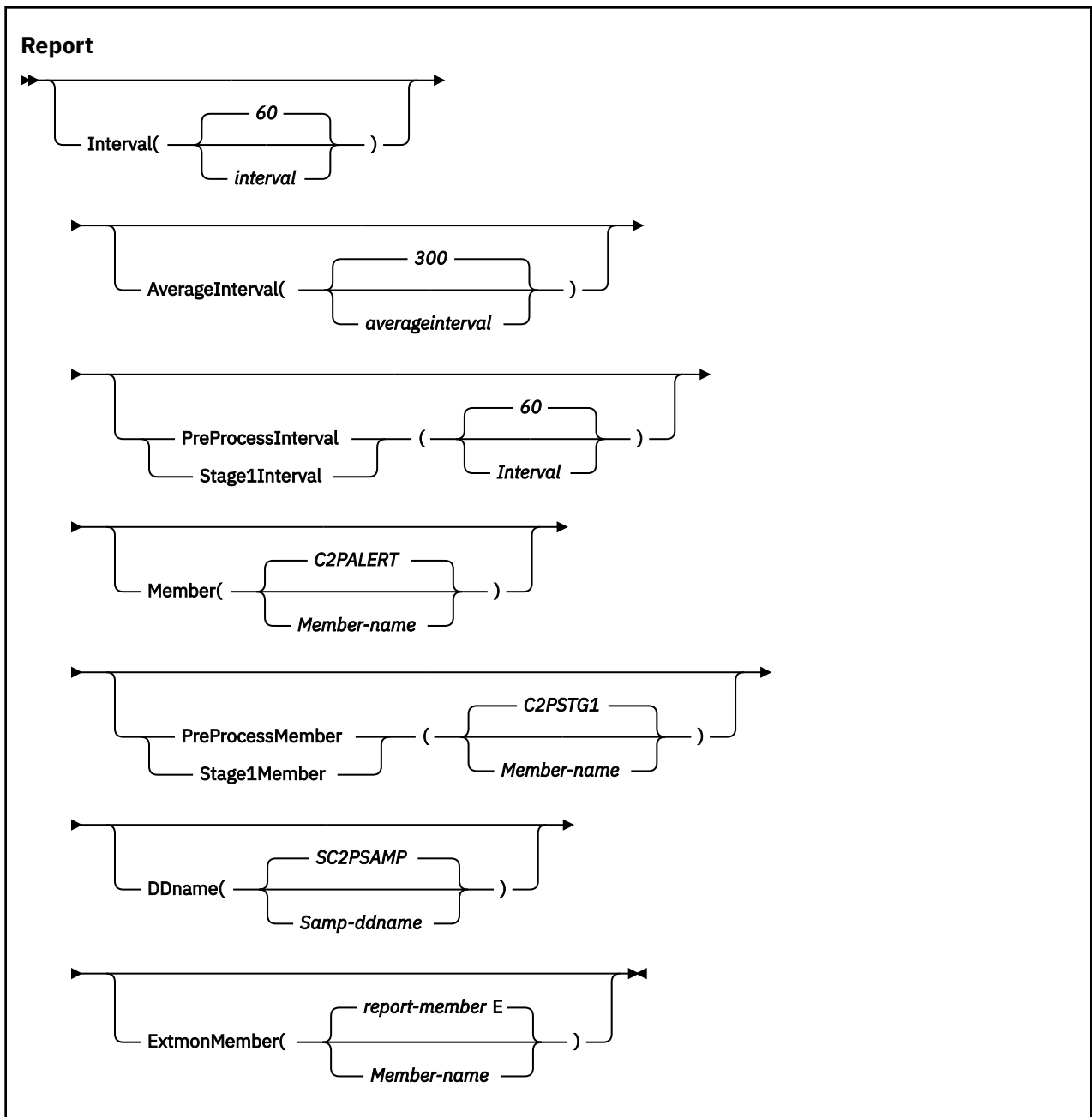
The default value of this option is SETSRVCLASS.

### **EXTMON**

Specifies that the Extended Monitoring process is to be used. It requires that the person who installed and configured the zSecure Alert software completes several configuration steps. These steps are described in [“Post-installation tasks” on page 109](#). The first subparameter specifies whether the process is ACTIVE or INACTIVE. The second subparameter specifies the retention period of the CKFREEZE snapshot data sets. CKFREEZE snapshot data sets that are older than the specified retention period are automatically deleted if the Extended Monitoring process is active. The default value for the RETAIN parameter is 24 hours.

## **REPORT command**

The REPORT command controls the timing of the reports and the source for the CARLa statements used for pre-processing environment information and report generation. The effects of the REPORT command might be delayed due to the cyclic nature of various tasks in zSecure Alert. For instance, a modified value for the *Interval* will only be used after expiration of the current interval. The REPORT command has the following syntax:



The keywords and variables have the following values:

**Interval**

Specifies the interval at which zSecure Alert analyzes the collected data and generate appropriate alerts. The value *interval* specifies the time interval in seconds. Valid time intervals are 10 - 3600 seconds. The default value is 60 seconds.

**AverageInterval**

Specifies the time over which zSecure Alert averages the occurrence of certain events for *moving window* analysis. This time is also called the history period. Generally, this period would be five times as long as *interval*. The numbufs parameter must be sufficiently large to capture all data for the *AverageInterval* period.

$$\text{Numbufs} > (\text{AverageInterval} / \text{Interval}) + 1$$

The value *AverageInterval* specifies the time in seconds. Valid time averaging periods are 10 - 9999 seconds. The default value is 300 seconds.

### PreProcessInterval or Stage1Interval

Specifies the interval at which zSecure Alert processes the information from the security database and the CKFREEZE file. The result of this processing is used as selection criteria for the regular record analysis. Because this process does not result in direct alert generation, but is only used as input for subsequent steps, it is called the STAGE1 CARLa process. To pick up the latest selection criteria, the reporting task will be refreshed after completion of the STAGE1 process. For the period that the STAGE1 process is active, operator REFRESH and COLLECT commands are postponed until the end of the process. The *Stage1Interval* must be specified in minutes, with valid values 10 - 1440. The default value is 60 minutes.

The best value for *Stage1Interval* is dependent on the frequency of updates to your system and to your security database.

### Member

Specifies the membername in the partitioned data set that is used for the data analysis. It contains the CARLa statements that generate the appropriate alerts, as specified for your installation.

### PreProcessMember or Stage1Member

Specifies the membername in the partitioned data set that is used for processing the security database and CKFREEZE file. It contains the CARLa statements that result in selection criteria used during the alert generation process. The output of the STAGE1 process must be explicitly included by the alert generation process.

### DDName

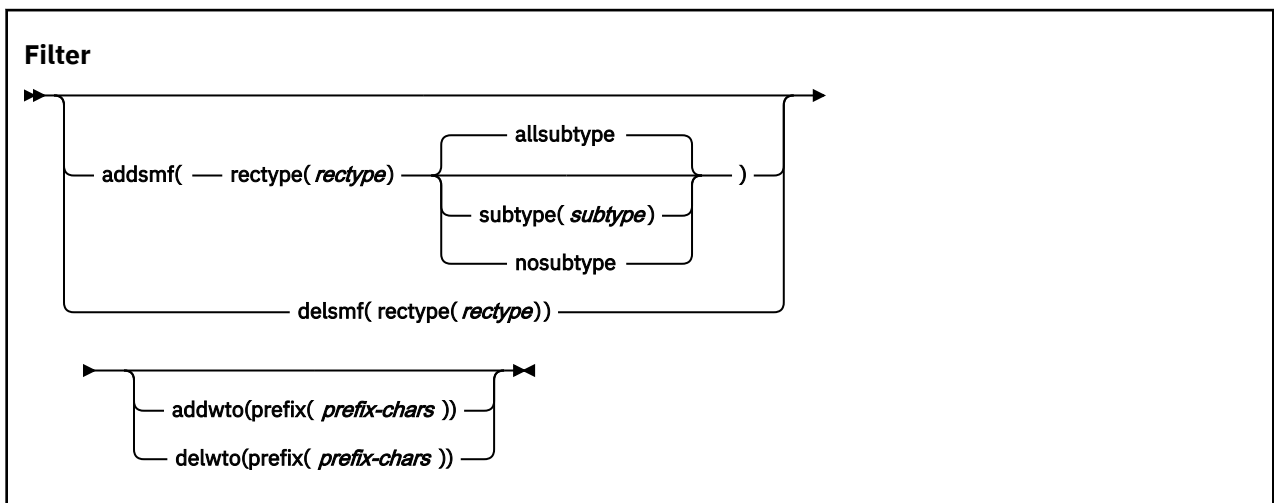
Specifies the JCL DD-name pointing to the partitioned data set containing CARLa statements used by zSecure Admin and Audit. It must contain at least the members indicated by *member* and *Stage1Member*.

### ExtmonMember

Specifies the member-name in the partitioned data set that is used for the Extended Monitoring alerts. This member contains the CARLa statements that are used to analyze the CKFREEZE snapshot data sets and create appropriate alerts. If the ExtmonMember option is not specified, or if no member-name is specified, a default member name is used. The default member name is built from the member name specified for the MEMBER keyword followed by the letter "E".

## FILTER command

The filter criteria are used to limit the amount of data collected in the in-memory buffers for further processing. By using the FILTER command, it is possible to eliminate unused events early in the process, thus increasing the overall efficiency. If there are no SMF and WTO filter criteria specified, all SMF records and WTO messages are collected for further processing. To avoid this situation, the zSecure Alert User Interface will generate dummy filters, that do not match any event. The FILTER command has the following syntax:



The following section describes the possible keywords and parameters.

## **ADDSMF**

Specifies the additional filter criterion to be used for SMF-records. You can repeat the FILTER command to specify as many filter criteria as you need. The criterion you specify is added to the already active criteria. The SMF-record type to be selected is specified by the *rectype* and *subtype* parameters. The available suboptions are:

### **Allsubtype**

Specifies that all SMF-record subtypes are included in the record filter (default). This specification can also be interpreted as the absence of any filtering on subtype. Subtypes are used for only SMF-record types 30, 80, 92, and ACF2. For all other SMF-record types, the subtype specification is ignored.

### **Rectype**

Specifies the SMF-record type that must be selected or that must no longer be selected. The *rectype* parameter must have a numeric value 0 - 2047, or the value ACF2 to specify records generated by ACF2.

### **Subtype**

Specifies the SMF-record subtype that must be selected. The *subtype* is only used for SMF-record types 30, 80, 92, and ACF2. For all other SMF-record types, the subtype is ignored. The value of *subtype* must be numeric or a single alphabetic character. The subtype is interpreted as follows:

#### **Rectype 30**

The *subtype* is the standard SMF-record subtype. Although currently SMF-Record type 30 only has defined subtypes 1 to 5, the range accepted by zSecure Alert is 1 - 8.

#### **Rectype 80**

The *subtype* is the RACF event code. For a complete list of RACF event codes, see RACF Auditor's guide. The range of values accepted by zSecure Alert is 1 - 255.

#### **Rectype 92**

The *subtype* is the standard SMF-record subtype. Although SMF-Record type 92 currently has defined only subtypes 1 - 17, the range accepted by zSecure Alert is 1 - 255.

#### **Rectype ACF2**

The *subtype* is the ACF2 record type. For a complete list of ACF2 subtypes, see the "SELECT/ LIST Fields" chapter in the *CARLa Command Reference*; see the ACF2\_SUBTYPE field in NEWLIST TYPE=SMF.

### **Nosubtype**

Specifies that the SMF-record subtype, as described previously for the Subtype keyword, must not be used as a selection criterion. Use of this keyword resets all subtypes previously specified for the indicated *rectype*.

## **DELSMF**

Specifies that you no longer want the specified SMF-record type to be selected. The SMF-record type is identified by the *rectype* parameter only. It is not possible to deactivate SMF-record selection per subtype.

## **ADDWTO**

Specifies the filter criteria used for the WTO-messages. You can specify up to 24 different filter criteria. Although you can specify message prefixes starting with C2P, most of the C2P messages are not captured. Only messages C2P0100, C2P0335, and the range C2P0900 to C2P0999 can be captured and used to trigger alerts.

## **DELWTO**

Specifies that you no longer want WTO message selection to occur for messages starting with *prefix-chars*.

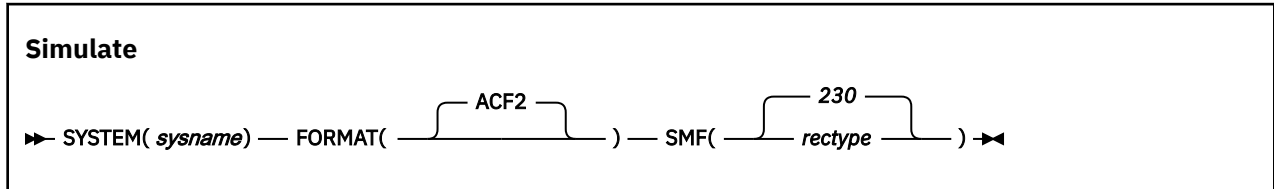
## **Prefix**

Specifies the first characters of the WTO message identifier. If you want to include all ICH messages, simply specify ICH. If you only want to include ICH408I messages, specify the full seven (7) characters of the message identifier. The maximum length of the message prefix is eight (8) characters. The minimum length is one (1) character.



## SIMULATE command

For normal operations, the SIMULATE command is not required because zSecure Alert uses the documented interface to obtain the SMF-record type used by ACF2. Only when this process fails, the SIMULATE command is required. The command has various keywords and required parameters that are currently not used by zSecure Alert. These keywords and parameters are included for consistency with the zSecure Admin and Audit syntax of the SIMULATE command. They can be used in a future version of zSecure Alert. The SIMULATE command has the following syntax:



The following section describes the possible keywords and parameters.

### System

Specifies the system name to which this SIMULATE command applies. Currently, the value for *sysname* is ignored. You must specify the SMF\_ID of the current system.

### Format

The only supported parameter ACF2 indicates that this SIMULATE command is used to specify ACF2 specific options.

### SMF

Specifies the SMF-record type for the ACF2 generated SMF-records. The parameter *rectype* must be numeric with a value 1 - 255. The default value is 230.

## Coexistence considerations

For migration purposes, the zSecure Alert configuration data set can be shared between z/OS images with different releases of zSecure Alert. However, share the configuration data set between different releases for only a limited amount of time, because new alerts and new functions are not available until all sharing systems have been upgraded. If you want to use new alerts and new functions, but do not want to upgrade all systems at the same time, temporarily break the sharing and assign different configuration data sets.

If you share the configuration data set, configure zSecure Alert only by using the lowest level ISPF interface in use. After a configuration data set has been upgraded, you can no longer make changes from the lower-level interface unless you back out the upgrade. Moreover, the lower-level zSecure Alert address space might or might not work correctly with a configuration that was created or maintained using the new ISPF interface.

Upgrading is supported from lower release versions that are still supported when the newer release is issued.

## Upgrade of zSecure Alert

When you use a higher level of the zSecure Alert configuration interface than the one used for the alert configurations present in your configuration data set, the following panel is displayed:

```

zSecure - Setup - Alert

UPGRADE process about to start for C2R.IP01.C2PCUST
Warning: The current zSecure Alert data set was created using downlevel
panels. There might still be a downlevel zSecure Alert using it. After
upgrade, a downlevel zSecure Alert can no longer use this data set and
customization will only be possible from the zSecure Alert release 1.12.0
(or newer) User Interface.

The following downlevel table has been found:
User interface level : 1.11.0
Table name           : C2PIUACD
Creation date        : 2011/03/08
Last change date     : 2012/05/10
Last change time     : 08:49:01
Last changed by      : ALERTU1

Select upgrade option
3 1. Upgrade from downlevel table
   2. Create new table
   3. Cancel upgrade process.

```

Figure 6. Setup Alert panel: Upgrading zSecure Alert

You can choose the following upgrade options:

### 1 - Upgrade from downlevel table

The Alert configurations are stored in ISPF tables. Choose this option if you decide to maintain your old configuration with the alerts selected and destinations for each alert. The old configuration tables will be converted to new format tables. If one of the configuration steps requires additional information, it can be set to the status of Req instead of the desired state of OK. In this case, you must provide this information by using the corresponding action command. Configuration step Ver, which means verify your configuration, can always be set to Req, which means "required," to refresh the alert code. After selecting this option, you can no longer configure zSecure Alert with a lower-level ISPF interface.

### 2 - Create new table

Select this option when you do not want to keep your old configuration. A clean configuration is in use and you must perform all configuration steps; that means all configuration steps have status Req. As with option 1, you can no longer configure zSecure Alert with a lower-level ISPF interface.

### 3 - Cancel upgrade process

Select this option when you have not yet upgraded all systems that share the configuration data set to the current software level.

## Backout of an upgrade

zSecure Alert checks for the presence of higher-release tables. When you try to configure zSecure Alert with a lower-level ISPF interface than the level of your configuration, the following panel is displayed:

```

zSecure - Setup - Alert          Row 1 to 1 of 1
Command ==> _____ Scroll ==> CSR

The current zSecure Alert data set is shared with a higher level User
Interface. The following uplevel table(s) are found. You should configure
zSecure Alert from the highest User Interface level, or delete the higher
level table(s) by using the D action command.
Warning: Deleting the higher level table(s) results in the loss of all
customization performed from the higher level User Interface!
-----
   Level Table          Created   Changed   ID
   _  1.7.0 C2PIUACC      2005/05/08 2005/05/10 11:32:50 ALERTU1
***** Bottom of data *****

```

Figure 7. Configuring the correct ISPF interface level

Use the **D** (delete) action command only when you want to fall back to the current level. After backing out, rerun Verify and Refresh under the old ISPF interface to make the backout effective for the zSecure Alert address space.

---

# Chapter 14. Setup and use of the zSecure Visual Server

Using the zSecure Visual Server establishes a secure connection directly with RACF. You can then use the zSecure Visual client, a Windows-based graphical user interface, for decentralized RACF administration from the Windows environment.

Use the information in the following sections to install, configure, and use the Visual Server.

---

## Setup of the Visual Server

The following sections provide information about the prerequisites and procedures for installing the zSecure Visual Server:

- [“Installation requirements” on page 133](#)
- [“Required system authorizations” on page 134](#)
- [“Owners, directories, and file systems preparation” on page 134](#)
- [“zSecure configuration for zSecure Visual” on page 135](#)
- [“zSecure Visual Server software” on page 136](#)
- [“Setup of a new zSecure Visual Server” on page 136](#)

### Installation requirements

zSecure Visual is one of the CARLa-driven components in the zSecure product family. For all CARLa-driven components, SMP/E installation is done concurrently. All CARLa-driven components use the zSecure configuration.

Before configuring or using zSecure Visual, you must complete the basic installation process documented in Chapter 4, “Installation of the software,” on page 9 and the *Program Directory: IBM Security zSecure CARLa-Driven Components*. Be sure to perform the following tasks during installation:

- Create and customize a library with the low-level qualifier CKRINST. You can find the setup jobs for Visual [here](#).
- The CKGRACF component must run APF-authorized. See [“APF authorization of the software” on page 18](#).
- Establish the CKGRACF daily job. See [“Requirements and rationale for running the daily CKGRACF job” on page 42](#).
- Ensure that both the CKGRACF and CKRCARLA programs are program-controlled. See [“Setting up Program Control and PADS access” on page 202](#) for more information.
- Ensure that support modules from libraries such as *hlq.SCEERUN* and *hlq.SCEERUN2* (where *hlq* is CEE by default) are program-controlled.

**Note:** Normally, these data sets are in the linklist and are members of the profile \* or \*\* in the PROGRAM class. You must verify the data sets that apply to your system.

- Enable the zSecure configuration for zSecure Visual. See [“zSecure configuration for zSecure Visual” on page 135](#) for instructions.
- You can run multiple instances of the server, and the instances can run different releases. See [“Upgrading an existing Server to zSecure Visual 2.5.0” on page 139](#). However, within each server instance, all of the following components must be at the same level:
  - JCL
  - REXX

- CARLa library
- Load modules
- The USS code that is extracted from the SCKRPAX library

Using different levels of these components is not supported.

## Required system authorizations

Before using zSecure Visual, you must perform the following tasks:

- Establish authorization to set up the required users, groups, directories, and file systems. For instructions, see [“Owners, directories, and file systems preparation” on page 134](#).
- Set up READ access to these FACILITY resources for the user who runs the Server Setup:
  - BPX.FILEATTR.APF
  - BPX.FILEATTR.PROGCTL
- Establish authorization to create and mount the dedicated file system needed for the zSecure Visual data. The file system can be HFS or zFS.
- Establish authorization to create entities in your system for job scheduling or automated operations, or for both, for the purpose of the production process.
- Set up UPDATE access to one of the procedure libraries of your Job Entry Subsystem and authority to set up STARTED profiles. These authorizations are required to set up the started task for the server. Alternatively, you can run the Server as a batch job. If you run the Server as a batch job, you must have the appropriate SURROGAT authority.
- Select and allow an available set of IP ports for each server. See [“TCP/IP Security” on page 138](#) for more information.
- Provide READ access on the C2R.SERVER.ADMIN resource to the users who add new workstations to the server (in the XFACILIT class, unless your installation has customized this. For more information, see [Appendix A, “Site module,” on page 191](#)). Job C2RZWADM uses group MYGROUP. See [“Setup of the server processes” on page 137](#). These users must also have a valid z/OS UNIX System Services home directory, unless the BPX.DEFAULT.USER has a valid home directory. Provide each user with a unique home directory so that they cannot read the generated install passwords of other users. Also, the default connect GROUP profile for these users must have an OMVS segment with a valid GID.
- The RACF userids of the RACF administrators who will use the Visual client, but who will not add new workstations to the server, also require a UID and GID. However, it is not necessary to define separate OMVS segments for these users. Instead, you can exploit BPX.DEFAULT or BPX.UNIQUE, depending on your z/OS system. Also, these users do not need home directories of their own.
- All users of the zSecure Visual client need READ access on the data set that is identified by the C2RWCUST DD statement in the server JCL, and to the data sets that are identified by the C2RWASSC member in that data set.

## Owners, directories, and file systems preparation

When the number of concurrently active clients exceeds the limit for a single server, you need multiple servers. Multiple servers can run in separate z/OS images or within a single z/OS image, sharing the file system where the software resides.

If you run multiple instances of the zSecure Visual Server, the servers can share the directory where the software resides, but each server instance must have its own instance-related data (that is, subdirectories run and log). As a result, an initial password (a one-time usable shared secret) that you generate for a client is valid only for that specific server instance. After the initial connect, the certificates are also valid only for that same server instance. Therefore, clients must access a server by an IP-address or DNS name that is always associated with the same server instance.

Using separate directories and file systems also eases your future upgrade of z/OS and the zSecure Visual software because you can upgrade the system and reinstall the zSecure Visual software while ensuring all data in place.

**Note:** Up to 315 clients can be connected to a server at the same time. To accommodate this many clients, the maximum number of open file descriptors allowed per process for the zSecure Visual Server userid must be raised to 1594 or more.

Because each file or directory in UNIX must have both an owning user and an owning group, you must assign owners. The following defaults are used in this documentation and the IBM-supplied jobs. You can adapt these defaults to fit the conventions of your installation:

<i>Table 8. Directory and file owner user and group id naming conventions</i>		
	<b>Software</b>	<b>Data</b>
Owning User	C2RUSER	C2RSERVE
Owning Group	C2RGROUP	C2RSERVG
Directory	/usr/lpp/c2r/V2R5M0	/u/c2rserve/server1
Mount point	/usr/lpp/c2r	/u/c2rserve
File system	OMVS.C2R.ZFS	OMVS.C2RSERVE.ZFS

As shown in [Table 8 on page 135](#), the data is owned by C2RSERVE, the default user under which the Server runs. The Server does not, however, own the files containing the software. Instead, the Server is granted READ and EXECUTE access to the software through the group permission bits and through a CONNECT to the group that owns the software files. The same access is required for people who are to use the client. Similarly, both the server and the users of the clients need READ access to the OS data sets where zSecure resides. The default high-level qualifier of these data sets is CKR.

In the same way, you can connect your security support and production control personnel to C2RSERVG to grant them access to the Server-owned data files because they might need to view log files.

**Note:** The default mount points in [Table 8 on page 135](#) do not coincide with the software and data directories. This configuration allows you to set up multiple Servers under a single userid. Similarly, you can install future software releases within a single file system. If you want separate file systems for each release or for each Server, you can use the IBM-supplied jobs as templates and run them multiple times.

Automount is commonly done for file systems that are used for home directories, such as /u/c2rserve, but typically not for software. For file systems that are not automounted, update the BPXPRMxx member in your parmlib to ensure that the file systems are mounted after subsequent IPLs.

## **zSecure configuration for zSecure Visual**

The default zSecure configuration is C2R\$PARM. You can have other configurations. The following zSecure configuration parameters are specifically for zSecure Visual:

- C2RWCUST
- C2RW131A
- C2RWIN
- C2RSERVE

**Note:** The FIPS 140-2 cryptography standard was replaced by the NIST 800-131A standard. As a result, the C2RWFIPS configuration parameter was replaced by the C2RW131A parameter.

Make sure that the data set that is identified by C2RWCUST contains all members that are required for the current release. For a new configuration, empty members are created by job CKRZPOST, but CKRZPOST makes no updates to a configuration that you might already have put effort into.

See [Appendix D, “Configuration parameters and members,” on page 205](#) for descriptions of these parameters and the members of C2RWCUST.

## zSecure Visual Server software

Use the information in this topic to run the zSecure Visual Server setup programs to install the software in the mainframe environment.

### Software location

As described in [“Owners, directories, and file systems preparation” on page 134](#), the default location for the software is `/usr/lpp/c2r/V2R5M0`. However, you can choose a different location. For example, you can remove the release number, or add a maintenance level to the path name.

Before running the setup jobs, update your zSecure configuration so that the C2RWIN parameter reflects your chosen software location. Specify the updated configuration in subsequent setup jobs.

### Owner and location preparation for the software

You can use job C2RZCZFS to prepare the file system where the software is to be installed.

- You might not want a new file system; for example, you might want to use an already mounted file system from a previous installation. To use an existing file system, comment out the jobsteps that create and mount a file system. However, you must still run the other jobsteps to set up the directories.
- Because the C2RZCZFS job mounts the file system, run it as root. This mount does not persist after subsequent IPL. Ensure that the file system is mounted when needed. For example, you can include the mount in your BPXPRMxxx member.
- The file system must be mounted with the SECURITY and SETUID attributes. These attributes are required because zSecure Visual runs as a daemon, and therefore requires a program-controlled environment.

For an upgrade installation, you normally do not have to prepare a new file system. However, create a new directory into which to unpack the software. You can then start using the upgraded software by changing the C2RWIN parameter in the zSecure configuration and restarting the Server.

### Unpacking the software

- Before running the C2RZWUNP job, supply the zSecure configuration that contains your customized C2RWIN parameter.
- Job step OS2ZFS copies the (SMP/E-installed) software into a UNIX file.
- After unpacking, the pax file is no longer needed, and you can discard it.

## Setup of a new zSecure Visual Server

The following sections provide information about the processes required to set up a new zSecure Visual Server:

- [“Setup of the userid and file system” on page 137](#)
- [“Updating the zSecure configuration: the Server root” on page 137](#)
- [“Setup of the server processes” on page 137](#)
- [“TCP/IP Security” on page 138](#)
- [“First time startup of the Server” on page 138](#)
- [“Upgrading an existing Server to zSecure Visual 2.5.0” on page 139](#)

## Setup of the userid and file system

Run the C2RZWUSR job as root because it mounts the file system and must transfer ownership of the created home directory to the userid of the server. The XFER jobstep transfers the ownership.

You can run multiple servers under the same userid, provided that each server has its own ServerRoot directory. Multiple servers under a single userid can run different releases of the software.

## Updating the zSecure configuration: the Server root

As outlined in “Owners, directories, and file systems preparation” on page 134, each zSecure Visual Server must have its own directory to use as a Server root. To address the required directory, edit the zSecure configuration for each Server that you set up. Normally, you use a subdirectory of the home directory of the userid that runs the Server. For example:

- Default userid that runs the server is C2RSERVE
- Home directory of the C2RSERVE user is /u/c2rserve
- Default Server root is /u/c2rserve/server1

For each Server that you want to prepare:

1. Prepare the zSecure configuration, and then use this configuration in the subsequent jobs.
2. Run job C2RZRUT to establish the Server root.

## Setup of the server processes

To run as a started task:

- You must copy JCL-procedures C2RSERVE, C2RSTOP, and C2RSLOG to a library that is part of your JES procedure concatenation. You also must copy the zSecure configurations for all Servers to the same library, because for started tasks, no JCLLIB is available.
- Ensure that the Server process runs under the intended userid. The process that stops a server (C2RSTOP) or prints the server log files (C2RSLOG) must run under the same userid as the Server itself.

As an alternative to running as a started task, you might choose to construct batch jobs to run the C2RSERVE, C2RSTOP, and C2RSLOG procedures. For example, you might want to use your job scheduling system to start and stop the zSecure Visual Server. Or, during setup, you might want to run the first-time start as a batch job. For considerations on batch jobs versus started tasks, see [“Making the software available for batch processes” on page 21](#).

Whether you select jobs or started tasks:

- Update the CONFIG=C2R\$PARM in the EXEC or PROC statement to reflect the zSecure configuration or configurations that you prepared for your Server or Servers. For a started task, consider using a System symbol as the configuration member name or part of the configuration member name.
- Procedures C2RSTOP and C2RSLOG must refer to the same zSecure configuration (the same Server root directory) as the Server that they are to operate on.
- Make sure that you leave the TIME=NOLIMIT specification in the JCL in place. The Server starter is a short-lived process, but the Server itself runs in a forked process, for which the MAXCPUIM has no effect. The CPU time limit is inherited from the parent.

Job C2RZWADM is supplied to establish the required access for STARTED, SURROGAT, FACILITY and XFACILIT resources. See [“Required system authorizations” on page 134](#). If you customized the Site module to use a resource class other than XFACILIT, change this job accordingly. See [Appendix A, “Site module,” on page 191](#) for information about customizing the Site module.

## TCP/IP Security

The Server must have permission to use the IP stack and the selected port. You can configure a base TCP port in job C2RZWINI. In addition, the Server uses port base+1. The Server also uses a set of ephemeral ports, but you do not have to reserve these ports.

In the PORT or PORTRANGE statement in your TCP/IP configuration, specify a SAF resource. For example:

```
PORTRANGE 8000 2 TCP * NOAUTOLOG SAF VISUAL
```

This statement restricts the use of TCP ports 8000-8001 to users that have at least READ access to the EZB.PORTACCESS.sysname.tcpname.VISUAL resource in the SERVAUTH class. For sysname, the MVS system variable SYSNAME is substituted. For tcpname, the TCP/IP job name is substituted.

Instead of \*, you can fully or partially specify the jobname or jobnames that you intend to use for the server and for the first-time server-start job; for example, C2R\*. However, with SAF active, there is usually no need to impose jobname restrictions.

If you have activated protection of unreserved ports in your TCP/IP stack, you must grant permission to use these ports to the userid under which you will run the Server. For more information about protecting unreserved ports, go to the z/OS Internet library at:

[www.ibm.com/systems/z/os/zos/library/bkserv/index.html](http://www.ibm.com/systems/z/os/zos/library/bkserv/index.html)

Select the version of z/OS that you are using then select **z/OS Communications Server -> IP Configuration Reference** in the **Contents** column.

In a multi-stack (CINET) environment, the zSecure Visual Server binds only to one stack at a time. To have a predictable IP address that your clients (and the SE.W transaction) can connect with, ensure that the same stack is used after each start of the server. For example, if the stack you want to use is named ABC, you can set up stack affinity by adding the following step to the C2RSERVE job before the C2RSERVE EXEC line:

```
//STEP0 EXEC PGM=BPXTCAFF,PARM=ABC
```

The Server, including the first-time start by job C2RZWINI, also needs at least READ access to the EZB.STACKACCESS.sysname.tcpname resource in the SERVAUTH class. For sysname, the MVS system variable SYSNAME is substituted. For tcpname, the TCP/IP job name is substituted.

## First time startup of the Server

Job C2RZWINI is supplied to start the Server for the first time and to establish a Certificate Authority. This job must run under the userid of the Server, and on the z/OS image where you intend to run the server. In a multi-system environment, either sysplex or more traditional multi-access spool, you might need to specify system affinity to ensure that the job runs on the correct system image.

Alternatively, you can run job C2RZWINI under a different userid, but in that case, after stopping the server, run job C2RZWXFR.

**Attention:** Never run job C2RZWINI when upgrading an already-established Server. Doing so invalidates all previously issued certificates.

After a while, you can see the following line in the <Server-root>/log/server.log) file:

```
P399M194V0.2.67L269A4S0E80:LCM: Initial certification completed successfully
```

After the Certificate Authority is established, run job C2RSLOG to print the server logs, and archive the output. IBM Software Support might request this output in case of problems.

Job C2RZWINI performs an initial start of the Server, but it does not terminate the Server although the job C2RZWINI itself terminates. For normal starting and stopping, use procedures C2RSERVE and C2RSTOP respectively, as described in [“zSecure Visual Server operations” on page 152](#).



## Upgrading an existing Server to zSecure Visual 2.5.0

### About this task

For any upgrade, you must make sure that your local copies of the JCL (for example, in your JES procedure library when running the Visual Server as a started task) match the level of other zSecure components. These local JCL copies include the zSecure configuration that you use for Visual Server. For a new server instance, job CKRZPOST has prepared your configuration, but this preparation is not done when upgrading, because the configuration contains your customization, which the CKRZPOST job does not overwrite.

### Procedure

To upgrade the zSecure Visual software on an existing server, follow these steps:

1. Unpack the server software into a new directory that is different from the directory where the previous level was unpacked.
2. Verify that the server userid has the required access to the new directory and the files in the directory.
3. Edit the zSecure configuration that your Server uses. Be sure that the C2RWIN parameter reflects the location of the new software.
4. Stop and then restart the server.

**Attention:** Do not run job C2RZWINI when upgrading. Doing so invalidates all previously issued certificates.

## Compatibility of IBM Security zSecure Visual and zSecure components

Use the guidelines in this topic to plan for an upgrade of IBM Security zSecure Visual.

To optimize zSecure Visual features, all related components must be the same version. For optimum performance, combine zSecure Visual client 2.5.0 with:

- z/OS V2R5
- CKRCARLA 2.5.0
- CKGRACF 2.5.0
- zSecure Visual server 2.5.0

Upgrading the zSecure Visual client does not require the client to be at the same release level as the zSecure Visual server. However, IBM does not support using previous releases of the Visual client with the current release of the Visual server. See [Table 9 on page 140](#).

First, upgrade the server to the latest release, and then begin installing the new client. Multiple instances of the server can exist while you manage the workload of upgrading all the client instances.

Multiple zSecure Visual client versions can coexist on the same workstation. For example, on a single computer you can install version 2.5.0 of the client without first removing version 2.1. In general, multiple client versions can exist concurrently on a single computer *if no port conflicts exist*:

- Configure a different local port number than the default to run multiple versions in parallel.
- Ensure that the port value for each client version corresponds to the port of the Visual server with which it communicates.

Multiple zSecure server instances with different versions are also supported if your configuration prevents port conflicts. For more information, see the *IBM Security zSecure CARLa-Driven Components: Installation and Deployment Guide*.

[Table 9 on page 140](#) lists available support based on the zSecure Visual server version.

Table 9. zSecure Visual client versions compatibility

Server Client	zSecure Visual 2.5.0	zSecure Visual 2.4.0	zSecure Visual 2.3.1	zSecure Visual 2.3.0
zSecure Visual 2.5.0	Supported	No formal support	No formal support	No formal support
zSecure Visual 2.4.0	Compatible	Supported	No formal support	No formal support
zSecure Visual 2.3.1	Compatible	Compatible	Supported	No formal support
zSecure Visual 2.3.0	Compatible	Compatible	Compatible	Supported

**Note:** *Compatible* means that new function is not supported in a down-level client.

## Making clients known to the server

To access the server, azSecure Visual client must have a local server definition and a corresponding client definition on the server. The mainframe environment provides limited support for the initial or incidental configuration of clients. After at least one client is installed and configured, use this client to further create and maintain the client definitions. See [“Authority to manage client definitions” on page 146](#) for information.

## Visual server access through ISPF

See [“Installation requirements” on page 133](#) for guidelines on setting up the zSecure configuration for zSecure Visual. For more information about zSecure configurations, see [Chapter 6, “Deployment of the software,” on page 23](#).

**Note:** The C2RWZINI job must have run against that particular server instance.

## Configuring the Visual Client

### About this task

The administrator uses the ISPF interface to configure the zSecure Visual client.

### Procedure

1. Go into IBM Security zSecure Admin on z/OS ISPF.
2. Enter **SE** (Setup), and select **W** (Windows configuration).
3. Use action **AP** and your client ID to create an initial password or password phrase.

If you use action **AP** now and lose or cancel the initial password (or phrase), or if its validity expires before the client is successfully installed, you can use action **P** to generate a new initial password (or phrase).

```

Menu          Options      Info      Commands      Setup
-----
zSecure Visual - Configuration
Command ==> _ start panel

1 1. Add, delete, or install zSecure Visual Windows client

Server . . . . . IP01 (IP or DNS)
Server base port . 8000___ (IP base port of server)

Act Agent id
AP 12.1. 100_____

Act must be A, D, P, C, AP (A=add D=delete C=cancel pwd P=new pwd)

```

Figure 8. Configuration screen for the zSecure Visual Windows Client

Your TSO session does not have to be on the system where the server is active. Consequently, you must select the server by a resolvable DNS name or IP address and port number.

If you specify an IP address, ensure that you use the same IP address that your clients will use.

**Note:** Do not use these addresses:

**Loopback address**

Do not use because every stack has its own copy of the loopback address.

**Dynamic VIPA address**

Do not use because such an address might move between stacks or even between z/OS images.

You must identify the client by its client ID. The client ID must match the ID that is used in the Server definition dialog on the client.

4. A pop-up panel prompts you to enter a user ID and corresponding password or password phrase:

```

Enter userid and password or password phrase

Userid . . . . . ADMIN___
Password/Phrase .

```

Figure 9. Userid and password configuration for zSecure Visual Windows client

If the logon is successful and the client exists, you receive the initial password that you must supply in the Server definition dialog on the client side. The initial password has a limited validity of seven days, or the duration of the server run. For cancelation of the password before its validity expires, see [“Canceling a password” on page 142](#).

If the password generation fails, a general error message is displayed in the right upper corner of the screen. A more descriptive error message is also displayed. For problem diagnosis see [“SE.W communication problems” on page 157](#).

5. Install the client on the personal computer by following the instructions in the *IBM Security zSecure Visual: Client Manual*.

The new client can be installed next to a previous release. Customization of the previous release is not used by the new release. However, you can copy previously defined servers, including their certificates, as described in the *IBM Security zSecure Visual: Client Manual*.

**Note:** When upgrading a 1.x server, existing certificates will automatically be converted to the new encryption standard for 2.x servers. It is not possible to create new certificates for a 1.x client on a 2.5.0 server.

**Results**

If the server behaves in an unexpected way, you can review the files in the log directory:

### **bbracf.log, server.log**

These files provide information about the latest run of the server.

### **bbracf.log0, ..., bbracf.log9**

These log history files correspond to previous runs of the server. There can be up to 10 log history files.

For additional information about debugging zSecure Visual client issues, see the *IBM Security zSecure Visual: Client Manual*

## **Canceling a password**

If you decide not to use an existing password for client installation, you can cancel the password by typing action **C** in the Windows configuration panel. To be effective, cancel the password before anyone actually uses the password to install the client. Before generating a new password, actions **P** and **AP** also cancel any password previously issued for the client.

## **Creating Visual Clients in bulk**

### **About this task**

The bulk-agent function also serves as a mass-password-reset; it does not test whether or not the agent IDs already exist.

The mass-add function can operate in two modes:

#### **Autogen =yes**

In this mode, the bulk process creates or overwrites a dataset with generated agent IDs and initial passwords, in a format such as:

```
614 >CFC51AF4A7  
615 >5171DCADCD
```

The numbers are equivalent to the number that can be filled in on the zSecure Visual Configuration panel, in the column after the fixed constant 12.1. You can use the generated list to tell each agent-user his or her initial password. See the information about adding or editing a server definition in the *IBM Security zSecure Visual: Client Manual*.

#### **Autogen = no**

In this mode, the bulk process expects as input a dataset such as the one shown. The passwords and the > characters can be omitted, or they can contain what a previous run left in the dataset. The dataset must be sequential and have record format F or FB.

## **Procedure**

1. To invoke the bulk agent, go to any zSecure product panel under ISPF and type the following command on the command line:

```
TSO C2RELSI BULK
```

2. Enter the following information in response to the prompt of line-mode dialog:
  - Base port number of the server
  - Whether or not you want Autogen mode
  - Agent number for the first agent and the number of agents you want to generate. These two numbers are required only in Autogen mode.
  - Dataset name for the zSecure Visual clients.
    - In non-Autogen mode, this dataset must already exist and contain the list of agents to be generated.
    - In Autogen mode, this dataset might or might not exist.

- At the end of the dialog, you are prompted for the zSecure Visual Administrator's userid and password. When you supply these items correctly, the list of initial passwords is displayed.

## Configuration of client authorities

By default, client authorities are checked using resources in the XFACILIT class. However, your installation might have chosen to use a different resource class for the zSecure-related resources. See Appendix A, “Site module,” on page 191. Resources that are checked by z/OS UNIX System Services (that is, the resources covered by BPX.\*\*\*) cannot be reconfigured. These resources are always checked in the FACILITY class.

## Profiles for assigning interface levels to users

Mostly, the menu options, buttons, and fields in the zSecure Visual client application are enabled or disabled based on the user-selectable interface level. The central administrator can configure which interface levels each user can select. There are several interface levels: Helpdesk, Connect, User, Access List, Group, and Full. The *IBM Security zSecure Visual: Client Manual* documents exactly what is allowed under each of these levels.

To deny interface levels to client users, grant NONE access to the profiles listed in the following table:

*Table 10. Profiles for assigning interface levels to users*

Profile	Interface level
C2R.CLIENT.INTERFACE.HELPDESK	Helpdesk
C2R.CLIENT.INTERFACE.CONNECT	Connect
C2R.CLIENT.INTERFACE.USER	User
C2R.CLIENT.INTERFACE.ACCLIST	Access List
C2R.CLIENT.INTERFACE.GROUP	Group
C2R.CLIENT.INTERFACE.FULL	Full

For compatibility reasons, discrete profiles are required for Interface levels. Interface levels for which no corresponding profile exists are available for all users of zSecure Visual.

The zSecure Visual client interface uses several more security resources to configure its functionality, as explained in the following topics. You can review a subset of these resources using the MYACCESS report output. To inspect the MYACCESS output for a user, use the following TSO command:

```
CKGRACF SHOW MYACCESS ID <id>
```

## Required access for generated commands

Although the menu options, button, check boxes, and fields in the client application are enabled or disabled based on the profiles described in “Profiles for assigning interface levels to users” on page 143, further permissions are required on the server side. Without these permissions, the commands that the client generates, depending on these buttons and check boxes, will fail on the server side. Therefore, whenever the central administrator grants a user a particular interface level, the administrator must also make sure that the user is granted access to the resources as specified in the following table:

*Table 11. Resources for role-based authorities*

Resource	Uac c	Helpdes k	Connec t	User	Access List	Group	Full
CKG.CMD.CMD.EX.ADDGROUP	n	n	n	n	n	u	u
CKG.CMD.CMD.EX.ADDSD	n	n	n	u	u	u	u

Table 11. Resources for role-based authorities (continued)

Resource	Uac c	Helpdes k	Connec t	User	Access List	Group	Full
CKG.CMD.CMD.EX.ADDUSER	n	n	n	u	u	u	u
CKG.CMD.CMD.EX.ALTDS	n	n	n	u	u	u	u
CKG.CMD.CMD.EX.ALTGROUP	n	n	n	n	n	u	u
CKG.CMD.CMD.EX.ALTUSER	n	n	n	u	u	u	u
CKG.CMD.CMD.EX.DELDS	n	n	n	n	n	u	u
CKG.CMD.CMD.EX.DELGROUP	n	n	n	n	n	u	u
CKG.CMD.CMD.EX.PERMIT	n	n	n	u	u	u	u
CKG.CMD.CMD.EX.RACMAP	n	r	r	r	r	r	r
CKG.CMD.CMD.EX.RALTER	n	n	n	u	u	u	u
CKG.CMD.CMD.EX.RDEFINE	n	n	n	u	u	u	u
CKG.CMD.CMD.EX.RDELETE	n	n	n	n	n	u	u
CKG.CMD.CMD.EX.SETROPTS	n	n	n	u	u	u	u
CKG.CMD.CMD.REQ.CONNECT	n	n	u	u	u	u	u
CKG.CMD.CMD.REQ.PERMIT	n	n	n	n	u	u	u
CKG.CMD.CMD.REQ.REMOVE	n	n	u	u	u	u	u
CKG.CMD.COMMENT	n	r	r	r	r	r	r
CKG.CMD.LIST	r	r	r	r	r	r	r
CKG.CMD.SHOW.MYACCESS	n	r	r	r	r	r	r
CKG.CMD.USER.REQ.PWDEFAULT	n	n	n	u	u	u	u
CKG.CMD.USER.REQ.PWNOHIST	n	n	n	n	n	n	u
CKG.CMD.USER.REQ.PWNORULE	n	n	n	n	n	n	u
CKG.CMD.USER.REQ.PWRESET	n	u	u	u	u	u	u
CKG.CMD.USER.REQ.PWSET	n	u	u	u	u	u	u
CKG.CMD.USER.REQ.PWSET.DEFAULT	n	n	n	u	u	u	u
CKG.CMD.USER.REQ.PWSET.EXPIRED	n	n	n	u	u	u	u
CKG.CMD.USER.REQ.PWSET.NONEXP	n	n	n	u	u	u	u
CKG.CMD.USER.REQ.PWSET.PASSWORD	n	n	n	u	u	u	u
CKG.CMD.USER.REQ.PWSET.PREVIOUS	n	n	n	u	u	u	u
CKG.CMD.USER.REQ.RESUME	n	u	u	u	u	u	u
CKG.CMD.USER.REQ.SCHEDULE	n	u	u	u	u	u	u
CKG.RAC.SCP.CONNECT.BASE.AUTH.USE	n	n	u	u	u	u	u
CKG.RAC.SCP.CONNECT.BASE.AUTH.*	n	n	n	u	u	u	u
CKG.RAC.SCP*.BASE.*	n	n	n	n	u	u	u
CKG.SCP.ID.**	n	n	n	n	n	n	u

- Required access levels are abbreviated with n for NONE, r for READ, and u for UPDATE.
- Specifically for CKG.CMD.USER.REQ.PWNOHIST and CKG.CMD.USER.REQ.PWNORULE, granting a user UPDATE access results in bypassing the password history and the password rules, respectively, as specified in the RACF SETROPTS settings.
- For all other resources in the table, granting higher access has no effect. However, do not grant ALTER access because this access gives a user full control over not only the resource, but over the profile as well.
- Generic profiles are supported.
- In addition to any profiles you decide to cover with resources in the previous table, create catch-many profiles CKG.CMD.USER.REQ.\*, CKG.CMD.\*\*, CKG.RAC.\*\*, CKG.SCP.ID.\*.SYS1.\*, and CKG.\*\*, with UACC=NONE, and empty access lists. In this way, you prevent new functions of future releases of zSecure Visual from inadvertently becoming available to delegated administrators.

## Profiles for schedule name selection lists

The schedules a user can create are defined with discrete profiles of the form CKG.SCHEDULE.<SCHEDULE NAME>. When creating schedules, the user can select available schedule names from a list. The schedule name \$DELETE allows the user to mark user profiles for deletion. See the examples in the following table for some suggested schedule names.

Profile	Uacc	Help desk	Connect	User	Access List	Group	Full
CKG.SCHEDULE.\$DELETE	n	n	n	u	u	u	u
CKG.SCHEDULE.GRPADMIN	n	n	u	u	n	u	u
CKG.SCHEDULE.HELPDESK	n	u	u	n	n	n	n
CKG.SCHEDULE.SYSADMIN	n	n	n	n	n	n	u

## Authorities required to duplicate a user

Usually at least group-special and CLAUTH(USER) authorities are required to duplicate a user. You must also have DATASET authorities to create aliases in the master catalog.

## Profiles to allow the Define Alias action

To allow the Define Alias action, create discrete profiles of the form CKG.UCAT.<USER CATALOG NAME>. These profiles are required because otherwise zSecure Visual has no way to know which user catalogs exist. When a zSecure Visual user is granted at least READ access to the profile representing the user catalog, the user can define an alias for a userid or groupid pointing to the catalog.

Profile	Uacc	Any role
CKG.UCAT.<USER CATALOG NAME>	n	nr*

**Note:** The user also need READ access to FACILITY STGADMIN.IGG.DEFDEL.UALIAS profile, or UPDATE access on the master catalog and the relevant user catalogs.

## Resource for RACF scoping

When the user has READ or higher access to the resource, the system extends the user's CKGRACF scope to the RACF scope for users with READ access. If the user has NONE access, the scope is not extended.

<i>Table 14. Resource to activate RACF scoping</i>		
Resource	Uacc	Any role
CKG.SCP.RACF	n	nr*

## Password change policy for zSecure Visual users

If your local policy requires users to specify a reason when changing a password, you can use the profile in the following table to enforce the policy. This profile is triggered when a user with NONE access to the profile attempts to change the password. In all other situations, specifying a reason is possible but not required. A discrete profile is required.

<i>Table 15. Profile to enforce password change policy</i>		
Profile	Uacc	Any role
C2R.CLIENT.EMPTYREASON.PWSET	n	n

## Segment editing for users

In order to edit segments, users require UPDATE access to the relevant resource for the class as seen in “Required access for generated commands” on page 143. (Specifically, they are CKG.CMD.CMD.EX.ALTUSER, CKG.CMD.CMD.EX.ALTGROUP, CKG.CMD.CMD.EX.ALTDSO and CKG.CMD.CMD.EX.RALTER.) In addition, users require UPDATE access to the necessary FIELD class resources (or System Special).

The following table shows the syntax of the resource that is used to control segment editing.

<i>Table 16. Resource that controls segment editing</i>		
Profile	Uacc	Any role
<CLASS>.<SEGMENT>.<FIELD>	n	u

## Authority to manage client definitions

A zSecure Visual client requires a local server definition and a corresponding client definition on the server in order to access the server through a safe channel. To set up a new channel, an initial password is needed once. To be able to manage the client definitions located on the server, the administrator must have READ or higher access on the C2R.SERVER.ADMIN resource. This access allows the administrator to create new client definitions, edit and delete existing ones, and also to generate initial passwords on any system the administrator can log on to. You might want to grant this authority to only a few people.

<i>Table 17. Resource to maintain zSecure Visual server and client definitions</i>		
Resource	Uacc	Any role
C2R.SERVER.ADMIN	n	r

## Profile for viewing system-wide RACF options

To view the system-wide RACF options, users must have READ access on the following discrete profile:

<i>Table 18. Profile for viewing system-wide RACF options</i>		
Profile	Uacc	Any role
C2R.CLIENT.SETROPTS	n	r

This profile is defined as a discrete profile in the XFACILIT class. If a zSecure Visual client user does not have READ access to the profile, the user cannot display the RACF SETROPTS settings.



## Implementing site-specific functions

---

zSecure Visual supports two site-specific functions:

- Presenting site-specific user data in the Visual Client; data that is user-only information, specific to your organization, like employee numbers and department codes.
- Calling site-specific REXX scripts from the Visual Client through its user interface in a manner that is transparent to the end user. This makes the Visual Client fully customizable to support any new function for your organization.

If you want to implement these site-specific functions, additional configuration of the Visual Server is required.

### Site-specific user data

Use the guidelines and settings in this topic to configure site-specific user data for zSecure Visual.

You can configure zSecure Visual to present *site-specific user data*. This data is user-only information that is specific to your organization (for example, employee numbers and department codes). You can then retrieve, display, and search on this data in the following Visual client panels:

- User properties dialog
- User table
- Find dialog

For example, you can configure Visual Server to retrieve personnel information that the Visual client displays in addition to the other fields that it normally displays.

To configure for the display of site-specific user data in the Visual client, perform the following tasks:

- Determine what information and information characteristics to display to your users:
  - Location of and information in the site-specific user datasets that you want to display in the Visual client.
  - Column order in which you want to display the user data in the Visual client.
  - Columns of data for which the users can perform search operations.
  - Whether to display the site-specific user information in addition to or instead of INSTDATA information.
- Create the Associations and Record format configuration files to specify the location and format of the site-specific user information. These files are described in this section.
- Specify the allocation of the data sets to the Visual client using the C2RWASSC dataset member of the C2RWCUST ALLOC parameter. See the C2RWCUST parameter in [Appendix D, “Configuration parameters and members,”](#) on page 205.

#### Associations configuration file

This file specifies the name of the data sets where the site-specific user information is staged for access by the Visual Server. The Associations configuration file contains a header row followed by one or more data file definitions and one or more record format file definitions.

##### Header row

The header row is specified using the capital letter H as the first character, followed by the version number of the site associations configurations file, which is currently 1.13:

```
H1.13
```

##### Data file definitions

Specifies the user ID for which you want to view data and the name of the data file that contains the data records:

```
User_id DATA DSN='data.file'
```

### **User\_id**

Specifies an individual user ID or a generic user ID for all users. Start each row with a capital U as the first character. Separate this keyword from the DATA keyword with one or more spaces.

#### **Individual user\_id**

Specify an individual RACF user ID to retrieve and present user information for a single user ID. This user ID must match the user ID that is logged onto the client in order to use the specified data file. Individual user IDs are useful for testing an initial setup of the data for display and for restricting access to specific users.

#### **Generic user\_id**

Specify a generic user ID to retrieve and present user information for all user IDs. Use an asterisk (\*) to specify a generic user ID (U\*). If an individual user ID is specified and verified, the generic user ID is not used. Generic user IDs are useful for presenting user information to a general population in your organization as part of normal operations.

### **DATA**

This keyword must precede the name of the data file containing the data records. Separate this keyword from the DSN parameter with one or more spaces.

#### **DSN='data.file'**

Specifies the name of the data file containing the data records for the specified user ID. Enclose the file name in single quotation marks.

### **Record format file definitions**

Specifies the user ID for which you want to view data and the name of the Record format configuration file:

```
User_id RECFORMAT    DSN='recordformat.file'
```

### **User\_id**

Specifies an individual user ID or a generic user ID for all users. Start each row with a capital U as the first character. Separate this keyword from the RECFORMAT keyword with one or more spaces.

#### **Individual user\_id**

Specify an individual RACF user ID to format retrieved user information for a single user ID.

#### **Generic user\_id**

Specify a generic user ID to format retrieved user information for all user IDs. Use an asterisk (\*) to specify a generic user ID (U\*).

### **RECFORMAT**

This keyword must precede the name of the Record format file. Separate this keyword from the DSN parameter with one or more spaces.

#### **DSN='record\_format.file'**

Specifies the name of the Record format configuration file. Enclose the file name in single quotation marks.

### **Example contents for Associations configuration file**

This example demonstrates the specification of two data set names: one entry for an individual user and one generic entry for all users.

```
H1.13
UDEMOUSER  DATA      DSN=' DEMO . DATA '
UDEMOUSER  RECFORMAT   DSN=' DEMO . FORMAT1 '
U*         DATA      DSN=' SERV#1 . DATA '
U*         RECFORMAT   DSN=' SERV#1 . RECfmt '
```

### **Record format configuration file**

The Record format configuration file specifies how to present the user information from the site-specific data file. The record format file has the following record types:

## FIELD key field

The syntax of this entry is:

```
*FIELD 'field_name' (field_start,length)
```

where:

### \*FIELD

Required. For CARLa to combine local site user data with RACF information, each row in the data file must contain a field that matches a value in the RACF database. This field is defined using the *field\_name*. When CARLa extracts the information from RACF, it uses the RACF value for the chosen field to look up the relevant row in the data file.

### *field\_name*

Required. Specifies which user profile field is used to look up the relevant row in the in the data file. Single quotation marks are required around the field name. You must include at least one space after \*FIELD to separate the field name.

### *field\_start,length*

Specifies the starting position and length of the field name in the data file that corresponds with the value specified as *field\_name*. Both values must be integers. Use the delimiters as shown in [Example contents for Record format configuration file using \\*FIELD](#); separate the integers with a comma and enclose both values in the parentheses.

## User ID key field

The syntax of this entry is:

```
*USERID (field_start)
```

where:

### \*USERID

Required. For CARLa to combine local site user data with RACF information, each row in the data file must contain a field that matches a value in the RACF database, namely, the RACF USERID. When CARLa extracts the information from RACF, it uses the RACF value for UserID to look up the relevant row in the data file. CARLa then uses the offsets defined (for example, Department at offset 29 in the examples in [Example contents for Record format configuration file using \\*USERID](#)) to extract and include in the values returned to the Visual client.

### *field\_start*

Required. Specifies the starting position of the user ID field. The field is always 8 characters long, so you do not specify the length of the field. (All user IDs are eight characters long.) You must include at least one space after the \*USERID prefix to separate the *field\_start* value.

## Column definitions

```
n 'column_title' (field_start,length) Y | N
```

*n*

Specifies the column sequence number, which indicates the order in which the columns are displayed in the Users profile table and the User properties form. Specify a single integer in the range 1-9. The maximum number of columns is 9.

### 'column\_title'

Specifies the name that is assigned to the displayed column. You must use single quotation marks; for example, 'Department'. You can specify up to 20 characters.

### (*field\_start,length*)

Specifies the starting position of the column and the width (length) of the field in the data file. Both values must be integers. You must use the delimiters as shown; separate the integers with a comma and enclose both values in parentheses. There is no validation for overlapping column definitions; the administrator is responsible for specifying the values correctly.

## Y | N

Specifies whether the column is enabled for searching (added to the search form). Specify Y to enable for searching or N to disable for searching. If you do not specify Y or N, the column is not enabled for searching.

### Installation data

```
*INSTDATA
```

If this field is not specified, the Visual client displays the site-defined columns as a replacement to the Installation data field in the Users table and the User properties dialog. Add the \*INSTDATA row to the record layout if you want to display your site-specific user information and the installation data information in the Visual client.

### Example contents for Record format configuration file using \*FIELD

This example demonstrates a layout of site-specific user information in four fields (or columns, depending on the dialog), with one searchable field (Employee). The order of the fields can be different from the order in which the column definitions are listed (which is specified by the column sequence number). The user information is displayed in addition to installation data (INSTDATA) information.

This example lists the fields according to the sequential offsets in the data (source) file:

```
*FIELD 'pgmname' (1,20)
1 'Employee No.' (21,7)
2 'Department' (29,5) Y
4 'Cost Center' (34,7)
3 'State' (42,3)
*INSTDATA
```

The Visual client reads the Record format configuration file to generate the corresponding CARLa commands that it sends to the Visual Server. The following example shows the contents of a data source file with fixed length records that are referenced by the fields in the previous example of a Record format configuration file:

Offsets:

	1	21	29	34	42
A. Name One	1000405	203420002451	NSW		
B. Name Two	0003050	300120002451	TAS		
C. Name Three	2030060	203420030288	NSW		
A. Name Four	2004078	300120002451	VIC		
B. Name Five	1000407	510630030288	SA		
C. Name Six	0060902	640620005624	WA		

### Example contents for Record format configuration file using \*USERID

This example demonstrates a layout of site-specific user information in four fields (or columns, depending on the dialog), with one searchable field (Employee). The order of the fields can be different from the order in which the column definitions are listed (which is specified by the column sequence number). The user information is displayed in addition to installation data (INSTDATA) information.

This example lists the fields according to the sequential offsets in the data (source) file:

```
*USERID          (1,8)
2 'Employee'     (9,20) Y
1 'Department'   (29,5)
4 'Cost Center'  (34,7)
3 'State'        (42,3)
*INSTDATA
```

This example lists the fields according to the desired order of presentation in the Visual client:

```
*USERID          (1,8)
1 'Department'   (29,5)
2 'Employee'     (9,20) Y
3 'State'        (42,3)
4 'Cost Center'  (34,7)
*INSTDATA
```

The Visual client reads the Record format configuration file to generate the corresponding CARLa commands that it sends to the Visual Server. The following example shows the record layout from a RACF data (source) file with fixed-length records that is referenced by the fields and offset locations in the example Record format configuration file:

```
offsets: 1      9      29      34      42
          C2RWQA47QA-00000047      500654300510 SA
          C2RWQA46QA-00000048      500654301610 TAS
          C2RWQA40QA-00000040      500654300510 WA
```

## Site-defined REXX scripts

Use the guidelines in this topic to customize zSecure Visual such that site-defined REXX scripts can be called from zSecure Visual through its user interface in a manner that is transparent to the end user. Currently, the REXX scripts can be executed at the local node only.

Starting from version 2.1.0, you can customize zSecure Visual to call site-defined REXX scripts through the zSecure Visual user interface. The process is transparent to the end user. For site-defined REXX scripts to be called from zSecure Visual through its user interface, the Visual Server must be configured with an association file that contains the site-defined scripts configuration information that the Visual Client can use. The association file is defined as the C2RSCRPT member of the C2RWCUST data set, with the site-defined scripts themselves also being members of this data set.

This is an example of such an association file:

```
2.1.0
$HOMEDIR USER OMVS "Create home directory"
$ROLEAB USER * "Add role ab"
$SCRIPT3 GROUP BASE "Disable passphrase"
$ALIAS USER * "Define alias to resource"
```

The first line of the association file contains a version identifier that is used to distinguish between different versions of an association file. The current version is 2.1.0. Only one version of the association file (the C2RSCRPT member) is supported. Future versions of the association file will be backward compatible with previous versions.

The subsequent lines are made up of the following fields:

### Script Name

The name of the C2RWCUST member that contains the site-defined script. To differentiate between site-defined scripts and other members of C2RWCUST, it is suggested to prefix members that contain the site-defined scripts with a '\$' character.

### Class

Represents the class that is to be provided as an input parameter to the script, such as USER or GROUP.

### Segment

Represents the segment which is to be provided as an input parameter to the script, such as BASE or OMVS. If segment selection is not required, then specify an asterisk (\*) for this column.

## Description

A short description of the site-defined script. This description is displayed as text for the corresponding Action menu item and the context-sensitive menu item in the user interface. The description has to be enclosed in double quotes. The description should be a word or a few words at most. Descriptions larger than 50 characters will be truncated.

The contents of the association file are not case-sensitive, except for the Description field.

This sample illustrates a REXX script that uses the DFSMS AMS (Access Method Services) command **DEFINE ALIAS** to define an alias to a resource name based on the value of the first character of the supplied key:

```
/* REXX */
/* In case CLASS=USER, the zSecure Visual client passes a key
   and a segment to the site-defined script */
/* The segment is not employed in this script */
parse arg class segment key
if class<>'USER'
then
do
  say 'CLASS must be USER'
  return 123
end
/* Derive a 'name' value from the key */
name = key
/* Derive a 'relate' value from the key */
if substr(key,1,1)='C'
then
do
  relate = "ICFCAT.C1"
end
else
do
  relate = "ICFCAT." || key
end
/* Build the 'define' argument */
define_argument = "alias (name('" || name || "') relate('" || relate
define_argument = define_argument || "'))"
/* Provide some feedback for when the 'define' fails */
say "define" define_argument
/* Execute the 'define' command */
address tso
define define_argument
/* pass TSO command return code to the Visual client; 0 = success */
return rc
```

## zSecure Visual Server operations

---

### Starting the Visual Server

You can start the Visual Server as a started task by issuing this command:

```
S C2RSERVE
```

or submit a batch job with SURROGAT authority as in SCKRSAMP(C2RJSERV). Using either method requires running the command or job under the proper server userid.

The server can be started only if z/OS UNIX System Services are available. If you want to use automated procedures to start the server, be sure that these procedures execute after the following system message has been received:

```
BPXI004I OMVS INITIALIZATION COMPLETE
```

Failure to wait for this message results in the symptoms described in [“Server startup problems” on page 155](#).

If you try to start the server twice for the same IP port, the second start command terminates.

## Visual Server logs to verify initialization

You can use one of the following methods to see whether initialization is ready:

- Use the ISPF OBROWSE command to look in the log file periodically:

```
OBROWSE <ServerRoot>/log/server.log
```

- Use the C2RSLOG procedure to copy the logs to JES pool space:

```
S C2RSLOG
```

## Stopping the Visual Server

To stop the server, issue the following command:

```
S C2RSTOP
```

You can also stop the server by canceling the parent task. The parent task is the one that has a proper step name (not \*OMVSEX).

## Problem determination

---

This section contains the following troubleshooting topics:

- [“Resources to resolve system problems” on page 153](#)
- [“Command to collect diagnostic information” on page 155](#)
- [“Server setup \(job C2RZWINI\) problems” on page 155](#)
- [“Server startup problems” on page 155](#)
- [“Server response problems” on page 156](#)
- [“zSecure Admin termination problems” on page 157](#)
- [“SE.W communication problems” on page 157](#)

## Resources to resolve system problems

You can locate information to help resolve system problems using any of the following resources:

- The **File about-server.box** in the run subdirectory provides information about the server as a whole. The same information is available on the client as the **Server information** option in the **Help** menu.

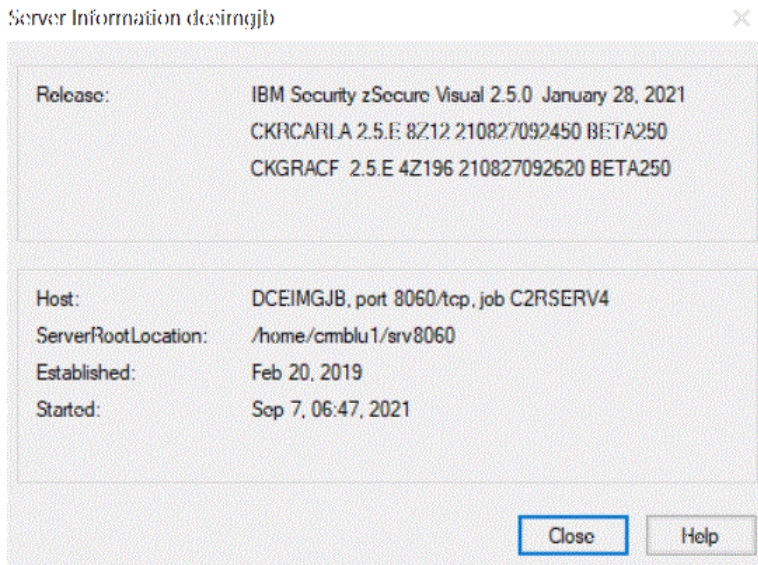


Figure 10. zSecure Visual Client Server Information dialog

The upper box shows the software releases that the server uses. The upper line corresponds to the release of the pax file. The other two lines provide the releases and build dates of the zSecure components CKRCARLA and CKGRACF as they were at the time the server was started.

**Note:** Do not upgrade these components while the server is active. If you do, the Server information does not display until you restart the server.

The lower box provides the following information about the server identity:

**Host**

The hostname of the server, its IP port as configured in job C2RZWINI, and the jobname of the address space that started the Server.

**ServerRootLocation**

The (possibly resolved) value of the C2RSERVE parameter in the zSecure configuration.

**Established**

The time that the server established itself as a Certificate Authority (job C2RZWINI).

**Started**

The time that the server was last started or restarted.

The text in the title bar does not come from the **about-server.box** file. It contains the name that was specified on the client side under the **File -> Configure** menu.

- The MVS syslog provides messages related to server start problems. In addition, you can often find messages about security violations (ICH408I).
- SMF can provide insight into security violations. SMF can also provide information about successful access based on the AUDIT option of the RACF profiles.
- Server logs are available in the `log` subdirectory within the Server root directory. This directory is identified by the C2RSERVE parameter in the zSecure configuration of the Server. The Server log directory is also identified on the Client side in the Server information box, through the **Help** menu. The `bbracf.log` and `server.log` files in the `log` subdirectory provide information about the latest run of the server. There is a history of 10 logs for each type; for example, the files `bbracf.log0`, ..., `bbracf.log9` correspond to previous runs of the server.
- For problems during SE.W, look in the home directory of the TSO user. See [“SE.W communication problems” on page 157](#).
- If the client behaves in an unexpected way, see the *IBM Security zSecure Visual: Client Manual* for more information.



- The SYSPRINT from the last CKRCARLA run, the CKGPRINT from the last CKGRACF run, and the commands issued are available through the client's communication window.

## Command to collect diagnostic information

The **c2rdiag** command can be run at any time; it does not matter whether the zSecure Visual Server is running. The collected information is stored in a dump file, **C2Rdiag\_dump\_XXXX.tar**, where XXXX represents a time stamp. The dump file can be transferred to IBM Software Support for troubleshooting.

Because the **c2rdiag** command needs information about all active processes in the system to collect diagnostic information, the command must be run under a userid with root authority (uid=0). Running under root authority ensures the necessary permission:

- READ and WRITE permission to the <Server Root> directory that is identified by the C2RSERVE parameter in the zSecure configuration file.
- READ and EXECUTE permission to the zSecure Visual Server software directory that is identified by the C2RWIN parameter in the zSecure configuration file.

## Collecting diagnostic information and sending to IBM for troubleshooting

### Procedure

Perform the following steps to collect diagnostic information and send the dump file to IBM.

**Note:** System log output (SDSF) is not captured by the **c2rdiag** command. If this information is considered relevant, you must supply an extract of the system log around the time of the suspected events.

1. Log on to the system with a userid that has root authority.
2. Open an OMVS command shell and navigate to the <Server Root> directory.
3. Run the command `./bin/c2rdiag`
4. Using binary mode, transfer the dump file, **C2Rdiag\_dump\_XXXX.tar**, to IBM
5. After IBM confirms receipt of the file, delete the dump files to prevent disk space from running out. The zSecure Visual Server cannot delete the files because they are root-owned.

## Server setup (job C2RZWINI) problems

The following error messages can occur during server setup:

### FSUM2078

This message might be issued if you did not create a home directory for the server userid.

### FOM0303I rsn=0924041A

The following message indicates you do not have READ access on the FACILITY resource BPX.FILEATTR.APF:

```
FOMF0303I CKGRACF: chattr() error: rv=-1, errno=8B, rsn=0924041A
```

### FOM0303I rsn=0924041B

The following message indicates you do not have READ access on the FACILITY resource BPX.FILEATTR.PROGCTL:

```
FOMF0303I ./bin/bbmini: chattr() error: rv=-1, errno=8B, rsn=0924041B
```

## Server startup problems

When the Server encounters a problem during startup, it produces a C2RW message. These messages are described in the *IBM Security zSecure: Messages Guide*. The following startup problems do not produce C2RW error messages.

- Attempts to start the server before z/OS UNIX System Services initialization is complete (that is, too soon after IPL) result in message ICH408I.

```
ICH408I USER(C2RSERVE) GROUP(C2R) NAME(ZSECURE VISUAL SERV)
        CL(FSOBJ )
        INSUFFICIENT AUTHORITY TO DUB
```

The task runs, but not as an z/OS UNIX System Services process, which makes it useless. If you receive this message:

1. Cancel the task.
  2. Wait for the BPX1004I message as described in [“Starting the Visual Server” on page 152](#).
  3. Start the task again.
- Attempts to run the server when you are not allowed to use the port number result in the following error message:

```
TCPIP Conn: can't bind to socket (errno 111)
```

In this case, you might have reserved the TCP/IP port numbers used by the server using parameters in the PROFILE.TCPIP dataset. The commands can look similar to:

```
PORT xxxx TCP C2RSERVE NOAUTOLOG
```

or

```
PORTRANGE xxxx yy TCP C2RSERVE NOAUTOLOG
```

In this case, the C2RSERVE jobname is the only ID allowed to open the port. Therefore, if the installation step runs with a different jobname, it receives a `bind() errno=111` message.

To avoid this problem, base your protection of TCP/IP ports on userid, rather than jobname. See [“TCP/IP Security” on page 138](#).

- Attempts to run the server while TCP/IP has not been started result in the following error message:

```
S8E220:TCPIP Conn: Socket error 112
```

In this case there might be problems with TCP/IP, or TCP/IP might not be active at all.

The server aborts itself when it cannot successfully run zSecure Admin. For example, failing to make CKGRACF and CKRCARLA program-controlled (see [“Installation requirements” on page 133](#)) can result in messages such as:

```
ICH420I PROGRAM CKGRACF FROM LIBRARY CKR.SCKRLOAD
        CAUSED THE ENVIRONMENT TO BECOME UNCONTROLLED.
BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR
        SERVER (BPX.SERVER) PROCESSING.
```

This message might be accompanied by a dump file (CEEDUMP.timestamp) in the run directory. A dump that is written for this reason can be discarded.

## Server response problems

If the server is not responding, first determine whether the server is waiting or spending CPU time to perform work. You can see this example with SDSF DA. The server normally is shown as 3 address spaces in SDSF (while idling). Possible causes for lack of response include:

- The client is using the wrong port numbers or machine name. There is a test connection button in the client to verify they at least are active. You can use the **netstat** command under TSO to see on which port the server is listening.
- The server log shows a number of messages:

```
E10:Crypt: Protocol violation. message from 12.1.4 and no secure channel
E18:Crypt: Unexpected message from 12.1.4 suspicious, so discarded
```

The probable cause of these messages is that the server agent has been stopped and then started again, while the client agent kept running. On a lightly loaded 30 MIPS machine, the client can connect to the server within 6 minutes, and six E10/E18 message pairs resulting from retransmissions by the

client are printed to the server log. A quicker way to recover is to end the **c2ragent.exe** task using the **End Process** button of the Windows Task Manager, and to close and restart the IBM Security zSecure Visual application.

Another cause might be a logon attempt just after configuring a client. In that case, a single E10/E18 message pair can be printed at the server side because a secure channel has not been completely set up yet. Recovery in this case only takes 1 minute on a fast machine. However, you can avoid the delay by waiting 15 seconds (on a fast machine) after client configuration before attempting a logon. In trace mode, the server displays the following message when it is ready to accept a logon:

```
E0:CA: Finished certifying Agent Keys
```

- The server log shows the following reconnect message:

```
E183:Route: reconnected from 12.1.4
```

This message indicates that two clients with identical agent ids are attempting to communicate with the server. Stop one of the **c2ragent.exe** processes with the Windows Task Manager. The processes most likely reside on a single computer but they might also reside on separate computers.

- The server log shows the following message:

```
E160:LCM: There are no valid LCM certificates. Please reconfigure the server
```

The most likely cause is that server initialization, job C2RZWINI, was not run successfully. Less likely, the server did not run in the last 9 months, so it did not refresh its certificates in time. Stop the server, run or rerun job C2RZWINI, and verify that the server successfully initializes as a Certificate Authority. See [“First time startup of the Server” on page 138](#).

If none of these items describe your situation and the problem is reproducible, you can start the server with the TRACE option:

```
S C2RSERVE,OPT=TRACE
```

Using the TRACE option results in bigger server logs that contain detailed timing information. To have IBM Software Support help with debugging those problems, send both the client log and the server log.

## zSecure Admin termination problems

After a zSecure Visual client logon, a few zSecure Admin transactions are performed to tailor the GUI to the user's authorities and to download the class descriptor table. Sometimes these actions fail and one of the following error messages displays:

- CKR0010 OPEN abend hhh-hh on file ddname

The OPEN for the indicated file failed. The ddname field might be empty or contain garbage. Also check that the user has at least READ access on the RACF database.

- CKR999I GETMAIN FAILED FOR HEAP name - INCREASE REGION

The CKRCARLA program terminates with CKR999I or CKR0999 when the program requires more virtual storage than allocated by USS. To resolve this problem:

1. Increase the maximum allowed virtual storage size for the Server's userid by specifying an ASSIZEMAX value in bytes in the OMVS segment for the server, as shown in the following example:

```
ALTUSER C2RSERVE OMVS(ASSIZEMAX(64000000))
```

2. Restart the server to make this change effective.

For additional information about this problem, you can also examine the MVS system log for security violation messages.

## SE.W communication problems

The SE.W communication is handled by the REXX C2RELSI program. This program creates four files in the home directory of the user:

### **C2RELSI.userid.LST**

The official response file. This file usually contains the install password generated. The password is normally just one line containing ten hexadecimal digits, as shown in the following example:

```
8337F93AD5
```

### **C2RELSI.userid.ERR**

The line mode output file, which usually contains only the userid and passwords prompts:

```
userid:password:
```

### **C2RELSI.userid.LSI**

The input file with commands for the server. For a **P** command, the input file would contain:

```
minigenerateinstallpassword(12.1.100)
echo(!R:)
```

### **C2RELSI.userid.LOG**

The software log file, which normally contains only the software level and open/close messages as illustrated in the following example:

```
<20010427 08:11:27 utc> P399M1V0.0L309A5S0E10:Opened C2RELSI.MYUSER.LOG.
Product: racfwin.product.server.app. Version: 1.4.
Builddate: 2001/04/23/13:02. Local time: Fri Apr 27 08:11:27 2001.
<20010427 08:11:28 utc> P399M1V0.0L164A2S0E20:Forced close of C2RELSI.MYUSER.LOG
<20010427 08:11:28 utc> P399M1V0.0L461A5S0E15:Closed C2RELSI.MYUSER.LOG
```

The following error messages might display at various stages of the SE.W communication process:

### **Failure to execute**

If the REXX C2RELSI cannot find the program **lsi**, or the current user is not allowed to execute it, the message **Failure to execute** is displayed in the upper right corner of the screen. If you press PF1 (Help), the long message explaining the cause of the error is displayed as shown in the following example:

```
/u/C2RSERVE/c2rserve/bin/lsi -t C2RELSI.MYUSER.LOG A:10.0.1.20:8011 C2RELSI.MYUSER.LSI -
errno=81 53B006C
```

The long message specifies an error number (*errno*) that provides information about the problem. The possible error messages and associated explanations are as follows:

#### **errno=81 594003D**

This error occurs when one of the directories in the path to the **lsi** executable is not found. The path is specified by the C2RWIN parameter in the zSecure configuration. To correct the problem, make sure that the path exists in the z/OS UNIX System Services zFS file system and that the path was used in job C2RZWUNP.

**Note:** The C2RWIN parameter is case-sensitive.

#### **errno=81 53B006C**

This error occurs when the location of one of the zSecure Visual programs is not found. To correct the problem, make sure that the path exists in the z/OS UNIX System Services zFS file system and that the path was used in job C2RZWUNP.

#### **errno=6F 5B400002**

This error occurs when the current user has no search access on a directory in the path to the **lsi** executable. This problem also shows up in the SYSLOG as an access violation:

```
ICH408I USER(MYUSER ) GROUP(MYGROUP ) NAME(VISUAL RACF ADMIN )
/usr/lpp/c2r/V2R3M1/lsi
CL(DIRSRCH ) FID(01E2D4E2F0F0F833F409000000000003)
INSUFFICIENT AUTHORITY TO LOOKUP
ACCESS INTENT(--X) ACCESS ALLOWED(OTHER ---)
```

To fix the problem, grant the user who is to run SE.W access to the directory where the zSecure Visual server code resides. In job C2RZWUNP, ownership of this directory was established as user

C2RUSER and group C2RGROUP (which you might have customized). CONNECT the userid who is to run SE.W to the owning group. Note that SE.W is only required to configure the first workstation. This workstation can then be used to configure subsequent workstations.

### **Cannot browse an empty file**

This ISPF error message can hide the original error message reporting on a failure to execute **lsi**. This message might display if the zSecure Visual server is not running yet.

### **an error has occurred**

If the password generation fails, this message is displayed in the upper right corner of the screen and is accompanied by one of the following more descriptive error messages.

#### **couldn't open session with bluebook adapter**

This descriptive message indicates that the server has not been started, or it has been started but it is not yet ready to accept a password generation request.

If the server has just been started, it is usually ready to generate a password after about 10 seconds on a lightly loaded 30 MIPS machine. If the same error message is displayed after a delay of a few minutes, the server might be unreachable or the IP number might be incorrect.

### **logon failed**

This message is displayed when the server accepts the password generation request but is still not ready to generate a password. To resolve this problem, wait a few seconds (on a 30 MIPS machine) after the failure message displays before attempting another password generation request. When the server is ready to generate a password, the following message displays in the server log:

```
E5:Dispatch: Started adapter 'RACF'
```

If the server runs in trace mode, it is ready to generate a password when the following trace message is printed twice:

```
E0: IpcSetState:setting state ( 6 -> 1 )
```

### **Must be numeric**

This message displays when the entered agent ID is not of the form 12.1.<NN>, where <NN> is a sequence of decimal digits. To fix this problem, enter an agent ID in the correct form (for example, 12.1.100).

### **Userid and password messages**

- Unknown userid <userid>.
- Userid <userid> is revoked.
- Invalid password.
- The password has expired.

### **Resource C2R.SERVER.ADMIN in the <class> class is not covered by a RACF profile.**

If this error occurs, you can see the following message in the JES SYSLOG:

```
ICH13003I C2R.SERVER.ADMIN NOT FOUND
```

### **EDC5139I Operation not permitted. Reason code: 00d8.**

This message and reason code indicate that the server userid has no READ access to the FACILITY resource BPX.SERVER. If this error occurs, you can see the following message in the JES SYSLOG:

```
ICH408I USER(C2RSERVE)
BPX.SERVER CL(FACILITY)
INSUFFICIENT ACCESS AUTHORITY
ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )
```

### **EDC5139I Operation not permitted. Reason code: 02af.**

This message and reason code indicate that one of the modules that is run under control of the Visual server cannot be loaded because it does not meet the Program Control requirements. See [“Installation requirements”](#) on page 133 on how to set up Program Control for the Visual server.

Also, search the SDSF syslog for messages that occurred around the time of the failure. For example, messages like the following may appear:

```
ICH420I PROGRAM CKRCARLA FROM LIBRARY CKR.SCKRLOAD CAUSED THE ENVIRONMENT TO BECOME UNCONTROLLED.  
ICH422I THE ENVIRONMENT CANNOT BECOME UNCONTROLLED.  
BPXP014I ENVIRONMENT MUST REMAIN CONTROLLED FOR DAEMON (BPX.DAEMON) PROCESSING.  
CSV042I REQUESTED MODULE CKRCARLA NOT ACCESSED. THE MODULE IS NOT PROGRAM CONTROLLED
```

In particular, messages ICH420I and CSV042I identify the module that does not meet the requirements. Find the PROGRAM profile that covers that module, find from which data set the module is to be loaded, and make sure that that data set is a member of the relevant PROGRAM profile.

**C2RW018I The resource class for zSecure security checks cannot be determined**

The CKRSITE module does not contain a valid security class. Such a class is required to determine the access of users to various resources. For information about the CKRSITE module, see [Appendix A, “Site module,” on page 191](#).

**<userid> has no READ access to C2R.SERVER.ADMIN resource in the <class> class.**

This message indicates that the userid does not have at least READ access to the C2R.SERVER.ADMIN resource. In the JES SYSLOG, you can see the following message:

```
ICH408I USER(ABCDEFGH)  
C2R.SERVER.ADMIN CL(FACILITY)  
INSUFFICIENT ACCESS AUTHORITY  
ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )
```

**The environment does not satisfy the requirements for program control.**

A required module is not program controlled. All load modules (and program objects) that are loaded in the Visual Server address space must be program controlled. Also, the file system that contains the Visual Server software must be mounted with the SECURITY and SETUID attributes. You can identify the uncontrolled module from message CSV0421I in the MVS syslog. See [“Installation requirements” on page 133](#) and [“Owner and location preparation for the software” on page 136](#). After establishing program control, you must restart the server.

**The agent has not been added with A or AP.**

This message indicates an attempt to generate a password for an unconfigured client. No password has been generated. Add the client as described in [“Configuring the Visual Client” on page 140](#).

---

## Chapter 15. Data preparation for SIEM

You can use zSecure to make z/OS event data available for SIEM applications such as IBM QRadar SIEM or Micro Focus ArcSight.

The zSecure Adapters for SIEM transform SMF records into a text format that the SIEM application can process, and adds information into these events that help the SIEM application interpret the data. This process is designed to produce an audit trail of z/OS events by copying large quantities of SMF records to the SIEM application. This function is also available in zSecure Audit.

There are two modes of operation for this 'full' enriched SMF feed: near real-time (sent using the UNIX syslog protocol), and by FTP file polling. Near real-time works better with real-time SIEM processing but also incurs more overhead during peak periods. FTP file polling allows you to postpone processing to a less busy time. In file polling mode, the SIEM application retrieves these text files according to a schedule that is configured on the SIEM console. For near real-time mode, the SIEM application must be configured to accept syslog traffic. The 'full' near real-time SMF feed can be collected by zSecure in two ways: directly by using SMF INMEM facility or using the zSecure SMF collector (CKQEXSMF).

You can also send alerts generated by zSecure Alert to the SIEM application. The alerts can be based on SMF or on other sources (for example based on the detection of system changes). Alerts are transferred near real-time to the SIEM application and are not dependent on any configured schedule. In zSecure Alert, specify the UNIX syslog format, and specify QRadar Unix syslog or ArcSight CEF via syslog as the recipient. For more information about zSecure Alert, see the *IBM Security zSecure Alert: User Reference Manual*.

---

### Prerequisites

After installing the software, you must also perform activities to create and modify the configuration. The following criteria must be met:

- If you use PARMLIB member IFAPRDxx to disable some zSecure components, either zSecure Adapters for SIEM or zSecure Audit must not be disabled. For details, see [“Enablement of license features” on page 18](#).
- The SCKRLOAD library must be APF-authorized. For details, see [“APF authorization of the software” on page 18](#).
- If you decide to use the direct SMF INMEM real-time interface, you must have z/OS 2.3 or higher installed, or a lower z/OS level with APAR OA49263, and set up the SMFPRMxx member to include the INMEM keyword and parameters. For more information, see [“Procedure for near real-time” on page 163](#). If you use the exit intercept method, you must configure CKQEXSMF as described in [“Assigning a userid and setting up the CKQEXSMF server started task” on page 174](#).

For these real time interfaces, QRadar SIEM might need to be updated to allow syslog input for the z/OS-related DSMs.

- You must set up a process to periodically refresh your CKFREEZE and UNLOAD data sets. See [“Use of a fresh CKFREEZE and UNLOAD each day” on page 42](#). Note that UNLOAD can only be used if the product has more than just the QRADAR\* entitlement (for example, AUDIT\* or ADMINRACF). However, if the product has only the QRADAR\* entitlement, an active or backup RACF database, a copy of the RACF database, an ACF2 backup database, or an inactive ACF2 database must be used.
- If you decide to use the file polling method to transport the LEEF data, you must have an active FTP (or SFTP) server on your z/OS image, so that QRadar SIEM can download those LEEF files.
- When using Transport Layer Security (TLS): ICSF must be active and the ICSF PKDS data set must be initialized when using cryptographic hardware. For more information, see *z/OS Cryptographic Services ICSF System Programmer's Guide* and *z/OS Cryptographic Services ICSF Administrator's Guide*.
- When using TLS: the certificate that is to be used must be added to the External Security Manager (ESM). The public key must be stored in the ICSF PKDS data set when using cryptographic hardware.

- When using AT-TLS, an AT-TLS policy must be created and activated. For more information, see section "Policy-based networking" in *z/OS Communications Server: IP Configuration Guide* and section "Policy Agent and policy applications" in *z/OS Communications Server: IP Configuration Reference*.

The zSecure configuration must contain the specific parameters for SIEM. For information, see ["Updating the configuration files for LEEF creation"](#) on page 166.

For instructions for installing and configuring zSecure, see the *Program Directory: IBM Security zSecure CARLa-Driven Components* and the first few chapters of the *IBM Security zSecure CARLa-Driven Components: Installation and Deployment Guide* (this manual).

## SMF records for the data collection process

---

Before starting the data collection process, you must:

1. Generate the SMF records. See ["Generating the SMF records"](#) on page 162.
2. Make the SMF records available for SIEM. See ["Make SMF records available to SIEM"](#) on page 163.

### Generating the SMF records

#### Before you begin

SMF processing must be turned on and appropriate records must be created and saved. The standard required SMF records are:

- 0, 7, 9, 11, 14, 15, 17, 18, 22, 26, 30, 36, 41, 42, 43, 45, 47, 48, 49, 52, 53, 54, 55, 56, 57, 58, 59, 61, 62, 64, 65, 66, 80 (RACF and Top Secret), 81 (RACF), 82, 90, most of 92, 118, and 119
- Selected subtypes of 102 that zSecure forwards to SIEM are Db2 IFCids 4, 5, 6, 7, 8, 9, 10, 22, 23, 24, 25, 55, 83, 87, 90, 92, 104, 105, 106, 107, 140, 141, 142, 143, 144, 145, 169, 177, 219, 220, 258, 270, 314, 319, 343, 361, 362, 370, 371, 373, and 404; see [Procedure](#)
- The CICS monitoring record type 110 subtype 1
- The ACF2 record type (site-defined number) if you have ACF2
- 83 subtype 4 for data from Linux® for IBM Z
- 83 subtype 5 for data from WebSphere® Application Server
- 83 subtype 6 for data from IBM Security Key Lifecycle Manager
- 83 subtype 7 for data from IBM Multi-Factor Authentication
- 123 subtype 1 for data from z/OS Connect

The following CARLa members specify the exact selection of which SMF records are to be transformed to text format:

- For QRadar: CKQLEEF and CKQLEEF L
- For ArcSight: CKQCEFG

These members can be updated by regular maintenance.

If you use installation-defined events, make sure to include the SMF records required in your CARLa member CKQCEF#C (for ArcSight) or CKQCES (for QRadar).

#### Procedure

- To generate SMF records for CICS transactions, set up and enable CICS monitoring.  
You can set up monitoring by data types and classes. For example, you can monitor the classes for exceptions, performance, and resources. To use CICS monitoring:
  1. Create a DFHMCTxy CICS Monitoring Control Table (MCT).
  2. Add MCT=xy to the System Initialization Table (SIT).



3. Run the CEMT INQ MON command to confirm or set one or both of the following:

- Monitoring on classes of monitoring data and options
- Classes of monitoring data and options

For more information, see the CICS Monitoring Facility documentation in the CICS Transaction Server library at <http://www.ibm.com/software/htp/cics/tserver/v53/library>.

You can also use the SET MONITOR command to change monitoring classes and options.

- For Db2, you must activate the Db2 trace in order to generate the required SMF records. Use the following commands.

These commands are intended as an example. In your installation, IFCIDs might already be logged to SMF by other traces. Verify and adapt these examples to meet the requirements of your installation.

```
-<subsysname> START TRACE(PERFM) DEST(SMF) CLASS(30) IFCID(6,7,8,9,10,19,22,90,  
107,177,314)  
-<subsysname> START TRACE(STAT) DEST(SMF) CLASS(30) IFCID(258)  
-<subsysname> START TRACE(AUDIT) DEST(SMF) CLASS(*)
```

## Make SMF records available to SIEM

If you decide to use the file polling method to transport the text-formatted SMF data, and are using SMF MAN datasets, do not specify your live SMF data sets as your only input. Doing so results in gaps: SMF records that are written between the last SIEM data collection run and a subsequent switch of the SMF data set, are missing.

### Procedure for near real-time

Using the direct near real-time SMF interface requires either of these input sources:

#### CKQEXSMF

The CKQEXSMF task can be used to intercept the SMF records using SMF exits. When using this method, you must ensure that the SMF record exits are enabled in your SMFPRM~~xx~~ parmlib member, and you must set up and start the CKQEXSMF started task. This method has no prerequisites on using SMF logstreams or INMEM resources.

#### SMF INMEM

When using SMF logstreams, you can add an in-memory resource name using SMF INMEM support. You define an additional in-memory destination of selected SMF records in your SMFPRM~~xx~~ parmlib member. The default INMEM name that zSecure uses is IFASMF.CKQRADAR. Also specify the required SMF types in the TYPE() parameter; for example:

```
INMEM(IFASMF.CKQRADAR,RESSIZMAX(128M),  
TYPE(0,7,9,11,14,15,17,18,22,26,30,36,41,42,  
43,45,47:49,52:59,61,62,64:66,80:83,90,  
92,102,110,118,119,230))
```

### Procedure for file polling

- If you are using SMF logstreams, the most convenient way to run data collection is by reading directly from a logstream. Make sure that the data collection for SIEM runs at least as frequently as the SMF retention period that you specified for your logstream. (You use the logstream administrative data utility, IXCMIAPU, to specify a logstream.) You might want to set up a dedicated logstream for this purpose.
- If you are using SMF data sets, you must prepare the input for the data collection during your SMF offload process. That is:

1. Add another DD-statement to your IFASMFDP program like:

```
//OUTDD2 DD  
DISP=(MOD,CATLG),DSN=your.prefix.D&YYMMDD..T&HHMMSS,UNIT=...,SPACE=...
```

2. Update the control statements for IFASMFDP to write simultaneously to your existing accumulation data sets and to the new data sets.

The collection process for SIEM can then use the DSNPREF parameter to retrieve the additional data sets and, after successful processing, delete these data sets.

The use of system symbols, as shown in this example, is supported only in started tasks and in job classes that have SYSSYM=ALLOW set. If you run your SMF offload as a batch job, you can use a generation data group (GDG). However, this approach has disadvantages in serialization. Consider converting to a started task.

- If you are creating daily SMF accumulation data sets and you intend to prepare SIEM data once a day, you can use the accumulation data sets as input. However, do not use, for example, a monthly accumulation data set as input for a daily preparation because, in that case, SMF records early in the accumulation are read multiple times. The high water mark (HWM) processing in zSecure skips records that were already processed, but the repeated reading takes processing resources. Especially when your accumulated SMF is written to tape, and the tape data set becomes multi-volume, reading the SMF accumulation might become prohibitive, both in processing resources and in contention for your tape drive and volume.
- If your data set contains records from multiple z/OS images, do not feed that data set directly into SIEM; this is not supported. Instead, run the collection process separately for each image and write each collection into a separate directory. Each SIEM collection process does not need to run under the z/OS image that it pertains to, as long as the collection process has access to the SMF, CKFREEZE, and UNLOAD from the image it pertains to. Use one of the following methods:
  - Specify an EXCLUDE statement in member CKQCEF#C (for ArcSight) or CKQCES (for QRadar). See “Updating the configuration files for LEEF creation” on page 166. Each collection process must have its own member in its zSecure configuration data set. With this method, the combined SMF is read multiple times.
  - First run a special CARLa job or jobstep against accumulation data sets and use the output from that job or jobstep as input for the separate instances of SIEM preparation. In the job, use SELECT statements to specify the SMF ID and UNLOAD statements to write the records to separate data sets for each z/OS image. With this method, the combined SMF is read only one time.
- When recovering a lost SMF interval, run a similar job to select the interval and SMF ID from your accumulation data sets.

## Setup of the collection process for QRadar

---

This section describes the configuration that is required to convert the SMF records to the LEEF format that QRadar uses. The near real-time and the file polling method are described separately.

### Before you begin

zSecure supplies two sets of members to achieve integration with QRadar through LEEF:

- The first set is for near real-time and uses the CKQ prefix and some variants ending in L (for 'Live').
- The second set also uses the CKQ prefix but without the L suffix.

Decide which data collection process you want to use, and select the appropriate set of members. Complete the following steps for your chosen process. For the near real-time process, zSecure provides a CKQRADAR procedure for running as a Started Task. For collecting and preparing data for the file polling process, zSecure supplies the CKQCLEEF procedure and the CKQJLEEF example batch job that uses that procedure. Files that the procedure creates contain a log of events, including security events. The permission bits of these files can be controlled using the UMASK statements in the CKQCLEEF procedure body. The default is UMASK=027, which excludes access for unspecified users.

The CKQCLEEF process runs a CARLa script from SCKRCARL (CKQLEEF) that essentially includes the following code:

```
imbed member=ckq0es
```

```

imbed member=ckqrenv esm=RACF

/* Primary selection of records */
NewList type=SMF name=SMFSEL DDname=CKREPORT
  imbed DDname=SMFHWIN
  imbed member=ckqxes
  summary system "|" date MinTime MaxTime "|" SMFdd count(10)

/* optional report : number of records per date per type */
NewList type=SMF DDname=CKREPORT ,
  title="Number of records per date per type"
  select likelist=SMFSEL
  summary type "|" date MinTime MaxTime "|" count(10)

/* optional report : number of records per type and SubType */
NewList type=SMF DDname=CKREPORT ,
  title="Number of records per type and subtype"
  select likelist=SMFSEL
  summary type SubType "|" count(10)

/* One mergelist for all record types for "device type" z/OS */
MergeList name=SMF DDname=CKQLOGZ
/* IPL - type 0 */
NewList type=SMF name=IPL nodup header=leef
  select likelist=SMFSEL type=0
  SortList datetime(nd) LEEF,
    | typ | '|' | dTF,
    datetime(Java_SimpleDate, 'devTime'),
    system('job'),
    recorddesc('sum')

...
imbed member=ckqces
...
EndMerge

```

The SMFSEL newlist acts as the central filter of SMF events. It applies the cut-off timestamp (also called the high water mark (HWM)) to eliminate records that might have been processed by a previous iteration of CKQCLEEF. The near real-time member CKQLEEF does not contain any code to process such a high water mark. The CARLa code also includes the following members:

### **CKQCES**

CARLa code that is to be processed at the start of processing. You can use this member, for example, when IBM or vendor software writes badly formatted SMF records that cause errors. In this case, IBM Software Support can supply a set of CARLa statements that can be used until the problems with the OEM vendor are solved or a more permanent solution is built.

### **CKQRENV**

Environmental specifications. This member applies only to RACF systems, where it specifies groups that represent privileged roles.

### **CKQXES**

This member excludes certain events; it contains installation-specific exclude statements. For more information see [“Record suppression exit” on page 171](#).

### **CKQCES**

CARLa code to customize the z/OS Log Source to also map installation-defined events. For example, you might have a product that creates its own SMF records.

## **Procedure**

1. Update the configuration files for LEEF data creation.
2. Check the contents of the general zSecure configuration member C2R\$PARM.
3. Optional: Customize contents of the text formatted SMF data.
4. Optional: If you want to use the file polling process, prepare a directory for the LEEF data.
5. Optional: If you want to use the near real-time process, set up a user ID for the Started Task.

## Updating the configuration files for LEEF creation

Use this topic to help you customize existing CKRPARM data set members for LEEF data creation (for QRadar SIEM).

### About this task

If you want to use a new zSecure configuration data set (often called CKRPARM, although you can use any data set name), run job CKRZPOST. See [Chapter 6, “Deployment of the software,” on page 23](#) for information.

You can use an existing CKRPARM data set, but if it was created by an older level of zSecure, some configuration members might be missing. If so, copy these members from the SCKRCARL and SCKRSAMP libraries for the near real-time process: CKQRENV, CKQSPECL, CKQCES, CKQXES and CKQOES

### Procedure

Now customize the members:

1. Adapt member CKQSPECL for near real-time, or CKQSPEC for file polling to specify your input and output:
  - Specify as input the active security database for the proper ESM or an UNLOAD and the CKFREEZE data set that you refresh every day with the C2RJPREP job. See [“Use of a fresh CKFREEZE and UNLOAD each day” on page 42](#). Using the active security database gives more up-to-date enrichment of the LEEF records, but requires READ access on the security database. If you are using Top Secret, remove the UNLOAD allocation, because that does not apply to Top Secret. If the product has only the QRADAR\* entitlement, then an active or backup RACF database, a copy of the RACF database, an ACF2 backup database, or an inactive ACF2 database must be used instead of UNLOAD. If the product has entitlements in addition to QRADAR\*, you can choose whichever you prefer.
  - Specify the IP address(es) of your SIEM system on the SYSLOGUDP or SYSLOGTCP parameter in CKQSPECL. You can specify a hostname that can be resolved, an IPv4 address, or, if IPv6 is enabled (dual mode TCP/IP stack), an IPv6 address.
  - For file polling, specify the SMF that you selected as input, either as the name of a logstream, or by using the DSNPREF parameter. See [“Make SMF records available to SIEM” on page 163](#). If you use the DSNPREF parameter, specify the DELETE parameter, so that the data sets that you created for this purpose during SMF offload are deleted after successful processing.
  - For file polling, specify the absolute path of the LEEF files. For log sources that do not occur in your system (for instance, for ACF2 when your system uses RACF, or the reverse), you can direct the output to /dev/null to avoid writing LEEF files that consist of only a zip header. For the other LEEF files, do not change the file name, and make sure that the directory part of the path matches the CKQPATH parameter.
  - For file polling, specify the absolute path for the time-based cutoff file. This file is the one identified as TYPE=SMFHWIN and TYPE=SMFHWOUT. Make sure that you specify the same file in both cases.  
**Note:** If you need to recover a lost SMF interval, blank out this file in order to prevent skipping the time period that you want to recover. After recovery is done, edit it back to the previous contents.
  - For file polling, the default audit settings for new files generate logs of successful and failed access. If your installation does not require logging of access to files containing LEEF data, modify the ALLOC FAUDIT specifications in CKQSPEC.
  - For file polling, specify output options if desired. The default options are in the zSecure-supplied sample members.
2. Adapt the environmental specifications.
  - For RACF systems, configure the privileged user groups in member CKQRENV. This member specifies groups that represent privileged roles. If a user is connected to a group listed in this member, events are annotated with the group name.
  - For ACF2 and Top Secret systems, this configuration does not apply.

**Note:**

- Do not start the process until you finish updating your configuration.
- The first run fails with a CKR0945 message because the cutoff file does not yet exist. There is no harm in this failure; the file is created during this first job or started task. You can restart the process.

## Setup of the collection process for Micro Focus ArcSight

This section describes the configuration that is required to convert SMF records to the Common Event Format (CEF) that ArcSight uses.

### About this task

zSecure supplies a procedure CKQCEF to cover all possible integration methods. A sample job CKQCEFJD is available in SCKRSAMP. Parameter members and sample exits all have names starting with CKQCEF.

The CKQCEF process runs a CARLa script from SCKRCARL (CKQCEFG) that essentially includes the following code:

```

imbed member=ckqcef#0 list /* initialization exit point */
/* environment parameters for RACF */
imbed member=ckqcef#r esm=RACF list
imbed member=ckqcef#a esm=ACF2 list
imbed member=ckqcef#t esm=TSS list

/* define fields */
....

/* default fields in header, can be overridden in parm member */
NewList type=SMF name=DEFAULTS outlim=0
/* required for TCP/UDP Syslog messages, blanks for log files.
  Extra blank is intentional separator, followed by | operator. */
Define CEF_datetime(16,cef_dt,noprefix) as datetime
Define CEF_hostname(5,noprefix) as system
Sortlist system

/* Primary selection of records */
NewList type=SMF name=SMFSEL DDname=CKREPORT PL=0
select likelist=SMFHW /* optional cut-off of older SMF */
imbed member=ckqcef#x list /* exclude SMF records exit point */
summary system('Syst') "|" date MinTime MaxTime "|" SMFdd count(10)

/* optional report : number of records per date per type */
NewList type=SMF DDname=CKREPORT PL=0,
title="Number of records per date per type"
select likelist=SMFSEL
summary type(4) "|" date MinTime MaxTime "|" count(10)

/* optional report : number of records per type and SubType */
NewList type=SMF DDname=CKREPORT PL=0,
title="Number of records per type and subtype"
select likelist=SMFSEL
summary type(4) SubType "|" count(10)

/* IPL - type 0 */
NewList type=SMF name=IPL syslog header=cef
select likelist=SMFSEL type=0
list CEF_datetime | CEF_hostname | CEF_header,
| header_type | '|IPL|0|' |,
system,
CEF_timestamp,
recorddesc
....

```

The SMFSEL newlist acts as the central filter of SMF events. It applies the cut-off timestamp (also called the high water mark (HWM)) to eliminate records that might have been processed by a previous iteration of CKQCEF. The actual CARLa code for generating and testing the HWM file is located in a separate member in SCKRCARL, CKQSMFHW. Similarly, member CKQSMFZZ prevents CARLa syntax errors when CKQSMFHW is not used. By including one or the other in the run-time parameters, the HWM mechanism is enabled or disabled, resp.

The CARLa code also includes the following installation exit members:

#### **CKQCEF#0**

CARLa code that is to be processed at the start of processing. You can use this member, for example, when IBM or vendor software writes badly formatted SMF records that cause errors. In this case, IBM Software Support can supply a set of CARLa statements that can be used until the problems with the OEM vendor are solved or a more permanent solution is built.

#### **CKQCEF#X**

This member excludes certain events; it contains installation-specific exclude statements. For more information, see [“Record suppression exit” on page 171](#)

#### **CKQCEF#C**

CARLa code to customize the SMF conversion process to also map installation-defined events. For example, you might have a product that creates its own SMF records.

#### **CKQCEF@A, CKQCEF@R, CKQCEF@T**

Environmental specifications. The members apply to ACF2, RACF and CA-TopSecret systems, resp.

For RACF systems, the sample code specifies groups that represent privileged roles, and causes these RACF group names to be included in the CEF requestContext extension field.

### **Procedure**

1. Update the configuration files for CEF data creation.
2. Check the contents of the general zSecure configuration member C2R\$PARM.
3. Customize contents of the text-formatted SMF data.
4. If you want to use the file polling process, specify dsnames for temporarily holding the CEF data, and a HWM data set for each LPAR.
5. If you want to use the near real-time process, set up a user ID for the Started Task.

## **Updating the configuration file for CEF data creation**

### **About this task**

If you want to use a new zSecure configuration data set (often called CKRPARM, although you can use any data set name), run job CKRZPOST. See [Chapter 6, “Deployment of the software,” on page 23](#) for information.

You can use an existing CKRPARM data set, but if it was created by an older level of zSecure, some configuration members might be missing. If so, copy member CKQCEFP from SCKRSAMP. Optionally, if you wish to customize the exits, copy the following members from the SCKRCARL library: CKQCEF@A, CKQCEF@R, CKQCEF@T, CKQCEF#C, CKQCEF#X, and CKQCEF#0.

### **Procedure**

Now customize the members:

1. Adapt member CKQCEFP for use by the CKQCEF started task. Alternatively, you can change the member name in the PROC statement of the CKQCEF member in PROCLIB.
  - a) Specify as input the active security database for the proper ESM or an UNLOAD and the CKFREEZE data set that you refresh every day with the C2RJPREP job.

See [“Use of a fresh CKFREEZE and UNLOAD each day” on page 42](#). Using the active security database gives more up-to-date enrichment of the CEF records, but requires READ access on the security database. If you are using Top Secret, remove the UNLOAD allocation, because that does not apply to Top Secret. If the product has only the QRADAR\* entitlement, then an active or backup RACF database, a copy of the RACF database, an ACF2 backup database, or an inactive ACF2 database must be used instead of UNLOAD. If the product has entitlements in addition to QRADAR\*, you can choose whichever you prefer.

- b) Specify the SMF that you selected as input, either as the name of a logstream, the CKQEXSMF exit name, a DD name or by using the DSNPREF parameter.

See “Make SMF records available to SIEM” on page 163. If you use the DSNPREF parameter, specify the DELETE parameter, so that the data sets that you created for this purpose during SMF offload are deleted after successful processing.

- c) For sending CEF messages using network packages, specify the IP address(es) of your SIEM system on the SYSLOGUDP or SYSLOGTCP parameter.

You can specify a hostname that can be resolved, an IPv4 address, or, if IPv6 is enabled (dual mode TCP/IP stack), an IPv6 address. Make sure the SIEM system listens to port 514 with the selected protocol (UDP or TCP), otherwise specify the port number too.

- d) For file polling or batch driven file transfer, do not use the OPTION SYSLOGTCP or SYSLOGUDP parameters, but specify OPTION SYSLOGTOFILE. The output data set must be allocated to DD name C2RSYSLG, with a JCL DD statement like this:

```
//* CEF formatted output, ASCII encoded, large data set
//C2RSYSLG DD DISP=(&DSTAT),DSN=&DPREF..&SYS..CEF,
//          UNIT=&UNIT,VOL=SER=&VOLSER,
//          SPACE=(32760,(1000,1000),RLSE,,ROUND),
//          LRECL=2052,RECFM=VB
```

The messages in this data set will be UTF8 (ASCII) encoded, so use file transfer in BINARY mode.

- e) When using an SMF accumulation data set as input, specify a UNIX file or z/OS data set to be used as cutoff file.

Use CARLa ALLOCATE commands or DD statements for DD=SMFHWIN and DD=SMFHWOUT. Make sure that you specify the same file or data set name in both cases:

```
//* high water mark keeps the last SMF time stamp, small data set
//SMFHWIN DD DISP=SHR,DSN=&DPREF..&SYS..SMFHWM
//SMFHWOUT DD DISP=SHR,DSN=&DPREF..&SYS..SMFHWM
```

The HWM function is switched on by an include statement in CKQCEFP:

```
imbed member=CKQSMFHW list      /* Generate cut-off file      */
```

or disabled:

```
imbed member=CKQSMFZZ nolist   /* No cut-off needed      */
```

**Note:** If you need to recover a lost SMF interval, blank out this file in order to prevent skipping the time period that you want to recover. After recovery is done, edit it back to the previous contents.

## 2. Adapt the environmental specifications.

For RACF systems, configure the privileged user groups in member CKQCEF@R. This member specifies groups that represent privileged roles. If a user is connected to a group that is listed in this member, events are annotated with the group name.

For ACF2 and Top Secret systems, specify SIMULATE commands in members CKQCEF@A and CKQCEF@T as needed.

## Sample batch jobs

The CKQCEF proc takes a parameter CKQPARM with the name of the CKQCUST member with run-time parameters.

The following JCL would send SMF records from a dump data set to an ArcSight server:

```
//SMFRUN EXEC CKQCEF,CKQPARM=BATPARMS
//SMF DD DISP=SHR,DSN=SMF.DUMP(0)
//CKFREEZE DD DISP=SHR,DSN=&DPREF..&SYS..CKFREEZE
```

With a member BATPARMS in CKQCUST:



```

    alloc type=RACF active /* only for RACF sites */
/* alloc type=ACF2 backup    only for ACF2 sites */

/* Or specify permanent SMF data sets.                                */
    alloc type=SMF         dd=SMF

/* Now specify CKFREEZE data set name to enrich events */
    alloc type=ckfreeze dd=CKFREEZE

/* Specification of syslog destination: DNS name or IP address      */
/* or like ::FFFF:127.0.0.1                                         */
    Option syslogtcp=MY.ARCSIGHT.SERVER

/* Prevent data loss                                                */
    Option MsgRc=(438,16) /* SMF input terminated: out of memory */
    Option MsgRc=(1788,4) /* ZFS unique-format auditids not unique */
    Option MsgRc=(2202,0) /* Conflicting jobname for ASID          */

/* Make sure we do not fail on start / restart due to a data set
   temporarily in use for a refresh or backup. You can adjust the
   maximum wait time in minutes.                                     */
    OPTION SERIALIZATION(ENQ(SYSDSN),WAIT,MAXWAIT(15))

    imbed member=CKQSMFZZ nolist /* No cut-off needed              */

```

Alternatively, parameters can be specified inline with the CKQPARM dd statement as illustrated by member CKQCEFJD in SCKRSAMP and CKRJOBS.

**Note:** CKQCEF internally uses the CARLa LIST command for SMF analysis and not SORTLIST; so it cannot make full use of the SMFCACHE function in zSecure. Fields like USERID may be missing from some CEF events for long running tasks when SMF records that contain the user ID were dumped in an earlier SMF data sets and CKQCEF now processes SMF records that lack the user ID. To reduce missing user ID fields in batch SMF processing, concatenate in JCL the previous SMF data set ahead of the data set that you need to process and use HWM to cut off the records from the first.

```

//SMFRUN EXEC CKQCEF,CKQPARM=HWMPARMS
//SMF DD DISP=SHR,DSN=SMF.DUMP(-1)
// DD DISP=SHR,DSN=SMF.DUMP(0)
//CKFREEZE DD DISP=SHR,DSN=&DPREF..&SYS..CKFREEZE
/* high water mark keeps the last SMF time stamp
//SMFHWM DD DISP=SHR,DSN=&DPREF..&SYS..SMFHWM
//SMFHWOUT DD DISP=SHR,DSN=&DPREF..&SYS..SMFHWM

```

## Optional customization of the SIEM data

This section describes optional configuration that changes the text formatted SMF data to better meet the installation's logging requirements.

Three installation exit members support changes to the contents of the text-formatted SMF data. This includes:

1. Changing the contents or formatting of fields for most record types, and other global changes.
2. Excluding events and record types from the audit trail.
3. Adding installation defined (custom) records to the audit trail.

### Initialization exit

QRadar member CKQ0ES.  
ArcSight member CKQCEF#0.

This member is included at the start of processing. You can use this member, for example, when IBM or vendor software writes badly formatted SMF records that cause errors. In this case, IBM Software Support can supply a set of CARLa statements to be used until the problems with the OEM vendor are solved or a more permanent solution is built.



You can also use this member to specify the right CCSID when translating, for example, double-byte character set (DBCS) characters to UTF-8. For example, for Japanese Latin extended Unicode, you can include the following CARLa statement:

```
OPTION MY_CCSID=1399
```

## Record suppression exit

QRadar member CKQXES.  
ArcSight member CKQCEF#X.

This member contains installation-specific EXCLUDE statements for the SMF selection. Each EXCLUDE command results in the exclusion of all SMF records that match in all newlists that generate text formatted records. You must be very precise with these EXCLUDE commands, or the effect might be more than what you aimed for. The following examples show CARLa EXCLUDE statements that can be used to suppress event data that user George caused:

- Suppress all events that *George* generates:

```
exclude user=George
```

- Suppress all events where RACF returned a return code 8 and possibly, in the future, other non-RACF records as well:

```
exclude user=George desc=viol
```

- To exclude only *George's* DATASET access violations:

```
exclude user=George event=access(viol) class=dataset
```

- To exclude all DATASET and RESOURCE access that is granted to *George* due to the operations attribute, but includes RACF commands that *George* issued due to operations:

```
exclude user=George event=access(success) racfauth=operations
```

The following examples show CARLa EXCLUDE statements that can be used to suppress events for a list of user IDs:

- To exclude all successful RACF event codes between 1 and 7 for specific user IDs:

```
exclude user=(known,user,ids), access<=read, desc=success event=allsvc(success)
```

- To exclude successful access recording for specific user IDs:

```
exclude user=(known,user,ids) access<=read, desc=success
```

Valid values for the ACCESS field in NEWLIST TYPE=SMF are: N/A, NONE, EXECUTE, READ, UPDATE, CONTROL, ALTER, or OWNER as described in the table "SMF record ACCESS field - available values"; see the field descriptions for the SMF newlist in zSecure CARLa Command Reference. By coding `access<=read`, you still include UPDATE and higher levels of access in the text-formatted records.

The following example shows a CARLa EXCLUDE statement that can be used to suppress events based on SMF record type. To exclude READ logging from SMF 14 or subtype 1 of SMF type 92 records:

```
exclude type=(14,92(1))
```

## Custom event exit

QRadar member CKQCES.  
ArcSight member CKQCEF#C.

This member can be used for CARLa statements to customize the text-formatted records to also map installation-defined events. For example, you might have a product that creates its own SMF records.

## Updating the configuration file for the near real-time process

---

### About this task

This section describes the additional configuration steps that are required when you use the near real-time process.

### Procedure

1. Specify the IP address(es) of your QRadar® or ArcSight® SIEM system on the SYSLOGUDP or SYSLOGTCP parameter in CKQSPECL or CKQCEFP.

You can specify a hostname that can be resolved, an IPv4 address, or, if IPv6 is enabled (dual mode TCP/IP stack), an IPv6 address. Specify the TLS Listen Port used by the QRadar Log Source on the SYSLOGTCP parameter in CKQSPECL when using TLS.

2. Specify the input source for the SMF records that you want to use.

The following options are possible:

#### INMEM

Specify the name of the INMEM resource that you have defined for use by CKQRADAR. The default INMEM name that zSecure uses is IFASMF.CKQRADAR.

#### CKQEXSMF

This specification requires the use of the GETPROC parameter on the ALLOC statement. Use the exact form as shown in the example:

```
alloc type=SMF ddname=smf0rec getproc=ckqio2pc
```

You must install and configure the CKQEXSMF started task. See [“Operation of the CKQEXSMF started task”](#) on page 176.

## Updating the configuration file for the file polling process for LEEF data

---

### About this task

This section describes the additional configuration step that is required when you use the file polling process for QRadar.

### Procedure

Verify the file retention period.

The CKQCLEEF procedure also contains a step to remove obsolete files in the UNIX System Services file system. Files in the designated directory are considered obsolete if they are older than a configured number of hours. The CKQFIN member specifies the number of hours that a file is considered valid; default is 72 hours.

## Check the contents of the general zSecure configuration member

---

The default configuration member that zSecure uses is C2R\$PARM, but you might have other, different versions. The zSecure configuration member must be available to the JCL that you use for the collection process. For a started procedure, this is a JES procedure library. For a job, you can use the JCLLIB statement to specify any other data set.

The C2R\$PARM member contains several SIEM-specific parameters:

### **CKQCUST**

The name of the data set that contains the configuration members that zSecure uses for the SIEM interface. Updating these members is described in the following sections:

- [“Updating the configuration files for LEEF creation” on page 166](#)
- [“Updating the configuration file for CEF data creation” on page 168](#)
- [“Optional customization of the SIEM data” on page 170](#)

### **CKQPATH**

The UNIX directory where the file polling process is to create its LEEF data for QRadar.

## **Assigning a userid and preparing a directory to store the LEEF data**

---

### **About this task**

If you use the file polling process to transport the LEEF data, the data is stored in the USS file system. This section describes the required steps for preparation of the UNIX System Service file system. If you use the SMF INMEM real-time interface process, you can skip this step.

If you run multiple data preparation processes (for multiple z/OS images), each process must have its own (sub)directory.<sup>1</sup> Like any UNIX directory, it must have both an owning user and an owning group. You probably want to use the home directory of the userid that runs the collection process (or a subdirectory of the home directory), and you probably want to use a dedicated file system.

zSecure provides jobs to create the required user and group, home directory, and file system:

- Jobs CKQAUSR for RACF
- Jobs CKQAUSA for ACF2
- Jobs CKQAUST for Top Secret

### **Procedure**

1. Select the job that applies to your External Security Manager (RACF, ACF2, or Top Secret).
2. Adapt the job according to your conventions for users, groups, uids, gids, and data sets.
3. Depending on your choice for a batch job or a started task, uncomment the actions that create a SURROGAT or a STARTED profile, (for RACF, or the ACF2 or Top Secret equivalent). This step ensures that the process runs under the designated userid.
4. Specify the size of the file system, based on the amount of SMF data that is generated between two consecutive retrievals by QRadar SIEM. Allow for a margin to accommodate for SMF peaks or QRadar outages.
5. Run the job. Make sure that the file system is mounted after subsequent IPLs. You might want to use the automount facility.

## **Assigning a userid and setting up the CKQRADAR or CKQCEF started task**

---

### **About this task**

This task is for near real-time QRadar and ArcSight support. For this method, the creation of the LEEF format data is best run by using a started task. Follow the steps in this section so set up the started task.

---

<sup>1</sup> Each collection process should have its own (sub)directory, so that each LPAR is represented by a unique file "maxdate".

## Procedure

1. Copy the CKQRADAR or CKQCEF procedure from SCKRPROC to the z/OS systems that you want to monitor in near real-time. Check the applicability of the SYSOUT class A, which is the default for the STATOUT and PRTOUT parameters.
2. In the configuration file (see [“Updating the configuration file for the near real-time process”](#) on page 172) you have specified the input source for the SMF records. If you have specified INMEM, you must make sure that the started task user ID has READ authority to FACILITY IFA.IFASMF.CKQRADAR or the INMEM resource that you specified. The SAF resource name uses the qualifier IFA in front of the INMEM resource name.
3. Additional authorizations are required when using TLS:
  - The started task user ID must have READ access to FACILITY IRR.DIGTCERT.LISTRING.
  - The started task user ID must have READ access to profiles in the classes CSFSERV and CSFKEYS when ICSF is used. For more information, see *z/OS Cryptographic Services ICSF Administrator's Guide*.

## Assigning a userid and setting up the CKQEXSMF server started task

---

### About this task

This task is for near real-time QRadar support. This task must be performed only if you decide to use the SMF Exit Collection method. Follow the steps in this section so set up the required CKQEXSMF started task.

For more information about starting and managing the CKQEXSMF started task, see [“Operation of the CKQEXSMF started task”](#) on page 176.

### Procedure

1. Set up a profile in the STARTED class to specify the ID that you want to use for the started task.
2. Copy the CKQEXSMF procedure from SCKRPROC to the z/OS systems that you want to monitor in near real-time. Check the applicability of the SYSOUT class A, which is the default for the ERROUT and PRTOUT parameters.

It is also possible to run the CKQEXSMF program directly under the MSTR subsystem. In that case, you must change the specification of CKQDEBUG to point to a pre-allocated dataset, comment out the ERROUT and PRTOUT parameters, and remove the SYSTSPRT and SYSPRINT DD statements. The following is an example of such a changed procedure:

```
//CKQEXSMF PROC REGSIZE=256M,      Region for zSecure SMF collector
//  CONFIG=C2R$PARM,              Configuration member
//  PPARM=CKQEXSMF                CKQEXSMF parameter member
//*
//          INCLUDE MEMBER=&CONFIG.
//CKQEXSMF EXEC PGM=CKQEXSMF,REGION=&REGSIZE,DYNAMNBR=100,
//          TIME=NOLIMIT,MEMLIMIT=8G
//STEPLIB DD DISP=SHR,DSN=&CPREFIX..SCKRLOAD
//PARMLIB DD DISP=SHR,DSN=&CKQCUST.(&PPARM)
//CKQDEBUG DD DISP=SHR,DSN=&DPREF..CKQDEBUG
```

3. Ensure that the required SMF exits have been defined in your SMFPRMxx in PARMLIB.

For z/OS 2.3 or higher, the SMF exit that must be enabled is IEFU86. For z/OS 2.2 and older, exits IEFU83, IEFU84, and IEFU85 must be enabled. Each exit point is used in a specific environment and for specific SMF records. If IEFU86 is not enabled on z/OS 2.3 or higher, zSecure uses the z/OS 2.2 exit routines as a partial fallback. However, this might cause a failure to collect certain events. Ensure that these exit points are enabled for the entire system and for all defined subsystems. For example:

```
SYS(EXIT(IEFU86))
```

or

```
SYS(EXIT(IEFU83,IEFU84,IEFU85))
```

Incorrect specifications in SMFPRMxx can lead to missing information about some events. For example, omitting OMVS(EXIT(IEFU84)) makes RACF access checks in z/OS UNIX invisible.

4. Start the CKQEXSMF started task before starting the CKQRADAR task that specifies input from the CKQEXSMF started task.

## Operation of the CKQRADAR and CKQCEF started task

---

Start the real-time SMF processing task with the following command:

```
START procname
```

The CKQCEF started task member supports an optional parameter CKQPARM=*membername* to use an alternate parameter member.

When the real-time security event monitoring starts in CKQRADAR, WTO message CKR2099 is issued with routing code 9, at the same time that message CKR0450 is written to the SYSPRINT. CKR2099 helps automated operations scripts keep track of the security event monitoring. CKR2099 is not issued during the hourly RESTART; the SMF feed has no gap during a restart.

When running, the real-time SMF processing started task accepts commands from the system operator. The operator commands that are enabled are STOP (or P) and MODIFY (or F). These operator commands are currently recognized only during input processing of SMF records. If the system generates few or no SMF records, the response to operator commands might be significantly delayed.

### **STOP *procname***

The program stops reading or waiting for SMF records. Records that have already been read are processed and when applicable, summary reports are created. The effect of the STOP command is similar to a regular end-of-file on the SMF record input source.

### **MODIFY *procname,action***

The MODIFY command requires an additional keyword to specify the required action. The following actions are supported on the MODIFY command:

#### **STOP**

Same as the STOP operator command.

#### **ATTN**

This action has the same effect as pressing the ATTN key in a TSO session. Current<sup>®</sup> processing is terminated, and output files are close normally.

#### **CANCEL**

Same as the ATTN command.

#### **DISPLAY**

This action requests information about the current status of SMF processing. Messages like CKR3014 are written by using a WTO (Write To Operator) macro to write messages to job and system logs. The messages are also visible in the job log and JES SYSLOG.

#### **RESTART**

This modify command can be used to request a restart of the program; for example, to pick up updated information in the security database or in a CKFREEZE file. Note that this deletes and replaces the current content of the output data sets of the currently running program unless they were routed to spool or allocated with DISP=MOD. After a manual F *jobname*, RESTART command, any restart interval that is set with OPTION RESTART\_INTERVAL is restarted.

#### **Other**

An unrecognized *action* results in message CKR3015.

## Operation of the CKQEXSMF started task

Start the zSecure SMF Collector started task using the following operator command:

```
START CKQEXSMF
```

This started task shares the zSecure SMF exits with zSecure Alert:

- If zSecure Alert is not yet active, the SMF exit routines are installed as Dynamic Exit routines.
- If zSecure Alert (C2POLICE) is already active, the exits installed by C2POLICE send selected events to the buffers that CKQEXSMF maintains and the other way around.
- If you stop either CKQEXSMF or C2POLICE, the exits continue to be used for the other task.
- If you stop both CKQEXSMF and C2POLICE, the exits are uninstalled.

It is important that the same level of the zSecure code is used for CKQEXSMF and C2POLICE. If you are also planning to use zSecure Alert (C2POLICE), you cannot start the C2POLICE and the CKQEXSMF tasks at the same time; you must separate the start times by a few seconds. You must also issue STOP commands with several seconds in between.

### zSecure SMF Collector START parameters

For normal execution of the zSecure SMF Collector (CKQEXSMF), you do not need to specify any startup parameters. By default, CKQEXSMF detects if it is already active and issues an appropriate error message before ending. CKQEXSMF is designed to use system resources effectively. If the CKQEXSMF started task has been shut down previously, the newly started task reuses those critical system resources that can be obtained only once and that cannot be returned to the system.

In some error situations, the CKQEXSMF started task fails to initialize. In these situations, you might need to specify one of the optional START parameters.

The following example shows a START command with the DEBUG parameter specified:

```
S CKQEXSMF , , ,DEBUG
```

#### DEBUG

Issues diagnostic messages during the first part of the initialization. These diagnostic messages can also be used to determine possible problems in processing the standard PARMLIB parameters. This setting is in effect until a subsequent DEBUG command is issued either from the operator console, or using PARMLIB.

#### FORCE

Forces initialization to continue regardless of a previous execution. During normal operation, it is not necessary to use the FORCE option. Use the FORCE option only when the CKQEXSMF started task cannot be started using other methods and IPLing the system is undesirable. In this case, also create a problem report so that the issue can be investigated.

#### DEBUG-FORCE

Activates both the DEBUG and FORCE options at startup. The started task procedure CKQEXSMF provided with zSecure also provides the PPARM parameter to specify the main PARMLIB member that initializes the CKQEXSMF started task. This parameter can be used to override the value that is specified in the procedure; the default value is CKQEXSMF.

### MODIFY command to monitor or modify the CKQEXSMF started task

When the CKQEXSMF started task is active, the console operator can monitor or modify operations using the MODIFY console command. You can use the F command as an alias for the MODIFY command. The following example shows a MODIFY command that displays the current status and options for CKQEXSMF:

```
MODIFY CKQEXSMF,DISPLAY
```

Ensure that the text after the comma is one of the supported operator commands for the CKQEXSMF started task. For information about these commands, see [“CKQEXSMF configuration statements” on page 177](#) and [“CKQEXSMF operator commands” on page 182](#).

## Stopping the CKQEXSMF started task

When you stop the CKQEXSMF started task, a CKQRADAR started task that uses the data that CKQEXSMF collects is also stopped. To stop CKQEXSMF, issue the STOP (alias: P) command from the console. For example:

```
P CKQEXSMF
```

The STOP command can also be issued as the parameter on the MODIFY command:

```
F CKQEXSMF ,STOP
```

For more detailed information about these commands, see [“CKQEXSMF configuration statements” on page 177](#) and [“CKQEXSMF operator commands” on page 182](#).

## Configuration of the zSecure SMF Collector using parmlib

By default, the PARMLIB DD statement refers to the CKQEXSMF member in the CKQCUST data set. The following commands are examples of commands that can be specified in parmlib:

### DEBUG

Diagnose problems.

### OPTION

Manage the in-memory data buffers.

### FILTER

Select required SMF record types.

### REPORT

Specify the frequency of certain debug messages.

The input parameters can be specified in the form of commands with keywords. Use TSO conventions when specifying these commands. For details about the CKQEXSMF parameter file, see [“CKQEXSMF configuration statements” on page 177](#) and [“CKQEXSMF operator commands” on page 182](#).

## CKQEXSMF configuration statements

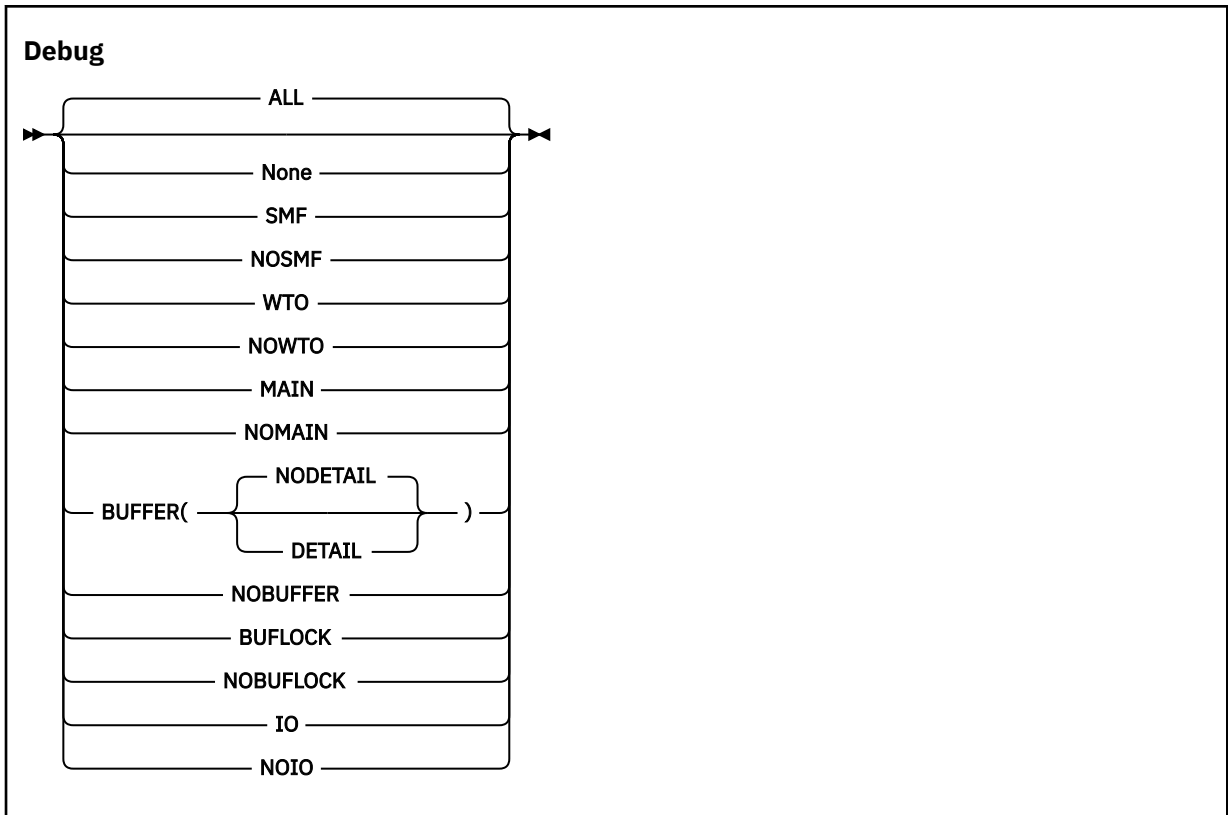
CKQEXSMF supports configuration statements that can be issued only once during initialization. These must be specified in the CKQEXSMF PARMLIB.

Of the configuration statements that are described in this section, only the OPTION and REPORT statements are required. If you want to obtain diagnostic information, use the DEBUG command.

### DEBUG command

The syntax of the DEBUG statements is provided in the following diagram.

**Note:** You can specify only one option at a time. To enable multiple debug options, issue the DEBUG command multiple times. The DEBUG command can be issued from the operator console or from the parmlib.



The keywords and variables have the following values:

**All**

This default level specifies that all diagnostic messages must be written to the console. Most of these messages are intended to assist during problem determination and are not intended for routine customer usage. Use the messages resulting from DEBUG BUFFER routinely to show statistics about the number of SMF records that were collected.

**None**

Deactivates creation of all diagnostic messages.

**SMF**

This option is intended for internal diagnostics. It currently has no effect.

**NOSMF**

This option is intended for internal diagnostics. It currently has no effect.

**WTO**

This option is intended for internal diagnostics. It currently has no effect.

**NOWTO**

This option is intended for internal diagnostics. It currently has no effect.

**MAIN**

Diagnostic messages that are related to mainline processing are written to the console. It includes responding to operator commands, initialization and management of all subtasks, and major buffer management functions.

**NOMAIN**

Diagnostic messages that are related to mainline processing are not written to the console. It includes responding to operator commands, initialization and management of all subtasks, and major buffer management functions.

**BUFFER**

Buffer usage statistics are written to the console, joblog, and syslog at the end of each reporting interval. These messages can be used to determine the number of SMF records that were



captured and the amount of storage that is required. If you include the `DETAIL` keyword, counts for the SMF records and subtypes are included. The following example illustrates resulting output:

```

+-----+
| CKQ0333I Buffer stats for buffer 08
| CKQ0325I Buffer stats: SMF(cnt,len) 00000256-00276854
| CKQ0544I Rectype Subtype Count
| CKQ0544I      14          2
| CKQ0544I      15          1
| CKQ0544I      30 total    4
| CKQ0544I          2          4
| CKQ0544I      42          6
| CKQ0544I      80 total    3
| CKQ0544I          2          1
| CKQ0544I          38          1
| CKQ0544I          39          1
| CKQ0544I          102         2
| CKQ0544I          119         1
+-----+

```

**NOBUFFER**

Buffer usage statistics are not written to the console.

**BUFLOCK**

This debug option is intended to assist in diagnosing the reason that a task could not save a record in the CKQEXSMF buffers. If such a situation occurs, an SVC dump is created of the address space where the event occurred. The BUFLOCK option is automatically disabled until the `DEBUG` command is issued again, either through an operator command or from PARMLIB.

Note that the SVC dump is not an indication that any error occurred. The dump is created only to assist in determining why the task could not save the record.

**NOBUFLOCK**

The BUFLOCK debug option is not used.

**IO**

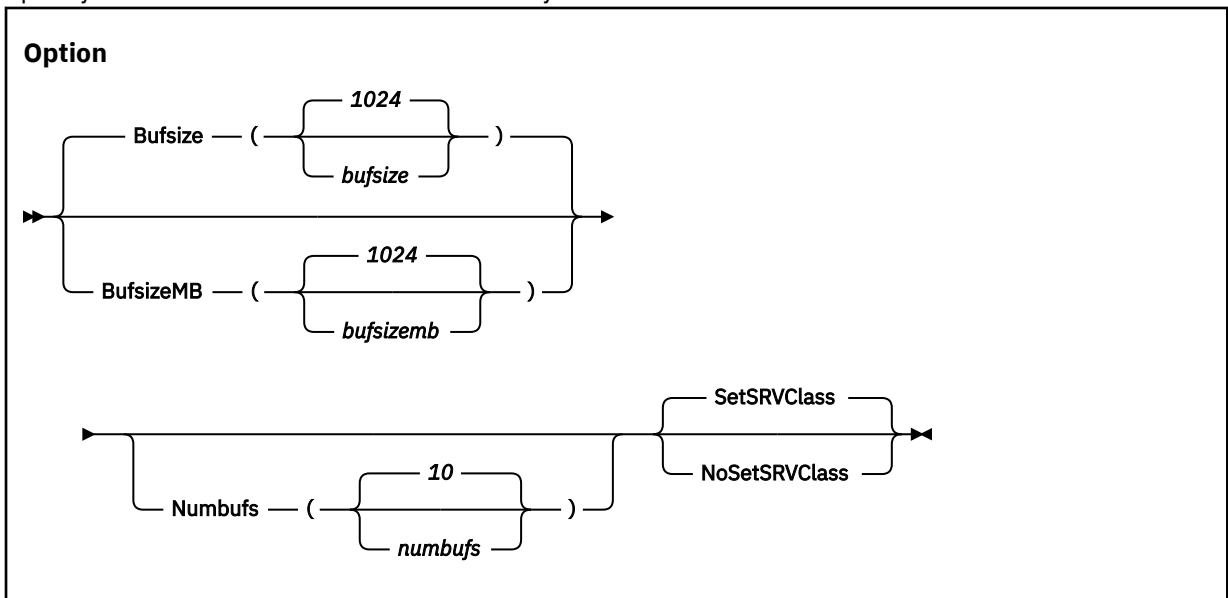
Specifies that all operations that the zSecure SMF Collector processes must be traced through SYSLOG. It might result in large numbers of WTO messages. This function is intended to be used by IBM support personnel to assist in diagnosing internal problems in the product.

**NOIO**

I/O diagnostic messages are not to be generated.

**OPTION command**

The `OPTION` command is only valid from the parmlib. The main purpose of the `OPTION` command is to specify the number and size of the in-memory data buffers.



The keywords and variables have the following values:

### **Bufsize or BufsizeMB**

The `Bufsize` or `BufsizeMB` keyword can be specified only when the `OPTION` statement is used during startup or during `RESTART` processing. It is ignored during `REFRESH` processing.

The `Bufsize` or `BufsizeMB` keyword can be specified only when the `OPTION` statement is processed from `PARMLIB` during startup or `RESTART` processing. `Bufsize` or `BufsizeMB` specifies the size of the in-memory buffers. In contrast to the buffers that are specified for `zSecure Alert` and `zSecure Access Monitor`, the `zSecure SMF Collector` buffers are used only to handle fluctuations in the number of `SMF` records. The buffers are also used to retain records during a possible `STOP` and `START` of the `CKQRADAR` started task. If more records must be retained than fit in the total buffer space, the oldest records are dropped.

If you use the `Bufsize` keyword, specify the required buffer size in kilobytes. If you use the `BufsizeMB` keyword, specify the size in megabytes. Valid sizes for the buffers are between 1 kilobyte - 1 gigabyte. The size that you specify is rounded up to the nearest megabyte. If you use both keywords in an `OPTION` statement, the program uses the last value that you specified. The buffers are allocated in 64-bit storage and count towards the specified `MEMLIMIT` of the started task. In general, it is more efficient to specify, for example, 10 buffers of 1 megabyte instead of 2 buffers of 5 megabytes. The size that is specified in the default `CKQEXSMF` `parmlib` member is 16MB.

### **Numbufs**

The `Numbufs` keyword can be specified only when the `OPTION` statement is used during startup or during `RESTART` processing. It is ignored during `REFRESH` processing.

`Numbufs` specifies the number of allocated buffers. The value `numbufs` must be between 2 - 32 inclusive. In contrast to the buffers that are specified for `zSecure Alert` and for `zSecure Access Monitor`, the `zSecure SMF Collector` buffers are used only to handle fluctuations in the number of `SMF` records. The buffers are also used to retain records during a possible `STOP` and `START` of the `CKQRADAR` started task. If more records must be retained than fit in the total buffer space, the oldest records are dropped.

The number of buffers that are specified in the default `CKQEXSMF` `parmlib` member is 32.

### **SETSRVCLASS, NOSETSRVCLASS**

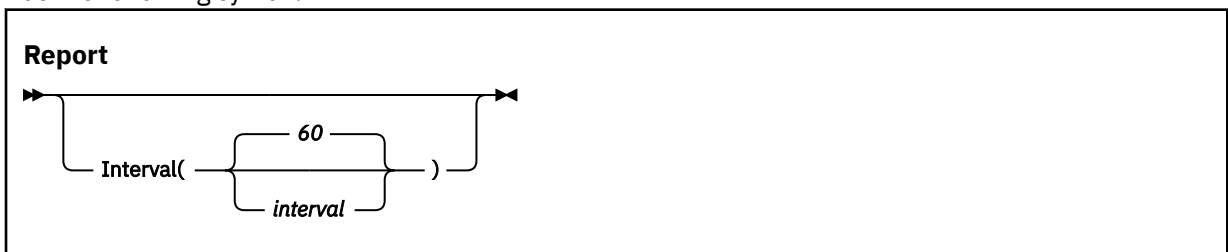
Determines whether the started task runs within the `SYSSTC` service class. The `SYSSTC` service class is intended for high priority system tasks. The collection of events and saving the collected data should be done in a timely manner. Therefore, a sufficiently high priority setting is important. If you must assign a lower dispatching priority, create an applicable `WLM` classification rule that assigns the required `WLM` service class. You must also specify `NoSetSRVClass` on the `OPTION` statement to prevent the program from resetting itself to the `SYSSTC` service class.

If you are running on a single processor system, careful analysis of your workload is required and using `NoSetSRVClass` might be preferred.

The default value of this option is `SETSRVCLASS`.

### **REPORT command**

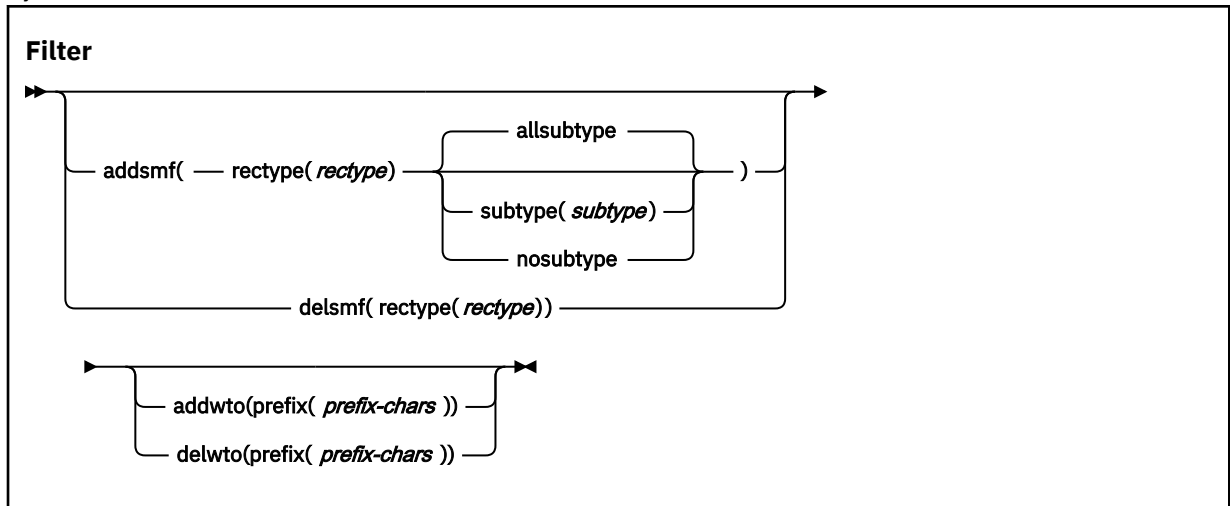
The `REPORT` command controls the timing of the `DEBUG BUFFER` messages. The `REPORT` command has the following syntax:



The keywords and variables have the following value: Interval. It specifies the time interval (in seconds) at which zSecure SMF Collector monitors the collection process and generates relevant messages. Valid time intervals are 10 - 3600 seconds. The interval specified in the default CKQEXSMF parmlib member is 300 seconds.

### FILTER command

The filter criteria are used to limit the amount of data that is collected in the in-memory buffers for further processing. By using the FILTER command, it is possible to eliminate unused events early in the process, thus increasing the overall efficiency. If there are no SMF filter criteria specified, all SMF records messages are collected for further processing. The FILTER command has the following syntax:



The following section describes the possible keywords and parameters.

### ADDSMF

Specifies the additional filter criterion to be used for SMF-records. You can repeat the FILTER command to specify as many filter criteria as you need. The criterion you specify is added to the already active criteria. The SMF-record type to be selected is specified by the `rectype` and `subtype` parameters. The available suboptions are:

#### Allsubtype

Specifies that all SMF-record subtypes are included in the record filter (default). This specification can also be interpreted as the absence of any filtering on subtype. Subtypes are used only for SMF-record types 30, 80, 92, and ACF2. For all other SMF-record types, the subtype specification is ignored.

#### Rectype

Specifies the SMF-record type that must be selected or that must no longer be selected. The `rectype` parameter must have a numeric value 0 - 2047, or the value ACF2 to specify records generated by ACF2.

#### Subtype

Specifies the SMF-record subtype that must be selected. The `subtype` is only used for SMF-record types 30, 80, 92, and ACF2. For all other SMF-record types, the subtype is ignored. The value of `subtype` must be numeric or a single alphabetic character. The subtype is interpreted as follows:

##### Rectype 30

The `subtype` is the standard SMF-record subtype. Although currently, SMF-Record type 30 has defined only subtypes 1 - 5, the range that zSecure accepts is 1 - 8.

##### Rectype 80

The `subtype` is the RACF event code. For a complete list of RACF event codes, see RACF Auditor's guide. The range that zSecure accepts is 1 - 255.

### Rectype 92

The *subtype* is the standard SMF-record subtype. Although SMF-Record type 92 currently has defined only subtypes 1 - 17, the range that zSecure accepts is 1 - 255.

### Rectype ACF2

The *subtype* is the ACF2 record type. For a complete list of ACF2 subtypes, see the ACF2\_SUBTYPE field in NEWLIST TYPE=SMF in the *CARLa Command Reference*.

### Nosubtype

Specifies that the SMF-record subtype, as described previously for the Subtype keyword, must not be used as a selection criterion. Use of this keyword resets all subtypes that were previously specified for the indicated *rectype*.

### DELSMF

Specifies that you no longer want the specified SMF record type to be selected. The SMF record type is identified by the *rectype* parameter only. It is not possible to deactivate SMF record selection per subtype.

### ADDWTO

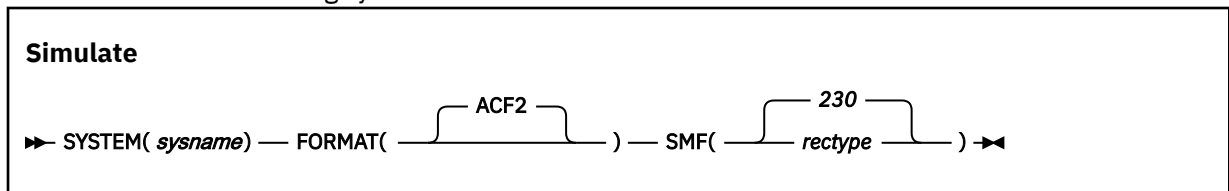
Specification of this keyword has currently no effect.

### DELWTO

Specification of this keyword has currently no effect.

## SIMULATE command

For normal operations, the SIMULATE command is not required because zSecure SMF Collector uses the documented interface to obtain the SMF-record type that ACF2 uses. The SIMULATE command is required only when this process fails. The command has various keywords and required parameters that zSecure SMF Collector currently does not use. These keywords and parameters are included for consistency with the zSecure Admin and Audit syntax of the SIMULATE command. The SIMULATE command has the following syntax:



The following section describes the possible keywords and parameters.

### System

Specifies the system name to which this SIMULATE command applies. Currently, the value for *sysname* is ignored. You must specify the SMF\_ID of the current system.

### Format

The only supported parameter ACF2 indicates that this SIMULATE command is used to specify ACF2 specific options.

### SMF

Specifies the SMF-record type for the ACF2 generated SMF-records. The parameter *rectype* must be numeric with a value 1 - 255. The default value is 230.

## CKQEXSMF operator commands

CKQEXSMF supports operator commands that have a direct effect on the running task.

When entered as an operator command, these commands do not require additional keywords. You can also abbreviate a command to the first four characters; for example, type DISP for the DISPLAY command.

### CRSH

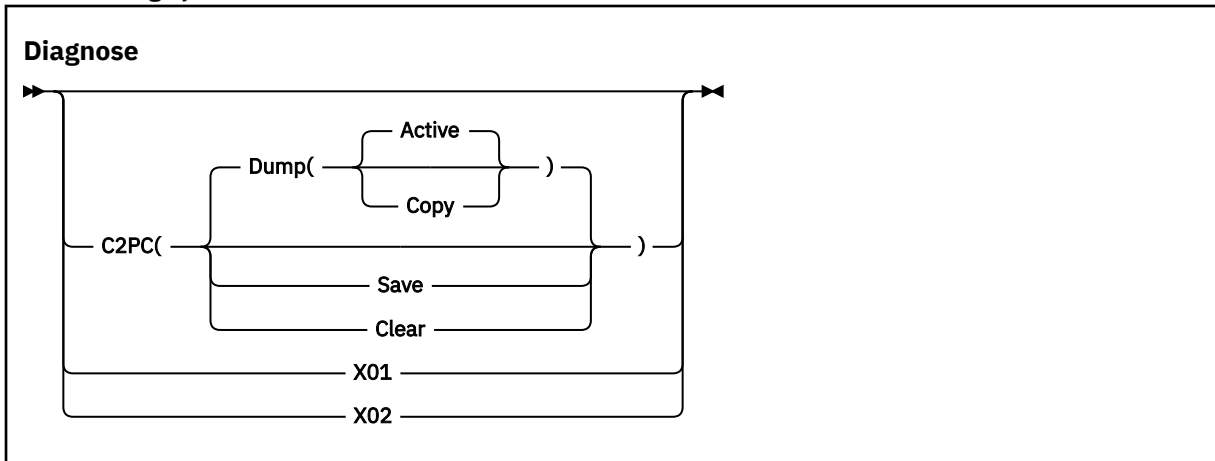
This command causes an immediate abend of the CKQEXSMF started task, without any recovery. It is intended for IBM internal testing only.

## DEBUG

Controls the diagnostic and monitoring messages that the program can generate. The command is effective immediately until a restart. For a complete description of the keywords for the DEBUG command, see [“DEBUG command” on page 121](#).

## DIAGNOSE

The DIAGNOSE command is intended for IBM support personnel to diagnose problems. It allows dumping internal control blocks and testing certain recovery scenarios. The DIAGNOSE command has the following syntax:



The keywords and parameters have the following values:

### C2PC

Information from the internal C2PC control block is to be saved or displayed on the operator console. This information is intended for IBM support personnel only. The following suboptions are available:

#### Dump

The Active or saved Copy of the C2PC data area is displayed on the operator console in dump format.

#### Save

The active C2PC data area is saved in the C2PC copy area.

#### Clear

The saved copy of the C2PC data area is cleared (reset to binary zeros).

### X01

This keyword is intended for IBM internal use only. It causes certain error situations which are used to test recovery actions.

### X02

This keyword is intended for IBM internal use only. It causes certain error situations which are used to test recovery actions.

## DISPLAY

Displays the current status and option settings for the zSecure SMF Collector. The display includes the current option settings, buffer space used, the number of the buffer currently in use, and the status of several error indicators if they are set. The DISPLAY command does not support any additional keywords.

## REPORT

Sets the values for the keywords that control the processing of the captured data. The new interval value is used after the current interval ends. For a complete description of the keyword for the REPORT command, see [“REPORT command” on page 127](#).

## RESTART

Gracefully shuts down the zSecure SMF Collector and immediately re-initializes the task. The address space in which the zSecure SMF Collector started task is running is not stopped and no additional console operator commands are needed to reactivate CKQEXSMF processing.

The main difference between a RESTART and a STOP command that are followed by a START command is the preservation of the Address Space ID (ASID). During the time that is required to process the RESTART command, SMF records are not collected. Also, any task that processes the collected SMF records directly from the CKQEXSMF task is terminated during the period that no SMF records are collected.

The RESTART command does not support any additional keywords.

### SIPL

Issue this command only at the request of IBM Software Support personnel or when explicitly required during release migrations. When the command is run, all in-memory data structures are freed, a system-level linkage index (LX) is lost, and the address space is marked as non-reusable. System level LXes are a limited resource that cannot be recovered without an IPL of the system. If you upgrade the zSecure SMF Collector program, the installation instructions might require you to shut down the previous version using this SIPL command.

The SIPL command does not support any additional keywords.

### STOP

Gracefully shuts down the zSecure SMF Collector started task. After the task ends, some memory remains reserved so that critical system resources can be used during a subsequent restart of the task. The effect of the STOP modify command is identical to that of the MVS STOP command.

The STOP command does not support any additional keywords.

## QRadar log source properties

In QRadar SIEM, a z/OS image is represented by a number of Log Sources: one for z/OS itself and one for RACF, ACF2, or Top Secret. In addition, if Db2 and CICS are active on your z/OS image, the image also contains Log Sources for these products.

Figure 11 on page 184 shows an example screen:

Log Source Name	<input type="text"/>
Log Source Description	<input type="text"/>
Log Source Type	IBM z/OS
Protocol Configuration	Syslog
Log Source Identifier	<input type="text"/>
Enabled	<input type="checkbox"/>
Credibility	5
Target Event Collector	eventcollector0 :: qradar
Coalescing Events	<input type="checkbox"/>
Incoming Payload Encoding	UTF-8
Store Event Payload	<input checked="" type="checkbox"/>

Figure 11. Log source properties configuration for QRadar

Perform the following steps in order:

1. Select the Log Source Type.

2. Select the Protocol Configuration:

- For Real Time, select **Syslog**.
- For Real Time using TLS, select **TLS Syslog**.
- For File Polling, select **Log File**.

QRadar might need an update to be able to select **Syslog** and/or **TLS Syslog**.

The window then changes and shows a different set of fields. For information about these fields, see *QRadar DSM Configuration Guide*.





---

## Chapter 16. Data preparation for Guardium Vulnerability Assessment

This topic describes the actions that you must take to use zSecure Audit to provide input for advanced auditing of your Db2 environment with IBM Guardium Vulnerability Assessment (Guardium VA).

Information about the RACF protection of Db2 objects and information about users and groups is loaded into Db2 tables for Guardium VA usage. This information is especially relevant if you use RACF groups as Db2 AUTHID or if you use the RACF Access Control Module DSNX@XAC. Guardium VA expects the data about a specific Db2 subsystem to be available in that Db2 subsystem itself. You can then directly combine the information from the Db2 catalog with the security information provided by zSecure. To load the data in a Db2 subsystem, zSecure provides examples in SCKRSAMP. If you want to modify and use these examples, copy them to another data set. Because the SCKRSAMP data set is SMP/E-controlled, future updates might overwrite your modifications.

When you have completed these steps and the appropriate Db2 tables have been loaded, you can benefit from the enhanced RACF information. In Guardium VA, you can use the Guardium VA entitlement reports and vulnerability tests that have names that start with zSecure.

To create and manage a Db2 database that contains the zSecure provided data to Guardium VA, you must take the following actions:

1. Create one or more Db2 databases.
2. Create one or more Db2 table spaces.
3. Create Db2 tables.
4. Load data into the tables.

The first three steps are the initial setup of the database. These steps are required only once, after you have initialized the data manager. When the tables are established, you can LOAD data into the Db2 database repeatedly. For example, you can refresh data each day. This is left up to the installation. At any time, you can delete your current table data. You can also manage these tables with regular Db2 utilities and SQL statements. The following sections describe the samples that can be used to create and load zSecure data for use by Guardium VA.

Because the example jobs directly interact with a local Db2 subsystem, they must be run on each system. Use of remote input sources or processing multiple systems in a single run is not supported. Before you run these jobs, ensure that the zSecure configuration (that is, member C2R\$PARM or a custom copy of that member) was customized with the correct values. You must also replace occurrences of !! by the correct level of Db2 in use, and the value !DSN! by the name of your Db2 subsystem. The provided example jobs use the SCKRSAMP data set. Change the name to the data set that you used for your modified copy of the example members.

The Db2 steps use the standard DSNUTILB Utility program and the DSNTEP2 productivity-aid example program. These two utility programs must be available and bound for the Db2 subsystem. For information on the installation of the DSNTEP2 sample program, see the section "Productivity-aid sample programs" in *Db2 for z/OS Utility Guide and Reference*.

The Db2 schema name that is used for this application is CKADBVA and all table names start with CKA. The schema name and the table names cannot be modified.

The example jobs that are provided are:

### **CKAJVA00**

A Db2 database names a collection of table spaces. The example job CKAJVA00 creates a database in Db2 to contain the table space and tables. A user with sufficient Db2 authorization to create a Db2 database must run this job. Guardium VA expects the information about a Db2 subsystem to be available in tables inside the Db2 subsystem. Therefore, you must run the CKAJVA00 job on each system that you want to analyze with Guardium VA.

The database is created with available defaults:

```
CREATE DATABASE CKADBVA;
```

You can change the name of the database to match your installation standards.

### CKAJVA01

A table space is one or more data sets in which one or more tables are stored. The example job CKAJVA01 has two steps. The first step uses the SQL **DROP** statement for the tables and table space to ensure that no remnants of a previous **LOAD** of the data remain. The second step creates the table space and tables. The first time that you run this job, the DROP step that drops the objects ends with return code 8. You can ignore this error. Alternatively, you can edit the job to run only the CREATE step. The job must be run by a user with sufficient Db2 authorization to drop and create these objects in the Db2 database that was created in example CKAJVA00. Guardium VA expects the information about a Db2 subsystem to be available in tables inside the Db2 subsystem. Therefore, you must run the CKAJVA01 job on each system that you want to analyze with Guardium VA.

The table space is created with available defaults:

```
CREATE TABLESPACE CKADBVA in CKADBVA;
```

You might have to specify allocation-related keywords to assign the correct *storagegroup*, or to increase available space for the tables, for example:

```
USING STOGROUP storagegroup  
PRIQTY 20000
```

A sample of the creation of a table and its index in the table space:

```
CREATE TABLE CKADBVA.CKA_OS_GROUP (  
    COMPLEX          CHAR(8)    NOT NULL,  
    GROUP            CHAR(8)    NOT NULL,  
    ADDITIONAL_INFO  VARCHAR(256),  
    PRIMARY KEY (COMPLEX, GROUP)  
)  
in CKADBVA.CKADBVA;  
CREATE UNIQUE INDEX CKADBVA.IDX_CKA_OS_GROUP ON  
    CKADBVA.CKA_OS_GROUP (COMPLEX, GROUP);
```

After creation of the tables and indexes, **GRANT** statements are issued to allow SELECT authority on the tables to user SQLGUARD (the Guardium VA SQL ID). For example:

```
GRANT SELECT ON CKADBVA.STATUS TO SQLGUARD;
```

You can change the name of the table space to match your installation standards. You cannot change the schema name and table names.

### CKAJVA99

This job also has two steps. The first step is a CKFCOLL step that collects the information from the Db2 catalog tables into a CKFREEZE data set. The second step uses CKRCARLA to create the JCL and input for a second job. By default, the second job is immediately submitted for execution. The second job has steps for each Db2 subsystem. It uses CKRCARLA to create an input file specific for a particular Db2 subsystem and uses DSNUTILB to load the file in that Db2 subsystem. It also contains some SQL steps for simple postprocessing of the data. These steps are repeated for each Db2 subsystem. You can update job CKAJVA99 to include or exclude specific Db2 subsystems with a CARLa **SELECT** or **EXCLUDE** statement at the indicated line. In that case, the generated job contains steps for the selected Db2 subsystems only. A user with sufficient Db2 authorization to LOAD and UPDATE the Db2 tables must run the CKAJVA99 job.

If you have multiple releases of Db2 active, you must create an instance of this job for each release. The **STEPLIB DD** statements must reflect the correct Db2 level for each release. In this situation, you must also use the sample **SELECT** or **EXCLUDE** statements to limit the Db2 subsystems to those matching the release of the STEPLIB libraries.

By default, the job that is generated by CKAJVA99 is immediately submitted for execution. If you want to first look at the generated job, you can redirect the output from DDNAME STAGE2 to either a data set or to SYSOUT.

The **LOAD** statements that are used to load the tables include keywords that reflect the static nature of the data that allows repeated loading of the tables. The following keywords are relevant:

```
LOAD DATA
      REPLACE
      REUSE
      LOG      NO
      NOCOPYPEND
```

For more information about the meanings of these keywords, see *Db2 for z/OS Utility Guide and Reference*.

The example jobs use as input the following members in SCKRSAMP:

#### **CKAVA000**

This member contains the CARLa **ALLOC** statements that are required to specify the RACF input source and the CKFREEZE data set.

#### **CKAVA001**

This member contains the CARLa statements that are used to create RACF user and group information records and the normal (non-effective) format of the Db2 object access matrix for a RACF system.

#### **CKAVA002**

This member contains the CARLa statements that are used to create the effective format of the Db2 object access matrix for a RACF system.

#### **CKAVA100**

This member contains the CARLa **ALLOC** statements that are required to specify the ACF2 input source and the CKFREEZE data set.

#### **CKAVA101**

This member contains the CARLa statements that are used to create ACF2 user and source-group information records and the normal (non-effective) format of the Db2 object access matrix for an ACF2 system.

#### **CKAVA102**

This member contains the CARLa statements that are used to create the effective format of the Db2 object access matrix for an ACF2 system.

#### **CKAVALDO**

This member contains SQL **LOAD** statements to load the Db2 tables.

#### **CKAVAMNO**

This member contains CARLa **INCLUDE** statements to include the applicable CARLa members based on the active External Security Manager (RACF or ACF2).

#### **CKAVASQ0**

This member contains SQL statements for required simple post-processing of the normal format access matrix.

#### **CKAVASQ1**

This member contains SQL statements for required simple post-processing of the effective format access matrix.

#### **CKAVASQ9**

This member contains SQL statements for recording the status of the loading process.

For more information about Db2 and the utilities, see the following documents:

- *Db2 for z/OS Utility Guide and Reference*
- *Db2 for z/OS SQL Reference*
- *Db2 for z/OS Administration Guide*



## Appendix A. Site module

Use of the Site module is optional. The product is fully operational without it. Job CKRZSITE in the SCKRSAMP library is supplied to assist you in creating the Site module.

Table 19 on page 191 list the zSecure options that can be customized in the Site module CKRSITE:

<i>Table 19. Site module options</i>			
<b>Parameter</b>	<b>Allowed values</b>	<b>Description</b>	<b>Default</b>
AUTH	SINGLE DOUBLE TRIPLE	Default multi-authority setting.	SINGLE
CUSTSPEC	Up to 100 characters of text	Site or customer-specific identifier. This parameter is included in the SYSPRINT output for various zSecure components.  Because the value is also used as the default CHECKPWD in zSecure Collect, choose a value that can be expected to be stable over a long period (10 years or more).	<none>
CLASS	SAF class name	Resource class for the zSecure security checks.	XFACILIT
KEEPCOMMAND	Numeric	Expiration period for CKGRACF queued commands.	7
KEEPAUDIT	Numeric	Period during which completed and expired queued commands and past scheduled actions are retained by CKGRACF. The value of this setting must be larger than the expiration period.	30
RESTRICT	Y N	Restricted mode. Alternatively, you can use the general resource CKR.READALL provided by the CLASS parameter instead of this setting.	N

If you perform a distribution-oriented installation, you might want to use different CKRSITE parameters for different z/OS images. This configuration can be set up by storing the CKRSITE modules in separate libraries for each image, and concatenating these libraries to the SCKRLOAD library, using either JCL or the WPREFIX/UPREFIX parameter. In line mode, the WPREFIX/UPREFIX parameters only support the SCKRCARL library. Also, note that storing CKRSITE directly in the SCKRLOAD library would be an SMP/E violation because the SCKRLOAD library is SMP/E-managed.



---

## Appendix B. Security setup for zSecure

The APF-authorized functions of zSecure are protected by resources in the XFACILIT class, unless you changed this setup. (See [Appendix A, “Site module,”](#) on page 191 for instructions).

zSecure uses SAF to configure menus, action commands, and to determine what data (profiles, rules, SMF records) users are allowed to see:

- To activate these controls, you can create catchall (backstop) profiles or rules for resources with high level qualifiers CKF, CKG, CKR, C2R, and C2X.
- To disable all functions for most users, specify default access NONE. Grant READ access only to those people entitled to use all functions of zSecure.
- To activate use of the zSecure Admin Command Logging function, define the profile CKR.CKXLOG.\*\* and profiles for the resources as described in [“Other ISPF interface controls”](#) on page 195.

All these resources use the XFACILIT resource class or the resource class that is configured in your CKRSITE module (see [Appendix A, “Site module,”](#) on page 191).

To enhance performance, zSecure does not always read the entire RACF database. To ensure that scope-related reports are correct, ensure the following:

- CKGRACF scope profiles must start with non-generic characters for the CKG qualifier.
- RACF delegated authority profiles must start with non-generic characters for the IRR qualifier.

For further refining, see the remaining information in this Appendix.

---

### ISPF interface controls

zSecure uses the system authorization facility (SAF) to determine the data (profiles, rules, SMF records) that users are allowed to see, and to configure menus. The scope of resources and data that users see is controlled using access to the CKR.READALL resource. See [Appendix C, “Restricted mode,”](#) on page 201. The ISPF user interface menus are determined through resources CKR.OPTION.\* and the available line commands are controlled using resources CKR.ACTION.\*.

### Resources that configure which options are shown

Protection can be defined for all menu options. Depending on the access a user has on the corresponding resource names, the menu options are displayed (READ access allowed) or hidden (no access). When an option is hidden, the user is not allowed to perform the option.

The resource names follow the same naming convention:

```
first qualifier: 'CKR'  
second qualifier: 'OPTION'  
third qualifier: main panel option  
fourth qualifier: secondary menu option  
etc.
```

So for the main panel, the resource names are:

```
CKR.OPTION.SE for SETUP  
CKR.OPTION.RA for RACF  
CKR.OPTION.AA for ACF2  
CKR.OPTION.AU for AUDIT  
CKR.OPTION.AM for ACCESS  
CKR.OPTION.EV for EVENTS  
CKR.OPTION.CR for COMMAND REVIEW  
CKR.OPTION.RE for RESOURCE  
CKR.OPTION.CO for CARLA  
CKR.OPTION.IN for INFORMATION  
CKR.OPTION.LO for LOCAL
```

The resource names for the SETUP panel are:

```
CKR.OPTION.SE.0 for OPTIONS
CKR.OPTION.SE.1 for INPUT FILES
CKR.OPTION.SE.2 for NEW FILES
CKR.OPTION.SE.3 for PREAMBLE
CKR.OPTION.SE.4 for CONFIRM etc.
```

The resource names for the SETUP DEFAULT panel are:

```
CKR.OPTION.SE.D.0 for SETUP DEFAULT OPTIONS etc.
```

**Note:** If the resource checked for an option is not protected through SAF, then the option will be shown.

You can allocate ddname C2RIMENU to a data set or terminal before starting zSecure, or select the Debug action commands option under Setup trace. (See the *User Reference Manual*.) For every menu option that is selected, a line describing the tested resource is written to the ddname. You can display the C2RIMENU ddname by using the C2RIMENU primary command.

Also, to see a complete and up-to-date list of available overview types, you can enter the FIELDS command from the command line on most zSecure panels. Select BUILTIN on the next panel to show the list.

Note that restrictions to access a panel option will **not** result in any SECURITY restrictions. If a user can define his own panels, or change the zSecure panels, the user will be able to perform all options. So strict SECURITY authorization must still be in place.

## Resources that configure which line commands are allowed

It is possible to configure which line commands are allowed on the various overview displays. This holds both for overviews from a database query and, for example, from a data set containing configurations for zSecure Alert. For each command, READ access to a resource CKR.ACTION.*overview-type.entity.action-character* is checked.

The different overview-type/entity combinations are:



AD.F:	ACF2_RULE	R1.\$:	REPORT_AC1
AI.I:	ACF2_INFOLINE	RA.\$:	RACF_ACCESS
AK.F:	ACF2_RULELINE	RA.C:	RACF_ACCESS (Custom display)
AL.L:	ACF2_LID	RA.D:	RACF_ACCESS (Data set profiles)
AR.I:	ACF2_INFORULE	RA.G:	RACF_ACCESS (Group information)
CE.\$:	CERTIFICATE	RA.R:	RACF_ACCESS (Resources)
CL.\$:	RACF CLASS	RA.U:	RACF_ACCESS (User information)
CP.\$:	CICS_PROGRAM	RC.D:	RACF (DATASET entities)
CR.\$:	CICS_REGION	RC.G:	RACF (GROUP entities)
CS.\$:	CICS_TRANSACTION	RC.M:	RACF (User ID summary)
DA.\$:	DB2_TABLESPACE	RC.R:	RACF (RESOURCE entities)
DB.\$:	DB2_BUFFERPOOL	RC.U:	RACF (USER entities)
DC.\$:	DB2_COLLECTION	RD.D:	REPORT_NONDEFAULT
DD.\$:	DB2_DATABASE	RN.D:	REPORT_REDUNDANCY
DE.\$:	DB2_VARIABLE	RO.D:	REPORT_OUTOFGROUP
DG.\$:	DB2_STOGROUP	RP.\$:	REPORT_PADS
DH.\$:	DB2_SCHEMA	RR.D:	REPORT_PROFILE
DJ.\$:	DB2_JAR	RR.R:	REPORT_PROFILE
DK.\$:	DB2_PACKAGE	RS.D:	REPORT_SENSITIVE
DN.\$:	DB2_PLAN	SC.D:	RACF REPORT_SCOPE (DATASET entities)
DO.\$:	DB2_ROUTINE	SC.R:	RACF REPORT_SCOPE (RESOURCE entities)
DQ.\$:	DB2_SEQUENCE	SD.\$:	SENSDSN
DR.\$:	DB2_REGION	SM.D:	SMF (DATASET entities)
DS.\$:	DSN	SM.G:	SMF (GROUP entities)
DT.\$:	DB2_TABLE	SM.L:	SMF (LOGONID entities)
DY.\$:	DB2_DATATYPE	SM.R:	SMF (RESOURCE entities)
EF.\$:	RUN_DD	SM.U:	SMF (USER entities)
FL.\$:	FIELD	SP.\$:	SPT
IC.\$:	SETOPTS_CLASS	ST.\$:	REPORT_STC
MB.\$:	MEMBER	TK.\$:	ICSF_TOKEN
MC.\$:	IMS_REGION	TR.D:	TRUSTED (data sets)
MP.\$:	IMS_PSB	TR.R:	TRUSTED (Resources)
MQ.\$:	MQ_REGION	TR.U:	TRUSTED (Users)
MT.\$:	IMS_TRANSACTION	UN.\$:	UNIX
QA.\$:	MQ_CHLAUTH	ZA.B:	Alert (Category selection)
QC.\$:	MQ_CONNECT	ZA.C:	Alert (Configuration selection)
QH.\$:	MQ_CHANNEL	ZA.D:	Alert (Destinations)
QI.\$:	MQ_INIT	ZA.R:	Alert (Selection)
QN.\$:	MQ_NAMELIST		
QO.\$:	MQ_AUTHINFO		
QP.\$:	MQ_PROCESS		
QQ.\$:	MQ_QUEUE		
QT.\$:	MQ_TOPIC		

You can allocate ddname C2RIMENU to a data set or terminal before starting zSecure, or select the "Debug action commands" option under Setup trace (see the *User Reference Manual*). For every action command that is selected, a line describing the tested resource is written to the ddname. You can display the C2RIMENU ddname by using the C2RIMENU primary command.

To change the action character or the description, see the *User Reference Manual*.

## Other ISPF interface controls

This section lists additional resources that control the way that the ISPF interface behaves:

### **CKR.CONTROL.MASK**

This resource controls the capability of users to ask for all profiles (no mask or name). If a matching profile is defined, and the user does not have at least READ access, most queries in the ISPF interface must specify at least three non-generic characters in the mask for the RACF profile or in an other query-specific field on the selection panel. An example is the RA.U query that requires at least three non-generic characters in the mask for the Userid or Name field. This resource forces the use of the 31-bit version of the CKRCARLA program in stead of the 64-bit version. It also forces the use of the **Use IO in preference to storage** option in **SE.0 Run Options**. The control function is inactive for users with READ access or higher.

### **CKR.CKXLOG.\*\***

This profile enables the zSecure Admin Command Logger functions in the zSecure Admin ISPF user interface. If this profile is defined, the CKR.CKXLOG.ID.SHOW and CKR.CKXLOG.ID.PROMPT resources can be used to control the individual command logger functions. In the absence of this profile, the command logger functions are not available in the zSecure Admin ISPF environment.

### **CKR.CKXLOG.ID.SHOW**

This resource controls which users can use the CKXLOGID primary command to set ticket information. READ access to this resource also activates use of the ticket ID and description in the command confirmation panels. This resource can be used to selectively implement the use of tickets to annotate commands that are logged using the zSecure Admin Command Logger (CKXLOG).

### **CKR.CKXLOG.ID.PROMPT**

This resource controls which users must use the CKXLOGID primary command or specify ticket information in the zSecure Admin command confirmation panels. The control is used for all users that have at least READ access to the matching profile. This resource can be used to selectively require the use of tickets to annotate commands that are logged using the zSecure Admin Command Logger (CKXLOG).

## **Access to the security database**

---

The userids using IBM Security zSecure need permission to read the security database (or perhaps a copy or an unload). However, this might create an exposure. [Appendix C, “Restricted mode,” on page 201](#) describes this type of exposure, and how to remedy it.

## **Authorization and userid mapping when using the zSecure Server**

---

Specific authorizations are required for using remote data and for routing commands.

- For remote data access, the user requires several authorizations:
  - The user must be authorized to access the remote destination.
  - The user must be authorized to use the remote data set using the zSecure Server.
  - For all data sets other than the live data sources, the user must have access to the data set.
- For routing commands, the user's required authorization depends on the routing method chosen:
  - For zSecure Server-based command routing, the user needs access to appropriate zSecure resources. These resources are described in this section.
  - For RRSF-based command routing, the user needs an approved user association. For information about the required RRSF authorizations, see the *RACF Security Administrator's Guide* and the *RACF Command Language Reference*.
  - For NJE-based command routing, the user needs authorization to route jobs to the remote system.

These authorizations must be defined before the user can access remote data or route commands.

Authorization is verified using the `userid` of the user. On the remote system, authorization is verified using a `userid` defined on that remote system. The `userid` on the remote system is obtained from a *mapping rule* that links the `userid` used to log on to a `userid` that is defined on the remote system. For more information about this `userid`-mapping rule, see [“Userid mapping” on page 198](#).

For remote destinations other than the current ZSECNODE, you can control authorization to the remote destination using profiles in the XFACILIT class, or using profiles in the RRSFDATA resource class. These profiles are used for data access as well as for command routing. The verification is done on the system where the user logs on. If the destination is a server defined on the current ZSECNODE, the user is automatically authorized to access the server. The profile in the XFACILIT class must match resource CKNADMIN.TONODE.<node-name>. In this resource name, the following qualifiers are used:

**CKNADMIN**

A fixed prefix.

**TONODE**

A fixed qualifier for the destination node verification.

**node-name**

The ZSECNODE that was specified for the remote data, or the ZSECNODE to which the ZSECSYS that was specified belongs.

If no profile is found in the XFACILIT class (unless you changed this; see Appendix A, “Site module,” on page 191), profiles in the RRSFDATA resource class are used. The profile in the RRSFDATA class must match resource DIRECT.<node-name>. The <node-name> has the same value as specified in the preceding list. If no RRSFDATA profile is found either, access to the remote system is allowed. If a profile is found, the user must have at least READ access to the resource.

On the destination server, you must implement a similar authorization for the source server. The user must have at least READ access. If the source server is defined on the same ZSECNODE as the current server, the user is automatically authorized to access the current server. The resource name used for the access verification is CKNADMIN.FROMNODE.<node-name>. In this resource, the qualifiers are:

**CKNADMIN**

A fixed prefix.

**FROMNODE**

A fixed qualifier for the source node verification.

**node-name**

The ZSECNODE assigned by the system administrator to the system where the user is running the query.

If no profile is found, the user is not authorized to access this node. Because this verification is done on the destination system, the userid onto which the logged-on userid is mapped must have at least READ access to the resource.

Authorization to use the remote data set can be controlled using profiles in the XFACILIT class. The profile in the XFACILIT class must match resource CKNDSN.<dstype>.<node-name>.<system-name>.<dsname>. In this resource name the following qualifiers are used:

**CKNDSN**

A fixed prefix.

**dstype**

Describes the type of data. It has the value that is used on the TYPE= keyword of the CARLa ALLOCATE statement. Example values are RACF, CKFREEZE, SMF, and UNLOAD. The special value DEFTYPE is used for all types that are defined using the CARLa DEFTYPE statement. The CARLa ALLOCATE statement can also be used to allocate an SMF log stream data set. In that case, the dstype has the value LOGR-SMF.

**node-name**

The ZSECNODE used for the system where the data set resides.

**system-name**

The ZSECSYS used for the system where the data set resides.

**dsname**

The name of the data set that is accessed remotely. For some data, the dsname used here is a placeholder, instead of the true data set name. The placeholders ACTIVE, PRIMARY, BACKUP and MANAGED are used for data sets where the exact data set name is not relevant. For remote

commands the *dsname* CKRCMD is used. The name MANAGED for a data set is a reserved name that does not result in the allocation of any real data set.

Because this verification is done on the destination system, the *userid* onto which the logged-on *userid* is mapped must have at least READ access to the resource.

Running remote commands is also controlled on the remote system using CKNDSN profiles in the XFACILIT resource class. The resource name is similar to those used for data set access:

CKNDSN.<*dstype*>.<*node-name*>.<*system-name*>.CKRCMD

The qualifiers have the same meaning as described in the preceding list. The last qualifier is:

#### **CKRCMD**

A fixed suffix to indicate that this resource describes the authority to run commands.

If the data set that is accessed is not ACTIVE, PRIMARY, BACKUP, MANAGED, or CKRCMD, the user must also have access to the data set itself. The zSecure Server opens the data set using the user's authorization, and normal DATASET access verification takes place. The user must have sufficient authorization to access this data set. If not, a regular access violation message is issued, followed by a 913 OPEN abend. The name MANAGED for a data set is a reserved name that does not result in the allocation of any real data set.

## Userid mapping

Because the user naming conventions can be different in different RACF databases, it is possible to implement *userid* mapping rules. Your installation might use one of several types of *userid* mapping. You can implement *userid* mapping in either of the following ways:

- Using a profile in the XFACILIT class (unless you changed this; see [Appendix A, "Site module," on page 191](#))
- Using existing RRSF user associations.

The profile in the XFACILIT class must match resource

CKNUMAP.<*source-nodename*>.<*source-userid*>.<*target-nodename*>.

In this resource name the following qualifiers are used:

#### **CKNUMAP**

A fixed prefix.

#### **source-nodename**

The zSecNode assigned by your system administrator to the system where you are running the query.

#### **source-userid**

The *userid* you used to log on.

#### **target-nodename**

The zSecNode used for the system where the data set resides, or where you want to run commands.

The profile must have an APPLDATA field that specifies the *userid* used for you on the remote system. Possible values for the APPLDATA are:

#### **=USERID**

The identity mapping is used.

#### **other**

The value for the target *userid*.

If the APPLDATA is missing or the first character is blank, the source *userid* is not accepted.

It is possible to use generic profiles for these mapping rules. In this way, a single profile can be used to map multiple *userids* on multiple systems.

If there is no *userid* mapping profile, zSecure uses existing RRSF user associations. Only approved associations where the logged-on user is a PEER or MANAGER-OF the remote *userid* are taken into

consideration. For more information about setting up RRSF user associations, see the *RACF Security Administrator's Guide* and the *RACF Command Language Reference*.

If inconsistencies in the RRSF user associations are encountered, the identity mapping is used.

If there is no CKNUMAP profile and also no RRSF user association, the identity mapping is used.

## Other security resources

zSecure issues SAF calls to configure menus and to limit use of its authorized functions. All SAF calls are in the XFACILIT class, unless you customized the CLASS option in the site module. (See [Appendix A, "Site module,"](#) on page 191.)

- Users of the zSecure Admin component who are to issue REMOVE USER commands need READ access to the STGADMIN.IGG.DELETE.NOSCRTCH and STGADMIN.IGG.DEFDEL.UALIAS facility resources, or ALTER on the master catalog and the relevant user catalogs. This is necessary in order to enable them to delete all components of VSAM data sets, and to delete catalog aliases.
- Users of the zSecure Admin component frequently create command streams in data sets. Because these data sets can contain passwords, be sure that they are erased upon deletion as shown in the following example:

```
ADDSD 'workprefix.C2R*.CKRCMD*. *' UACC(NONE) ERASE
ADDSD 'workprefix.C2R*.CKR2PASS.*' UACC(NONE) ERASE
```

Where *workprefix* is the prefix for ISPF work data sets, as specified in the WORKPREF parameter in the zSecure configuration, see [Appendix D, "Configuration parameters and members,"](#) on page 205.

- When running zSecure under ACF2, the C2RIMENU program must be enabled to perform SAF calls:

```
INSERT SAFDEF.C2RIMENU PROGRAM(C2RIMENU)
RB(C2RIMENU) NOAPFCHK ID(C2RIMENU)
RACROUTE(REQUEST=AUTH,CLASS=XFACILIT,STATUS=ACCESS)
```

If this is not done, all panel options will be visible to every user, resulting in error messages when they try to use options they are not allowed to. If you changed the Site module to use a resource class other than XFACILIT, you should adapt the above SAFDEF accordingly.

- Users of the CKGRACF REFRESH command need access to the resource C2GRACF in the APPL class. This includes the userid that runs the daily CKGRACF job. For additional information about this job, see ["Requirements and rationale for running the daily CKGRACF job"](#) on page 42. The APPL class has a default RC=4 so that the program can run without a covering profile. However, if an APPL profile exists that covers the C2GRACF resource (\*, for example), READ access is required.
- Users running CKRCARLA in an APF-authorized environment requires READ access to resource CKR.CKRCARLA.APF.
- Users of the RACF Exit Activator program need UPDATE access on C2X.exitname where *exitname* is the name of the exit-module as described in the RACF System Programmer's Guide. It is the name that the corresponding module would have if dynamic activation would not be used. For example, the resource for the RACROUTE REQUEST=AUTH postprocessing exit is C2X.ICHRCX02.

## Resources that specify which data can be seen or updated

For the READALL resource, see [Appendix C, "Restricted mode,"](#) on page 201. For other resources in this category, see the *User Reference Manual* for your zSecure product.

## Security checks related to zSecure Collect

For zSecure Collect related security checks, see the *IBM Security zSecure Admin and Audit for RACF: User Reference Manual*. If you are using zSecure Audit for ACF2 or Top Secret, see the *User Reference Manual* for those products.



---

## Appendix C. Restricted mode

zSecure can be used in two distinct modes: *unrestricted mode* and *restricted mode*.

- In unrestricted mode, all information in the RACF or ACF2 security databases is included in reports that can be viewed from the ISPF panels or from printed reports.
- In restricted mode, only data within the user's scope is reported. For example, in restricted mode a Help Desk operator cannot create the same reports or view the same data that a Central Administrator can.

The choice between these two modes of operation can be made for the entire installation, or per user, or per group.

The net effect of allowing a user unrestricted access is that you have effectively created a "read-only" unscoped auditor. That is, the user can see the same data and options as a user with the RACF AUDITOR or ACF2 AUDIT attribute, but cannot change any global options or auditor settings.

---

### Conditions for restricted mode

Restricted mode is determined in the following manner:

1. If the user specifies SIMULATE RESTRICT, restricted mode is activated.

This is noted in one of the message variants that give some information about what would happen without the SIMULATE RESTRICT:

```
CKR0031 Restricted mode by simulation, although user userid has privilege privilege
CKR0031 Restricted mode by simulation, although user userid READ access to class profile
CKR0031 Restricted mode by simulation for user userid, although no profile class profile
```

2. If one of the ALLOC statements refers to a remote node and the (possibly mapped) user ID there has no SPECIAL, AUDIT, or ROAUDIT privilege and no READ permit on CKR.READALL in the security database on that remote node, restricted mode is activated. The class for the CKR.READALL resource on the remote node is determined by the site module on the remote node. See [Appendix A, "Site module," on page 191](#).

This is noted in one of the message variants that give some information about what would happen without the remote node restriction:

```
CKR0031 Restricted mode by remote node, although user userid has privilege privilege
CKR0031 Restricted mode by remote node, although user userid READ access to class profile
CKR0031 Restricted mode by remote node for user userid, although no profile class profile
```

If both "[1](#)" on page 201 and "[2](#)" on page 201 are the case, then the messages use simulation and remote node, instead of giving the similar messages twice.

3. Else, if the user has any of the attributes SPECIAL, AUDITOR, ROAUDIT in the system running the query, unrestricted mode is granted. This is noted with message

```
CKR0031 Unrestricted mode active, user userid has privilege privilege
```

4. Else, access to the CKR.READALL resource is tested, (in the XFACILIT class, unless you changed this; see [Appendix A, "Site module," on page 191](#)). If the SAF call takes a decision (through profile or class default RC):

- a. NO access enforces restricted mode. This is noted with message:

```
CKR0031 Restricted mode active, user userid no READ access to class profile
```

b. READ or higher access grants unrestricted mode. This is noted with message:

```
CKR0031 Unrestricted mode active, user userid READ access to class
profile
```

5. Else, (that is, access to CKR.READALL is undecided) restricted mode is in effect when at least one of the following is true:

a. The live system is RACF and the user has only access to (part of) the input via PADS, as described in “Setting up Program Control and PADS access” on page 202. This applies to security database, CKFREEZE, SMF, DEFTYPE, and access monitor input files. This is noted with message:

```
CKR0031 Restricted mode active because of [PADS | program pathing] user
userid
```

b. The Site module (see Appendix A, “Site module,” on page 191) has been configured to specify restricted mode. This is noted with message:

```
CKR0031 Restricted mode active by installation option; user userid
```

6. Else, unrestricted mode is granted. This is noted with message:

```
CKR0031 Unrestricted mode active; user userid
```

**Note:** Access is undecided when there is no covering profile in RACF and there is, in the Site module, a resource class with a default return code of 4. The default class is XFACILIT, which has a default return code of 8, meaning that access is forbidden.

Users that have the SPECIAL, AUDITOR, or ROAUDIT attribute on the executing system operate in unrestricted mode by default as explained above, but can be made restricted by using SIM RESTRICT. When using SIM RESTRICT, global audit fields will be invisible to the SPECIAL user who does not also have AUDITOR or ROAUDIT.

When accessing a remote system through CKNSERVE, a similar decision is taken based on the system-wide properties and CKR.READALL permit of the mapped user ID on the remote system.

When analyzing multiple, possibly remote, security databases in one run, keep in mind that 'Restricted mode' is a condition that is not specific to a complex or file. It applies to all complexes and applicable files after it is activated based on insufficient user authority on any one local or remote system where a query is directed.

When running in restricted mode, the scope used is smaller than for trusted reporting in unrestricted mode. As a consequence, the reports exploiting scope-related information might contain fewer records. This reduced scope is the same one that you get by adding SUPPRESS REASON=(SELFCONNECT, PWDCHANGE, WARN, NOPROFILE, CKGRACMAP CKGRACDCERT) to a scope report. A message CKR2245 is issued to document this if you are using a function that also needs the scope trees.

## Effects of restricted mode: the user's scope

---

The user's scope in restricted mode is evaluated from the database that is being examined. If the input source contains no security database, the database on the live system is used for scope evaluation.

For a RACF input source, you can enlarge the user's scope by granting READ access to (some) CKG scope resources. This way, you can define auditors with fine-grained access authorizations. See the *User Reference Manual* for more information. Note that granting access is always done on a system image that uses that input as a live database, so it has no effect on older unloads.

## Setting up Program Control and PADS access

---

### About this task

This information applies only if your live system uses RACF.

The greatest level of security is achieved by making use of Program Access to Data Sets (PADS). Without PADS, users can use, for example, ISPF Browse to examine the RACF database, or even copy the database



to a system where they can run in unrestricted mode. However, due to the way that RACF implements Conditional Access, this option is also the most cumbersome to use. As an alternative to PADS, you can exploit the zSecure Server in self-connect mode to access the security database. See [“Use of the zSecure Server to limit the need for access to the security database”](#) on page 59.

You can combine PADS access, or access through the zSecure Server, with the use of the CKR.READALL resource to override restricted mode for selected (or all) users.

If you want to set up zSecure for operation using Conditional Access or PADS mode, you must define profiles in the program class, and activate RACF program control. Many installations perform most of these steps as part of the implementation of UNIX System Services (USS).

## Procedure

Use the following steps to set up Conditional Access or PADS mode:

1. Acquaint yourself with the principles of Program Control and Program Access to Data sets, as documented in the *RACF Security Administrator's Guide* (SA23-2289 for z/OS RACF).
2. Determine if your installation is using RACF Program Control in BASIC mode or in ENHANCED mode.
  - If your system uses BASIC Program Control mode, you should add the required PROGRAM profiles with a command like:

```
RDEF PROGRAM CKR* ADDMEM('CKR.SCKRLOAD'//NOPADCHK)
```

- If your system uses ENHANCED Program Control mode (available as of z/OS 1.4), you can add the required PROGRAM profiles with commands like:

```
RDEF PROGRAM CKR* ADDMEM('CKR.SCKRLOAD'//NOPADCHK) APPLDATA('MAIN')
RALT PROGRAM ** ADDMEM('CKR.SCKRLOAD'//NOPADCHK)
```

If you use different load libraries (for example, you might have created multiple load libraries for multiple versions of the Site module, and concatenate these to your main zSecure load library), you must specify ADDMEM for each of the load libraries that you need to have program-controlled.

If you have set up alias names for load modules, also create profiles that cover the alias names.

On older systems, the volume serial number must be inserted between the slashes (//) in the previous example commands. This can lead to problems if your data set is located on an SMS managed volume. In that case, to prevent SMS from moving the data set to a different volume, ensure that the data set is assigned to storage class that has the Guaranteed Space attribute (or on a non-SMS managed volume).

If you intend to use zSecure interactively via TSO/ISPF, also add program profiles for some other executable modules of zSecure. In this example, \*\* is used to describe all relevant load modules. Alternatively, you can add the entire library to the definition of program profile \* or \*\* as in the example.

3. Add an authorization group to be used for users authorized to access the database in PADS mode.
4. Add a conditional access list to the profile describing your RACF databases. You might restrict users to using the back-up database; you might first need to add a profile for this.

A sample command is:

```
PE 'databaseprofile' WHEN(PROGRAM(CKRCARLA)) ID(authgroup)
```

5. Ensure that program control is active by issuing SETROPTS LIST, check that the output specifies WHEN(PROGRAM). If it does not, schedule introduction of program control. (Review the current contents of the PROGRAM class first.)

Program control is activated as follows:

```
SETROPTS WHEN(PROGRAM)
```

6. You might need to add a program profile describing operating system modules, depending on how much your site moved from link list to LPA. (For LPA modules you do not need program profiles.)

Generally the commands issued are:

```
RDEF PROGRAM * ADDMEM('SYS1.LINKLIB'//NOPADCHK) UACC(READ)
RALT PROGRAM * ADDMEM('SYS1.CMDLIB'//NOPADCHK)
RALT PROGRAM * ADDMEM('SYS1.MIGLIB'//NOPADCHK)
RALT PROGRAM * ADDMEM('SYS1.CSSLIB'//NOPADCHK)
RALT PROGRAM * ADDMEM('cee.version.SCEERUN'//NOPADCHK)
RALT PROGRAM * ADDMEM('TCP/IP.SEZALINK'//NOPADCHK)
RALT PROGRAM * ADDMEM('TCP/IP.SEZATCP'//NOPADCHK)
```

If you want to run *interactively* in PADS mode, you must also add the ISPF and PDF link list libraries. When you load a *dirty* (non-controlled) module, you probably need to log on again for your next try. Sometimes, leaving ISPF and invoking ISPF through TSOEXEC can be sufficient to regain a controlled environment.

```
RALT PROGRAM * ADDMEM('CKR.SCKRLOAD'//NOPADCHK) /* IBM Security zSecure */
RALT PROGRAM * ADDMEM('FAN130.SEAGLMD'//NOPADCHK) /* REXX */
RALT PROGRAM * ADDMEM('ISP.SISPLD'//NOPADCHK) /* ISPF/PDF */
```

If you define program \* or \*\* for all modules in SYS1.LINKLIB, also consider creation of two more specific profiles for the programs ICHDSM00 and IRRDPI00 with a reduced UACC. These two programs check the existence of a matching program profile to allow users to execute the program. If no program profile exists, only auditors are authorized to execute ICHDSM00 (DSMON). If, however, a generic profile \* has been defined with a UACC(READ) all users are authorized to execute ICHDSM00. Therefore, it is a good idea to also issue the following commands:

```
RDEF PROGRAM ICHDSM00 UACC(NONE) ADDMEM('SYS1.LINKLIB'/'*****'/NOPADCHK)
RDEF PROGRAM IRRDPI00 UACC(NONE) ADDMEM('SYS1.LINKLIB'/'*****'/NOPADCHK)
PE ICHDSM00 CLASS(PROGRAM) ID(your-auditors) ACCESS(READ)
PE IRRDPI00 CLASS(PROGRAM) ID(your-dynamic-parse-initialization-userid)
ACCESS(READ)
```

For more information about these commands, see the section about Program Control in the *RACF Security Administrator's Guide*.

7. To activate a change to any PROGRAM profile in the system, you must issue:

```
SETROPTS REFRESH WHEN(PROGRAM)
```

8. First try PADS access through batch jobs. If this works, you can move on to interactive access. First try to get it working with a minimal, clean environment: issue the CKR command (or your local copy) immediately after logging on to TSO, before you start ISPF. When invoked in this way, CKR frees file ISPLLIB to be sure to prevent dirty modules, and invokes the program as the primary ISPF application through the TSOEXEC command.

You must be aware that other ISPF applications (like SDSF) can create an environment that cannot be cleaned up even with the TSOEXEC command. In this situation, you might have to log on again. After you ensure that it works in the cleanest case, you can try to add back your own ISPF environment piece by piece to get a usable work environment and to see where you start getting 913 or 306 abends.

Each intercepted 913 abend produces a debugging display of the Job Pack Queue module in the SYSPRINT file. (You can review this with the SYSPRINT primary command under ISPF.)

You can also use the RACF ICH420I messages to determine the cause for the dirty environment.

---

## Appendix D. Configuration parameters and members

As described in “[Creating zSecure configuration data sets](#)” on page 25, job CKRZPOST creates a starter configuration for zSecure. You can create additional configurations if you need them. For example, you might want to create separate configurations for each z/OS image, for each community, or to provide a dedicated configuration for zSecure Alert, zSecure Visual, zSecure Adapters for SIEM, and so on.

The configuration must use JCL SET statements so that it can be used by both the batch and ISPF-interface. To enable the ISPF interface to interpret the SET statements, each parameter must be specified in a separate SET statement. Parameters are not case-sensitive except for UNIX filenames and DESC parameters which are case-insensitive insofar as they are used by the ISPF interface. Parameters that are used in JCL must conform to JCL-standards.

The INCLUDE statement (same syntax as in JCL) is supported by the ISPF interface to include members from the configuration data set. In the ISPF interface, the configuration data set is never part of a concatenation, unlike in JCL. The configuration data set can be useful to store common parameters in a common member and to override the required parameters as needed. For example, users of the Helpdesk configuration need to start zSecure at the RA.H option. The configuration for all the other options do not need to be changed from the default settings. You can create this configuration by overriding the C2REMAIN invocation in your copy of CKR (see further), or by specifying a separate configuration such as:

```
// INCLUDE MEMBER=your-common-member  
// SET STARTTRX='MENU(RA.H)'
```

The last occurrence of an assignment overrides any previous assignments.

The ISPF interface supports the symbol &SYSUID. (including the period), system symbols as specified in the active IEASYMxx parmlib member, and all symbols accepted by the REXX MVSVAR function. Other JCL parameters and continuation lines are not supported by the ISPF interface. Although unsupported and obsolete parameters can result in messages on the primary menu, they do not otherwise affect the functionality, so you can share configurations among unlike releases.

All parameters that are listed for the configuration can also be coded (in TSO/E convention) as overrides when invoking C2REMAIN. Typically, users do not need to use C2REMAIN. Instead, they can start the REXX exec CKR, or your copy of it. For example, you can create a REXX exec CKRQ, intended for simple administrative tasks, that adds only the overriding parameter STARTTRX (MENU (RA . Q)). Symbolic parameters are not supported as part of overrides, you should instead resolve parameters within (your copy of) CKR.

The version of CKR shipped with zSecure supports the same overrides. For instance, in order to run Setup default against a particular SE.D data set you can invoke:

```
CKR PROFDSN('HELPDESK.CKRPROF')
```

The following parameters are supported:

### **BLKSIZE**

BLKSIZE for VB data sets. If system-determined block size (SDB) is supported on your system, you can leave this out, or specify BLKSIZE=0. Otherwise, it is suggested to specify BLKSIZE=23476 if work data sets are created on 3380 compatible devices or BLKSIZE=27998 for 3390 compatible devices.

### **CKACUST and CKACUSV**

These parameters specify the data set names of the "compliant authorized ID population" members for zSecure Audit option AU.R - Rule-based compliance evaluation. CKACUST specifies a fixed blocked data set and CKACUSV specifies a variable blocked data set. Users can concatenate their own CKACUST and CKACUSV libraries in front of the library that is specified in the configuration member, and in the zSecure product library with option CO.1 or SE.8. See the topic "CO.1 LIBRARIES - Data set selection" in the User Reference Manual for your zSecure product.

## CPREFIX

This parameter specifies the beginning qualifiers of the data sets that contain the zSecure software. To allow your configurations to be use across software upgrades, consider using aliases for your configurations.

Alternatively, keep the release identifier in the data set name and change it when you want to switch all users from one release to the next. However, if you have multiple configurations, you need to update them all.

## C2PACPRM

The name of the data set containing parameters for the Access Monitor function configuration parameters. The data set is normally created by job CKRZPOST and referenced by the PARMLIB DD-statement and the SC2PCUST DD-statement in the JCL of the Access Monitor address space. For details, see [Chapter 10, “Setup of zSecure Admin Access Monitor,”](#) on page 61.

## C2PCUST

Only required when you use the zSecure Alert component. The alert CARLa and zSecure Alert parameter file are generated by the ISPF interface and written to this data set. Alert definitions are stored in ISPF tables, which are written to this data set as well. This data set should be allocated to SC2PSAMP DD of the zSecure Alert started task. See the zSecure Alert manual for more information.

## C2POLICE

The name of the zSecure Alert started task. The default name is C2POLICE. Only required when you use this component.

## C2RSERVE

C2RSERVE is the Server root directory for a Visual Server instance; all data that a particular Server uses is anchored in this directory. If you run multiple Servers, each Server must have its own value for C2RSERVE. To meet this requirement, you usually create a separate zSecure configuration for each Server. Alternatively, you can use a common zSecure configuration for all Servers and use a System symbol as part of the C2RSERVE parameter.

This parameter applies only to the zSecure Visual component. For additional information, see [Chapter 14, “Setup and use of the zSecure Visual Server,”](#) on page 133.

## C2RWCUST

This parameter specifies a data set for local CARLa and CKGRACF control statements for the Visual Server. For a new server, job CKRZPOST has filled in the data set name where your main configuration (for example, C2R\$PARM) resides. See [“Creating zSecure configuration data sets”](#) on page 25 for information about running CKRZPOST.

The following members are available for you to customize, if desired:

### C2RWEXG1

CKGRACF commands to be imbedded during a client-initiated transaction

### C2RWEXR1

CARLa to be imbedded during a client-initiated transaction

### C2RWASSC

The C2RWASSC member specification is required *only* if you are implementing the presentation of site-specific user data in the Visual client. See [“Site-specific user data”](#) on page 147 for information about configuring the display of site-specific user data in the Visual client. This member specifies the location of the Associations configuration file in a CARLa ALLOCATION (ALLOC) statement. The format of the statement is:

```
ALLOC TYPE=SITE_ASSOCIATIONS DSN='associations.file'
```

where:

### SITE\_ASSOCIATIONS

The Visual client uses this required keyword to look up the contents of the specified Associations file. The Associations file specifies the name of the customer data files and presentation format files that are used to display site-defined user information in the Visual client.

**associations.file**

Specifies the site-defined name of the Associations file.

For a new Server, empty members are created by job CKRZPOST. For an existing server, CKRZPOST makes no updates, so that it does not overwrite configurations that you customized. Make sure that you run the Visual Server with JCL at the same level as the rest of the server coding.

If you use the zSecure Multi-system feature with a non-default token for the Multi-system Server, add an OPTION statement with your token to members C2RWEXG1 and C2RWEXR1. See [Chapter 9, “Setup for remote data access and command routing,”](#) on page 49.

This parameter applies only to the zSecure Visual component. For additional information, see [Chapter 14, “Setup and use of the zSecure Visual Server,”](#) on page 133.

**C2RW131A = [ON | OFF]**

When this parameter is ON, encryption between the Visual Server and its clients is NIST 800-131A-compliant. This means that encryption algorithms that do not comply with the NIST Special Publication 800-131A (issued by the USA National Institute of Standards and Technology) are no longer accepted.

When upgrading, do not set C2RW131A to ON until all clients have upgraded to at least the 2.1.0 level of the zSecure Visual Client software, and have then upgraded their certificates during the client-server connection.

**Note:** Normally, certificate upgrade takes no more than 30 minutes of client-server connect time. However, no upper time limit exists. Running the server in non-NIST 800-131A mode for a few days or weeks is normally sufficient for all clients to upgrade their certificate. If any workstations do not complete their upgrades during the non-NIST 800-131A period, it is most likely due to not connecting at all, rather than too long a computation. A new initial password is required for these workstations. See [“Making clients known to the server”](#) on page 140.

When this parameter is OFF, older encryption algorithms are accepted. Use this setting for:

- Clients that connect using a zSecure Visual 1.13.0 or 1.13.1 level of the client software.
- The first connection after upgrading an old client to zSecure Visual 2.1.0 or higher, if you want to continue using the existing certificate.

If you set C2RW131A to ON before the first-time connection of an upgraded client, the client cannot use or upgrade its old certificate. If this situation occurs, you can switch C2RW131A back to OFF and then stop and restart the server. If those actions are unacceptable, you must issue a new initial password to the client. See [“Making clients known to the server”](#) on page 140.

**Note:** After changing the value of the C2RW131A parameter, you must restart the server to make the change take effect.

This parameter applies only to the zSecure Visual component. For additional information, see [Chapter 14, “Setup and use of the zSecure Visual Server,”](#) on page 133.

**C2RWIN**

C2RWIN is the directory where the zSecure Visual Server software resides. Multiple Servers can share this directory, provided that they run the same level of the product. Because the zSecure Visual Server does not write into this directory, you can mount the file system where the software resides read-only after you complete the product installation.

The C2RWIN parameter is also required when using zSecure Visual under ISPF to configure the first client. See [“Making clients known to the server”](#) on page 140.

This parameter applies only to the zSecure Visual component. For additional information, see [Chapter 14, “Setup and use of the zSecure Visual Server,”](#) on page 133.

**CKNSVPRM**

This parameter specifies a data set for configuration statements for the zSecure Server. For a new server, job CKRZPOST has filled in the data set name where your main configuration (for example, C2R\$PARM) resides. See [“Creating zSecure configuration data sets”](#) on page 25 for information about running CKRZPOST.

This data set must contain the two members indicated by the PPARM and PCOMMON parameters in the zSecure Server procedure, as described in [Chapter 9, “Setup for remote data access and command routing,”](#) on page 49.

#### **CKQCUST**

This parameter applies to the data preparation for SIEM. See [“Setup of the collection process for QRadar”](#) on page 164.

#### **CKQPATH**

This parameter applies to the data preparation for SIEM. See [“Setup of the collection process for QRadar”](#) on page 164.

#### **DATACLAS, STORCLAS, MGMTCLAS, TEMPUNIT**

Data class, storage class, management class and generic or esoteric unit name for allocation of ISPF work data sets. With SMS, these parameters are passed to your ACS-routines. The parameter TEMPUNIT is only used when temporary data sets are being allocated, see the *User Reference Manual* for your zSecure product. When TEMPUNIT is left empty, or is not specified, the value of the UNIT parameter is used.

#### **DPREF**

Prefix for data sets created by batch jobs in SCKRSAMP.

#### **EARLYWRN**

A list of userids who are to receive messages when unsupported or obsolete parameters are in the configuration member. If you specify multiple userids, separate them with commas. Enclose the entire list of user IDs in quotation marks, 'ADMIN1,ADMIN2,ADMIN3' for example. If this parameter is left empty, all users receive these messages.

#### **INIT**

Can be specified as YES (or RESET) or NO to indicate whether the settings from a previous session by the same user are to be kept or not.

#### **YES/RESET**

Reset all parameters to their defaults. These values are either the system defaults or the values that have been set with SETUP DEFAULT.

#### **NO**

Values from the previous session (user values) are used.

#### **JES**

This parameter specifies the JES level. The value can be 2 or 3, referring to JES2 and JES3, respectively. This parameter affects the type of JCL generated by zSecure.

#### **LIBDEF**

The LIBDEF parameter indicates whether an ISPEXEC LIBDEF must be issued for the zSecure user and common libraries. The value can be YES or NO. The default value is YES. When LIBDEF=NO is coded, the libraries must be preallocated to ISPF. DDNAME ISPTABLE is always allocated by an ISPEXEC LIBDEF and thus cannot be pre-allocated.

#### **PROFDSN**

The PROFDSN parameter specifies the data set used to customize the ISPF interface. The contents of the specified data set are updated by the SETUP DEFAULTS statement. Specify a partitioned data set with LRECL=80 and RECFM=FB. The SETUP DEFAULT options are disabled if this data set is not specified. If the data set specified PROFDSN is specified, but the data set is not available, the ISPF interface is aborted in order to prevent users inadvertently working without the intended settings (for example, with the wrong input).

#### **STARTTRX='MENU(menu)' | STARTTRX='CMD(command)'**

The transaction you want to run when the ISPF interface is started.

#### **MENU:**

Any menu, up to two levels, that is valid on the default primary menu, like AU.S or CO.

#### **CMD:**

A zSecure or TSO command that is valid on the default primary menu, like CARLA or RESULTS. Multiple commands can be specified, separated by a semicolon.

## **SIMESM={RACF|ACF2|TSS}**

This optional parameter makes the ISPF interface (in particular, the configuration panels for zSecure Alert) behave as if it ran with RACF, respectively. ACF2 as the active External Security Manager. This way, you can configure, for example, a zSecure Alert configuration for ACF2 while running in a TSO-session under RACF. By default, the active External Security Manager is used.

## **SYS**

Identifies the system to analyze. It is used as a qualifier when creating CKFREEZE and UNLOAD data sets, and for intermediate data sets where generated commands etc. are stored. The install process updates the sample configuration with the SMF system id, prefixed with an S in case the SMF system id starts with a digit (which would result in invalid data set names). This parameter must be modified if you distribute the CKRPARM data set from your installation system to another system.

## **UNIT**

The generic or esoteric unit name for UNLOAD, CKFREEZE, and permanent ISPF work data sets, SYSDA, DISK, or DASD, for example.

## **UPREFIX**

Can be specified to indicate installation or user-specific zSecure libraries that must be used in addition to the common data sets. The ISPF interface searches for additional libraries. If the libraries exist, they are concatenated in front of the corresponding zSecure common libraries. The library names searched for are:

```
&UPREFIX. .SCKRPLIB
&UPREFIX. .SCKRMLIB
&UPREFIX. .SCKRSLIB
&UPREFIX. .SCKRTLIB
&UPREFIX. .SCKRCLIB
&UPREFIX. .SCKRLOAD
&UPREFIX. .SCKRCARL
&UPREFIX. .SCKRCJPN
&UPREFIX. .SCKRMJPN
```

If you have old data sets (with SC2RPLIB as the low-level qualifier instead of SCKRPLIB, for example), and want to continue to use them, rename the data sets or create aliases. Instead of a single UPREFIX parameter, you can also specify a comma-separated list, enclosed by single quotes.

In line mode, the UPREFIX parameter supports only SCKRCARL library.

## **USRDATA**

The USRDATA parameter can be used to specify installation-defined user data fields that are stored in USER profile. The specified fields are shown in the IBM Security zSecure Group Administration displays. If you store phone and social security numbers (SSN) numbers in user data fields with this name, you could code:

```
USRDATA='PHONE SSN'
```

## **VOLSER**

Can be used to specify the volume serial on which to allocate UNLOAD, CKFREEZE, and work data sets.

## **WORKLLQ**

Low-level qualifier to be appended to permanent work data set names.

## **WORKPREF**

Can be specified to set the prefix of the ISPF work data sets. When not specified, the prefix is constructed by evaluating the SYSPREF variable as set with the TSO PROFILE command. When SYSPREF is not empty and also unequal to SYSUID, the prefix is set to `syspref.sysuid`.

Otherwise, work data sets names start with `sysuid`.

The WORKPREF parameter setting allows you to have unique prefixes for all users because the work data sets cannot be shared. This result can be achieved by using the SYSUID. (with trailing period) variable as shown in the following example:

```
WORKPREF='&SYSUID..C2R'
```

## WPREFIX

The WPREFIX parameter specifies installation or workgroup-specific zSecure libraries that must be used in addition to the common data sets and user data sets. (See the UPREFIX configuration parameter.) The ISPF interface searches for additional libraries. If these libraries exist, they are concatenated front of the corresponding zSecure common libraries. The library names searched are:

```
&WPREFIX. .SCKRPLIB
&WPREFIX. .SCKRMLIB
&WPREFIX. .SCKRSLIB
&WPREFIX. .SCKRTLIB
&WPREFIX. .SCKRCLIB
&WPREFIX. .SCKRLOAD
&WPREFIX. .SCKRCARL
&WPREFIX. .SCKRCJPN
&WPREFIX. .SCKRMJPN
```

If you have old data sets (with SC2RPLIB as the low-level qualifier instead of SCKRPLIB, for example) and want to continue to use them, rename, or create aliases. Instead of a single WPREFIX parameter, you can also specify a comma-separated list, enclosed in single quotes. In line mode, the WPREFIX parameter supports only the SCKRCARL library.

## DESC, CKFREEZE, UNLOAD, SMF

These parameters describe a set of input files to be available to all users of this zSecure configuration. The input files specified become the default set. These parameters work only in combination. That is, if you code DESC, also code at least one of the following: CKFREEZE, UNLOAD or SMF. Only specify SMF if zSecure Audit is included in your license. The input file set is the active set for all users who do not reset their input source on entry to zSecure to the one they used in the last session. Resetting the input source is default behavior, so for new users the parameter DESC/CKFREEZE/UNLOAD/SMF is only effective if you also change the setup default to no reset. This value can be changed using the Setup Default option. The Setup Default option is also the preferred way to customize the set of Input Files.

For example, if you code the following statements:

```
SET  DESC='Daily refreshed input files'
SET  CKFREEZE='sys2.cnr.daily.ckfreeze'
SET  UNLOAD=''
SET  SMF='sys2.cnr.daily.smf'
```

zSecure uses the CKFREEZE and the SMF data sets indicated along with the Live primary RACF database.

For more information about setting up the defaults, see [Appendix E, “Configuring the ISPF interface,” on page 211.](#)



---

## Appendix E. Configuring the ISPF interface

This section provides information about setting the default options for zSecure ISPF panels and other ISPF-related functions.

- [“Setup of default options for user groups \(Setup menu\)” on page 211](#)
- [“Configuring zSecure Admin to create new userids in the RACF database” on page 222](#)
- [“Locally defined functions” on page 222](#)

---

### Setup of default options for user groups (Setup menu)

---

Setup default (SE.D) is the preferred way to customize options for groups of users. It updates the data set that the PROFDSN parameter identifies so that you can create different default settings for separate groups of users. You can run Setup default against a new data set for testing purposes, and rename or copy the data sets when you are done, so your users are not affected by an incomplete change. If you use only a single PROFDSN data set, Setup default sets system-wide options. If no default settings are present in the PROFDSN data set, standard (zSecure-shipped) settings are used. To prevent corruption of the user interface, restrict access to the Setup default menu option, and only grant update access to the PROFDSN data set to staff that understand this process. For additional information about restricting access to menu options, see [“Resources that configure which options are shown” on page 193](#)

To run Setup default, start the ISPF interface with the selected PROFDSN data set. You can complete this task by selecting a configuration that specifies PROFDSN=*selected.data.set*, or by using PROFDSN(*selected.data.set*) as an override when invoking C2REMAIN.

Next, run SE . D. A panel like the normal SETUP panel is shown. After you change and exit the SETUP DEFAULT panel, you are prompted with the following panel:

```
zSecure defaults, Profdsn: SELECTED.DATA.SET

Choose:      (N=only new users will use defaults)
              (Y=all users will receive new defaults)
              (D=discard all changes,
               not possible for INPUT FILES and NLS)
```

If you want users to use the default settings each time they enter a zSecure session, specify INIT=YES in their configuration file.

New users always use the default system settings. If you choose N, they are the only ones to use the new settings. If you choose Y, the new settings are used by all users of this PROFDSN the next time they start IBM Security zSecure. Use D to discard all changes made, except for the NLS and INPUT FILES. Changes for these options cannot be discarded.

**Note:** Only the files defined in SE . D . 1 are changed, added, or deleted; the other files added to SE . 1 remain the same.

Therefore, choosing Y does not affect any other addition or modification to a user's configuration.

### Setup (default) National Language Support (SE.D.N)

zSecure includes the functionality to select the language for panel displays. In addition to selecting the language, you can also customize the text on the panel using the selected language. Currently, language specifications are limited to selection panel options. For additional information see [“Selecting a different language” on page 213](#) and [“Customizing individual menu options” on page 213](#).

zSecure provides the following support for DBCS:

## Input fields

- DBCS characters for input fields that support DBCS are accepted. For example, the RACF programmer name and Installation Data fields accept DBCS characters for input fields.
- DBCS characters for input fields in CARLa like COPY NEWNAME and NEWDATA are accepted and work.
- DBCS characters are accepted in modifiable fields, also known as overtypable fields, and are not garbled.
- The CKGRACF REASON field accepts data in DBCS.
- DBCS characters are accepted as input when they are entered as part of a quoted string or comment. They are not generally supported outside of quoted strings except in specific cases like the ISPF FIND primary command, and CARLa (Auditing and Reporting program language) scan strings.

## Command support

- Primary commands that contain DBCS strings are accepted on all panels.
- The SELECT FIELD= SCAN= command works for DBCS strings. However, DBCS strings must be enclosed in quotes.
- ISPF Edit and Browse sessions allow for mixed mode which means that editing DBCS strings works correctly.
- The FIND command supports DBCS search values as long as the search value is enclosed in delimiters. Delimiters can be either single quotation marks (') or double quotation marks (").

## ISPF menu, display, and report panels

- With option SETUP NLS, the Japanese language can be selected. If this option is selected, some user interface items are displayed in Japanese including: the Main Menu, the RA.H menu option, action commands, and action bars.
- DBCS strings on ISPF panel displays and Japanese reports generated by zSecure are displayed correctly.
- Complete DBCS strings on customer written or adapted reports are displayed correctly. Truncated fields might not display correctly.
- zSecure displays and reports that include audit concerns, as well as option AU.V and AU.S have been translated into Japanese. Except for menu options, most other panels are still in English.
- NLS tables containing DBCS characters are processed correctly.
- ISPF messages are translated into Japanese.

## Formatting

- Uppercase translation leaves DBCS alone.
- DBCS strings with WORDWRAP take into account language restrictions on line breaks when possible (for Japanese only).
- E-mail with a UTF-8 format attachment correctly contains DBCS translated characters, if the user passes the proper (mixed DBCS) CCSID.
- XML in UTF-8 format correctly contains DBCS translated characters, if the user passes the proper (mixed DBCS) CCSID. However, if stylesheet imbedding is used, then a stylesheet in the user's CCSID must be used.
- CCSID stylesheet support: Users must use a stylesheet included in their CCSID. Stylesheet CCSID 939 and CCSID 1047 can be used interchangeably. Stylesheet CCSID 1388 does not work.
- zSecure exploits the z/OS support for JIS X 0213:2004.

## Limitations

- All CKRCARLA messages and help text are still completely in English. ISPF messages are translated into Japanese.

- IP\_PORT audit concerns are not translated into Japanese.
- Long DBCS INSTDATA display as formatted by RACF LISTUSER is garbled on the MI panel, but also garbled by LISTUSER so accepted as correct.
- Text in generated e-mails (for example, subject lines) cannot contain DBCS characters.
- Uppercase translation leaves DBCS alone. That is, PRINT CAPS only works for NEWLISTs that contain DBCS with a LANGUAGE statement.
- ISPF service calls require commands to be issued in uppercase when the terminal mode is 3277KN or 3278KN. zSecure uses lowercase ISPF commands. To prevent the zSecure UI from failing, zSecure changes the ISPF terminal mode to 3278 dynamically at startup. Because the terminal mode is set for the entire TSO session, it is also active for non-zSecure applications running in a split screen session. To indicate that the terminal mode has been changed, the following warning message is issued:

```
ISPF terminal type changed from 3278KN to 3278.
Terminal type will be set back to 3278KN after exiting the
zSecure UI. Please note that while zSecure is active, all logical
screens (SPLIT SCREEN) will also use ISPF terminal type 3278.
```

Upon exit, the terminal mode is switched back to the original setting.

## SE.D.N panels

When you choose **N** from the SETUP panel, the following panel is displayed:

```

Menu   Options   Info   Commands
-----
zSecure Suite - Setup - NLS

Command ==>

Change language to
1 1. English
   2. Dutch
   3. French
   4. German
   5. Italian
   6. Portuguese
   7. Spanish
   8. Japanese
   9. Other
  0. zSecure
Language used: User English

Action on language items
1 1. No action (only use specified language)
   2. Reset all items to company default
   3. Reset all items to IBM Security zSecure default
   4. Customise menu items
   5. Customise action line commands
```

Figure 12. SETUP - NLS panel

### Selecting a different language

From the panel shown in [Figure 12 on page 213](#), you can define the language to use on the zSecure menu panels. The **IBM Security zSecure** language option is defined for easier communication with IBM Software Support software support to address questions or problems. When you contact IBM Software Support, you will be asked to change the NLS support option to IBM Security zSecure. After completing the support call, you can switch back to your (own) installation defined language. The **Other** language is for any other, not specified, language.

To reset a language to the default option (zSecure-defined) or to your company-selected option, specify one of the **reset** options in the **Action on language items** list. Options **2** and **3** allow you to change all language items except Japanese and zSecure languages.

### Customizing individual menu options

For all languages except Japanese and zSecure, selecting Option **SE.N** Action **4** opens the panel shown in [Figure 13 on page 214](#) so that you can customize the menu items. The text displayed on this panel is dependent on your own specifications. If you select Option **5**, the action line commands are displayed so that you can customize them. Options 4 and 5 are not available for the Japanese and zSecure languages.

**Note:** For information about double-byte character set (DBCS) support in zSecure, see [“Setup \(default\) National Language Support \(SE.D.N\)”](#) on page 211.

```

Menu          Options      Info      Commands      Setup
-----
zSecure Suite - Setup - NLS          Row 1 to 24 of 337
Command ===>          Scroll ===> CSR
Specify action for menu item(s): Edit, Insert, Copy, Delete
Standard options  Option and text as shown on panel

- SE          SE Setup          Options and input data sets
- SE 0        0 Run              Specify run options
- SE 1        1 Input files      Select and maintain sets of input data set
- SE 2        2 New files        Allocate new data sets for UNLOAD and CKFR
- SE 3        3 Preamble         CARLA commands run before every query
- SE 4        4 Confirm          Specify command generation options
- SE 5        5 View             Specify view options
- SE 6        6 Instdata         Customize installation data appearance
- SE 7        7 Output           Specify output options
- SE 8        8 Command files    Select and maintain command library
- SE 9        9 Certificates     Specify templates for new digital certific
- SE A        A Alert          Configure zSecure Alert
- SE A A      A Alert          Select and customize alerts
- SE A E      E E-mail        Configure e-mail address lists
- SE A P      P PCI          Configure PCI data sets, userids, and grou
- SE A S      S Sensitive    Configure sensitive resources, userids, an
- SE B        B Collections  Select and maintain collections of input s
- SE U        U User defined  User defined input sources
- SE N        N NLS          National language support
- SE T        T Trace        Set trace flags and CARLA listing for diag
- SE V        VM VM Files    Copy RACF/VM database (VM only)
- SE W        W Windows      zSecure Visual RACF configuration
- SE D        D Default      Set system defaults
- SE D 0      0 Run              Specify run options

```

Figure 13. SETUP - NLS panel showing all menu items

The zSecure Suite - Setup - NLS panel shows a table with all available zSecure menu options. The table includes items that you are not allowed to use because of limitations in your license or system specifications. You can **edit, copy, insert**, or **delete** any of the items of this table. The order shown in the Setup panel determines the order of the menu items that displays when you use the zSecure product panels. To add a line below the current line, type **I** in the selection field. To move an item, add the item in the correct location, then delete the old item.

Each time you enter a line-command, the following panel is displayed:

```

Menu  Options  Info  Commands
-----
zSecure Suite - Setup - NLS
Command ===>
Official option . . SE 0          (also used for profile checking)
Specify action for menu item
1 1. Use as specified below      2. Delete menu item
   3. Reset to system defaults    4. Reset to IBM Security zSecure defaults
Menu option . . . . 0          (as displayed on user menu)
Short description . . Run
Long description . . Specify run options
Command (or MENU) . CMD(%C2REDFLL NO &C2RNSE0)
Panel for "MENU" . . _____ New menu . . N (Y/N)

Press ENTER to continue.

```

Figure 14. SETUP - NLS panel shown after selecting a line-command

On this panel, the following fields can be specified:

**Official option**

This field specifies the link to the SAF resource which is checked for this menu option. It can be a three-level deep specification. The specification also determines which menu the option are on (that

is, specifying RA 4 2 results in the option being placed on the RA.4 menu). The official option can only be changed when the menu option is initially defined. That is, the option can only be changed with **copy** or **insert**, not with the **edit** line command. For user added menu items, except on the LO menu, the Official option fields must contain at least one of the following characters: @, # or \$. Otherwise, the added option is deleted during an NLS upgrade.

### Action

The action determines what to do with the menu option. If you used the **delete**-line-command, the action is initialized to one of the following:

Option **2** indicates that pressing ENTER deletes the menu option.

Option **3** resets the table-item to your systems default. If there are no system defaults, zSecure defaults are used.

Option **4** resets the item to the original zSecure settings for the language you specified.

### Menu option

The menu option as it is to be displayed on the panel. User changes to default menu options are not propagated during an NLS upgrade. For user options, the Menu option must match the Official option. A user-defined Menu option cannot be the same as an existing first level (main menu) option or another menu option in the same menu.

### Description

Specifies the short and long description as displayed on the product menu panels. Descriptions can contain ISPF variable names. These variables are resolved when the menu is displayed during product operation. User changes to default menu options are *not* propagated during an NLS upgrade.

### Command (or MENU)

Specifies the command to be performed when the option is specified. The value can be a command (%CMD) or panel as normally specified in the ISPF panel body. To indicate that the next panel is a menu, specify MENU xx, where xx is the official option under which the menu items are defined. So, **MENU SE** displays a menu containing all the items with an official option SE xx. **MENU SE D** displays a menu containing all the items with the official option SE D zz. To display the next menu on a different panel than the main panel, specify your own panel in the field **panel for menu**. Except for the LO (local) option, user changes to the command field are *not* propagated during an NLS upgrade.

### Panel for "MENU"

Panel to be selected if the menu option is taken.

### New menu

This option can be used to add a primary menu option to be either shown on a new menu (**Y**) or expandable on the main menu (**N**). When expandable, leave the panel for the MENU field blank and use MENU option name for Command (or Menu), MENU R@ for example.

After you press Enter, the following panel is displayed:

```

Menu  Options  Info  Commands  Setup
-----
zSecure Suite - Setup - NLS
Command ==> -----

Display Menu option for any of the following programs
/  RACF Admin
/  RACF Audit
/  RACF Report
/  CKGRACF
/  ACF2 Admin
/  ACF2 Audit
/  ACF2 Report
-  TSS Audit
-  ZSCC
-  Visual
-  Alert
Select any of the following options
/  Menu option is supported on z/OS
/  Menu option is supported on z/VM
/  Menu option is a z/OS analyzing option
/  Menu option is a z/VM analyzing option
-  Menu option is only available on ISPF 5.0 and up

```

Figure 15. Setup - NLS panel showing the Display Menu option

On this panel, the following fields can be specified:

**Display menu item for any of the following programs**

This field specifies the licensed functions where the menu item is to be used. Table 20 on page 216 shows the relationship between product (license) and function:

Table 20. Relationship of products to licensed functions in NLS

Product	RACF Admin	RACF Audit	RACF Report	ACF2 Audit	ACF2 Report	TSS Audit	CKGRACF	Z S C C
zSecure Admin	/						/	
zSecure Audit for RACF		/	/					
zSecure Audit for ACF2				/	/			
zSecure Audit for Top Secret						/		
Z Security and Compliance Center		/	/	/	/	/		/

Except for the LO (local) option, user changes to licenses are *not* propagated during an NLS upgrade.

**Menu option is supported on z/OS**

Tag this field when functions used by this menu item are supported on z/OS.

**Menu option is supported on z/VM**

Tag this field when functions used by this menu item are supported on z/VM.

**Menu option is a z/OS analyzing option**

Tag this field when the menu item is used for analyzing z/OS data.

### Menu option is a z/VM analyzing option

Tag this field when the menu item is used for analyzing z/VM data.

### Menu option is only available on ISPF 5.0 and up

Tag this field when functions used by this menu item are only supported on ISPF 5.0 and up (CUA attributes, for example).

## Customizing action line commands

You can use option 5, Customize action line commands, to add, remove, and edit line commands to be used in reports in the zSecure user interface. The option is primarily intended to allow localization of line commands, but it is also possible to define new line commands for functions not included in zSecure. A line command can have two types of actions:

- A line command can call an ISPF panel or call zSecure built-in display or command generation routines. A called ISPF panel can in turn return generated RACF commands, return a value to call built-in routines after all, or return a value to call another panel or REXX.
- A called panel or REXX can use several predefined ISPF variables to obtain its data, and by using CARLa-mapped fields, to obtain values from the CARLa fields used to define the display (either shown on the display or hidden using the nondisplay modifier).

It is possible to configure line commands as a block, but certain restrictions apply. For example, it is not possible to use CARLa-mapped fields. For more information, see [Map CARLa fields into ISPF variables](#).

Selecting option 5 and pressing enter shows the list of all defined line commands as well as where they can be used primarily. A line command has two primary controls to say that it is valid on certain records: NEWLIST TYPE and ENTITY TYPE. ENTITY TYPE can be mostly viewed as, for example, USER (for instance, in type=RACF), DATASET (for instance, in type=RACF), and others.

You can configure line commands to be valid or not valid only in certain places, for example, on certain RACF classes or segments. If they are restricted by this type of limiting criteria, they won't show up in the menu shown as a result of the / line command. The panel below shows only the primary settings, not the additional restrictions by class or segment.

Menu	Options	Info	Commands	Setup
zSecure Suite - Setup - NLS      Row 33 to 66 of 379				
Command ===> _____ Scroll ===>				
CSR_				
Specify action for menu item(s): Edit, Insert, Copy, Delete				
Standard options    Option and text as shown on panel				
-	RC D AC	AC Access	Access Check for one userid or group	
-	RC D C	C Copy	Copy data set profile	
-	RC D D	D Delete	Delete data set profile	
-	RC D D	D Delete	Delete data set segment	
-	RC D E	E Event	Display event logging	
-	RC D L	L List	RACF listdsd command	
-	RC D LD	LD List profile	RACF listdsd DSNS command	

Figure 16. Setup - NLS panel showing the action item list

- The first column lists the NEWLIST type for which the line command is valid.
- The second column lists the entity for which it is valid.
- The third column lists the action of the line command.
- The fourth column lists what the user types in as the line command; in combination with the sixth column (description), this is what is displayed when you enter / as line command.
- The fifth column is only there to help you differentiate between the line commands.

Selecting the first line command when you type an E on the input field above the line command results in the following panel.

```

Menu   Options   Info   Commands
-----
zSecure Suite - Setup - NLS
Command ==> -----
Newlist type . . . . RC          (i.e. RC for RACF)
Entity type  . . . . D          (i.e. U for USER)
Action . . . . . AC          (i.e. C for COPY)

Specify action for menu item
  1 1. Use as specified below      2. Delete menu item
  3. Reset to system defaults     4. Reset to IBM Security zSecure defaults

Used action . . . . AC          (as displayed on user menu)
Used block action. . --          (optional; requires special support)
Short description . Access
Long description . . Access Check for one userid or group
Panel . . . . . C2RP&CKREREL.AC@ (panel to use for action specification)

/ Map CARLa fields into ISPF variables
_ Specify classes and segments for which this action is valid or not valid

Press ENTER to continue.

```

Figure 17. Setup - NLS panel showing newlist types, entity types, and actions

The following fields are defined on this panel:

**Newlist type**

The newlist type on which the line command is valid.

**Entity type**

The entity type on which the line command is valid.

**Action**

The action identifier. It is used to identify the resource that will be checked when restricting through profiles in the XFACILIT class. This field is also used as identifier for calling zSecure built-in display or command generating routines. Only line commands where Action contains a @, #, or \$ character will be kept during an NLS table upgrade.

**Specify Action for menu item**

This option can be used to delete the line command or reset it to defaults.

**Used action**

The actual characters typed on the zSecure panel to perform the line command. These are the letters used for a single command, in case the line command also allowed blocked line commands.

Changing **Used action** for a zSecure built-in entry will not be kept during an NLS table upgrade.

**Used block action**

The actual characters to be typed on the zSecure panel to indicate the start and end of a block of records, to be processed in one go. When this field has a value, all single record line commands are also processed in one go.

A block action always guarantees that the predefined ISPF variables that can be used by any called panel or REXX has the same value over all the records in the block. If the block crosses, for example, several complexes, it is split-up in multiple calls to the panel or REXX, presenting the specified panel multiple times.

Because CARLa-mapped fields are very likely to have different values for records in a block, zSecure disallows such fields when a block action is defined.

**Description**

A short and long description for the line command. Both are shown on the previous panel. The long description will be used on the menu displayed when performing the / line command.

Note that updates to the zSecure default set of line commands are not kept during an NLS table update.



## Panel

This field contains the panel that will be called when the line command is typed. If the field is empty, the built-in action, configured by **Action** is called.

The variable &CKREREL is used by zSecure to differentiate between either pull-down and non-pull down panels.

## Map CARLa fields into ISPF variable

This option allows using values from the display (either shown or hidden) as data for the shown panel. Using this is disallowed when a Used block action has been specified. If you select this field, continue at [Map CARLa fields into ISPF variables](#).

## Specify classes and segments for which this action is valid or not valid

Depending on the Entity type used by the record, this option allows to specify a list of classes and a list of segments on which this line command is valid or not valid. If you select this field and not the **Map CARLa fields into ISPF variable** field, continue at [Specify classes and segments for which this action is valid or not valid](#).

If you have not selected either of these last two fields when you press **Enter**, [Figure 15 on page 216](#) is shown.

## Map CARLa fields into ISPF variables:

If you selected the **Map CARLa fields into ISPF variables** field on the previous Setup NLS action panel ([Figure 17 on page 218](#)), the following panel is displayed when you press **Enter**.

```
Menu  Options  Info  Commands
-----
zSecure Suite - Setup - NLS
Command ==>> _____
Specify CARLa - ISPF matches
CARLa      ISPF      CARLa      ISPF
KEY        CKRRPROF  _____  _____
_____
_____
_____
_____
_____
_____
Press ENTER to continue.
```

*Figure 18. Setup - NLS panel for configuring field matches between CARLa-variables and ISPF-panel variables*

The **CARLa** column contains the name of the CARLa variables from your query. The **ISPF** column contains the name of the ISPF variables used in your panel. Ensure that the CARLa variables are present in your query. Otherwise, an error message is displayed after the panel has been displayed when performing the action.

## Specify classes and segments for which the action is valid or not valid:

If you selected the **Specify classes and segments for which this action is valid or not valid** field on the previous Setup NLS action panel ([Figure 17 on page 218](#)), the following panel is displayed when you press **Enter**.

```

Menu          Options      Info      Commands      Setup
-----
zSecure Suite - Setup - NLS
Command ==> -----
Newlist type . . . : RC
Entity type   . . . : R
Action       . . . : D

Specify classes for which this action is valid:
-----
Specify classes for which this action is not valid:
-----
Specify segments for which this action is valid:
BASE-----
Specify segments for which this action is not valid:
-----

Press ENTER to continue

```

Figure 19. Setup - NLS panel for configuring field matches between CARLa-variables and ISPF-panel variables

On this panel, you can configure for which classes or segments a line command is valid or not valid. You can create different actions for the same Used action depending on class and segment, by defining multiple line command entries for the same NEWLIST type, Entity type, and Used action. For example, one delete command (D) for a specific resource class (specified in the **Specify classes for which this action is valid** field) and another delete command (D) for all other classes (leaving this panel empty).

The display shows a line command D (Action, so the built-in identifier) that is valid on most record elements, except on displays of the BASE segment.

Depending on the entity type of the line command this allows the following:

Table 21. Entity types versus allowed selects / excludes

Entity type	for	Allows to select / exclude
D	DATASET	Segment
G	GROUP	Segment
R	General resource classes	Class and segment
U	USER	Segment
M	Multi-complex summary	Class and segment
Other		None

The system searches the NLS table for the most-matching entry. Meaning, if class and segment are both specified and correct, such entry applies even if there is an entry specifying only a correct class. If there is no such unique match, the first element that matches most is used.

## Setup (default) Installation defined names (SE.D.I)

Some data set and profile names are customizable and as such might vary from site to site. This option provides a means of telling the program the names used at your site.

The following panel is shown.

```

Menu      Options  Info      Commands
-----
zSecure Suite - Setup - Installation
Command ==>
Installation specific names
JES/328X data set mask . SY51.JSXLOG.**_____ (EGN mask)

```

Figure 20. Setup - Installation panel showing the JES/328X data set mask

**JES/328X data set mask**

Specify an EGN mask that covers the names of your JES/328X log data sets. The mask is only meaningful if you use JES/328X for remote printing. This mask is used by the JES/328X definitions and log data sets report, option RA.3.D. See the *User Reference Manual* for more details.

**Setup (default) Command files (SE.D.8)**

The SE.D.8 menu option allows you to allocate and select an existing library for subsequent use. When used together with **SE.D** it can be used to set a company wide default. Initially, it contains only DD:CKRCARLA for the product sample library. With the **I** line command you can insert new data set names. To activate a set use the **S** line command.

```

Menu      Options  Info      Commands
-----
zSecure Suite - Setup - Command file Row 1 from 3
Command ==>                               Scroll ==> CSR
Select sample library or work with a library (E, R, I, or D)
Sample library
- DD:CKRCARLA_____
***** Bottom of data *****

```

Figure 21. Setup - Command file panel showing the library selection

Each line in the library selection list must contain a data set name using TSO conventions, or DD: followed by an allocated filename. The library marked selected is used by the options of the **CO** Commands menu. Concatenations are not supported.

The following line commands can be used.

Table 22. Line commands used with the selected library	
E	Show the members in this library
D	Delete the line from the selection list, the data set is <b>not</b> deleted
I	Insert an empty line following this line, it is not selected automatically
R	Repeat name in this line
S	Select the library for subsequent use

When you have selected a data set, you can call the member list with the **E** line command, or option **CO.2**.

**Retaining your Setup default data when upgrading zSecure**

The PROFDSN data set in use is specified in the PROFDSN parameter in your configuration. For additional information, see [Appendix D, "Configuration parameters and members,"](#) on page 205.

# Configuring zSecure Admin to create new userids in the RACF database

---

## About this task

**Note:** You only need to perform this procedure if you use zSecure Admin.

zSecure Admin can be used to create new userids in your RACF database. If these new userids are to use TSO and ISPF, you need to specify a user catalog, an ISPF profile data set, and a DATASET for all data set that the new user owns.

## Procedure

To specify these values, make the following updates to the specified members in the `hlq.ckrparm` configuration data set:

1. To select the user catalog for this configuration, update member C2RSMUMA to specify an alias in the master catalog that points to one of your user catalogs.
2. To specify an ISPF profile data set for the new user, update member C2RSMUMP in the configuration data set to your UNIT and data set naming conventions.
3. To specify a DATASET profile for all data sets that the new user owns, you might need to update member C2RSMUMH to conform to the data set naming conventions you implemented during the installation and distribution procedures.

The alias and the ISPF profile are created when you use the MT function (Manage TSO information) on the RA.U display, not when you copy a user. In a multi-image environment the alias and ISPF profile data set are required in each z/OS image. Because ISPF profile data sets are not supposed to be shared, you might want to use different names in each z/OS image.

## Locally defined functions

---

The LOCAL option is designed to be customized by personnel who are experienced in writing ISPF dialogues. Panel names starting CKRP3C\* are reserved for user customized help panels. New or modified features can be added to the NLS tables as sub options of option **LO** (see Section “Setup (default) National Language Support (SE.D.N)” on page 211). The LO panel has space for a maximum of 12 options. If you modify or replace the panel used by option **LO** (C2RP3C@@) make sure you do not to overwrite your changes when you upgrade zSecure. The suggested way of adding or changing the supplied panels is to concatenate your own libraries in front of the zSecure product libraries. These libraries can be specified in your zSecure configuration using the WPREFIX or UPREFIX parameters, see Appendix D, “Configuration parameters and members,” on page 205). When the zSecure-supplied panel C2RPxC@@ is replaced by a user panel, make sure that the following lines are added to this user panel:

```
IF (.RESP=END)
  &C2RCOMM = 'MAIN'
  VPUT C2RCOMM SHARED
```

These lines are necessary for properly returning to the main menu.

The options provided on the LO panel are meant to serve as examples:

### P - Panel

Causes the panel specified in the panel field to be selected using the ISPF SELECT PANEL service. The panel must be a valid selection panel (That is, make sure to set ZSEL .)

### C - Command

Causes the command entered in the Command field to be selected using the ISPF SELECT CMD service.

### R - Start CKRERUN

Starts REXX exec CKRERUN using `CMD(%CKRERUN PANEL(*))`. No input from the panel is used. CKRERUN reads CARLa commands from variable CKRCMDV in the shared pool and displays the

results. Multiple lines of commands can be separated in CKRCMDV by the new line delimiter character 'x'15'. As supplied, CKRCMDV is not filled by option LO.

## Command generation

The CKRERUN program displays the specified dialog panels, runs the command, and displays the results. You can use this program from your own programs. CKRERUN is called with the following parameters:

**PANEL(panel1 panel2)**

selection and (optional) result panels

**RESULT(panel)**

show result panel

**REUSE(files)**

do not clear specified files

**SYSIN(member)**

(also) included member of CKRCARLA

**HELP(panel)**

member of SCKRPLIB for use in BROWSE

**PERFORM(command)**

TSO command to be called instead of CKRCARLA

If your installation defines its own ISPF panels, a user exit can be called to generate TSO commands for example. These commands are displayed to the user so that they can be confirmed, executed, and queued like the TSO commands generated by zSecure.

To configure the user exit, call the CKRERUN command from an ISPF panel with the PERFORM and PANEL parameters:

```
CKRERUN PERFORM(xxxx) PANEL(yyyy)
```

xxxx is the command which implements the installation-defined actions, for example:

```
/* ADDALIAS REXX */
push "DEFINE ALIAS (NAME(' ' || uuser || ' ') RELATE(' ' || ucat || ' '))"
'EXECIO 1 DISKW CKRCMD (FINIS'

'EXECIO 0 DISKW CKREPORT (FINIS OPEN'
```

The EXECIO to CKREPORT ensures that the user does not need browse the CKREPORT file. (The RESULTS panel displays the first non-empty file from CKREPORT, CKRCMD, CKR2PASS, and SYSPRINT.)

The ISPF panel yyyy could be:

```
%----- Define catalog alias for userid -----
%COMMAND ==>_ZCMD

+Userid          ==>_USER  +
+Usercatalog     ==>_UCAT          +(no quotes)

)PROC
  VPUT (USER UCAT) SHARED
  &CKRNEXT = &Z          /* no continuation panel */
)END
```

Figure 22. Example of the ISPF panel yyyy

Set CKRNEXT to the membername of the next panel to display, or clear the variable to indicate that this is the last panel. When CKRNEXT is empty, the function defined by the PERFORM statement is executed, or, if PERFORM was not specified, zSecure is run. In the latter case, the variable CKRCMDV is passed to zSecure to perform the user specified option.



---

## Appendix F. Interfacing with IBM Security Verify Adapter for RACF with optional zSecure support

IBM Security Verify Governance (ISVG) can be used to view access data that is defined on a z/OS RACF system, using the IBM Security Verify Adapter for RACF with optional zSecure support. This adapter can implement zSecure CARLa scripts to collect basic information about users and groups and the access that the users and groups have to resource profiles. The zSecure CARLa scripts that the adapter can use query the NEWLIST TYPE=RACF\_ACCESS fields. The adapter can also be used to modify user attributes.

For an overview of all operations and attributes that the IBM Security Verify Adapter for RACF supports, see [IBM Security Verify Governance Application Adapters](#) on IBM Documentation.

The zSecure CARLa scripts that the IBM Security Verify Adapter for RACF uses are released or updated as part of the IBM Security Verify Adapter for RACF (not as part of the zSecure product).





# Appendix G. Troubleshooting: Large size of CKFREEZE data sets

For many zSecure reports and analyses, a CKFREEZE data set is needed. As described in “Capacity planning information” on page 31, there are different "types" of CKFREEZE data sets, depending on the purpose of the information. Another way to look at CKFREEZE data sets is to list the actual types of data that is present in the CKFREEZE data set. The data is represented by multiple record types and subtypes. You can create a report with statistics about these records in either option **SE.1** or **IN.F**:

- The detail information of a set-of-files. The detail information is shown when you use the E or B line command in option **SE.1**. Use the L line command on the individual data sets listed to show the record statistics for any CKFREEZE data set.
- **IN.F** shows a CKFREEZE line in the @SECDATA report. Use the L line command to show the record statistics for the CKFREEZE data sets that are currently selected for reporting.

The L line command can also be used on other types of zSecure input data sets (SMF, ACCESS - for RACF only, and UNLOAD - for RACF and ACF2 only). For a full overview, see section "SE.1 SETUP - Input files" in the *zSecure (Admin and) Audit User Reference Manuals*.

Aside from the record statistics, the report also includes information about the CKFCOLL run that created the CKFREEZE data set. An example output is shown in the following figure:

```

zSecure Suite Display Selection                               Line 1 of 4
Command ==> ----- Scroll==> CSR

  Name      Summary Records Title
-  RUNTIME      1          1 Runtime state information
-  COMMANDS     1          1 Collect parameters and commands
-  MESSAGES    266        266 Runtime messages
-  SPACE        64 1000428 Space usage
***** Bottom of Data *****

```

Figure 23. Overview of reports for a CKFREEZE data set

When selecting space, and, for example, sorting on bytes with "sort bytes d", it is possible to see the largest contributors to space in the data set.

```

Space usage                                               Line 1 of 64
Command ==> ----- Scroll==> CSR
                                22 Nov 2019
08:36
  Type      Records      Bytes Bytes Contents
62          133459      33521837 33M52 UNIX file
41           16189      5810369 5810k HSM migration control record
42           12035      3665385 3665k HSM backup control record
61           10497      2441619 2442k UNIX directory
54            95       2391791 2392k PDSE DESB
0C +         8814      1691113 1691k Security information
50            5933      1625642 1626k PDS directory blocks
40           11824      1469549 1470k Catalog record
31            7677      1143873 1144k VTOC data set
33           2381       997642 997k6 VVR/NVR
C2 +         5974       625166 625k2 DB2/MQ record
53            343       567789 567k8 IEFJOBS member jobcard
35 +          853       366726 366k7 Security information
C0 +          967       228448 228k4 CICS record
1E            830       162680 162k7 Address space DD information
32            37        151293 151k3 VVCR/VVCN/VVCM

```

Figure 24. Overview of the SPACE report for a CKFREEZE data set

CKFREEZE data sets can become large because of the amount of data collected. You can conserve disk space and reduce CPU processing time by turning off CKFREEZE features that are not required for daily tasks. You can also save disk space by using SMS compression for the CKFREEZE data set.

For more information, see "zSecure Collect for z/OS" in the *zSecure (Admin and) Audit User Reference Manuals*.

## Skip collection of certain records

In the [Overview of the SPACE report](#) for a CKFREEZE data set, the largest contributors to the size of the CKFREEZE data set are record types 62 and 61 (for UNIX information), 41 (for DFSMSHsm MCDS information), and 42 (for DFSMSHsm BCDS information). Depending on the type of reports that you want to create using this CKFREEZE data set, you might want to use the following options to skip collecting certain records:

### **BCD=NO**

The Backup Control Data set (BCD) data is currently mainly used for determining whether discrete data set profiles must be removed. These discrete data set profiles are found in various **AU.V - Verify** functions and are reported in the RACF profiles report (**RA.3.1**). The information is also used to report usage of data set encryption keys.

### **MCD=NO**

The Migration Control Data set (MCD) data is currently mainly used for determining whether discrete data set profiles must be removed. These discrete data set profiles are found in various **AU.V - Verify** functions and are reported in the RACF profiles report (**RA.3.1**). The information is also used to report usage of data set encryption keys.

### **UNIX=NO**

UNIX data requires substantial amounts of disk space (all directories, owner data, and file permissions are stored). UNIX data is used for TRUSTED reports, SENSITIVE data sets reports (zFS and HFS data set sensitivity), and auditing UNIX file systems. SMF records pertaining to UNIX files often do not contain the path to the file; the CKFREEZE information can be used to show the appropriate path. Without the UNIX data, most reports still give usable, though incomplete, results. When you are not auditing the files in the zFS or HFS file systems, you can disable UNIX data collection.

This information is used in the following NEWLIST types: UNIX, TRUSTED, REPORT\_SENSITIVE, DSN, SENSDSN, and SMF.

Other options can be used to disable data collection for PDS(E) directories, Db2 catalog contents, CICS resource definitions, and similar. For more information see the "zSecure Collect command reference" in the *zSecure (Admin and) Audit User Reference Manuals*.

## SMS compression

SMS compression can also reduce the amount of disk space that is needed to store the CKFREEZE data sets. To use this compression, you can either create a DATACLAS in SMS that specifies COMPACTION, or set a system default data set COMPRESS option in PARMLIB. Examine whether compressing the data sets improves overall performance. Reading a compressed data set uses less I/O but can require more CPU.

## Notices

---

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
224A/101

11400 Burnet Road  
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

## Trademarks

---

IBM, the IBM logo, and [ibm.com](http://ibm.com)<sup>®</sup> are trademarks or registered trademarks of International Business Machines Corporation, in the United States and/or other countries. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on [ibm.com/legal/copytrade](http://ibm.com/legal/copytrade).

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

VMware, the VMware logo, VMware Cloud Foundation, VMware Cloud Foundation Service, VMware vCenter Server, and VMware vSphere are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are registered trademarks of Hewlett Packard Enterprise, International Business Machines Corporation and Quantum Corporation in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.



---

# Index

## A

access levels, verifying [29](#)  
Access Monitor  
    C2PAMP member [64](#)  
    cleanup of RACF exits [74](#)  
    collect data [37](#)  
    collect UNIX event data [68](#)  
    configuration [62](#)  
    configuring function using parmlib [71](#)  
    data sets used [34](#)  
    defining collection of detail data [65](#)  
    defining consolidation files [66](#)  
    defining data collection files [66](#)  
    defining permissions [63](#)  
    defining security resources [63](#)  
    defining started task parameters [64](#)  
    detail data, defining users or classes [65](#)  
    EventsToAlert [65](#)  
    managing data collection [64](#)  
    modifying STC [71](#)  
    monitoring STC [71](#)  
    Operating [70](#)  
    operator command [76](#)  
    passing VERIFY event records [65](#)  
    preparing the JCL for the started task [63](#)  
    process data [37](#)  
    RACF VERIFY [65](#)  
    required data sets [64](#)  
    requirements for installation [62](#)  
    resolving data storage problems [72](#)  
    resolving memory problems [72](#)  
    sample parameter file [64](#)  
    setting up [61](#)  
    START parameters [70](#)  
    started task [38](#)  
    starting STC [70](#)  
    stopping STC [71](#)  
Access Monitor configuration command  
    DEBUG [78](#)  
    OPTION [80](#)  
    REPORT [86](#)  
Access Monitor operator command  
    CONSOLIDATE [76](#)  
    CRSH [76](#), [116](#)  
    DEBUG [76](#)  
    DIAGNOSE [76](#)  
    DISPLAY [77](#)  
    REPORT [77](#)  
    RESTART [77](#)  
    SIPL [77](#)  
    STOP [78](#)  
access to data, controlling [57](#)  
ACF2 reporting  
    verifying [29](#)  
action line commands  
    customizing [213](#)

ADDSMF  
    FILTER command [129](#)  
ADDWTO  
    FILTER command [129](#)  
Alert  
    ddnames [107](#)  
ALL  
    DEBUG command [121](#)  
analytics preprocessing  
    C2PACMON configuration files [69](#)  
    configure CDP [70](#)  
    customization [69](#)  
    prepare directory [69](#)  
APF authorization [18](#)  
Application Transparent Transport Layer Security  
    and secure communication [56](#)  
AT-TLS  
    and secure communication [56](#)  
    and zSecure Server [56](#)  
auditor  
    Creating a read-only auditor [201](#)  
AVERAGEINTERVAL  
    configuration [118](#)  
    REPORT command [127](#)

## B

B8ROPT options module [99](#)  
back out upgrade of zSecure Alert [132](#)  
Backup Control Data set [39](#)  
batch processes [21](#)  
BCD [39](#)  
bind failure  
    during Visual Server startup [155](#)  
BLKSIZE [205](#)  
BPX1004I  
    waiting on [152](#)  
BUFFER  
    DEBUG command [121](#)  
buffer size  
    configuration [119](#)  
Buffer size  
    OPTION command [125](#)  
buffer space for zSecure Alert events [41](#)  
buffer usage  
    monitor [119](#)  
BUFSIZE  
    configuration [119](#)  
    OPTION command [125](#)

## C

C2PACPRM [205](#)  
C2PAMJOB member [65](#)  
C2PAMP [64](#)  
C2PAMPCL member [65](#)  
C2PAMRCL member [65](#)

- C2PCUST [205](#)
- C2POLICE [205](#)
- C2PXDEF1 preamble member [114](#)
- C2R [201](#)
- C2R.CLIENT.EMPTYREASON.PWSET profile [146](#)
- C2R.CLIENT.SETROPTS profile [146](#)
- C2R.SERVER.ADMIN resource [146](#)
- C2R\$PARM [24](#), [27](#), [172](#)
- C2R\$PARM (Command Logger) [44](#)
- C2R\$PARM member [26](#)
- C2R2131A switch [139](#)
- c2rdiag** command [155](#)
- C2RELSI
  - files C2RELSI.userid.\* [157](#)
- C2REMAIN [27](#)
- C2RIISPF member [14](#)
- C2RIMENU [193](#), [194](#)
- C2RJPREP [42](#)
- C2RJXRFR [42](#)
- C2RSDFLT [26](#)
- C2RSERVE
  - started task or job [152](#)
- C2RSMUMA [222](#)
- C2RSMUMH [222](#)
- C2RSMUMP [222](#)
- C2RW131A [205](#)
- C2RWASSC member, C2RWCUST [205](#)
- C2RWCUST [205](#)
- C2RWEXG1 member, C2RWCUST [205](#)
- C2RWEXR1 member, C2RWCUST [205](#)
- C2RWIN [205](#)
- C2RZCZFS [136](#)
- C2RZWINI job [138](#)
- C2RZWRUT job [137](#)
- C2RZWUSR job [137](#)
- C2XACTV program [75](#)
- C2XEXITS [205](#)
- capacity planning
  - CKFREEZE [227](#)
  - CPU time [37](#)
  - DASD storage [36](#)
  - virtual storage [37](#)
  - zSecure Admin [36](#)
  - zSecure Alert [40](#)
  - zSecure Audit [38](#)
- CBPDO, installing as part of [9](#), [10](#)
- CEF
  - customize [168](#)
  - sample batch jobs [169](#)
- CEF data creation [168](#)
- checklist, installation [1](#)
- CHECKSPOOLSIZE setting [48](#)
- CKACUST [23](#), [25](#), [26](#), [205](#)
- CKACUSV [23](#), [26](#), [205](#)
- CKFREEZE
  - data set types [32](#)
  - fresh [42](#)
  - space requirement criteria [34](#)
- CKFREEZE data sets [36](#)
- CKFREEZE, manage size [227](#)
- CKG.UCAT profile [145](#)
- CKG905I [19](#)
- CKGRACF
  - checking [29](#)
- CKGRACF (*continued*)
  - related error messages [19](#)
- CKGRACF, daily [42](#)
- CKNSVPRM
  - zSecure configuration file [49](#)
- CKQ0ES, CKQCEF#0 [170](#)
- CKQCEF
  - operation [175](#)
- CKQCEF started task [173](#)
- CKQCES, CKQCEF#C [171](#)
- CKQEXSMF
  - configuration statements [177](#)
  - make SMF records available [163](#)
  - operation [176](#)
  - operator commands [182](#)
- CKQEXSMF server started task [174](#)
- CKQLEEF [164](#)
- CKQRADAR
  - operation [175](#)
- CKQRADAR started task [173](#)
- CKQXES, CKQCEF#X [171](#)
- CKR [20](#)
- CKR program [20](#)
- CKR REXX exec [27](#)
- CKR.READALL resource [201](#)
- CKR962F [19](#)
- CKRERUN program [223](#)
- CKRINST library
  - creating from the SCKRSAMP library [5](#)
  - definition [5](#)
  - updating members [15](#)
- CKRJOBS [23](#)
- CKRPARAM [23](#)
- CKRPARAM data sets [20](#)
- CKRPROF [23](#)
- CKRZSITE job [191](#)
- CKRZUPDI member [12](#)
- CKRZUPDZ job [15](#)
- CKX962F [19](#)
- CKXLOG [42](#)
- CKXPARAM (Command Logger) [44](#)
- COLLECT
  - MODIFY command [116](#)
- collection process, setup [164](#), [167](#)
- COLLECTSTCNAME
  - MODIFY COLLECT command [116](#)
  - OPTION command [125](#)
- COLLECTTIME
  - MODIFY COLLECT command [116](#)
  - OPTION command [125](#)
- Command Logger
  - C2R\$PARM [44](#)
  - CKXPARAM [44](#)
  - Command Logger
    - policy [46](#)
    - configure ISPF user interface [47](#)
    - configure zSecure Command Verifier [47](#)
    - define log stream [44](#)
    - enable calling CKXLOGID from APF environment [47](#)
    - operation [45](#)
    - setup [42](#)
    - STC procedure [45](#)
    - STC user ID [45](#)
    - USERID/STARTED profile [45](#)



- command routing
  - support for [49](#)
- compatibility of zSecure Visual components [139](#)
- compression, zEDC [72](#)
- configuration
  - assigning to batch jobs [27](#)
  - assigning to started tasks [27](#)
  - assigning to TSO/ISPF users [27](#)
  - CKQCEF started task (SIEM) [173](#)
  - CKQEXSMF [177](#)
  - CKQEXSMF server started task (SIEM) [174](#)
  - CKQRADAR started task (SIEM) [173](#)
  - file polling [172](#)
  - overview [5](#)
  - z/OS Compliance Integration Manager [89, 92](#)
- configuration commands
  - z/OS Compliance Integration Manager [94](#)
- configuration data set
  - creating [23, 25](#)
  - customizing [26](#)
  - definition [5](#)
  - maintaining during upgrade [26](#)
  - making available [27](#)
  - purpose [23](#)
- configuration parameters under z/OS [205](#)
- configuration, Visual Server
  - bulk creation of clients [142](#)
  - canceling a password [142](#)
  - client-server communication [140](#)
  - existing client [140](#)
  - introduction [140](#)
- configure ISPF user interface (Command Logger) [47](#)
- configure zSecure Command Verifier (Command Logger) [47](#)
- CONSOLIDATE operator command [76](#)
- CPREFIX [205](#)
- CPU time
  - zSecure Alert events [41](#)
  - zSecure Audit reports [40](#)
- CPU time, capacity planning [37](#)
- create APF-authorized program calling CKXLOGID [47](#)
- CRSH operator command [76, 116](#)
- custom event exit (SIEM) [171](#)
- customization
  - analytics preprocessing [69](#)
  - C2PACMON configuration files [69](#)
  - CDP [70](#)
  - prepare directory [69](#)
- customize for near real-time [172](#)
- customizing
  - action line commands
    - language options [213](#)
  - installation parameters [11](#)
  - menus
    - language options [213](#)

## D

- daily CKGRACF [42](#)
- DASD storage
  - CKFREEZE data sets [39](#)
  - types of data [31](#)
  - zSecure Alert reports [40](#)

- DASD storage, capacity planning [36](#)
- data compression, zEDC [72](#)
- data set
  - configuration [5, 23](#)
  - installation, security [7](#)
  - naming conventions [7](#)
  - set access level [7](#)
  - specifying user catalog [8](#)
  - zSecure configuration [5](#)
- data set, migrating [131](#)
- data, controlling access [57](#)
- DATACLAS [205](#)
- daylight saving time [41](#)
- DBCS
  - support [211](#)
- DDNAME
  - REPORT command [127](#)
- ddnames
  - Alert [107](#)
- DEBUG
  - MODIFY command [116](#)
  - START command [115](#)
- DEBUG BUFFER [119](#)
- DEBUG command
  - ALL [121](#)
  - BUFFER [121](#)
  - IO [121](#)
  - MAIN [121](#)
  - NOBUFFER [121](#)
  - NOIO [121](#)
  - NOMAIN [121](#)
  - NONE [121](#)
  - NOSMF [121](#)
  - NOWTO [121](#)
  - SMF [121](#)
  - WTO [121](#)
- DEBUG configuration command [78](#)
- DEBUG operator command [76](#)
- default options [211](#)
- Define Alias action [145](#)
- define log stream (Command Logger) [44](#)
- definition [59](#)
- DELSMF
  - FILTER command [129](#)
- DELWTO
  - FILTER command [129](#)
- deployment
  - overview [5](#)
  - zSecure software [23](#)
- DESC [205](#)
- DIAGNOSE command (Alert) [124](#)
- DIAGNOSE operator command [76](#)
- diagnostic information for Visual Server [155](#)
- dirty
  - modules [204](#)
- DISPLAY
  - MODIFY command [116](#)
- DISPLAY operator command [77](#)
- domain name resolution, TCP/IP [48](#)
- double byte character set
  - support [211](#)
- DPREF [205](#)
- duplicate a user [145](#)

## E

E10:Crypt: Protocol violation (E10) [156](#)  
E18:Crypt: Unexpected message [156](#)  
EARLYWRN [205](#)  
education [xiii](#)  
enable calling CKXLOGID from APF environment [47](#)  
Environment refresh  
    configuration [118](#)  
    REPORT command [127](#)  
error during install  
    FOMF0303I [155](#)  
    FSUM2078 [155](#)  
error during SE.W  
    an error has occurred [157](#)  
    couldn't open session with bluebook adapter [157](#)  
    EDC5139I Operation not permitted [157](#)  
    file system mounted with NOSETUID or NOSECURITY  
    [157](#)  
    ICH13003I [157](#)  
    Invalid password [157](#)  
    logon failed [157](#)  
    Must be numeric [157](#)  
    no READ access to resource [157](#)  
    resource is not covered by a RACF profile. [157](#)  
    the agent has not been added with A or AP [157](#)  
    The password has expired [157](#)  
    Unknown userid [157](#)  
    Userid is revoked [157](#)  
errors  
    errno=6F 5B400002 [157](#)  
    errno=81 53B006C [157](#)  
    errno=81 594003D [157](#)  
    Must be numeric [157](#)  
    starting the Visual Server twice [152](#)  
example jobs for Guardium VA [187](#)  
extended monitoring [113](#)

## F

FACILITY  
    BPX.FILEATTR.APF [155](#)  
    BPX.FILEATTR.PROGCTL [155](#)  
    CKG.CMD. [143](#)  
    CKG.RAC. [143](#)  
    CKG.SCHEDULE. [145](#)  
    CKG.SCP [143](#)  
fast installation [9](#)  
file polling [163](#)  
file polling, configuration file [172](#)  
FILTER  
    MODIFY command [116](#)  
FILTER command  
    ADDSMF [129](#)  
    ADDWTO [129](#)  
    DELSMF [129](#)  
    DELWTO [129](#)  
    NOSUBTYPE [129](#)  
    PREFIX [129](#)  
    RECTYPE [129](#)  
    SUBTYPE [129](#)  
FOMF0303I [155](#)  
FORCE  
    START command [115](#)

formal installation [9](#)  
FORMAT  
    SIMULATE command [131](#)  
FSUM2078 [155](#)

## G

general configuration member [172](#)  
Guardium VA [187](#)

## H

hlq.ckrparm data set [222](#)

## I

IBM  
    Software Support [xiii](#)  
    Support Assistant [xiii](#)  
IBM.HCKR250.F1 [11](#)  
ICH408I  
    access to BPX.SERVER [157](#)  
    access to C2R.SERVER.ADMIN [157](#)  
    during Visual Server startup [155](#)  
    insufficient authority to lookup [157](#)  
IEFU83 exit [102](#), [112](#)  
IEFU84 exit [102](#), [112](#)  
IEFU85 exit [102](#), [112](#)  
IFAPRDxx parmlib member [18](#)  
IKJTSOxx [19](#)  
INIT [205](#)  
initialization exit (SIEM) [170](#)  
installation  
    as part of CBPDO [9](#), [10](#)  
    as part of Server Pack [9](#), [10](#)  
    as part of System Pack [9](#), [10](#)  
    C2R [201](#)  
    checklist [1](#)  
    checklist for RACF-Offline [99](#)  
    distribution to other images [17](#)  
    distribution-oriented [5](#), [23](#)  
    fast [9](#)  
    formal [9](#)  
    methods [9](#)  
    multiple source media [10](#)  
    overview [5](#)  
    preparing for [7](#)  
    RACF Offline [99](#)  
    restricted mode [201](#)  
    roadmap [1](#)  
    single [5](#)  
    single media [9](#)  
    verifying [29](#)  
    verifying with tasks [29](#)  
    Visual Server [133](#)  
    z/OS Compliance Integration Manager  
    [89](#)  
installation jobs  
    for customizing installation parameters [11](#)  
    obtaining [11](#)  
    sample JCL [11](#)  
    zSecure-supplied [11](#)  
installation parameters

installation parameters (*continued*)  
    customizing [11](#)  
    updating [12](#)  
installation setup  
    data set naming [7](#)  
    security planning [7](#)  
    user catalog [7](#)  
interface level profiles [143](#)  
INTERVAL  
    configuration [118](#)  
    REPORT command [127](#)  
IO  
    DEBUG command [121](#)  
ISPF  
    checking base functions [29](#)  
    checking menu configuration [29](#)  
    configuring interface [211](#)  
    location of components [14](#)  
    reset language [213](#)  
    SE.D.8 panel [221](#)  
    SE.D.I panel [220](#)  
    select language [213](#)  
ISPF command tables [18](#)  
ISPF interface controls (security setup [193](#), [195](#))  
ISPTCM [19](#)

## J

JCL installation sample [11](#)  
JCLLIB [27](#)  
JES [205](#)  
JES/328X data set mask [220](#)  
jobname information, setting up collection [65](#)

## L

LEEF [161](#)  
LIBDEF [205](#)  
line commands  
    customizing [213](#)  
Line commands  
    restrict [194](#)  
LOCAL option [222](#)  
localhost  
    Not found in SE.W [157](#)  
Log Event Enhanced Format (LEEF) [161](#)

## M

MAIN  
    DEBUG command [121](#)  
MAXMAILBYTES setting [48](#)  
MEMBER  
    REPORT command [127](#)  
Menu options  
    in ISPF [193](#)  
MGMTCLAS [205](#)  
MODIFY command  
    COLLECT [116](#)  
    DEBUG [116](#)  
    DISPLAY [116](#)  
    FILTER [116](#)  
    REFRESH [116](#)

MODIFY command (*continued*)  
    REPORT [116](#)  
    RESTART [116](#)  
    SIPL [116](#)  
    STOP [116](#)  
mount attribute  
    Visual Server's home file system [157](#)  
moving window [119](#)  
Moving window  
    configuration [118](#)  
    REPORT command [127](#)  
multi-system support  
    function [49](#)  
MYACCESS report [143](#)

## N

National Language Support [211](#)  
near real-time SMF feed [161](#)  
near real-time, configuration file [172](#)  
NIST 800-131A cryptography standard [139](#)  
NOBUFFER  
    DEBUG command [121](#)  
NOIO  
    DEBUG command [121](#)  
NOMAIN  
    DEBUG command [121](#)  
NONE  
    DEBUG command [121](#)  
NOSECURITY  
    mount attribute causes problem [157](#)  
NOSETUID  
    mount attribute causes problem [157](#)  
NOSMF  
    DEBUG command [121](#)  
NOSUBTYPE  
    FILTER command [129](#)  
NOWTO  
    DEBUG command [121](#)  
number of buffers  
    configuration [119](#)  
Number of buffers  
    OPTION command [125](#)  
NUMBUFS  
    configuration [119](#)  
    OPTION command [125](#)

## O

operation  
    CKQCEF [175](#)  
    CKQRADAR [175](#)  
operation (Command Logger) [45](#)  
operation, CKQEXSMF [176](#)  
operator commands  
    z/OS Compliance Integration Manager  
    [94](#)  
operator commands, CKQEXSMF [182](#)  
OPTION command  
    BUFSIZE [125](#)  
    COLLECTSTCNAME [125](#)  
    COLLECTTIME [125](#)  
    NUMBUFS [125](#)

OPTION configuration command [80](#)  
OPTION statement  
    for zSecure Server [51](#)  
    syntax [51](#)  
options module, B8ROPT [99](#)

## P

PADS mode  
    installation [201](#)  
PARMLIB [114](#)  
password change profile [146](#)  
policy (Command Logger) [46](#)  
port of entry information, setting up collection [65](#)  
post-installation tasks [17](#)  
PREFIX  
    FILTER command [129](#)  
prepare data for Guardium VA [187](#)  
problem determination for Visual Server [153](#)  
problem-determination [xiii](#)  
Proclib [21](#)  
PROCLIB [27](#)  
production, setup for [31](#)  
PROFDSN [205](#), [211](#)  
PROFDSN data set [221](#)  
Program Control and PADS access for RACF [202](#)  
Program directories [ix](#)  
program directory  
    CARLa-driven components [8](#)  
    RACF-Offline [8](#)  
Protocol violation [156](#)  
publications  
    zSecure Manager for RACF z/VM [ix](#)  
    zSecure Suite [ix](#)

## Q

QRadar  
    log source properties [184](#)  
QRadar SIEM  
    customize data set members [166](#)  
    prerequisites for setup [161](#)  
    SMF records [162](#)  
QRadar-specific parameters [172](#)

## R

RACF exits  
    cleanup [74](#)  
RACF Offline  
    activating [99](#)  
    installing [99](#)  
RACF scoping [145](#)  
RACF VERIFY [65](#)  
RACF-Offline  
    activating [101](#)  
    check enablement [103](#)  
    installation [99](#)  
    minimal testing [102](#)  
    run as TSO command [101](#)  
    SMF exits [102](#)  
    testing [102](#)  
Real (Resolved) Pathname [80](#), [85](#)

Real (Resolved) PatHName [80](#)  
record suppression exit (SIEM) [171](#)  
RECTYPE  
    FILTER command [129](#)  
REFRESH  
    MODIFY command [116](#)  
release, verifying supported [7](#)  
remote data access  
    support for [49](#)  
REPORT  
    MODIFY command [116](#)  
REPORT command  
    AVERAGEINTERVAL [127](#)  
    DDNAME [127](#)  
    INTERVAL [127](#)  
    MEMBER [127](#)  
    STAGE1INTERVAL [127](#)  
    STAGE1MEMBER [127](#)  
REPORT configuration command [86](#)  
REPORT operator command [77](#)  
reporting interval [119](#)  
Reporting interval  
    configuration [118](#)  
    REPORT command [127](#)  
reports  
    functions to display [29](#)  
requirements  
    programming [8](#)  
    space [8](#)  
resources required [31](#)  
RESTART  
    MODIFY command [116](#)  
RESTART operator command [77](#)  
RESTRICT  
    line commands [194](#)  
restricted mode  
    CKR.READALL resource [201](#)  
    Program Control and PADS access for RACF [202](#)  
    source determining evaluation [202](#)  
    specifying usage [201](#)  
roadmap, installation [1](#)  
RPH [80](#), [85](#)  
run as started task [137](#)

## S

sample batch jobs (CEF) [169](#)  
sample jobs for Guardium VA [187](#)  
SB8RLNK library [99](#)  
SB8RSAMP library [99](#)  
SCKRJOBS data set [31](#)  
SCKRPROC data set [21](#)  
SCKRSAMP data set [5](#), [31](#)  
SDE [161](#)  
SE.D [211](#)  
SE.D.8 panel [221](#)  
SE.D.I panel [220](#)  
SE.D.N panels [213](#)  
SE.W  
    trouble shooting [157](#)  
SECURITY  
    mount attribute required [157](#)  
security data, types [34](#)  
security, disabling [58](#)

segment editing [146](#)  
self-connect mode [59](#)  
Server Pack, installing as part of [9](#), [10](#)  
ServerToken keyword [55](#)  
SETUID

mount attribute required [157](#)  
setup  
CKQCEF started task (SIEM) [173](#)  
CKQEXSMF server started task (SIEM) [174](#)  
CKQRADAR started task (SIEM) [173](#)  
file polling (SIEM) [172](#)  
for SIEM, overview [161](#)

Setup Alert panel [131](#)

Setup default [211](#)

setup default data [221](#)

SIEM

custom event exit [171](#)

customize [170](#)

generate SMF records [162](#)

initialization exit [170](#)

make SMF records available [163](#)

record suppression exit [171](#)

setup

CKQCEF started task [173](#)

CKQEXSMF server started task [174](#)

CKQRADAR started task [172](#), [173](#)

overview [161](#)

SMF logstream [163](#)

storage setup for LEEF data [173](#)

SIEM data customization [170](#)

SIMESM [205](#)

SIMULATE command

FORMAT [131](#)

SMF [131](#)

SYSTEM [131](#)

SIPL

MODIFY command [116](#)

SIPL operator command [77](#)

Site module [17](#), [191](#)

site-specific data, configuring [147](#)

site-specific functions, zSecure Visual [147](#)

site-specific script, configuring [151](#)

Siteinfo file [157](#)

SMF

DEBUG command [121](#)

SIMULATE command [131](#)

SMF Collector (CKQEXSMF) [161](#)

SMF exits

clean up [117](#)

zSecure Alert [112](#)

SMF feed collection [161](#)

SMF filter

FILTER command [129](#)

SMF INMEM

make SMF records available [163](#)

SMF records

file polling [163](#)

generate [162](#)

make available [163](#)

near real-time [163](#)

SMP/E RECEIVE [11](#)

SMTP server settings [48](#)

socket error

during Visual Server startup [155](#)

STAGE1INTERVAL

configuration [118](#)

REPORT command [127](#)

STAGE1MEMBER

REPORT command [127](#)

START command

DEBUG [115](#)

FORCE [115](#)

Started task [21](#)

starting the Visual Server [152](#)

STARTTRX [205](#)

STC procedure (Command Logger) [45](#)

STC user ID(Command Logger) [45](#)

STOP

MODIFY command [116](#)

STOP command [114](#)

STOP operator command [78](#)

stopping the Visual Server [153](#)

STORCLAS [205](#)

SUBTYPE

FILTER command [129](#)

Support Lifecycle [105](#)

SYS [205](#)

SYSSTC [64](#), [118](#)

SYSTCPD [48](#)

SYSTEM

SIMULATE command [131](#)

System Data Engine (SDE) [161](#)

System Pack, installing as part of [9](#), [10](#)

system problems for Visual Server [153](#)

system resources [31](#)

## T

TCP/IP domain name resolution [48](#)

TCP/IP Security [138](#)

TCP/IP.DATA [48](#)

TCPIP error [111](#)

during Visual Server startup [155](#)

TCPIP error [112](#)

during Visual Server startup [155](#)

TEMPUNIT [205](#)

TRACE

server option [156](#)

training [xiii](#)

troubleshooting [xiii](#)

TSO Authorized Command [101](#)

TSO command tables [18](#)

TSEXEC

to obtain controlled environment [204](#)

## U

Unexpected message (E18) [156](#)

UNIT [205](#)

UNLOAD

fresh [42](#)

upgrade zSecure Alert [131](#)

UPREFIX [205](#)

user catalog, specifying [8](#)

userid mapping strategies [198](#)

USERID/STARTED profile (Command Logger) [45](#)

USRDATA [205](#)

## V

- verification of target and distribution libraries [17](#)
- virtual storage [35](#)
- virtual storage, capacity planning [37](#)
- Visual client
  - configure site-specific data [147](#)
  - configure site-specific script [151](#)
- Visual Server
  - configuration parameters [135](#)
  - configure site-specific data [147](#)
  - configure site-specific script [151](#)
  - first time startup [138](#)
  - how to start [152](#)
  - installation requirements [133](#)
  - installing [133](#)
  - multiple [134](#)
  - options [156](#)
  - required system authorizations [134](#)
  - response problems [156](#)
  - run as batch job [137](#)
  - Server root [137](#)
  - setup for new [136](#)
  - startup problems [155](#)
  - stopping [153](#)
  - TCP/IP Security [138](#)
  - upgrading [139](#)
- Visual Server configuration
  - canceling a password [142](#)
  - client-server communication [140](#)
  - existing client [140](#)
  - introduction [140](#)
- VOLSER [205](#)

## W

- WLM service class [64](#), [118](#)
- WORKLLQ [205](#)
- WORKPREF [205](#)
- WPREFIX [205](#)
- WTO
  - DEBUG command [121](#)
- WTO filter
  - FILTER command [129](#)

## Z

- Z Security and Compliance Center [89](#)
- z/OS Compliance Integration Manager
  - commands [94](#)
  - configuration [89](#), [92](#)
  - installation [89](#)
  - setting up [89](#)
- zEDC data compression [35](#), [72](#)
- ZSECNODE statement
  - for zSecure Server [51](#)
  - syntax [53](#)
- ZSECSYS statement
  - for zSecure Server [51](#)
  - syntax [54](#)
- zSecure
  - authorizations for remote data access [196](#)

- zSecure (*continued*)
  - authorizations for routing commands [196](#)
  - data presentation controls [193](#)
  - ISPF interface controls [193](#), [195](#)
  - line commands [194](#)
  - presentation option controls [193](#)
  - resource access requirements [199](#)
  - restricted mode [196](#)
  - security setup guidelines [193](#)
  - userid mapping [198](#)
- zSecure Admin
  - capacity planning [36](#)
  - premature termination [157](#)
- zSecure Alert
  - address space [114](#)
  - authorizations [110](#)
  - back out upgrade [132](#)
  - capacity planning [40](#)
  - data sets [111](#)
  - extended monitoring [113](#)
  - migrate data set [131](#)
  - security resources [110](#)
  - Setup Alert panel [132](#)
  - SMF exits [112](#)
  - started task [109](#), [114](#)
  - startup from upgrade [105](#)
- zSecure Audit
  - capacity planning [38](#)
- zSecure Collect
  - checking [29](#)
- zSecure configuration data set
  - creating [23](#), [25](#)
  - customizing [26](#)
  - definition [5](#)
  - maintaining during upgrade [26](#)
  - making available [27](#)
  - purpose [23](#)
- zSecure configuration file
  - CKNSVPRM symbol [49](#)
  - for multi-system support [49](#)
- zSecure Server
  - and AT-TLS [56](#)
  - authorization for userid [50](#)
  - configuration [49](#)
  - configuration statements [51](#)
  - disabling security [58](#)
  - function [49](#)
  - installation [49](#)
  - installed software [49](#)
  - MODIFY command [55](#)
  - operator commands [54](#)
  - security definitions [50](#)
  - self-connect mode [59](#)
  - START command [55](#)
  - STOP command [56](#)
- zSecure Visual
  - C2RZWUSR job [137](#)
  - client definition [146](#)
  - discreet profiles [145](#)
  - installation location [136](#)
  - unpack [136](#)
  - See also* system-wide option access
- zSecure Visual client

- zSecure Visual client (*continued*)
  - authorities [143](#)
  - interface level profiles [143](#)
- zSecure Visual Server
  - diagnostic information [155](#)
  - problem determination [153](#)
  - resources for problem solving [153](#)
  - send diagnostic information to IBM [155](#)
  - setup topics [133](#)
- zSecure Visual, site-specific functions [147](#)
- zSecure-Server
  - configuration
  - members [49](#)
  - OPTION [51](#)
  - ZSECNODE [51](#)
  - ZSECSYS [51](#)









Part Number:

(1P) P/N: