Security zSecure CARLa-Driven Components
Version 2.1.0

Installation and Deployment Guide

SC27-5638-00

Security zSecure CARLa-Driven Components
Version 2.1.0

Installation and Deployment Guide

SC27-5638-00

Note
Before using this information and the product it supports, read the information in "Notices" on page 239.

September 2013

This edition applies to version 2, release 1, modification 0 of IBM Security zSecure Admin (product number 5655-N16), version 2, release 1, modification 0 of IBM Security zSecure Audit for RACF (product number 5655-N17), version 2, release 1, modification 0 of IBM Security zSecure Audit for ACF2 (product number 5655-N17), version 2, release 1, modification 0 of IBM Security zSecure Audit for Top Secret (product number 5655-N17), version 2, release 1, modification 0 of IBM Security zSecure Visual (product number 5655-N20), version 2, release 1, modification 0 of IBM Security zSecure Alert (product number 5655-N21), IBM Tivoli Compliance Insight Manager Enabler for z/OS (product number 5655-N22), and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

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™ and Tivoli® Security Information and Event Manager 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
- Tivoli Compliance Insight Manager Enabler for z/OS®

Intended audience

This 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.

What this publication contains

This 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.

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

Access to publications and terminology

This section provides:

- A list of publications in the "IBM Security zSecure library."
- Links to "Online publications" on page x.
- A link to the "IBM Terminology website" on page x.

IBM Security zSecure library

The following documents are available online in the IBM Security zSecure library:

- *IBM Security zSecure Release information*

  For each product release, the release information topics provide information about new features and enhancements, incompatibility warnings, and documentation update information for the IBM Security zSecure products. You can obtain the most current version of the release information at http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.zsecure.doc_2.1/welcome.htm.

- *IBM Security zSecure CARLa-Driven Components Installation and Deployment Guide*, SC27-5638

  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 ACF2
  - IBM Security zSecure Visual for RACF
  - IBM Tivoli Compliance Insight Manager Enabler for z/OS
- *IBM Security zSecure Admin and Audit for RACF Getting Started*, GI13-2324

  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.
- *IBM Security zSecure Admin and Audit for RACF User Reference Manual*, LC27-5639

  Describes the product features for IBM Security zSecure Admin and IBM Security zSecure Audit. Includes user instructions to run the features from ISPF panels, RACF administration and audit user documentation with both general and advanced user reference material for the CARLa command language and the SELECT/LIST fields. This manual also provides troubleshooting resources and instructions for installing the zSecure Collect for z/OS component. This publication is only available to licensed users.
- *IBM Security zSecure Audit for ACF2 Getting Started*, GI13-2325

  Describes the IBM Security zSecure Audit for ACF2 product features and provides user instructions for performing standard tasks and procedures such as analyzing Logon IDs, Rules, and Global System Options, and running reports. The manual also includes a list of common terms for those not familiar with ACF2 terminology.
- *IBM Security zSecure Audit for ACF2 User Reference Manual*, LC27-5640

  Explains how to use IBM Security zSecure Audit for ACF2 for mainframe security and monitoring. For new users, the guide provides an overview and conceptual information about using ACF2 and accessing functionality from the ISPF panels. For advanced users, the manual provides detailed reference information including message and return code lists, troubleshooting tips, information about using zSecure Collect for z/OS, and details about user interface setup. This publication is only available to licensed users.
- *IBM Security zSecure Audit for Top Secret User Reference Manual*, LC27-5641

  Describes the IBM Security zSecure Audit for Top Secret product features and provides user instructions for performing standard tasks and procedures.
- *IBM Security zSecure Alert User Reference Manual*, SC27-5642

  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.
- *IBM Security zSecure Command Verifier User Guide*, SC27-5648

  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.
- *IBM Security zSecure CICS Toolkit User Guide*, SC27-5649

Explains how to install and use IBM Security zSecure CICS® Toolkit to provide RACF administration capabilities from the CICS environment.

- *IBM Security zSecure Messages Guide*, SC27-5643

  Provides a message reference for all IBM Security zSecure 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.

- *IBM Security zSecure Quick Reference*, SC27-5646

  This booklet summarizes the commands and parameters for the following IBM Security zSecure Suite components: Admin, Audit, Alert, Collect, and Command Verifier. Obsolete commands are omitted.

- *IBM Security zSecure Visual Client Manual*, SC27-5647

  Explains how to set up and use the IBM Security zSecure Visual Client to perform RACF administrative tasks from the Windows-based GUI.

- *IBM Security zSecure Documentation CD*, LCD7-5373

  Supplies the IBM Security zSecure documentation, which contains the licensed and unlicensed product documentation. The *IBM Security zSecure: Documentation CD* is only available to licensed users.

- *Program Directory: IBM Security zSecure CARLa-Driven Components*, GI13-2277

  This program directory is intended for the system 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 Tivoli Compliance Insight Manager Enabler for z/OS. Program directories are provided with the product tapes. You can also download the latest copy from the IBM Security zSecure documentation website at http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.zsecure.doc_2.1/welcome.html.

- *Program Directory: IBM Security zSecure CICS Toolkit*, GI13-2282

  This program directory is intended for the system 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 directories are provided with the product tapes. You can also download the latest copy from the IBM Security zSecure documentation website at http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.zsecure.doc_2.1/welcome.html.

- *Program Directory: IBM Security zSecure Command Verifier*, GI13-2284

  This program directory is intended for the system 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 directories are provided with the product tapes. You can also download the latest copy from the IBM Security zSecure documentation website at http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.zsecure.doc_2.1/welcome.html.

- *Program Directory: IBM Security zSecure Admin RACF-Offline*, GI13-2278

  This program directory is intended for the system 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 are provided with the product tapes. You can also download

the latest copy from the IBM Security zSecure documentation website at http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.zsecure.doc_2.1/welcome.html.

Online publications

IBM posts product publications when the product is released and when the publications are updated at the following locations:

IBM Security zSecure library
The product documentation site (http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.zsecure.doc_2.1/welcome.html) displays the welcome page and navigation for the library.

IBM Security Systems Documentation Central
IBM Security Systems Documentation Central provides an alphabetical list of all IBM Security Systems product libraries and links to the online documentation for specific versions of each product.

IBM Publications Center
The IBM Publications Center site (http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss) offers customized search functions to help you find all the IBM publications you need.

IBM Terminology website

The IBM Terminology website consolidates terminology for product libraries in one location. You can access the Terminology website at http://www.ibm.com/software/globalization/terminology.

Related documentation

If you are using IBM Security zSecure products in a RACF environment, you can find RACF user and reference information in several IBM manuals. The RACF commands and the implications of the various keywords can be found in the *RACF Command Language Reference* and the *RACF Security Administrator's Guide*. Information about writing other RACF exits can be found in the *RACF System Programmer's Guide*. Information about auditing RACF can be found in the *RACF Auditor's Guide*. You can access this documentation from the z/OS internet library available at http://www.ibm.com/systems/z/os/zos/bkserv/.

For information about incompatibilities, see the **Incompatibility** section under **Release Information** on the IBM Security zSecure documentation website at http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.zsecure.doc_2.1/welcome.html.

Table 1. Further information about RACF administration, auditing, programming, and commands

Manual	Order Number
z/OS V1 Security Server RACF Command Language Reference	SA22-7687
z/OS V1 Security Server RACF System Administrator's Guide	SA22-7683
z/OS V1 Security Server RACF Auditor's Guide	SA22-7684
z/OS V1 Security Server RACF System Programmer's Guide	SA22-7681
z/OS MVS™ System Commands	SA22-7627

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 following IBM Education website at http://www.ibm.com/software/tivoli/education.

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 http://www.ibm.com/software/support/probsub.html.

Statement of Good Security Practices

IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. 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 installation road map provides an overview of the steps to install, configure, and deploy a new installation of IBM Security zSecure.

About this task

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 6.
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 27.
   e. Make the zSecure configuration data sets available and establish security for each configuration:
      1) Assign the configurations to the appropriate TSO/ISPF users.

         See “Assignment of configurations to TSO/ISPF users” on page 27.

      2) Assign the configurations to the appropriate batch jobs and started tasks.

         See “Assignment of configurations to batch jobs and started tasks” on page 28.

      3) Establish security for each configuration to control access to product functions and data.

         See Appendix B, “Security setup for zSecure,” on page 201.

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, 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 42.
d. Refresh the CKFREEZE file.
   See “Use of a fresh CKFREEZE and UNLOAD each day” on page 42.
e. Set up the RACF Exit Activator.
   See “Setup of the RACF Exit Activator” on page 43.
f. Set up your own version of the New Password exit.
   See “Use of the zSecure New Password Exit with other New Password exits” on page 44.
g. Ensure that TCP/IP domain names can be resolved.
   See “TCP/IP domain name resolution” on page 44.
h. Check the settings for the SMTP server.
   See “SMTP server considerations” on page 44.

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 45.
   a. Install, configure, and activate the zSecure Server.
      See “Platform support for the zSecure Server” on page 45.
   b. Specify the remote data sets for use in CKRCARLA or the ISPF User Interface.
      See “Operator commands for the zSecure Server” on page 51.
   c. Perform setup for routing RACF and selected non-RACF commands to other systems.
      See “Setup for secure communication using AT-TLS” on page 53.
9. Set up components such as:
   - zSecure Admin Access Monitor.
     See Chapter 10, “Setup of zSecure Admin Access Monitor,” on page 59.
   - RACF-Offline.
     See Chapter 11, “Setup of RACF-Offline,” on page 77.
   - zSecure Alert.
     See Chapter 12, “Setup of zSecure Alert,” on page 83.
   - zSecure Visual Server.
     See Chapter 13, “Setup and use of the zSecure Visual Server,” on page 113.
   - Change Tracking.
     See Chapter 14, “Setup of Change Tracking,” on page 145.
   - Data preparation for QRadar® SIEM.
     See Chapter 15, “Data preparation for QRadar SIEM,” on page 151.
   - A z/OS Agent for Tivoli Security Information and Event Manager.
     See Chapter 16, “Setup of Tivoli Compliance Insight Manager Enabler for Tivoli Security Information and Event Manager,” on page 159.

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."

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 6.

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

The CKRINST library is a working data set that contains zSecure-supplied sample jobs to help you install and set up the software. 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

zSecure uses configuration data sets to support distribution-oriented installation where you install the software once and then deploy it on multiple systems. These 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

Complete the following tasks before you begin installing zSecure:

- "Verification of the release"
- "Naming and securing the zSecure data sets"
- "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 http://www.ibm.com/software/support/lifecycle/index_a_z.html#T. Also, review the release information posted in the zSecure information center for information about the latest product updates and any incompatibility warnings. See http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.zsecure.doc_2.1/welcome.htm.

Naming and securing the zSecure data sets

Before you begin the zSecure installation process, be sure to do the following tasks:

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.CKRvrm`. In this way, you can create data sets such as DEADSYS.CKR.SCKRLOAD or `DEADSYS.CKR.CKRvrm.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 12.

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.

Specification of a user catalog for the zSecure data sets (optional)

For easy access to the zSecure software from multiple z/OS images, specify a user catalog for the zSecure data sets. You can use an existing catalog or create a new one. In this way, you can easily access the software from multiple z/OS images. In addition, you can easily connect the user catalog to the master catalog when you must replace your master catalog during a z/OS software upgrade. The TSO command to appoint a user catalog is:

```
DEFINE ALIAS (NAME('your-high-level-qualifier') REL('your-appointed-usercatalog'))
```

Without an alias definition, 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 in the IBM Security zSecure information center. See http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.zsecure.doc_2.1/welcome.htm.

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 the Security zSecure Admin RACF-Offline function because the components for RACF-Offline must be installed into an SMP/E zone that already contains z/OS. Also, when you want to use the RACF-Offline component, you must install all CARLa-driven components in the same zone where z/OS is installed, and you cannot distribute among unlike z/OS levels. If you intend to use the RACF-Offline function, use the formal installation method. For information about installing RACF-Offline, see Chapter 11, "Setup of RACF-Offline," on page 77.

- **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."
- 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 10
- "Installing from multiple installation media" on page 10

Installing from a single installation media

About this task

The following procedure describes the general installation process, based on a single installation media (tape or download) containing all zSecure products that you have ordered.

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 12.
3. If you are installing from DASD, adjust the SMPPTFIN DD statement in the RECEIVE step as described in the *Supplemental Installation Instructions for Performing an SMP/E Installation from DASD* document, which you can obtain from IBM Software Support.
4. Run the CKRZINST job found in the CKRINST installation library created in step 1.

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 12.
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.

   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 12.

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.HCKR210.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 from http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.zsecure.doc_2.1/samples.html.

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

```
//STEP1 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//* //TAPEIN DD DSN=IBM.HCKR210.F1,UNIT=tunit,
//* //          VOL=SER=volser,LABEL=(x,SL),
//* //          DISP=(OLD,KEEP)
//* //FILEIN DD DSN=IBM.HCKR210.F1,UNIT=SYSALLDA,DISP=SHR
//OUT    DD DSNAME=your-prefix.CKRINST,
//          DISP=(NEW,CATLG,DELETE),
//          VOL=SER=dasdvol,UNIT=SYSALLDA,
//          SPACE=(TRK,(30,5,15))
//SYSUT3 DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN DD *
  COPY INDD=xxxxIN,OUTDD=OUT
```

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

- Uncomment either the `//TAPEIN` or the `//FILEIN DD` statement, depending on your distribution medium, and delete the other statement. Then, in the `SYSIN DD` statement, change the value of the `INDD` parameter from `xxxxIN` to `TAPEIN` or `FILEIN`, depending on which statement you specified.
- Add a job card and change the lowercase parameters to uppercase values to meet your requirements before submitting.
- For the `OUT DSNAME`, specify the high-level qualifier for the output data set name. The default low-level qualifier is `CKRINST`. Keeping the default qualifier is a good idea because it is the name used to refer to this data set throughout the zSecure documentation.

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 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 and using the Change Tracking function. 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 13
- "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 forzSecure 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 ****************************************
TapeUnit                          = 3480
PrefixForTargetLibraries          = CKR
VolumeForTargetLibraries          =
PrefixForDistributionLibraries    = CKR.DLIB
VolumeForDistributionLibraries    =
JclLib                            = Yes
SmpeTargetZone                    = CKR210T
SmpeDistributionZone              = CKR210D
PrefixForSmpeGlobalZone           = CKR.SMPE.G
PrefixForSmpeOtherData            = CKR.SMPE
SmpeCsiAndSmptlibVolume           = SMS001
//******* Products/features to install *********************************
AdminRACF                         = No
AuditRACF                         = No
AuditACF2                         = No
AuditTopSecret                    = No
AlertRACF                         = No
AlertACF2                         = No
VisualRACF                        = No
ComplianceInsightManagerRACF      = No
ComplianceInsightManagerACF2      = No
ComplianceInsightManagerTopSecret = No
ComplianceInsightManagerDB2       = No
ComplianceInsightManagerCICS      = 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 esoteric or generic unit name for tape units.

   If you are installing from DASD, leave this parameter as is. Instead, adapt job CKRZINST as described in “zSecure-supplied installation jobs” on page 11.

   ```
   TapeUnit                          = 3480
   ```

3. Specify the high-level qualifiers for the data sets where 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 embedded 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                    = CKR210T
   SmpeDistributionZone              = CKR210D
   ```

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.

    ```
    SmpeCsiAndSmptlibVolume           = 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
    ComplianceInsightManagerRACF      = No
    ComplianceInsightManagerACF2      = No
    ComplianceInsightManagerTopSecret = No
    ComplianceInsightManagerDB2       = No
    ComplianceInsightManagerCICS       =  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.ISPMENU
ISPSLIB ISP.ISPSENU
ISPPLIB ISP.ISPPENU
ISPTLIB ISP.ISPTENU
```

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 `ISPPENU` 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:
   - C2R8*xxxx* messages. These messages are described in the *IBM Security 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 ISPPLIB 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 'ISPMLIB' 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"
- "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 199. 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 run zSecure, update the parmlib member IFAPRD*xx* to enable the appropriate license features.

Procedure

1. Copy the required PRODUCT statements for zSecure enablement from SCKRSAMP library member CKRZPROD.
2. Paste the statements in the IFAPRD*xx* 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.

APF authorization of the software

For most purposes, the program object library containing zSecure must be APF authorized. APF authorization affectszSecure 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, Access Monitor for zSecure Admin, and the IBM Tivoli Compliance Insight Manager Enabler for z/OS
  - 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.

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

Updating the TSO command table
About this task

To run the zSecure Admin CKGRACF program from TSO, use the following procedure:

Procedure

1. Add the CKGRACF program name to the authorized AUTHCMD and AUTHPGM tables in the SYS1.PARMLIB member IKJTSO*xx*. 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 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 IKJTSO*xx* 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 (SC28-1872).

Updating the ISPF TSO Command Table
About this task

If CKGRACF is run from ISPF, its use is logged in the SPFLOG*x*.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
*
      ISPMTCM 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
       ISPMTCM  FLAG=32,ENTNAME=CKGRACF  TSO COMMAND, AUTH, NOLOG
* END CARD. STATEMENTS AFTER THIS CARD WILL BE IGNORED
       ISPMTCM  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 (SC34-4257).

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 223.

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.
- Embed the SCKRPROC data set in your system procilib concatenation.
- Copy the procedures to your system procilib 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 28.

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:
    - C2RI0CMD
    - C2RI0IOC
    - C2RI0SMF
    - C2RI0UNL
    - 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.
- The Tivoli Compliance Insight Manager Enabler for z/OS for Tivoli Security Information and Event Manager is usually operated as a started task, although it is not required. If you use this component, copy procedures C2EAUDIT and C2ECSTOP. If you use an event source with a LIVE or INTERCEPT strategy, and your SMF offload runs as a started task, you must also copy procedure C2ECLSMF. This procedure supports regular SMF offload processing. See "Event Source and User Information Source properties" on page 186.
- 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 28.

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 select which products and features are 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

To support distribution-oriented installation where you install the zSecure software once and then deploy it on multiple systems, zSecure uses 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 lists the configuration data sets that can be customized for each deployment.

Table 3. zSecure configuration data sets

Data set name	Description
your.prefix.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.
your.prefix.CKRPARM	This data set contains the main configuration member C2R$PARM. You can create other configuration members as well. The CKRPARM data set also contains the REXX CKR (adapted to your naming convention). Copy and adapt this member to a SYSPROC or SYSEXEC data set. See "Making the software available to TSO/ISPF users" on page 20.
your.prefix.CKRPROF	ISPF tables. If you intend to use the ISPF transactions under SE.D, specify this data set or a copy of it as the PROFDSN parameter in the C2R$PARM member. See "Setup of default options for user groups (Setup menu)" on page 223).
your.prefix.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.

The CKACUST data set is created and filled with job CKAZCUST available in the the CKRINST library or the SCKRSAMP library.

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.IPO1.CKRPROF'
//  SET SYS=IPO1
```

See Appendix D, "Configuration parameters," on page 213 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." If you are upgrading to a new version or installing zSecure again, see "Maintenance of existing zSecure configuration data sets" on page 27.

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"
- "Customization of zSecure configuration data sets" on page 26
- "Maintenance of existing zSecure configuration data sets" on page 27
- "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 12.
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 embed 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."

7. Optional: Only for zSecure Audit users that use option AU.R - Rule-based compliance evaluation. Update job CKAZCUST following the comments in the JCL and submit job to create the CKACUST library.

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 235. 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 parameter and update the data set name.

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 223.

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.

Incompatibility: Beginning with IBM Tivoli zSecure version 1.8.1, the names of ISPF variables within skeletons `C2RSMUMA`, `C2RSMUMH`, `C2RSMUMP`, `C2RSDFLT`, `C2RSJOB`, and `C2RSJOB3` were changed. Consequently, you cannot run job CKRZPOST for an upgrade installation because the private copies of these skeletons in your C2RPARM data sets (as CKRPARM was called before IBM Tivoli zSecure version 1.8.1) do not contain the new members. To update your copies of the C2PARM data sets, manually inspect the new skeletons (supplied in the SCKRSLIB library) and merge the new variables into your (customized) skeletons.

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"
- "Assignment of configurations to batch jobs and started tasks" on page 28
- Appendix B, "Security setup for zSecure," on page 201

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 *IBM Security 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

Use the following information to verify that your installation is working correctly.

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 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 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 `userid` 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(systerm) 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 C2AJIVP 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 28, 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 in 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.

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: CKFREEZE data sets are used for various purposes. They are created by the CKFCOLL program. 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. To be able 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 *IBM Security zSecure Admin and Audit User Reference Manual*.

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. For zSecure 1.13 and higher, all program and transaction information for all CICS and IMS™ systems is collected. For zSecure 1.13.1 and higher, 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
```

Regular Audit

This CKFREEZE data set has all information that is needed for the 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
```

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
```

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=N,SMS=N,TCPIP=N,MOD=N,CICS=N,IMS=N,NJE=N,S=V=NONE,NOXMEM
```

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:

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

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 SMFPRM*xx* 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.

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.

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." 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.

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: These copies 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 events/second * 60 seconds * 100 bytes = 6 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 CPU SU.

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 (SDC) * 40100 (CPU factor) * 2 (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 (events) * 1 (SU) / 40100 (CPU factor) = 0.0005 seconds.
- The total number of service units reported for your application will be 802000 (base) + 20 (events) *1 (SU) * 10 (SDC) = 804000 service units.

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 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 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: If you are doing a detailed analysis of your complete environment, use a full-size CKFREEZE data set. 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)

TCPIP=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: This 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 events/second * 60 seconds * 1000 bytes = 30 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 (SDC) * 40100 (CPU factor) * 2 (sec) = 802000 service units
- After starting zSecure Alert, the amount of CPU time needed to collect and record the information for these events is 50 (events) * 1 (SU) / 40100 (CPU factor) = 0.002 seconds.
- The total number of service units reported for your application will be 802000 (base) + 50 (events) * 1 (SU) * 10 (SDC) = 807000 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.

For the z/OS Agent for Tivoli Security Information and Event Manager, there are DST considerations. See “Setting z/OS UNIX time zones” on page 162.

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, embed 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. Examples of such processes are the generation of QRadar

LEEF data and the generation of Tivoli Security Information and Event Manager 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 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.

Setup of the RACF Exit Activator

The RACF Exit Activator program, C2XACTV, provides dynamic exit support for some RACF exits. The main purpose of C2XACTV is to install exits required by various zSecure products. For example, the zSecure New Password exit, ICHPWX01, enables SMF audit processing to track all password changes including those passwords changed through logon, signon, or a JOB statement in JCL. If you run z/OS 1.6 or earlier, RACF only logs password changes through the ALTUSER and PASSWORD commands.

In most cases, you do not need to control the exits explicitly using the RACF Exit Activator program:

- For zSecure Alert and the Tivoli Compliance Insight Manager Enabler for z/OS, you can control activation of the New Password Exit using the `C2XEXITS` parameter. See Appendix D, “Configuration parameters,” on page 213. If either of these products activates the New Password Exit, the resulting SMF records are also available to zSecure Audit for RACF.
- If you use the Access Monitor function in zSecure Admin, the relevant exits are activated or deactivated automatically when the Access Monitor is started and stopped.

However, if you are only using zSecure Audit for RACF, and your z/OS release is 1.6 or older, you must start the RACF Exit Activator program to activate the New Password exit. To create the additional SMF records that log all password change events, the exit must be activated once after each IPL.

To start C2XACTV and activate the exit, use the following zSecure supplied C2RCACTV procedure. Edit the data set name and parameters in the procedure to reflect the values used at your site.

```
// JCLLIB ORDER=(MY.CKRPARM,CKR.SCKRPROC)
//  EXEC  C2RCACTV,CONFIG=MYCONFIG,PARM='DYNEXIT ACTIVATE ICHPWX01'
```

With the C2XACTV program, you can control the exits using the commands provided by the program, although it is not usually required. For additional information about the commands, see the C2XACTV program documentation in the *IBM Security zSecure Admin and Audit for RACF: User Reference Manual*.

Note: The RACF Exit Activator program offers complete support for RACF pre-, main-, and post-exits; therefore, if you already have your own RACF exit routines in place, they are retained as subexits. Be sure to verify that the exit routines can coexist.

Use of the zSecure New Password Exit with other New Password exits

To supply your own version of the New Password exit, in addition to the one provided by the RACF Exit Activator, point to the exit from an appropriate PROGxx parmlib member. For example:

```
EXIT ADD EXITNAME(C2X.ICHPWX01) MODNAME(your-module-name)
         DSNAME(your-library) STATE(ACTIVE)
```

To avoid confusion, do not use the name ICHPWX01 for the New Password exit defined for your installation. That name is already used by the zSecure RACF New Password Router itself.

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 the z/OS information center for the z/OS release you are using and see **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. Beginning with zSecure Version 1.12, 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.

Note:

1. If you are using zSecure Version 1.11 or earlier, you can still report on data from multiple systems. However, you cannot obtain the information directly from a remote system. Instead, you must transfer the data sets containing information obtained from each system to a central system. Then, those copied data sets can be used as input to the CKRCARLA program, or as input files in the ISPF User Interface using the `SETUP FILES` function.
2. Support for working with data obtained directly from remote systems and remote command routing using either zSecure services or RACF RRSF services is available beginning with zSecure Version 1.12.

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 "Platform support for the zSecure Server."
- Specification of the remote data sets for use in CKRCARLA or the ISPF User Interface. See "Operator commands for the zSecure Server" on page 51.
- Setup for routing RACF and selected non-RACF commands to other systems. See "Setup for secure communication using AT-TLS" on page 53.

Platform support for the zSecure Server

The zSecure Server runs on z/OS Version 1.9 or later. Running on earlier versions of z/OS (such as z/OS Version 1.8) is not supported, and network transport protection using the Application Transparent Transport Layer Security (AT-TLS) functions is unavailable on earlier z/OS systems.

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 `OWNSYS` and the `SERVERTOKEN`. For more information about the `OPTION` statement see "Configuration file OPTION statement" on page 48.

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."

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 CKNRAC1 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 53 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 two optional statements.

- 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

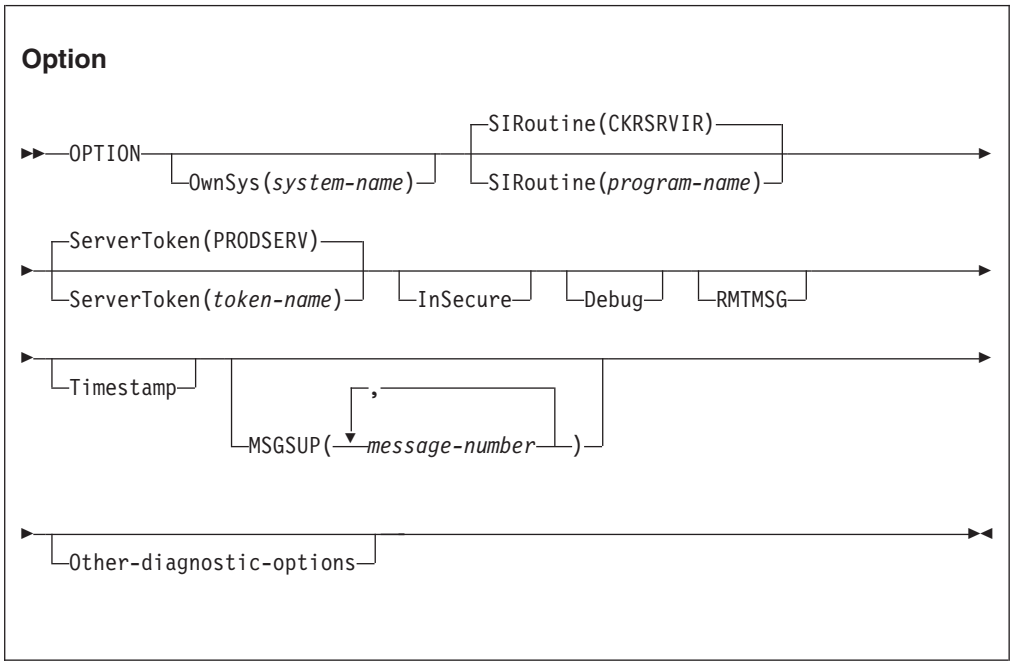
needed to describe your environment. In most cases, users specify (or default) the ZSECNODE in their zSecure UI setup, while some users might use the ZSECSYS to reference specific data sets that are present only on a specific system.

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. The zSecure-Server configuration file does not **need** to include the OPTION statement to specify its own system name. The value is derived automatically during startup, based on the IP address of the active system. In this way, all systems in the network can use the same configuration file. If multiple servers are defined that use the same IP address, use of the OPTION statement with the OWNSYS keyword is preferred. In its absence, one of the systems with a matching IP address (or host name) is chosen. Which system is used is unpredictable.

Configuration file OPTION statement

The OPTION statement specifies server-specific values. The OPTION statement is not required, but it is best to use it if multiple servers are used on the same system. 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.

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.

SIRoutine

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

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.

RMTMSG

This keyword can be used to signal the zSecure Server to include the SYSPRINT and SYSTERM 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.

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.

Timestamp

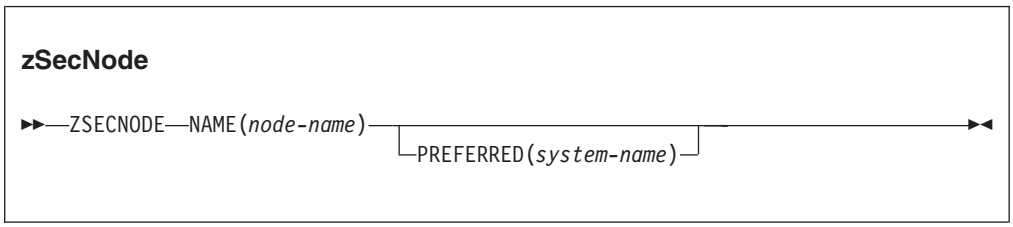
This keyword specifies that messages issued in the server CKNPRINT output file are prefixed with a timestamp. The timestamp information is shown in UTC, and uses a fixed format.

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 STORAGEGC.

Configuration file ZSECNODE statement

The ZSECNODE statement is used to specify the names of the nodes in the zSecure Servernetwork. 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.



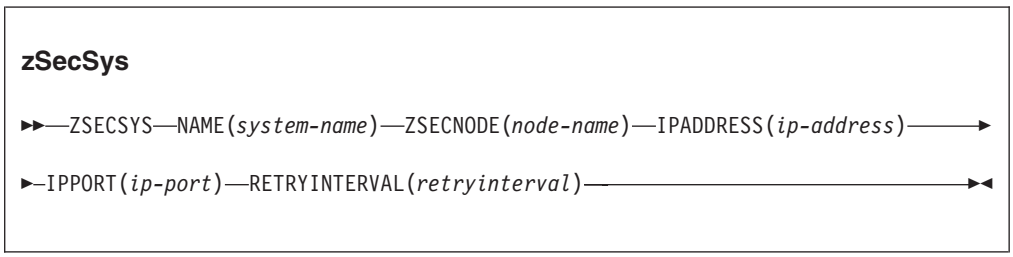
The keywords and parameters have the following meanings:

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 to describes the zSecure systems. This statement defines the systems where zSecure Servers are running. If a preferred system for the ZSECNODE 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

Use operator commands to manage the zSecure Servers. 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

Issue the MODIFY command to request an action by the zSecure Server program. You can use the following actions:

DEBUG
Diagnostic messages are printed in the 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

Issue STOP command to stop the zSecure Server in a controlled way. 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 CKNRAC2 to create a root certificate yourself. If you use job CKNRAC2:

1. Run job CKNRAC2 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 CKNRAC3 to import it into all other RACF databases.
3. After importing, delete the data set. If you do not, 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 CKNRAC4 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 203. 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 PDR2SYS. 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 4. Security-related settings

Area	Subarea	Field	Setting	Effects
TTLSRule	TTLSGroupAction	TTLSEnabled	on	Enforces use of certificate
	TTLSKeyringParms	Keyring	value	Specifies the name of the keyring
	TTLSEnvironmentAdvancedParms	ClientAuthType	SAFCHECK	Specifies that certificate must match a RACF user
	TTLSConnectionAdvancedParms	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
	CKNDAMIN.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 210. 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 (as in multi-system mode). Actual reading of the security database (or any other data) is done by the server address space, not by the original user.

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, follow 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.
- 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 60
- "Operation of the Access Monitor" on page 65
- "Access Monitor function command reference" on page 69

Considerations when upgrading from a previous release of Access Monitor

If you are upgrading from a previous release of Access Monitor, 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 Admin Access Monitor started task (C2PACMON) since the last IPL of the system, no specific upgrade steps are required.

- 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.1.13.0>
  //C2XPRINT DD SYSOUT=*
  //C2XIN    DD *
   DYNEXIT RECOVER ICHRCX02
   DYNEXIT RECOVER ICHRDX02
   DYNEXIT RECOVER ICHRFX04
  ```

  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 can immediately start C2PACMON using the previous release software.
- 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.

Installation and post-installation requirements

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 12.
- The `AdminRacf` license must be enabled. See "Enablement 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.
  - Do not use a zSecure configuration that was created with zSecure 1.10 or older. The 1.10 configuration does not have all the members and parameters required for Access Monitor.
- To use the Access Monitor on multiple z/OS images, use a separate CKRPARM data set for each image. 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 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.

Configuration of Access Monitor

To start the Access Monitor and begin data collection and consolidation, perform the configuration steps described in the following sections:

- 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.ICHRFX04, and C2X.ICHRIX02. 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.

You can use Job C2PZRIN0 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.

Required Access Monitor data sets

The data sets used for the daily collection and the daily consolidation process 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.

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.

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

This section describes the parameter members required to manage data collection and consolidation of the Access Monitor records. 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.

For more information about configuration parameters, see "Configuration commands" on page 71.

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. An example is 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 line highlights that the second column must start in position 10 of the record.

  ```
  C2PAMJOB
     ----+----1----+----2
     IBMUSER  YES
     C2PSUSER NO
  ```

  The first column contains a USERID for which jobname information is controlled. The second column can contain the value `YES` or any other value. Jobname information is collected only for those users for which the value `YES` has been specified. For users that are not included in the C2PAMJOB member, or that have any value other than `YES` specified, jobname information is not collected. Be sure that all information in this member is specified in uppercase.

- 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 70.

Definition of data collection and data consolidation files

Several times during the day, the collected Access Monitor records are saved to disk at each SMF interval. The default SMF interval for the INTVAL parameter in the SMFPRMxx member in PARMLIB is 30 minutes. The collected data is stored in a data set specified and allocated during the configuration process.

Once a day, the collected Access Monitor files are consolidated. 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. This data consolidation is automatically done every night at the specified consolidatetime. The collected data is stored in a data set specified and allocated during the configuration process.

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(584) 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(584) 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 names in member C2PAMCLT must end in `D&LYR2&LMON&LDAY.T&LHR&LMIN`. This results in a timestamp formatted as `Dyymmdd.Thhmm`.
- The specification for the data set names in member C2PAMCNT must end in `D&LYR2&LMON&LDAY`. 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 or MGMTCLAS, you can use them here.
- 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.

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:

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 71.

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."

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 70.

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 71.

Configuration of the Access Monitor function using parmlib

You can control the Access Monitor function by setting parameter values in the parmlib. By default, the 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 62.

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 62. 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.

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   ICHRFX04
DYNEXIT DEACTIVATE   ICHRDX02
```

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.ICHRCX02 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.ICHRCX02 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 73. 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

Using the Access Monitor OPTION statement, you can specify the method used by the exit router module to invoke the functional sub exits. The Access Monitor OPTION statement is used during initialization of the started task. If you want to switch the sub exit calling method, you can use the C2XACTV program. The DEACTIVATE function removes the exits from storage. Using the CSVDYNEX or DIRECT 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 DIRECT mode:

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

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

Access Monitor function command reference

Operators can manage the RACF Access Monitor function from the operator console or by specifying configuration commands in the Access Monitor started task parameter file C2PAMP.

- For information about operator commands, see "Operator commands" on page 70.
- For information about configuration commands, see "Configuration commands" on page 71.

Operator commands

You can specify the following Access Monitor operator commands in the MODIFY console command. When entered as an operator command, these commands do not require additional keywords. You can also abbreviate the command to the first four characters; for example, type `CONS` for the CONSOLIDATE command.

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 71.

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 74.

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.

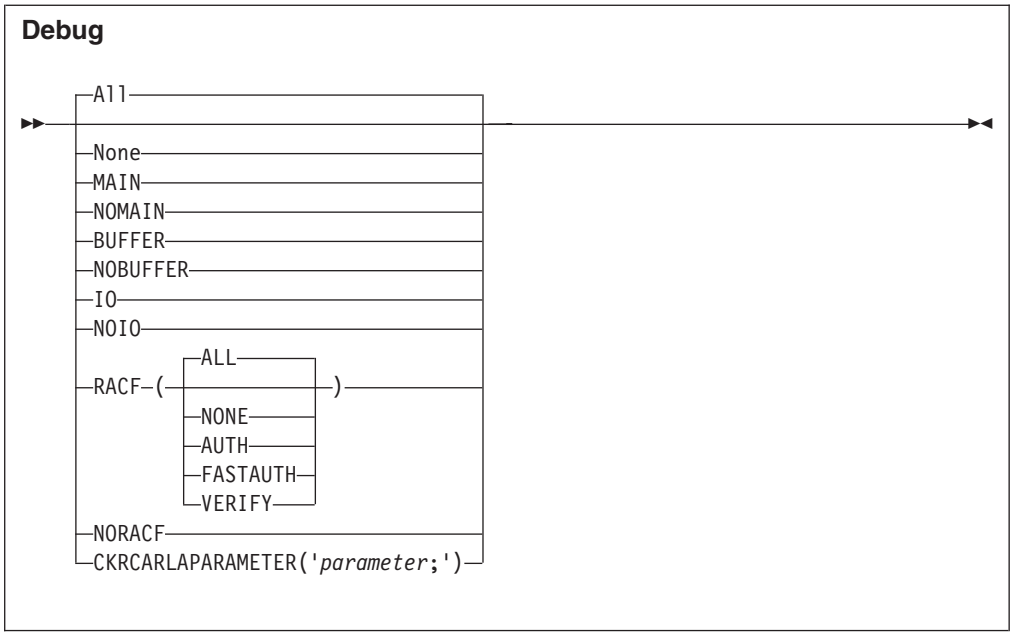
Configuration commands

Use the configuration commands described in this section to control Access Monitor operation with the Access Monitor started task parameter file C2PAMP. Normally, only the OPTION and REPORT commands are required. If you want to obtain diagnostic information, use the DEBUG command.

DEBUG command

Use the DEBUG command to specify diagnostic options for the Access Monitor function. 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.

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.

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.

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.

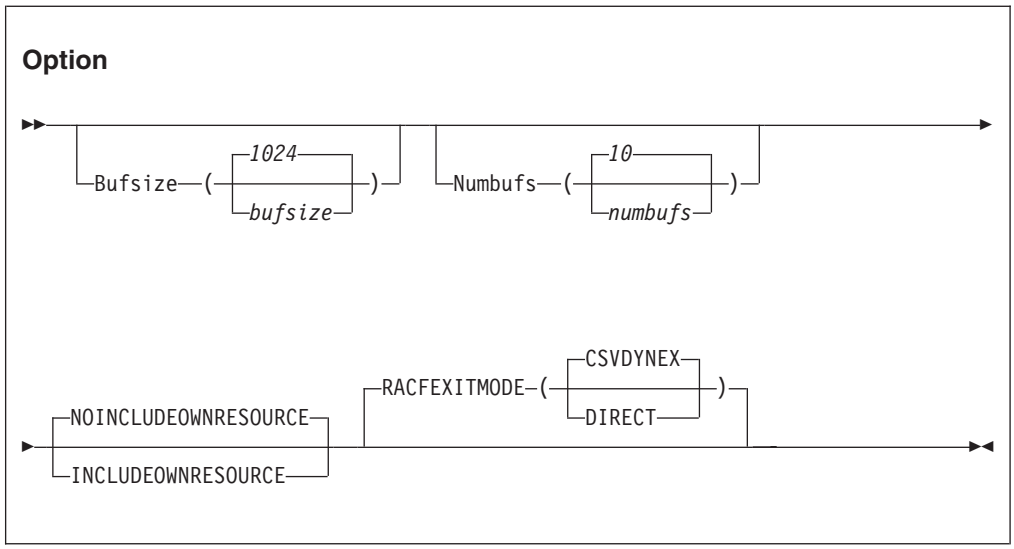
NORACF
Messages for events related to the RACF data collection process are not issued. This function is intended to help IBM Software Support personnel diagnose internal problems in the product.

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

Use the OPTION statement to specify startup options for the Access Monitor started task. The OPTION statement is only valid from the parmlib. When the parmlib is processed due to CONSOLIDATE processing, the OPTION statement is ignored. 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:



The keywords and variables have the following values:

Bufsize
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, the buffer containing the oldest data is used instead.

The value *bufsize* specifies the size of the buffers in kilobytes. Valid values for *bufsize* are between 1 and 16384, resulting in buffer sizes of 1 KB and 16 MB respectively. Using multiple buffers during periods of high activity significantly reduces the required *bufsize*.

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.

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 at the request of IBM Software Support personnel to troubleshoot your system. The default for this option is NOINCLUDEOWNRESOURCE.

RACFEXITMODE
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:

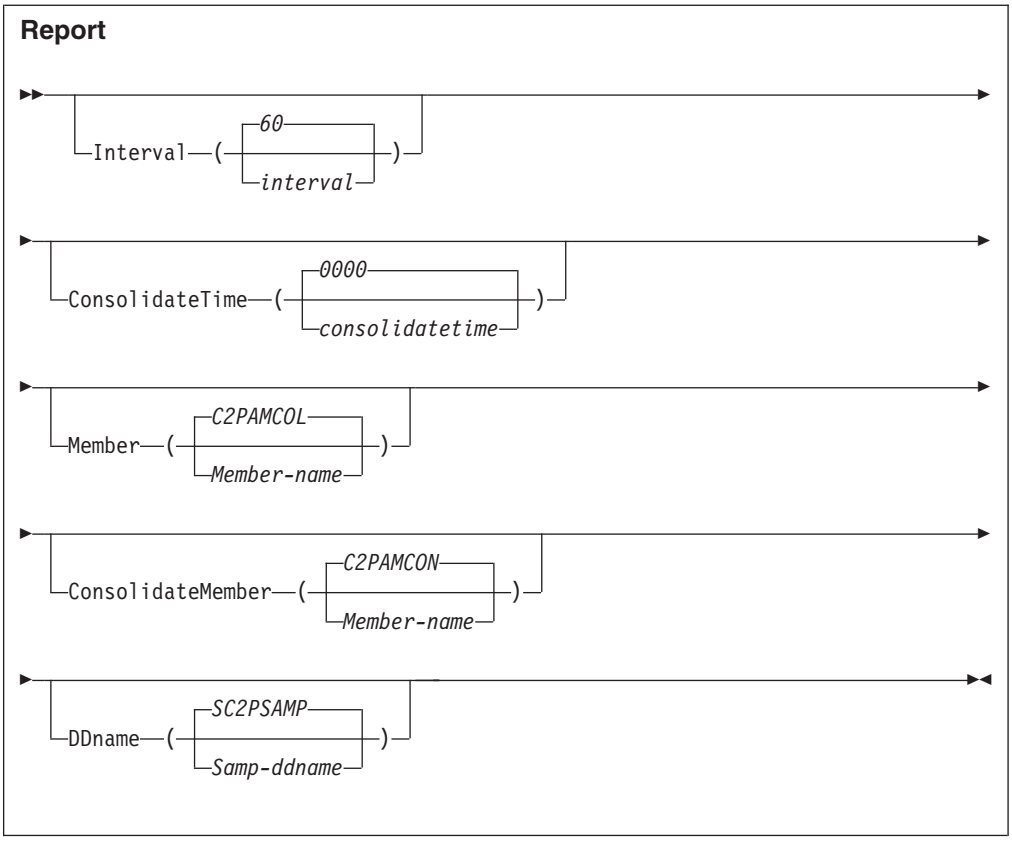
CSVDYNEX
This keyword indicates that the RACF exit router module uses z/OS dynamic exit services for calling the functional sub exits. This option provides additional flexibility and recovery for the called sub exits.

DIRECT
This keyword indicates that the RACF exit router module uses a direct branch instruction to call the functional sub exits. This option provides a fast path to the called sub exits.

If the **RACFEXITMODE** keyword is not specified, or if no value is specified, RACF exits are called using z/OS dynamic exit services.

REPORT command

Use the REPORT command to specify data processing options for the Access Monitor function. 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, you do not need to change the default value.

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*.

Chapter 11. 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*. RACF-Offline must be installed in the SMP/E zones used for the base z/OS products such as RACF. 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."

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

Use the following checklist to track the tasks completed during the RACF-Offline installation and activation process. To perform each task, see the procedure shown in the table.

Table 5. Installation checklist

Step	Description	Job name	Status
1	Install RACF-Offline. See the *Program Directory: IBM Security zSecure Admin RACF-Offline*.		
2	Build the default Options module (B8ROPT)	B8RJOPT	
3	Update Parmlib member for APF library	B8RSPROG	
4	Update Parmlib member for TSO Authorized Commands (Optional)	B8RSIKJ	
5	Verify Parmlib member for SMF Exits		
6	Define Minimal RACF authorizations for testing	B8RJRDF	
7	Test RACF Offline	B8RJTST	
8	"Check for RACF-Offline enablement" on page 82		

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 ID to identify the SMF records for those commands that are executed against an offline database.

      ```
      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:

1) 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.
2) 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 best results, use different SMF-ID as shown in the example B8ROPT source provided with the product.

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 a member for the program library to the PARMLIB member.
      ```
      APF ADD DSNAME(SYS1.SB8RLNK)                SMS
      ```
   b. Run the T PROG=*xx* operator command to update the 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 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. Run the T PROG=*xx* operator command to update the 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 the B8RACF command to the list of APF authorized commands that can be executed from TSO.

1. In parmlib member IKJTSO*xx*, add the B8RACF command to the AUTHCMD list.

   ```
   B8RACF              /*  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(*xx*) 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 SMFPRM*xx* 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

You can prepare for RACF-Offline testing by defining a limited set of authorization profiles rather than defining detailed profiles. At this 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 B8RJRDF 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
//SYSPRINT  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 IFAPRD*xx* in PARMLIB.

- If RACF-Offline is enabled, or not defined in IFAPRD*xx*, 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 IFAPRD*xx* member.

```
OWNER('IBM CORP')
        NAME('zSecure Admin')
        ID(5655-T01)
        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 12. 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

Before installing, verify that the product and release are current and supported on the platform that you intend to use. See http://www.ibm.com/software/support/lifecycle/index_a_z.html#T for the IBM Software Support Lifecycle.

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, seeChapter 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=
   //* Parameters used for zSecure Alert and
   //* Compliance Insight Manager Enabler
   //* SET C2XEXITS=ACTIV
   ```

   As shipped, these parameters are commented out. You must uncomment them and supply your own choices for the zSecure Alert configuration data set, for the name of the address space, and for the exit.

zSecure Alert address space overview

zSecure Alert runs as a started task. To collect all information for alert generation, it dynamically defines SMF exits and dynamically installs an ICHPWX01 RACF exit on RACF systems. 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*x* data sets. The records are passed unmodified to possible other SMF exits and subsequent SMF processing. Only SMF records specified in the active SMFPRM*xx* parmlib member can be captured by SMF exits.

On RACF systems, most of the password changes made by users are not recorded in SMF. RACF did not define any flag or indicator in the SMF records that a user changed their password during LOGON, SIGNON, or through the JCL of a batch job. They are all referred to as LOGON from now on. To overcome this issue, zSecure Alert activates a RACF new password exit (`ICHPWX01`). This new password exit creates an SMF record like what is created by the PASSWORD command, for every successful password change made during LOGON. To enable use of this exit, zSecure Alert dynamically installs and activates the included RACF new password exit. This new password exit does not overlay an existing RACF new password exit, but retains the existing routine as a subexit.

The EMCS console captures all WTO messages directed to the hardcopy set, typically the same as the SYSLOG. It does not likewise 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 called *stage 1*, which prepares the queries included in the actual reporting step.

During each reporting interval, the data is passed to the zSecure Audit for analysis and report generation. Reporting can be done through e-mails, text messages to pagers or cell phones, WTOs, which could for instance be captured by an automated operations package, or SNMP traps, which could in turn be captured by Tivoli Compliance Insight Manager or Tivoli Security Information and Event Manager or a network console like Netview or Tivoli Enterprise Console. Note that when commands are directed by RRSF or CPF, SMF records — and consequently, alerts — are generated on the sending and on the receiving systems.

At each reporting interval, zSecure Alert can also create a small snapshot CKFREEZE data set. This snapshot data set is used to detect any changes in selected system or security settings. If a system or security setting has changed, an alert can be issued. These types of alerts, which are based on the detection of changes to the system and security settings, are called Extended Monitoring alerts. See the information about deciding which alerts to activate in the *IBM Security zSecure Alert: User Reference Manual* for more explanation.

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 Admin and Audit for RACF: User Reference Manual* for details on the full power of 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 only possible if these EXITS have been enabled by the SMF parameter member in PARMLIB. You must ensure that the exit routines IEFU83, IEFU84 and IEFU85 have been enabled for SYS and for all subsystems. For more information about specifying EXITS in SMFPRM, see the relevant chapter in the *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 started task by a `RESTART` command. See "zSecure Alert operator commands" on page 94 for details about the `RESTART` command.

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 zSecure Alert captures all WTO messages by default, 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 92. Other commands are documented in "Other commands" on page 100. 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 *IBM Security zSecure Alert: User Reference Manual*.
- 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

This section describes how to create security resources and add permissions for zSecure Alert. You must set these 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.
- Started task names and security resources to assign the required userids and group or logonid to these started tasks.
- On RACF systems, the userid for C2POLICE must have UPDATE access to the XFACILIT resource C2X.ICHPWX01. 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 199.
- Started task output protection.
- Data set names and profiles covering them. This might include PROGRAM profiles when PADS access is being used.

For the users who configure zSecure Alert, the following security requirements apply:

- READ access to the data sets where the zSecure Admin and Audit software resides.
- Access to the zSecure Admin and Audit-specific resources that determine which options and actions are available. See Appendix B, “Security setup for zSecure,” on page 201.
- 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 READ access to the following resources:

  ```
  OPERCMDS: MVS.MODIFY.STC.C2POLICE.C2POLICE
  OPERCMDS: MVS.MCSOPER.<userid>*  (the actual resource depending 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. Without this access, both of the following items 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 or the zSecure Alert address space is restarted.

  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.
- If you are upgrading from 1.10 or earlier, rerun the ALLOC jobstep from job C2PZAIN1 and comment out all DD-statements except SYSPRCKF and C2PEMFRB.
- 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.

  As of zSecure 1.13, the Alert configuration data set must contain a member called C2PXPARM. You can use either of the following methods to create this member:

  – Run the Setup Alert transactions under ISPF and Verify and Activate your Alert configuration. However, for a shared configuration, do not perform this action until all Alert instances that use the configuration data set run on the new release.
  – Copy member C2PXPARM from the SCKRSAMP library.

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.
- If you are upgrading from zSecure Alert 1.4.5, be aware that some data set names have changed. You can decide on one of the following options:
  - Rename the old data sets.
  - Discard the old data sets and use the new ones.
  - Keep using the old data sets under their old names. In this case you must adapt the JCL and possibly your zSecure Admin and Audit configuration.

  Before making this decision, see "Coexistence considerations" on page 109.

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 SMFPRM*xx*. Also, the SMF dynamic exits are invoked only if the exits are enabled in SMFPRM*xx*. Before continuing, identify the SMF records that are required for your selected alerts, and enable the necessary SMF exits in your SMFPRM*xx*. 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 *IBM Security zSecure Alert: User Reference Manual*. The SMF exits that must be enabled are IEFU83, IEFU84 and IEFU85. Each exit point is used in a specific environment and for specific SMF records. Ensure that these three exit points are enabled for the entire system and for all defined subsystems.

```
SYS(EXIT(IEFU83,IEFU84,IEFU85))
```

Incorrect specifications in SMFPRM*xx* can lead to failure to detect some alert situations.

zSecure Alert provides an option to generate additional SMF records for every password change made during LOGON or Signon, depending on how the application calls it, and during start of a batch job. You can control generation of these additional SMF records through the C2XEXITS parameter of the zSecure configuration. See Appendix D, "Configuration parameters," on page 213.

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('your.prefix.DATA.CKFREEZE.D&LYR2.&Lmon.&Lday..T&LHR.&LMIN.') -
mod 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 Dyymmdd.Thhmm.
   - You can specify any leading qualifiers you want, as long as the data set name after symbol substitution remains valid.
   - The qualifier DATA in the sample data set name can be replaced by S&sysclone. to reflect the system where the snapshot data set is created.
   - 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 `COMMND`*xx*. 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 93 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 by 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 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 94 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. Because the actual Alert process is the second jobstep in the JCL, you must assign parameters to that jobstep. The use of the fourth positional parameter in the START command is not supported.

```
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.

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. You can only use the FORCE option if you cannot start zSecure Alert; avoid IPLing the system. Since you never need the use of the FORCE option during normal operation, create a problem report with your supplier of zSecure Alert.

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

zSecure Alert operator commands

During execution of the zSecure Alert started task, it is possible for a console operator to communicate with zSecure Alert by issuing MVS MODIFY commands. The zSecure Alert operator commands that might be used in the MODIFY console command are described in the following list:

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 current instance of the zSecure Admin and Audit subtask will be ended. The `stage 1` CARLa subtask and the reporting subtask can both be restarted. Because the alert generating reporting task can only be restarted after completion of the `stage 1` CARLa environment information processing task, it takes several minutes before the refresh is completed.

The REFRESH command does not support any additional keywords.

During the time the zSecure Collect started task is running, the REFRESH command is accepted but most processing is delayed. During this time, the reporting subtask keeps on running. When the collect task has finished, the `stage 1` preprocessing subtask is restarted. When the preprocessing subtask has finished, the reporting subtask is restarted.

COLLECT

This commands 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*.

The COLLECT command does not support any additional keywords.

While the zSecure Collect started task is running, the reporting task keeps on running so alerts are issued as usual and the `stage 1` CARLa environment information processing task is held until the collect task is ready.

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 might not 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 the processing of the captured data. You can use the new values the next time when you need them. When the command is issued by the operator MODIFY command, that time can be almost instantaneous, or never. For instance, a new value for the Interval or AverageInterval parameter, can be used at the next Interval. On the other hand, a new value for the Stage1Interval parameter might never be used, because it is only referenced at the end of the current Stage1Interval, at which time it might be overwritten with the value from Parmlib. For a complete description of all keywords, see "REPORT command" on page 105.

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 107.

Note: If this command is issued by an operator MODIFY command, the specified keywords can only be effective until the next REFRESH. A refresh might be executed as the result of an operator modify command, 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 100.

Note: If this command is issued by an operator MODIFY command, the specified keywords can only be effective until the next REFRESH. A refresh might be executed as the result of an operator MODIFY command, or automatically at the end of each Stage1interval.

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 102.

Cleanup and deactivation of SMF exits

Normally C2POLICE cleans up its environment, even when canceled. If C2POLICE is stopped and for some reason the zSecure Alert SMF exits are still active, you can use the following procedure to deactivate them. See the IBM *MVS System commands* manuals for the command syntax and *MVS Planning: Operations* for command authorizations.

First, issue the following command to determine whether the SMF exit module is active and under what exit names:

```
d prog,exit,mod=c2psmfu8
```

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) SYSTSO.IEFU83    SYSSTC.IEFU83    SYSASCH.IEFU83
EXIT(S) SYSJES2.IEFU83   SYSJES3.IEFU83   SYSTSO.IEFU84
EXIT(S) SYSJES3.IEFU84   SYSASCH.IEFU84   SYSJES2.IEFU84
EXIT(S) SYSSTC.IEFU84    SYSTSO.IEFU85    SYSSTC.IEFU85
EXIT(S) SYSASCH.IEFU85   SYSJES2.IEFU85   SYSJES3.IEFU85
```

Then deactivate for **each** exit by name:

```
setprog exit,modify,en=<exit name>,mod=c2psmfu8,state=inactive
```

On RACF systems, zSecure Alert also defines three new exit points as part of the dynamic installation and activation of the RACF ICHPWX01 exit:

- C2X.ICHPWX01.PRE
- C2X.ICHPWX01
- C2X.ICHPWX01.PST

In most situations, you do not need to explicitly remove these exits and the associated routines since they only provide additional functionality. If you want to remove these exit points, check the section on the use of C2XACTV in *IBM Security zSecure Admin and Audit for RACF: User Reference Manual* for a complete removal process. If you only want to deactivate the creation of additional SMF records, but want to keep the dynamic exit functionality, you can also use the following set of commands.

First determine whether these RACF exit points are active and if they have any modules associated with them:

```
d prog,exit
```

The output looks like the following example:

```
CSV460I 15.40.28 PROG,EXIT DISPLAY 491
EXIT              DEF EXIT              DEF EXIT              DEF
CSVDYLPA           E  CSVDYNEX           E  IEASDUMP.QUERY     E
...
BPX_IMAGE_INIT     E  BPX_PREPROC_TERM   E  EZBTCPIPSMTPEXIT   E
C2X.ICHPWX01.PRE   E  C2X.ICHPWX01       E  C2X.ICHPWX01.PST   E
```

Then request the module names:

```
D PROG,EXIT,EXITNAME=C2X.ICHPWX01.PST
CSV461I 15.41.19 PROG,EXIT DISPLAY 500
EXIT             MODULE   STATE MODULE   STATE MODULE   STATE
C2X.ICHPWX01.PST C2XPWXPS   A
```

There must not be an exit routine associated with the C2X..ICHPWX01.PRE exit point. It is defined for future extensions. If your organization already has an ICHPWX01 exit routine installed in LPALIB, it is now associated with the exit point C2X.ICHPWX01. You might not want to deactivate this exit point. If you want to deactivate the SMF record creation routine, you can issue the following command:

```
setprog exit,modify,en=C2X.ICHPWX01.PST,mod=c2xpwxps,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*.

Filters

As indicated in "Configuration" on page 86, 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. 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 `stage-1` preprocessing subtask obtains current information about the system environment and user attributes. This task is carried out hourly by default. If you do not like outdated information, you must process the security database and the CKFREEZE file every reporting interval. However, it is not necessary. Since obtaining a new I/O configuration image is a costly process, zSecure Collect is normally scheduled to run each day at a particular time to refresh the CKFREEZE file. However, it is also possible to have zSecure Alert dispatch this task by the operator command MODIFY C2POLICE,COLLECT.

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 then 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) 00000814-00307855
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. You must increase the buffer size such that the C2P0334 message is issued only a few times per day, and never for two consecutive intervals.

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 amount of virtual storage specified in the zSecure Alert started task JCL. The region parameter in the JCL must be at least 64 MB larger than the total buffer space specified by *bufsize* and *numbufs*.

Other commands

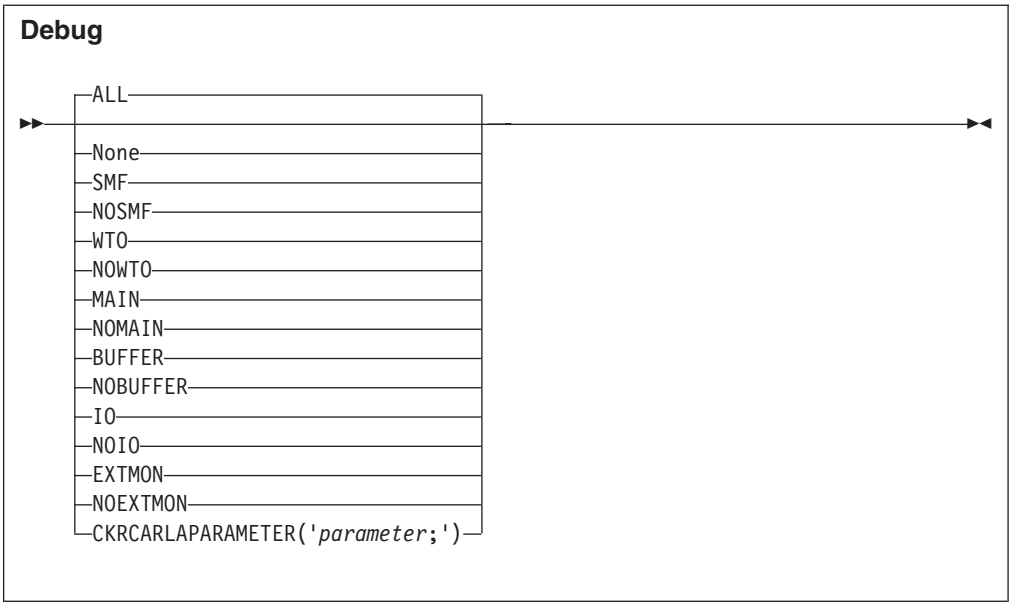
The following commands are not normally required. The DEBUG command enables you to obtain diagnostic information. You can enter these commands in the C2PXPARM member in the Alert configuration data set; see “Required data sets” on page 89.

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 can be used to specify zSecure Alert startup options. It 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.

NOBUFFER
Buffer usage statistics are not written to the console.

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.

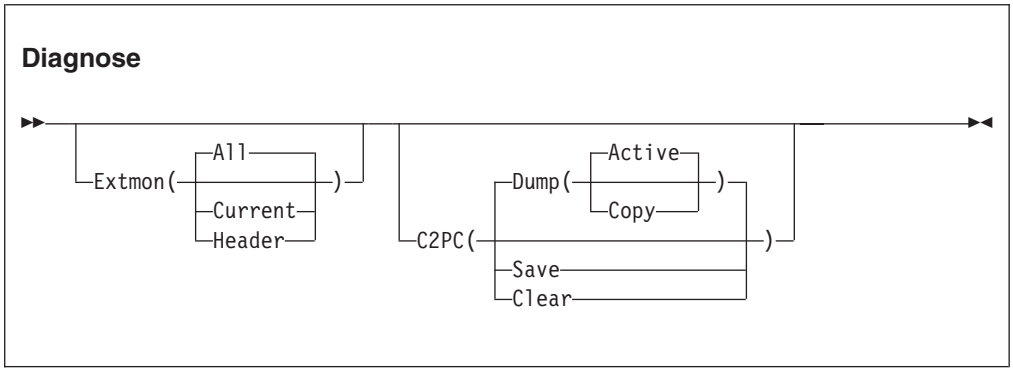
EXTMON
Diagnostic messages pertaining to Extended Monitoring alert processing are to be written to the operator console.

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 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 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:

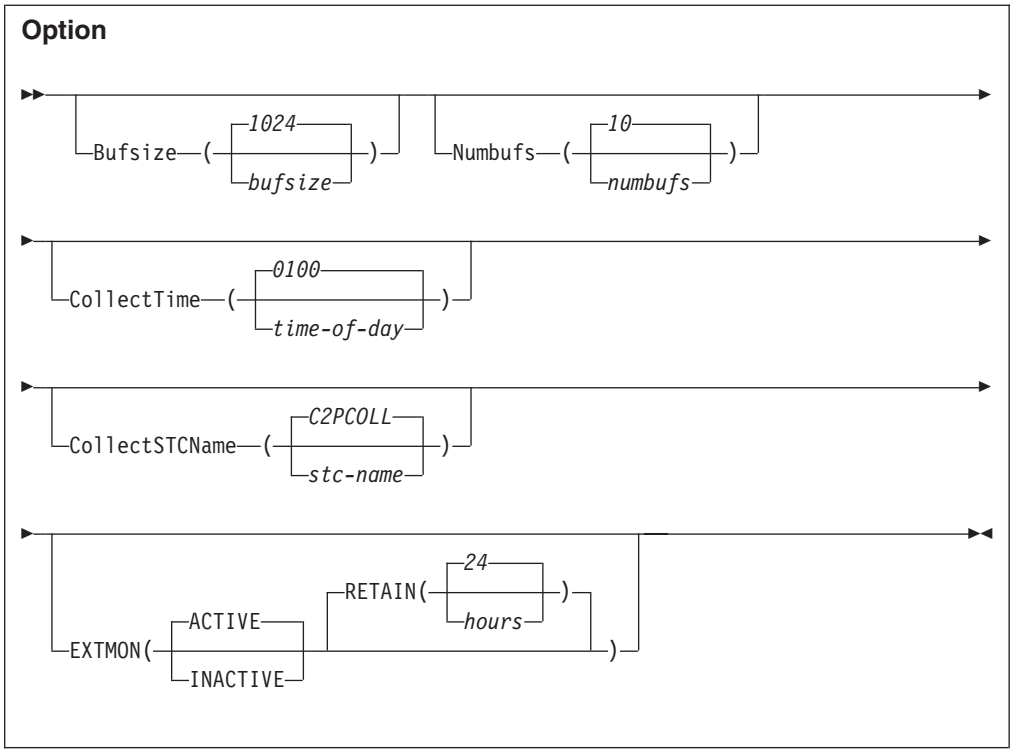
Dump The Active or saved Copy of the C2PC data area is displayed on the operator console in dump format.

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).

OPTION command

The OPTION command can be used to specify zSecure Alert startup options. The OPTION command is only valid when it is part of the PARMLIB statements during initial start or during RESTART processing. The OPTION command is ignored during REFRESH processing. The main purpose of the OPTION command is to specify the number and size of the in-memory data buffers. It has the following syntax.



The keywords and variables have the following values:

Bufsize

Specifies the size of the in-memory buffers used for storing the WTO messages and SMF records during the *interval* period. The buffer must be 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. All records of the current reporting interval are lost. The value *bufsize* specifies the size of the buffers in kilobytes. Valid values for *bufsize* are 1 - 16384, resulting in buffer sizes of 1 KB and 16 MB respectively. Use of the overflow buffers, also called extended buffering, significantly reduce the required *bufsize*. See "Configuration guidelines and performance implications" on page 97 for guidelines on selecting an appropriate buffer size for your installation.

Numbufs

Specifies the number of buffers allocated. The value *numbufs* must be 2 - 32. 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 97 for guidelines on selecting an appropriate number of buffers for your installation.

CollectTime

Specifies the time of day that the zSecure Collect started task 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.

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 must not be started at all.

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.

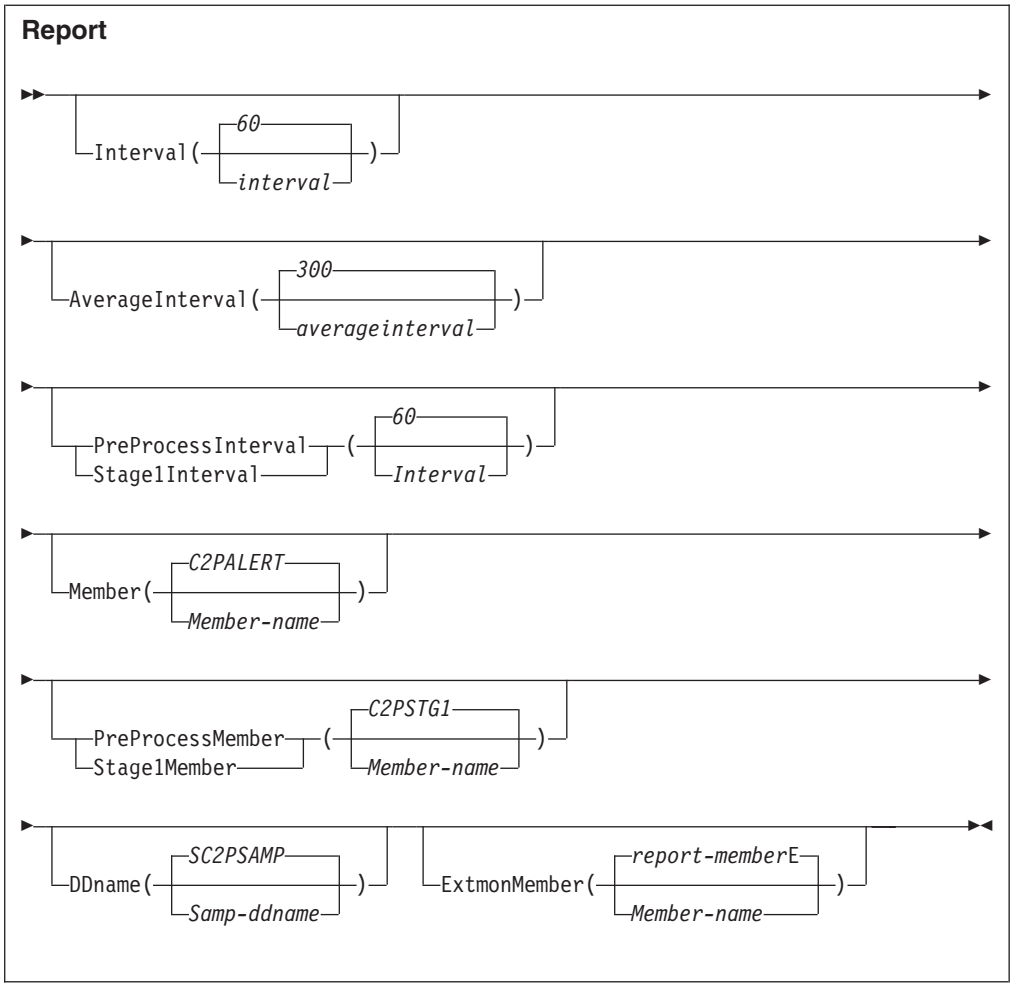
The zSecure Alert started task needs sufficient authorization for the start command. Follow the steps described in "Prerequisites for configuring and using zSecure Alert" on page 83 to define the necessary profiles.

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 87. 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 is used to specify zSecure Alert reporting options. It 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.

```
Numbufs > (AverageInterval / 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.

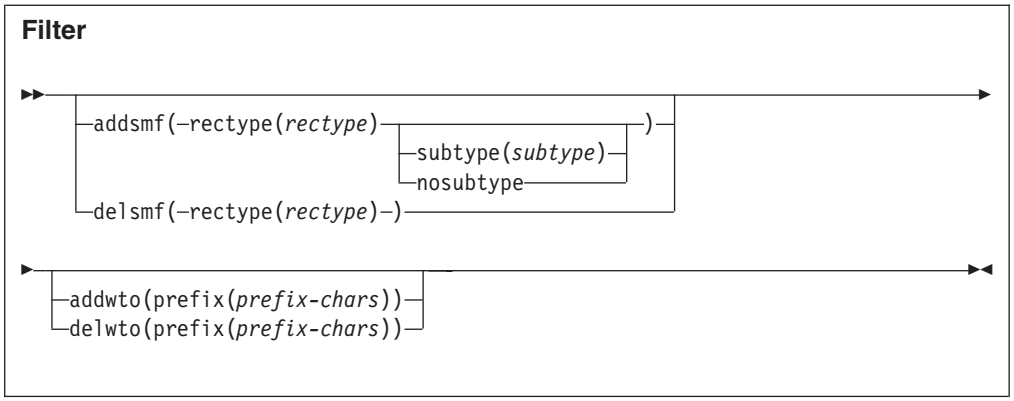
PreProcessMember or Stage1Member
Specifies the `membername` in the partitioned data set that is used for processing the security database and CKFREEZE file. It contains the CARLa statements that result in selection criteria used during the alert generation process. The output of the STAGE1 process must 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 command is used to specify zSecure Alert filter criteria. 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.

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.

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 - 255, 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 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 ACF2
The *subtype* is the ACF2 record type. For a complete list of ACF2 record types, see the ACF2 documentation. The range of values accepted by zSecure Alert is any single alphabetic character or a number 1 - 255.

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*.

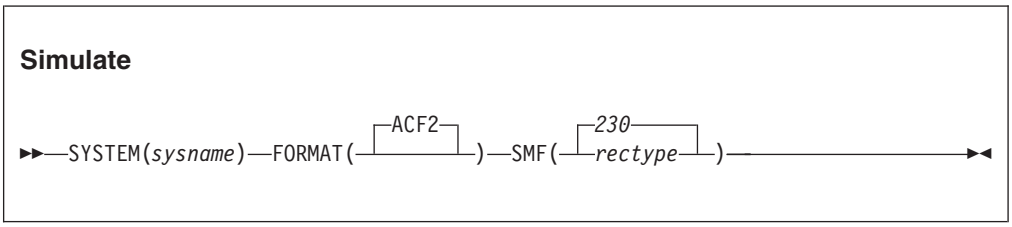
ADDWTO
Specifies the filter criteria used for the WTO-messages. You can specify up to 24 different filter criteria.

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

The SIMULATE command is used to specify the ACF2 SMF-record type for those situations where the SMF-record type cannot be obtained automatically. 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 1.9.0
(or newer) User Interface.

The following downlevel table has been found:
User interface level : 1.4.5
Table name           : C2PIUACA
Creation date        : 2005/03/08
Last change date     : 2005/03/10
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 5. 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.

Note: This check is not performed with zSecure Alert 1.4.5. Therefore, do not use the 1.4.5 level ISPF interface to configure zSecure Alert any more.

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 6. 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 13. 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

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 for running the daily CKGRACF job" on page 43.
- Ensure that both the CKGRACF and CKRCARLA programs are program-controlled. See "Setting up Program Control and PADS access" on page 210 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 116 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 2.1.0" on page 119. 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 115.
- 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 118 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 199). Job C2RZWADM uses group MYGROUP. See "Setup of the server processes" on page 117. 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.

  Note: The `C2RWCUST DD` statement is required starting with zSecure 1.12.

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:

Table 6. Directory and file owner user and group id naming conventions

	Software	**Data**
Owning User	C2RUSER	C2RSERVE
Owning Group	C2RGROUP	C2RSERVG
Directory	/usr/lpp/c2r/V2R1M0	/u/c2rserve/server1
Mount point	/usr/lpp/c2r	/u/c2rserve
File system	OMVS.C2R.ZFS	OMVS.C2RSERVE.ZFS

As shown in Table 6, 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 6 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 BPXPRM*xx* 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.

See Appendix D, "Configuration parameters," on page 213 for descriptions of these parameters.

zSecure Visual Server software

Use the following information 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 115, the default location for the software is /usr/lpp/c2r/V2R1M0. 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 BPXPRM*xx* 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

Job C2RZWUNP is supplied to unpack the zSecure Visual Server software.

- Before running this 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"
- "Updating the Security zSecure configuration: the Server root"
- "Setup of the server processes"
- "TCP/IP Security" on page 118
- "First time startup of the Server" on page 119
- "Upgrading an existing Server to zSecure Visual 2.1.0" on page 119

Setup of the userid and file system

Job C2RZWUSR is supplied to set up the userid for the zSecure Visual Server, its zFS, and its home directory. 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, provided that all of the servers run zSecure Visual 1.8.1 or higher.

Updating the Security zSecure configuration: the Server root

As outlined in "Owners, directories, and file systems preparation" on page 115, 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 C2RZWRUT 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 `MAXCPUTIM` 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 114. 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 199 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.

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 134.

Upgrading an existing Server to zSecure Visual 2.1.0

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.

zSecure Visual 2.1.0 introduces a C2RW131A switch, which allows you to enforce communication to be compliant with NIST 800-131A. However, be aware that older Visual clients might not be enabled, so you likely want to roll out the new version before you actually enforce compliance. Even if the switch is set OFF, communication to clients that do support a compliant protocol will be compliant.

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.

   Make sure that the C2RW131A parameter value is set to `OFF` until all clients have upgraded to at least the 2.1.0 level of the zSecure Visual client.
3. Edit the zSecure configuration that your Server uses. Be sure that the C2RWIN parameter reflects the location of the new software.
4. Stop then restart the server. If you are upgrading from a zSecure Visual Server that does not support NIST 800-131A compliant protocols to one that does, wait until you see the following messages in the `server.log` file in the log subdirectory before attempting to connect a client:
   - `E160:LCM: The LCM certificate in current use, certificate, is not NIST 800-131A compliant. A new LCM certificate will be generated in about 300 seconds.`
   - `E130:CA: The CA certificate in current use, certificate, is not NIST 800-131A compliant. A new CA certificate will be generated in about 300 seconds.`
   - `E130:CA: The CA certificate in current use, certificate, is NIST 800-131A compliant.`
   - `E160:LCM: The LCM certificate in current use, certificate, is NIST 800-131A compliant.`

   Note that the order of these messages is important. The messages about compliant LCM certificates might occur multiple times, but ignore all messages before the message that says the CA certificate is NIST 800-131A-compliant. The first "LCM-compliant" message that comes after the "CA-compliant" message indicates that the server is ready to have clients connected to it.

   Attention: Do not run job C2RZWINI when upgrading. Doing so invalidates all previously issued certificates.

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 127 for information.

Visual server access through ISPF

To access the server from the zSecure (Admin) ISPF panels, you must have a zSecure configuration that is enabled for zSecure Visual. See “Installation requirements” on page 113. 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

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** to create a client and an initial password. You can also use action **A** for now and use action **P** at a later time. If you use action **AP** now and lose or cancel the initial password, or its validity expires before the client is successfully installed, you can use action **P** to generate a new initial password.

```
 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 7. 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. You are prompted to enter a userid and corresponding password:

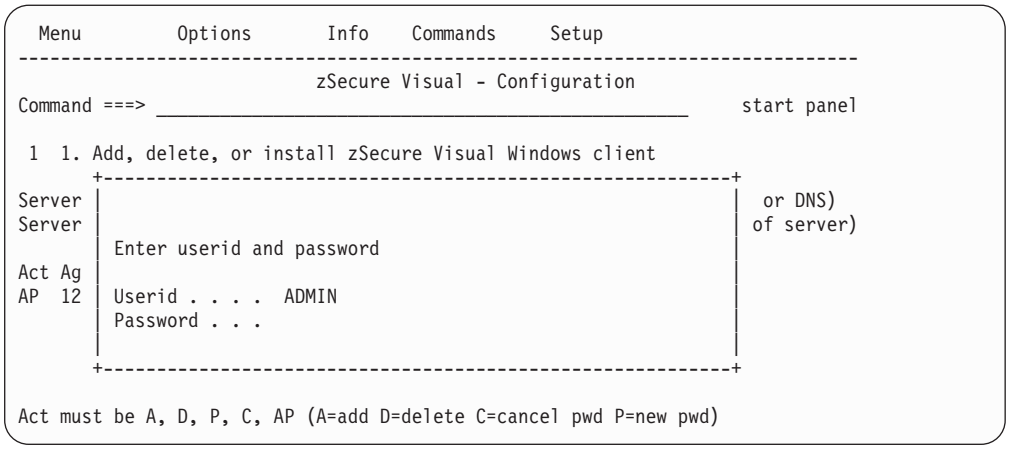


Figure 8. Userid and password configuration for zSecure Visual Windows client

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."

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 140.

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: Certificates existing prior to the server upgrade will automatically be converted to the new encryption standard for the 2.1.0 server. It is not possible to create new certificates for a lower level client on a 2.1.0 level 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

To add to many zSecure Visual agents (workstations) without supplying a password for each one, abulk-agent function is supplied. This 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.
3. 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 199. 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

You can configure what each user can do with the zSecure Visual client interface. Mostly, the menu options, buttons, and fields in the 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 7. 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," 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 8. Resources for role-based authorities

Resource	Uacc	Helpdesk	Connect	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
CKG.CMD.CMD.EX.ADDUSER	n	n	n	u	u	u	u
CKG.CMD.CMD.EX.ALTDSD	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

Table 8. Resources for role-based authorities (continued)

Resource	Uacc	Helpdesk	Connect	User	Access List	Group	Full
CKG.CMD.CMD.EX.DELDSD	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	r	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.

Table 9. Profiles to provide schedule name selection list

Profile	Uacc	Help desk	Con-nect	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

Duplicating a user requires several RACF authorities. Usually at least group-special and CLAUTH(USER) are required. 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.

Table 10. Profile to allow the Define Alias action

Profile	Uacc	Any role
CKG.UCAT.<USER CATALOG NAME>	n	nr*

Resource for RACF scoping

To control access to RACF scoping, use a profile that covers the resource listed in the following table. 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.

Table 11. Resource to activate RACF scoping

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.

Table 12. Profile to enforce password change policy

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 124. (Specifically, they are CKG.CMD.CMD.EX.ALTUSER, CKG.CMD.CMD.EX.ALTGROUP, CKG.CMD.CMD.EX.ALTDSD 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.

Table 13. Resource that controls segment editing

Profile	Uacc	Any role
<CLASS>,,<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.

Table 14. Resource to maintain zSecure Visual server and client definitions

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:

Table 15. Profile for viewing system-wide RACF options

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

The administrator uses these guidelines and settings 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,” on page 213.

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, which is currently 2.1:

```
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:

```
Uuser_id DATA DSN='data.file'
```

U*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:

```
Uuser_id RECFORMAT DSN='recordformat.file'
```

U*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 'pgmrname' (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

Starting from version 2.1.0, it is possible 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.

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 in 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.

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

Use the information in the following sections to start, stop, and view logs for the Visual Server.

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 138.

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 spool 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

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.

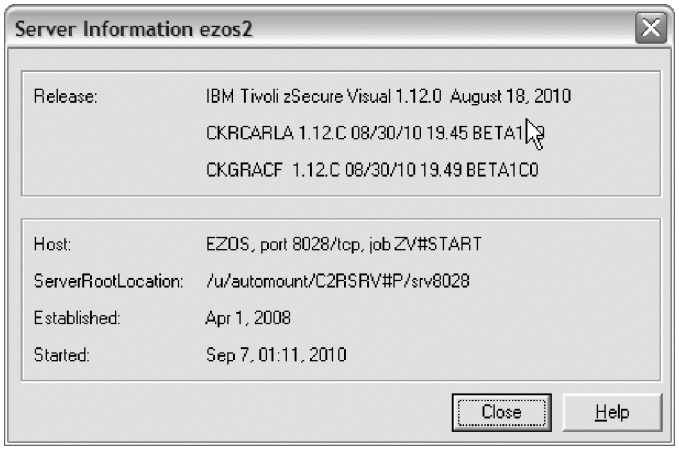


Figure 9. zSecure Visual Client Server Information dialog

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 140.
- 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

Use the **c2rdiag** command to collect diagnostic information about zSecure Visual Server and its environment. The 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 SERVR)
    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 135.
  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 118.

- 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 113) 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 119.

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 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/V1R8M1/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 113 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 199.

<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(ABCDEFG)
  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 113 and “Owner and location preparation for the software” on page 116. 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 121.

Chapter 14. Setup of Change Tracking

The Change Tracking system is a specialized function that monitors changes in system parameters and security settings against a verified base. Changes can be selectively approved, rejected, or deferred through an ISPF interface.

You can run Change Tracking for a single system image, or combine multiple images into a single view in order to have centralized Change Tracking administration.

Data sets required for Change Tracking

Table 16 lists the data sets required to run Change Tracking:

Table 16. Required data sets for Change Tracking

Short name	ISPF access	Batch access	Full name	Remarks
Configuration file	READ	READ	See "Assignment of configurations" on page 27	For each monitored system image, a separate configuration file is required. All these configurations must have the same DPREF parameter. Typically, the only parameter that would be different across monitored system images is SYS. All these configuration files must reside within a single partitioned data set, included in the JCLLIB statement of the Change Tracking jobs. By default, this PDS is CKR.CKRPARM.
Master file	READ (UPDATE)	UPDATE	&DPREF..CT.CKACDATE	UPDATE from ISPF is required only to remove systems from the CT administration.
Local Setup tables	READ (UPDATE)	READ	&DPREF..CT.CKACTAB	Updated by ISPF-transaction Setup Change track (SE.C)
Verified base	UPDATE	READ (UPDATE)	&DPREF..CT.&SYS..CKAVERIF	For each monitored system image, a separate Verified base is required. For an initial run only, the batch process updates this data set.
Exceptions file	UPDATE	UPDATE	&DPREF..CT.CKAEXCEP	
Defer file	UPDATE	-	&DPREF..CT.CKADEFER	
Input	-	READ	&DPREF..&SYS..CKFREEZE &DPREF..&SYS..UNLOAD	For each monitored system image, periodically refreshed CKFREEZE and UNLOAD are needed.

Table 16. Required data sets for Change Tracking (continued)

Short name	ISPF access	Batch access	Full name	Remarks
Intermediate files	-	CREATE DELETE	&DPREF..CT.&SYS..CKATSYSI &DPREF..CT.&SYS..CKATCOMP &DPREF..CT.&SYS..CKATPRIN &DPREF..CT.&SYS..CKATEXCP &DPREF..CT.&SYS..CKATREPP &DPREF..CT.&SYS..CKATREPS &DPREF..CT.&SYS..CKATSIMS &DPREF..CT.&SYS..CKATSYSD	For additional information, see the details of job CKAJTSYS. See Job CKAJTSYS.

Create the Master, Exception and Defer files, and the Local Setup tables with job `CKAJTCT1` residing in data set `CKRJOBS`.

Only one of each of these data sets is required, even when multiple system images are monitored. To use any of the configurations described in Table 16 on page 145, adapt the `JCLLIB` and `INCLUDE` statements in this job.

Verified bases are generated with job `CKAJTCT2`. For each monitored system image, a separate Verified base is required. Adapt the `JCLLIB` and `INCLUDE` statements to use all the configurations of the monitored system image. That is, job `CKAJTCT2` is to be run once with each configuration.

Set up the Change Tracking security environment in such a way that the Change Tracking jobs and the intended users of the ISPF component have the required access to these data sets. See the `CKAJTCT3` job for an example of the JCL to set up the appropriate access. Ensure that the security resources you create are all subject to your security policy, such as choices between generic and discrete profiles.

Setup of the daily batch suite

Change Tracking uses its data sets as follows:

- In batch, periodically refreshed CKFREEZE and UNLOAD data sets are checked against the verified base.
- Changes are available in an ISPF interface (AU.C), where they can be selectively approved. Approved changes are added to the verified base. However, changes can also be Rejected or Deferred.

For simplicity, it is assumed that all Change Tracking processes are run under a single z/OS image. Fresh CKFREEZE and UNLOAD data sets must be accessible from this image. However, creating a CKFREEZE data set can only be done from within an image itself. If you do not have shared DASD in place, you can make the CKFREEZE data sets available to Change Tracking through any transfer method (tape, NJE, FTP), as long as the transfer method preserves `LRECL=X` and does not truncate or wrap long records. If you use FTP, you must use both the EBCDIC and BLOCK options.

Similarly, an UNLOAD data set can be created from any system image that has access to a security database, but for best results, create it under the highest level image that uses it. You can transfer UNLOAD data sets in the same way you transfer CKFREEZE data sets.

For information about how to create CKFREEZE and UNLOAD data sets, see “Use of a fresh CKFREEZE and UNLOAD each day” on page 42.

For a shared security database, normally only a single UNLOAD data set is created. However, the Change Tracking jobs assume `&DPREF..&SYS..UNLOAD` for its data set name, which would be different for each system image. You do not have to create multiple UNLOAD data sets of the same database, however. Instead, you can copy the UNLOAD with a program like IEBGENER, or simply create an alias. For example, when images IPO1 and IPO2 share a security database, and you created an UNLOAD with job C2RJPREP on image IPO1, you can create the alias with:

```
DEFINE ALIAS (NAME('yourprefix.IPO2.UNLOAD') RELATE('yourprefix.IPO1.UNLOAD'))
```

The Change Tracking batch suite consists of the following:

- Job CKAJTSYS, which calls the CKACTSYS procedure, updates the Master and the Exceptions file. Run this job once (serially) for each monitored system image. This job can run on a system image that is different from the monitored system image. Do not run CKACTSYS until refreshed CKFREEZE and UNLOAD data sets are available. That is, these data sets have been created or refreshed and if needed, transferred to the system image where the Change Tracking jobs are run.

  Adjust the JCLLIB and INCLUDE statements in this job to use the configuration of the image whose data is to be processed. In addition, set the INIT#CT parameter to EQ the first time a particular image is processed by Change Tracking. This setting immediately promotes the entire configuration into the verified base, which prevents the entire configuration from being signaled as exceptions. After the first run, set INIT#CT back to NE.

  Job CKAJTSYS creates intermediate data sets when needed. For information about intermediate data sets, see Table 16 on page 145. The intermediate data sets are usually removed after the job completes, but you can choose to retain them for diagnostic purposes. To retain them, use the TREMOVE=NE option when you start the procedure CKACTSYS.
- Job CKAJTSRT, which calls the CKACTSRT procedure, removes duplicates from the exception files. Duplicate information can be added as a result of re-running jobs in the event of a job or systems failure. Run this job after all CKAJTSYS jobs are finished to ensure that the duplicate information is removed from the exceptions file.

  Update JCLLIB and INCLUDE statements in this job to use the configuration of any of the monitored systems.
- Job CKAJTM calls the CKACTPRT and CKACTM procedures. This job produces a printable report of the Change Tracking exceptions files and sends this report as an email memo. This job can optionally be scheduled to run after job CKAJTSRT.

  Update JCLLIB and INCLUDE statements in this job to use the configuration of any of the monitored systems. You must also add the email addresses of the intended email recipients to this job.

  You can use procedures CKACTPRT and CKACTM separately. For example, to print a report instead of emailing it, start the CKACTPRT job with the parameter RPTTYPE set to SPOOL. In the combined job CKAJTM, this value is overridden with RPTTYPE=FILE, because CKACTM needs a file as input for email transmission.

For all these jobs, it is assumed that you do not directly update in the SCKRSAMP data set because directly updating that data set violates the distribution-oriented installation rules. Instead, it is assumed that you copy the required jobs to your own data set such as one that contains jobs for your job scheduling software, and update your local copy.

Procedure CKACTSYS invokes procedure CKAC. As a result, SCKRPROC members C2RI* are used to allocate the UNLOAD and CKFREEZE files. Because z/OS does not allow overriding DD statements in nested procedures, you cannot use CKACTSYS if your naming convention does not fit with the C2RI* members. In that case, you must code your own CKAC invocations.

Because the Change Tracking suite must run serially, you might want to combine them into a single job. For instance, if you run Change Tracking over the monitored images IPO1, IPO2, and IPO3, you can combine jobs CKAJTSYS and CKAJTSRT as follows:

```
//JCLLIB   JCLLIB ORDER=(your.prefix.CKRPARM,
//         CKR.SCKRPROC)
//*
//*  Process input from systems IPO1, IPO2 and IPO3. Each of
//*  these needs to run under its own configuration.
// EXEC CKACTSYS,INIT#CT=NE,CONFIG=CFG#IPO1
// EXEC CKACTSYS,INIT#CT=NE,CONFIG=CFG#IPO2
// EXEC CKACTSYS,INIT#CT=NE,CONFIG=CFG#IPO3
//*
//*  Remove duplicates. This runs only once, under any of
//*  the above configurations.
// EXEC CKACTSRT,CONFIG=CFG#IPO1
```

Change Tracking with the ISPF interface

The ISPF interface for Change Tracking consists of the following panel options:

AU.C Inspects signaled changes and carries out follow up actions. This task is described in the User Reference Manual. To access this task, the user must have READ access on the XFACILIT resource `CKR.OPTION.AU.C`. For information about using a different resource class, see Appendix A, "Site module," on page 199.

The user of AU.C also requires READ access to resources covered by `CKR.ACTION.CH.*` (for actions on the Exceptions overview) and `CKR.ACTION.CT.*` (for actions on the System overview). See "Resources that configure which options are shown" on page 201.

SE.C Maintains tables that job CKAJTSYS uses; for example, data sets that you consider sensitive. By default, Change Tracking considers sensitive data sets to be the same data sets as CARLa REPORT SENSITIVE. This task is described in the User Reference Manual. To access this task, the user must have READ access on the XFACILIT resource `CKR.OPTION.SE.C`. For information about using a different resource class, see Appendix A, "Site module," on page 199.

Job CKAJTCT3 offers an example to set up RACF profiles for this. However, be sure that the security resources you create are subject to your security policy, such as choices between generic and discrete profiles.

To use the ISPF interface for Change Tracking, you must have a configuration with an appropriate `DPREF` parameter. For instance, you can use any of the configurations used by job CKAJTSYS.

Change Tracking interface to an external change management system

Every time a change is rejected, deferred, or confirmed, the REXX CKAECHGM program is called. This REXX program is the Change Tracking interface to an external change management system, such as IBM Information Management. By default no checking is done. The interface must be provided by the installation. This exit point is designed to validate a change management number.

Change tracking for previous levels of data sets

Previous levels of zSecure (before version 1.8.1) used different data sets names. Jobs CKAZCTU1/CKAZCTU2 (in SCKRSAMP) are supplied to assist you in converting. Job CKAZCTU2 is intended for the CKAVERIF data set and must be run separately for each system that you want to monitor. Job CKAZCTU1 is intended for the other data sets and needs to be run only once. See the instructions in the jobs.

Chapter 15. Data preparation for QRadar SIEM

You can use zSecure to make z/OS event data available for QRadar SIEM.

For QRadar SIEM, a z/OS image contains a number of Log Sources: for z/OS itself and 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. On the z/OS image, you must set up a zSecure process to transform SMF records into the Log Event Enhanced Format (LEEF) that QRadar expects. This process includes enrichment of the raw SMF data with data from your system configuration and security database.

Device Support Modules (DSMs) on QRadar SIEM retrieve these LEEF files, according to a schedule that is configured on the QRadar console.

You can also send alerts generated by zSecure Alert to QRadar SIEM. The alerts can be based on SMF or on other sources (for example based on the detection of system changes). Alerts are transferred real-time to QRadar SIEM and are not dependent on any configured schedule. In zSecure Alert, specify the UNIX syslog format, and specify QRadar SIEM as the recipient. For more info on zSecure Alert, see IBM Security zSecure Alert.

Prerequisites

Be sure to meet the following requirements before you set up the data collection process:

- Before you set up the data collection process for QRadar, you must complete the basic, shared part of the zSecure installation process. After installing the software, you must also perform activities to create and modify the configuration. The following criteria must be met:
  - PARMLIB member IFAPRD*xx* must enable at least the zSecure Audit component. 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.
  - 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.
- You must have an active FTP (or SFTP) server on your z/OS image, so that QRadar can download the LEEF files.
- The zSecure configuration must contain the specific parameters for QRadar SIEM. For information, see "Updating the configuration files" on page 155.

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 152.

2. Make the SMF records available for QRadar. See "Making SMF records available to QRadar" on page 153.

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 (DB2 IFCids 4, 5, 6, 7, 8, 9, 10, 22, 23, 24, 25, 55, 83, 87, 90, 92, 104, 105, 107, 140, 141, 142, 143, 144, 145, 169, 177, 219, 220, 258, 270, 314, and 319)
- 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 System z
- 83 subtype 5 for data from WebSphere Application Server
- 83 subtype 6 for data from IBM Security Key Lifecycle Manager

The exact SMF record selection is specified in the CARLa member C2ELEEF. This member can be updated by regular maintenance.

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 DFHMCT*xy* 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 Information Center at http://www.ibm.com/software/htp/cics/tserver/v42/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(6,7,8,9,10,22,90,107,
              177,314)
-<subsysname> START TRACE(STAT) DEST(SMF) CLASS(30) IFCID(258)
-<subsysname> START TRACE(AUDIT) DEST(SMF) CLASS(*)
```

- If you use installation-defined events, make sure to include the SMF records required by your CARLa member C2EQCES.

Making SMF records available to QRadar

Before you begin

Do not specify your live SMF data sets as your only input. Doing so results in gaps: SMF records that are written after you run the QRadar job (or started task), but before the next SMF switch, would be missing

Procedure

- 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 QRadar 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 QRadar 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. 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 QRadar 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. zSecure skips records that were already processed, but the excess reading costs 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 QRadar; this is not supported. Instead, do one of the following:
  - Specify an EXCLUDE statement in member C2EQ0ES. See “Updating the configuration files” on page 155.
  - 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 QRadar 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.
- 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

zSecure supplies job C2EJQRLF and procedure C2ECQRLF.

- If you want to run the collection process in batch, configure your job scheduling product to submit a copy of job C2EJQRLF. As an alternative, you can arrange for your SMF offload process to submit this job.
- If you want to run the collection process as a started task, include procedure C2ECQRLF and the zSecure configuration member in a procedure data set for your Job Entry Subsystem (JES), and have it started by your automated operation, or by a JES autocommand.
- Customize your copies of the JCL according to the conventions used in your installation. In particular, specify the zSecure configuration member of your choice. The default is C2R$PARM, but if you use several functions of the zSecure Suite, you probably want separate configurations for each function.
- Store your configuration member where the JCL for the job or started task can access it. 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.

  QRadar-specific parameters that you must supply are:

  C2EQCUST
  The name of the data set that contains configuration members C2EQENV, C2EQSPEC and C2EQFIN, see "Updating the configuration files" on page 155.

  C2EQPATH
  The UNIX directory where the collection process is to leave its data.

Note:

1. Do not start the process until you finish updating your configuration.
2. 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. Just rerun.

Assigning a userid and preparing a directory to store the LEEF data

About this task

You must assign and prepare a UNIX directory where the LEEF data is stored for retrieval by QRadar. 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 three different jobs to create the required user and group, home directory, and file system:
- Job C2EQAUSA for ACF2
- Job C2EQAUSR for RACF
- Job C2EQAUST 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.

Updating the configuration files

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, members C2EQENV, C2EQSPEC, and C2EQFIN might be missing. If so, copy these members from the SCKRCARL and SCKRSAMP libraries.

If you want further customization, also copy members C2EQ0ES, C2EQXES, and C2EQCES.

Procedure

Now customize the members:

1. Adapt member C2EQENV to specify your input: specify the UNLOAD and 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. If you are using Top Secret, remove the UNLOAD allocation, because that does not apply to Top Secret.
2. Adapt member C2EQSPEC to specify your input and output:
   a. Specify the same UNLOAD and CKFREEZE data set that you specified in member C2EQENV.
   b. Specify the SMF that you selected as input, either as the name of a logstream, or by using the DSNPREF parameter. See "Making SMF records available to QRadar" on page 153. 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. 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`; this action avoids 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 C2EQPATH parameter.
   d. 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.
   e. If desired, specify output options. The default options are in the zSecure-supplied sample member.
3. Adapt member C2EQFIN to specify the retention period (in hours) for the LEEF files.

4. Optionally, adapt members C2EQ0ES, C2EQXES, and C2EQCES as follows:

   C2EQ0ES
   CARLa 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 to use until the problems with the OEM vendor are solved, or a more permanent solution is built.

   You can also exploit this member to specify the right CCSID when translating, for example, double byte character set (DBCS) characters to UTF-8. For instance, for Japanese Latin extended Unicode, you can include the following CARLa statement:

   ```
   OPTION MY_CCSID=1399
   ```

   C2EQXES
   CARLa EXCLUDEs for the SMF selection.

   C2EQCES
   CARLa to customize the z/OS Log Source to also map installation-defined events. For example, you might have a product in place with its own SMF records.

QRadar log source properties

On the QRadar Console, specify the properties of the Log source. Figure 10 shows an example screen. For more information, see the *Configuring DSMs Guide* for QRadar.

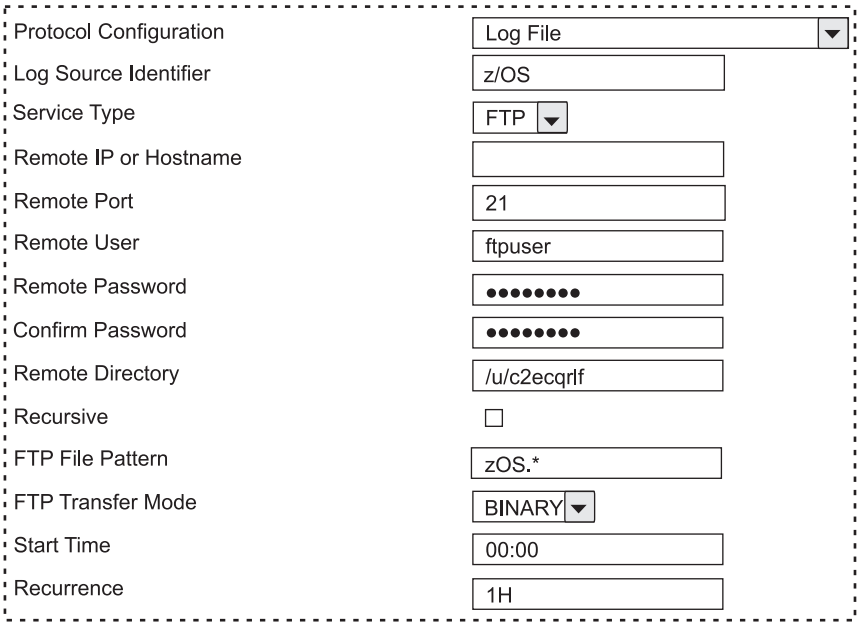


Figure 10. Log source properties configuration for QRadar

QRadar z/OS-specific event properties

QRadar shows a number of normalized event properties automatically. Additional fields can be added as custom properties in the QRadar user interface.

A list of appropriate fields that you want to consider is listed in the QRadar technote at https://qmmunity.q1labs.com/node/2615

Chapter 16. Setup of Tivoli Compliance Insight Manager Enabler for Tivoli Security Information and Event Manager

If you run Tivoli Security Information and Event Manager and want it to include your z/OS system images, you must set up an Agent.

Normally, each Agent processes the following data sources from the z/OS image where it is active:

- An Event Source (ES). For z/OS, the data is SMF data.
- A User Information Source (UIS). For z/OS, this consists of the security database and CKFREEZE.

However, under certain conditions multi-image Agents are supported. See "Performance and multi-image considerations" on page 181.

Note: Tivoli Compliance Insight Manager Enabler for z/OS 2.1 requires Tivoli Compliance Insight Manager Version 8.5 or any version of Tivoli Security Information and Event Manager. For full functionality, use the most recent version of zSecure and the most recent version of Tivoli Security Information and Event Manager. In this chapter, unless otherwise stated, Tivoli Security Information and Event Manager refers to supported versions of Tivoli Compliance Insight Manager and Tivoli Security Information and Event Manager.

Overview

Tivoli Security Information and Event Manager uses the event data that is created through normal SMF processing. The agent component, Tivoli Compliance Insight Manager Enabler for z/OS, copies selected SMF data to a file that is stored in UNIX System Services (USS, formerly called OMVS) and then passes the data to the Tivoli Security Information and Event Manager Server. The Agent consists of the following processes:

- The Agent. This process provides a secure communication channel to the Tivoli Security Information and Event Manager Server. It is typically started soon after IPL and only stopped in preparation for the next IPL.
- The User Information Source actuator. This process collects data from the security database and from the CKFREEZE data set.
- The Event Source actuator. This process reads live or accumulated SMF data and generates an extract available to be used by the Agent. The original SMF data is not deleted, changed, or moved, and so it remains available for other functions that need this data; for example, chain-of-evidence and non-security processes that require SMF records as input. The ES actuator also references the UIS data.

Authorizations and expertise required during configuration

Several of the Tivoli Compliance Insight Manager Enabler for z/OS configuration steps require specific authorizations and expertise, or assistance from someone who has the authorizations and expertise. This person might be your system programmer, your security administrator, your network administrator, your USS expert, or your production planner. In the following list, the authorizations that are required depend in part on choices that you make:

- You must have authorization to set up the required users, groups, directories, and file systems. See "Owners, directories, and file systems" on page 164.
- You must select an available set of IP ports for the Agent. On the Management Console, specify a port number for the Agent to use. See "Secure connection setup" on page 177. The Agent might also try to find and open one port in the dynamic or private port range 49152 through 65535. For an Agent on z/OS, the default port is 5992. See "TCP/IP security" on page 163 for information about port and stack protection.
- For best results, be sure that each Agent uses a dedicated file system (HFS or zFS) for its data. Authorization is required to create and mount this file system.
- Depending on your choice for started tasks or batch jobs, authority is required to set up `STARTED` or `SURROGAT` profiles or the equivalent in ACF2 or Top Secret. See "Starting and stopping the Agent" on page 169.
- For started tasks, you must also have update access to one of the procedure libraries of your Job Entry Subsystem.
- Because the Tivoli Compliance Insight Manager Enabler for z/OS is intended as a production process, you might require authorization to create entities in your system for Job Scheduling, Automated Operations, or both.

Before you decide how to configure the Agent and set up the required authorizations, read the information that follows.

System requirements

This section describes the requirements to install and use a z/OS Agent for Tivoli Security Information and Event Manager.

Installation and configuration criteria

Before you use a z/OS Agent for Tivoli Security Information and Event Manager, you must complete the basic, shared part of the zSecure installation process. After installing the software, you must also perform the post-installation activities to create and modify the configuration. The following criteria must be met:

- PARMLIB member `IFAPRD`*xx* must enable the Tivoli Security Information and Event Manager RACF Enabler for z/OS component. 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.
- The zSecure configuration must contain the specific parameters for Tivoli Compliance Insight Manager Enabler for z/OS. For information, see "Parameters for the z/OS Agent" on page 167.

For instructions for installing and configuring zSecure, see the first few chapters of this installation manual. For best results, use a dedicated configuration. See "Assignment of configurations" on page 27. For information about how to use the zSecure configuration member, see "Preparation of a new Agent" on page 167.

Note: For step-by-step instructions for installing the z/OS agent for Tivoli Security Information and Event Manager, go to the IBM Support Web site at http://www.ibm.com/software/support/probsub.html. Search for a technote about installing Tivoli Compliance Insight Manager Enabler for z/OS.

The Tivoli Compliance Insight Manager Enabler for z/OS exists in several separately orderable features: for RACF, ACF2, TSS, DB2®, and CICS. If you have

more than one of these features, it is important to install them all in a single SMP/E zone, together with the zSecure base. If you also have zSecure components that are not part of Tivoli Compliance Insight Manager Enabler for z/OS (such as Audit, Alert, and Visual), also install them in that same SMP/E zone. Installing in this way gives you a single set of libraries with all the required capabilities.

A single instance of the Agent, using these full-capability libraries, collects all the events for Tivoli Security Information and Event Manager at one time, reading the SMF only once. From the Tivoli Security Information and Event Manager side, the Event Sources for RACF, ACF2, TSS, DB2, and CICS appear as a single audited machine, and a single Event Source (and a single User Information Source).

This single Event Source supplies data for all features, and a single trigger from the Tivoli Security Information and Event Manager server collects them.

SMF data

To generate data, 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, 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), 90, most of 92, 118, and 119
  - Selected subtypes of 102 (DB2 IFCids 4, 5, 6, 7, 8, 9, 10, 22, 23, 24, 25, 55, 83, 87, 90, 92, 104, 105, 107, 140, 141, 142, 143, 144, 145, 169, 177, 219, 220, 258, 314, and 319)
  - The CICS monitoring record type 110 subtype 1
  - The ACF2 record type (site-defined number) if you have ACF2

  The exact SMF record selection is specified in the CARLa member C2ELES. This member can be updated by regular maintenance.
- To generate SMF records for CICS transactions, CICS monitoring must be enabled and set up. 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: monitoring on classes of monitoring data and options, classes of monitoring data and options, or both.

  For more information, see the CICS Monitoring Facility documentation in the CICS Transaction Server Information Center at http://www.ibm.com/software/htp/cics/tserver/v31/library/.

  The SET MONITOR command can also be used 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:

```
-<subsysname>  START TRACE(PERFM) DEST(SMF) CLASS(30) IFCID(6,7,8,9,10,22,90,107,177,314)
-<subsysname>  START TRACE(STAT) DEST(SMF) CLASS(30) IFCID(258)
-<subsysname>  START TRACE(AUDIT) DEST(SMF) CLASS(*)
```

  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.

- If you use installation-defined events, make sure to include the SMF records required by your CARLa member C2EICES. For additional information about installation-defined events, see "CARLa members that support adding CARLa statements (optional)" on page 171.

Setting z/OS UNIX time zones

Procedure

If the Agent is used to audit systems in different time zones, use the following steps to ensure that the USS time zones are correct:

1. Set each hardware clock on each system to Universal Time Coordinate (UTC).
2. Set the local timezone in your PARMLIB member `CLOCK`*xx*. This change is not required for the Agent. However, if you had previously set the hardware clock to local time, changing this setting keeps the timestamps for other messages and applications (that is, the non-UNIX processes) consistent.
3. Set the `TZ` variable to the local time zone. You can set the `TZ` variable in the `/etc/profile` file or in the `.profile` of the user that runs the Agent.

   Caution:

   - Set the hardware clock on each system to the same value as the hardware clock of the Tivoli Security Information and Event Manager Server to ensure consistency in the reports. See the following examples:
     - For Eastern Standard Time in the United States, set the `TZ` variable to `"EST5CDT"`.
     - For Mountain Standard Time in the United States, set the `TZ` variable to `"MST7CDT"`.
     - For Central Europe Time, set the `TZ` variable to `"CET1CEST,M3.5.0,M10.5.0"`. (This time is one hour ahead of UTC, with daylight saving time starting the last Sunday in March and ending the last Sunday in October).
   - If your site does not refresh the CKFREEZE often, it is a good idea to schedule your CKFREEZE refresh on Sunday, or to schedule an additional CKFREEZE refresh during the night when daylight saving time begins or ends, after the switch from CEDT to CET, from EDT to EST, or after other changes. Depending on the collection strategy you choose, the refresh can be done using job C2RJPREP, or it can be part of UIS-Collect. See "Use of a fresh CKFREEZE and UNLOAD each day" on page 42. Also see "Event Source and User Information Source properties" on page 186 for information about collection strategies.

     The reason for this suggestion is that the Agent passes the time zone of the monitored system to the Tivoli Security Information and Event Manager Server, so that the server can display the time correctly. The Agent obtains its time zone from the CKFREEZE data set, rather than from the live system because the Agent does not have to run under the z/OS image that it monitors. You can find an unexpected amount of work happening between 2:00 a.m. and 3:00 a.m. because if things work automatically, this time interval is skipped (beginning of DST), or used twice (end of DST).

Unicode requirement

The z/OS Agent requires Unicode support. You can verify whether Unicode is available with the operator command `D UNI,ALL`.

If Unicode is unavailable, look in SYS1.PARMLIB or other concatenated parmlibs for a member CUNUNI*xx*. There is no default member shipped by IBM; it must be built by the person who configures Unicode. If the member is present, Unicode can be activated with the command SET UNI=xx. Consider activating Unicode automatically at IPL time by specifying UNI=xx in your IEASYSxx parmlib member.

If member CUNUNI*xx* is not present, you or your system programmer must build it. The member can contain code as simple as the code in the following example: IMAGE CUNIMG00; REALSTORAGE 25600;

The IMAGE keyword refers to another member in PARMLIB that you can compile with the image generator CUNMIUTL. The following job step provides an example:

```
//CUNMIUTL EXEC PGM=CUNMIUTL
//SYSPRINT DD   SYSOUT=*
//TABIN    DD   DISP=SHR,DSN=SYS1.SCUNTBL
//SYSIMG   DD   DISP=SHR,DSN=SYS1.PARMLIB(CUNIMG00)
//SYSIN    DD   *
    CASE NORMAL;              /* ENABLE TOUPPER AND TOLOWER */
     CONVERSION 1047,0037;    /* EBCDIC     -> EBCDIC 037   */
     CONVERSION 0037,1047;    /* EBCDIC 037 -> EBCDIC       */
     CONVERSION 1208,0037;    /* utf-8      -> EBCDIC 037   */
     CONVERSION 0037,1208;    /* EBCDIC 037 -> utf-8        */
     CONVERSION 1047,1208;    /* EBCDIC -> utf-8            */
     CONVERSION 1208,1047;    /* utf-8 -> EBCDIC            */
```

For additional information about setting up Unicode, see the IBM manual *Support for Unicode™: Using Conversion Services*(SA22-7649). The CUNUNI*xx* member is described in the "*MVS Initialization and Tuning Reference*.

TCP/IP security

The Agent and the first-time start by job C2ECNNCT both require permission to use the IP stack and the selected port. In the PORT or PORTRANGE statement in your TCP/IP configuration, specify an SAF resource as shown in the following example:

```
PORTRANGE 5992 2 TCP * NOAUTOLOG SAF C2EAGENT
```

This statement restricts the use of TCP ports 5992-5993 to users that have at least READ access to the EZB.PORTACCESS.sysname.tcpname.C2EAGENT 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 job name or names that you intend to use for the server, for the first-time server-start job (for example, C2E*), or both. However, with SAF active, there is usually no need to impose job name restrictions.

The Agent and the first-time start by job C2ECNNCT also require 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.

If you have activated protection of unreserved ports in your TCP/IP stack, you must grant the userid under which you will run the Agent permission to use these ports. For more information about protecting unreserved ports, go to the z/OS information center at http://publib.boulder.ibm.com/infocenter/zos/v1r11/index.jsp and see **Communications Server -> IP Configuration Reference**.

Domain name resolution

The z/OS agent requires that the localhost domain name can be resolved to the 127.0.0.1 IP address, either through your domain name server or through the hosts file.

To check whether the localhost domain name can be resolved, issue the command `ping localhost` on the TSO command line. If successful, the result is similar to these lines:

```
CS V1R12: Pinging host LOCALHOST (127.0.0.1)
Ping #1 response took 0.005 seconds.
```

Consult your TCP/IP documentation for information about how to update your DNS or hosts file.

Owners, directories, and file systems

To support multiple instances of the Tivoli Compliance Insight Manager Enabler for z/OS, use separate directories and separate file systems for software and data. For example, multiple Agents can run in separate z/OS images or within a single z/OS image, sharing the file system where the software resides. Or, you might want to send your z/OS system audit data to multiple independent Tivoli Security Information and Event Manager servers. By providing separate directories and file systems, you can upgrade z/OS without reinstalling Tivoli Security Information and Event Manager. With this configuration, you can also upgrade the zSecure software without affecting the audit data.

Because each file or directory in UNIX must have both an owning user and an owning group, you must assign owners. The following default settings are used in this documentation and in the zSecure-supplied jobs. You can adapt these settings to the conventions used by your installation:

Table 17. Default owners, directories, and file systems

	Software	**Data**
Owning User	`C2RUSER`	`C2EAUDIT`
Owning Group	`C2RGROUP`	`C2EGROUP`
Directory	`/usr/lpp/c2e/v`*x.y*	`/u/c2eaudit/actuatr1`
mount point	`/usr/lpp/c2e`	`/u/c2eaudit`
file system	`OMVS.C2E.ZFS`	`OMVS.C2EAUDIT.ZFS`

As shown, the data is owned by `C2EAUDIT`, the (default) user under which the Agent runs. However, the Agent does not own the files containing the software. Instead, the Agent is granted READ and EXECUTE access to the software through the group permission bits and through one of the following settings, depending on which system you are on:

- A `CONNECT` to the group that owns the software files on a RACF system
- A `TGR` resource rule on ACF2 systems
- An ADDTO of the owning group on Top Secret systems.

In the same way, you might want to connect your security support and production control personnel to `C2EGROUP` in order to grant them access to the Agent-owned data files so that these users can view the log files, for example.

Note: The default *mount points* in Table 17 on page 164 do not coincide with the software and data directories. You can therefore set up multiple Agents under a single userid. Similarly, you can install future software releases, or future zSecure products, within a single file system. If you do want separate file systems for each release, or for each Agent, you can use the zSecure-supplied jobs as templates, running them multiple times if necessary.

Automount is commonly set up for file systems that are used for home directories, such as `/u/c2eaudit`, but is not usually set up for software. For file systems that are not automounted, update the `BPXPRMxx` member in your parmlib to ensure that the file systems will be mounted after subsequent IPLs.

Although owning users and groups in Table 17 on page 164 are represented by user and group names, it is the numeric UID/GID that is recorded for UNIX files. Therefore, it is not required to create a unique user for the single purpose of owning files. In particular, if your installation standards require the files that contain the zSecure software to be root-owned, no unique userid is required. You can submit job C2EUNPAC from a userid that has `UID=0` in the OMVS segment. The Agent does not use `SETUID`, so there is no security risk in making the software root-owned.

Installation of the Tivoli Compliance Insight Manager Enabler for z/OS software component

To install the Tivoli Compliance Insight Manager Enabler for z/OS software component, use the instructions provided in the following sections:

- "Preparation of the owner and location for the software"
- "Uploading the pax file" on page 166
- "Unpacking the software" on page 166

Preparation of the owner and location for the software

Job C2EZCZFS for RACF, C2EZCZFA for ACF2, and C2EZCZFT for Top Secret are supplied to prepare the location where the software is to be installed.

- To use a file system from a previous installation that is already mounted, rather than creating a new file system, you can comment out the job steps that create and mount it. However, you must still run the other job steps, in order to set up the directories.
- Adapt the job (either C2EZAUSR, C2EZAUSA, or C2EZAUST) to specify the names of the user and group you selected and specify a unique UID/GID. For ACF2, also specify all fields that are relevant for the UID-string on the INSERT statement for the logonid.
- The default location is `/usr/lpp/c2e/V8R0M0`. You can choose a different location; for example, `/opt/c2e/V8R0M0`. You can also remove the release from the path name, or add a maintenance level to the path name. During the setup process, the installation directory is referred to as `C2ESW`.

  The supplied job C2RZCZFS uses `/usr/lpp/c2e` as the mount point to accommodate additional zSecure software that might also use this file system. The remaining parts of C2ESW are created by job C2EUNPAC.
- The job mounts the file system and must therefore run as root, but the file system does not remain mounted after a subsequent IPL. Be sure that the file system will be mounted when needed. For example, you can include the mount in your `BPXPRMxx` member.

For an upgrade installation, you normally do not have to prepare a new file system. However, create a new directory for unpacking the software so that you can use the upgraded software easily by changing the C2ESW parameter in the configuration and restarting the Agent.

Uploading the pax file

Upload file ibm.tsiem.actuator.pax.Z (or the C2EPAX.Z file if you are using the precursor product, Tivoli Compliance Insight Manager) from the zOS directory on the Tivoli Security Information and Event Manager CD into an HFS or zFS file on your z/OS system. On the z/OS system you can select any path name because, after unpacking, the file is no longer needed.

If uploading directly into an HFS or zFS file on your system is not possible or not allowed, you can transfer into an OS data set. This data set must be pre-allocated and sequential, and it can have record format FB, record length 80, and any block size that your system supports.

Be sure to transfer in **binary** mode.

Unpacking the software

Job C2EUNPAC in the installation library is supplied to unpack the Agent software.

- The job can run either under the user and group that are to own the files that will contain the software (through SURROGAT), or it can run under a user that has uid=0:
  - When using SURROGAT, specify USER=C2RUSER,GROUP=C2RGROUP or the values that you chose for file ownership in the JOB statement. In this case, you can skip the CHOWN job step. For information about file ownership selections, see "Owners, directories, and file systems" on page 164.

    On ACF2 systems, you might want to use JOBFROM. However, using JOBFROM is not a good idea. The JOBFROM attribute is much more powerful than is required: the user with JOBFROM can submit jobs to run under any logonid. With the appropriate SUR resource rules, you can use the SURROGAT method. (The appropriate rules are provided with the local CLASMAP entry for SURROGAT. By default the CLASMAP translates SURROGAT into SUR, but the local system programmer or ACF2 administrator might have chosen any other three-character combination for the CLASMAP.)

    A LID must have access to all necessary OMVS groups by means of R-TGR resource rules. The default OMVS group for the user is also derived from the GROUP field of the LID record.
  - To ensure that the files are not root-owned when a user is running as root (a user with uid=0), specify your substitutions for C2RUSER and C2RGROUP in the CHOWN job step, and make sure to run that job step.

    Note: A non-root user who can use su to act as root through the FACILITY/BPX.SUPERUSER permission does not work. In this case, a user with uid=0 is required.
- If you transferred the pax file into an OS data set, rather than an HFS or zFS file, uncomment job step OS2ZFS and supply the name of the uploaded data set.
- Grant the user that runs C2EUNPAC access to the pax file. To run job step OS2ZFS, you must have WRITE access; otherwise READ access is sufficient.
- Set PAXFILE to the path to which you uploaded the pax file, and C2ESW to the location where the Agent software is to reside. Path names are case-sensitive.

After successful completion of this job, you can delete the pax file and, if applicable, the OS data set that was input to job step OS2ZFS. These files are no longer be needed.

Preparation of a new Agent

To prepare the new Agent for Tivoli Security Information and Event Manager, use the instructions in the following sections:

- "Parameters for the z/OS Agent"
- "Preparing the Agent-running userid and its workspace"
- "Preparing the Agent root" on page 169
- "Starting and stopping the Agent" on page 169
- "CARLa members that support adding CARLa statements (optional)" on page 171
- "Strategies for collecting SMF event data" on page 172

Parameters for the z/OS Agent

Uncomment and, if necessary, edit the actuator-specific parameter section in your zSecure configuration (default `C2R$PARM`). Parameters specific for the z/OS Agent are `C2ECUST`, `C2EPATH`, `C2ESW`, `C2ELVPFX`, and `C2ELVLLQ`. The parameters are documented in Appendix D, "Configuration parameters," on page 213.

If you intend to run multiple Agents, each Agent needs its own C2EPATH and C2ELVPFX parameters; therefore, in principle you must have a separate zSecure configuration for each Agent. However, see "Same Agent definition on multiple images" on page 184. Multiple Agents can share the C2ESW parameter if they run the same level of the Tivoli Security Information and Event Manager software.

Preparing the Agent-running userid and its workspace

About this task

During an upgrade installation, you usually do not need to prepare the userid and its workspace. Instead, be sure that you do not inadvertently change the UID or GID of an existing user or group, because existing data would then be inaccessible. When migrating from the OS/390® to the z/OS Agent, read this section carefully before deciding on the required actions.

All processes that constitute a single Agent must run under a single RACF userid, ACF2 logonid, or Top Secret acid. These processes include starting, stopping, and first-time connect. Multiple Agents can run under separate userids or under the same userid, but each Agent must have its own AgentRoot.

zSecure provides three different jobs to create the Agent-running user and group, home directory, and optional file system:

- C2EZAUSA for ACF2
- C2EZAUSR for RACF
- C2EZAUST for Top Secret

These jobs grant the Agent-running user access to the software through the software-owning group, C2RGROUP by default. They also set ownership and umask, thereby granting group C2EGROUP READ access to log files that the Agent creates. Include your support staff in the C2EGROUP. To review the default values for other settings, see "Owners, directories, and file systems" on page 164.

Procedure

To prepare the Agent:

1. Adapt the job for the application being installed to specify the names of the user and group you selected and specify a unique UID/GID. For ACF2, specify all fields that are relevant for the UID-string on the INSERT statement for the logonid.
2. If necessary, adjust the ASSIZEMAX value (maximum address space size) depending on the amount of SMF data per chunk.

   To measure the data, run CKRCARLA in a batch job using SCKRSAMP library member C2ELES against a typical amount of SMF data and allow for a margin. The change does not take effect until the Agent is stopped and started.
3. Set up the Agent-running user with the access permissions in the following list.

   Note: It is suggested that you grant these permissions by connecting the userid to the appropriate group or groups. For example, connect the user who runs the Agent process to the group that owns the Agent software.

   - ALTER access to data sets &C2ELVPFX..** so that the Agent can clean up these data sets after processing them. This data set mask also includes the Agent's dedicated CKFREEZE data set.
   - Unconditional READ access to the input data sets specified or implied on the Management Console.

     The Agent must have READ access to all data sets that are explicitly specified.
   - If the Agent contains an ES that has no SMF specified, then READ access to the live SMF is required.

     The live SMF can be the MANx data sets or (starting with z/OS 1.9) the SMF log stream, accessed through the LOGR subsystem.
   - If the Agent contains a UIS that has no security database specified, then READ access to the primary RACF database or backup ACF2 database is required.
   - If the Agent contains a UIS that has no CKFREEZE data set specified, then the Agent maintains (reads and writes) its own dedicated CKFREEZE data set, &C2ELVPFX..CKFREEZE&C2ELVLLQ. This data set is already covered by the previous bullet point.
   - If the Agent contains a UIS that has no CKFREEZE data set specified, the Agent must also have permission to run zSecure Collect in order to maintain the &C2ELVPFX..CKFREEZE&C2ELVLLQ data set. For information, see the *User Reference Manual* for your zSecure product.
   - READ access to the data sets where your zSecure configuration and your license reside.
   - READ access to resource CKR.CKRCARLA.APF and CKR.READALL in the XFACILIT class. This might be a different class, if you configured it in that way. See Appendix A, “Site module,” on page 199.
   - If the PROCACT class is audited, the Agent userid must have READ access to the UNIXPRIV resources SUPERUSER.PROCESS.GETPSENT and SUPERUSER.PROCESS.KILL.

Language Environment runtime options

If the Language Environment® runtime options on your system are different from the default options, system errors can occur. To avoid this problem, check your system Language Environment runtime options, and specifically the ALL31, HEAP

and STACK options. The default values for these options are either specified in parmlib member CEEPRM*xx* or customized as described in the *z/OS Language Environment Customization* manual.

You can also check the options by including the following Language Environment option in the .profile file of the UNIX user that runs the Agent:

```
export _CEE_RUNOPTS="RPTOPTS(ON)"
```

Be sure that the Language Environment options in effect for the ALL31, HEAP and STACK are:

```
ALL31(ON)
HEAP(32K,32K,ANYWHERE,KEEP,8K,4K)
STACK(64K,64K,ANYWHERE,KEEP,512K,128K)
```

If you cannot change the Language Environment options to the defaults, you can override these options at run time. Add the following line to the .profile file of the UNIX user that runs the Agent:

```
export _CEE_RUNOPTS="ALL31(ON) HEAP(,,ANYWHERE,) STACK(,,ANYWHERE,)"
```

Preparing the Agent root

About this task

Job C2EAROOT in the SCKRSAMP library is supplied to create and prepare the Agent root directory and some OS data sets that the Agent needs. Before submitting this job, complete these tasks.

Procedure

- Change the USER parameter in the JOB statement to the user that is intended to run the Agent process.
- In the JCLLIB and INCLUDE statements, specify the zSecure configuration that contains your C2EPATH and C2ESW parameters.
- If you intend to use a LIVE User Information Source, adjust the size of the CKFREEZE data set. See "Event Source and User Information Source properties" on page 186. Typically, the required size varies by about 2 MB for each online DASD volume. If you are certain that you will never use a LIVE UIS, you can remove this allocation.

Starting and stopping the Agent

Before you begin

Warning: Do not start the Agent job or procedure until you have run the full set of actions as described in "Secure connection setup" on page 177.

Also, make sure that none of these processes is started before OMVS is fully initialized; that is, before the system issues message BPXI004I OMVS INITIALIZATION COMPLETE. Starting the processes before OMVS is initialized might fill your zFS file system with error messages.

About this task

The following procedures and jobs are supplied to start and stop the Agent:

Table 18. Procedures and jobs for starting and stopping the Agent

Job or Started Task	Member in SCKRPROC	Member in SCKRSAMP	When typically run
Agent starter	C2EAUDIT	C2EJSTRT	Soon after IPL, but after start of OMVS
Agent stopper	C2ECSTOP	C2EJSTOP	Only in preparation of a system shutdown for next IPL, or for maintenance

Whether you run processes as batch jobs or started tasks, ensure that you do the following:

- Update the `CONFIG=C2R$PARM` in the `EXEC` or `PROC` statement to reflect the zSecure configuration that you prepared for this Agent. For a started task, consider using a system symbol as part of the configuration member name.
- Make sure that you leave the `TIME=NOLIMIT` specification in the JCL in place. The Agent starter is a short-lived process. However, the Agent itself runs in a forked process for which the `MAXCPUTIME` value in the `BPXPRM`*xx* member has no effect. The CPU time limit is inherited from the parent.

Do not stop the Agent by canceling it or its parent process. Canceling one of these processes can result in the loss of z/OS events that are being sent to the Tivoli Security Information and Event Manager Server. To shut down the Agent gracefully, use the C2ECSTOP procedure or the C2EJSTOP job. The Agent can then flush its data properly.

Depending on the value for the `C2XEXITS` parameter in your configuration, the Agent starter also establishes the IBM Security zSecure-supplied exit `C2XPWX01`. For information about this exit and how to use it, see "Setup of the RACF Exit Activator" on page 43, and the *User Reference Manual* for your zSecure product. If you are using an ACF2 or Top Secret system, the parameter `C2XEXITS` has no effect.

For additional information about the `C2XEXITS` parameter, see Appendix D, "Configuration parameters," on page 213.

Procedure

- For processes that you want to run as a started task, copy the appropriate SCKRPROC member to a started task library connected to your JES2 or JES3. If desired, you can adjust the member name for your copy. Under ACF2, copy to names that do not match the intended logonid. Then, ACF2 will search for a GSO STC record matching the procedure name, and use the logonid specified there. This logonid does not require the STC attribute. An advantage of using GSO STC records is that you can run multiple STCs under the same logonid, which automatically fulfills the requirement for the same (UNIX) UID/GID. Also, not requiring the STC attribute, in combination with SUR resource rules, allows you to run batch jobs (for example, C2ECNNCT) under the same logonid.

  Also, copy your zSecure configuration member into a started task library, because it is included by a `JCL INCLUDE` statement.

  In order to have the started tasks run under the desired userid, create one or more `STARTED` profiles for RACF), GSO STC records for ACF2, or definitions in the STC record for Top Secret.

For example, the commands are:

– In RACF:

```
RDEF STARTED C2E* OWNER(SYS1) STDATA(STUSER(C2EAUDIT))
```

– In ACF2:

```
SET C(GSO) INSERT STC.C2EAUDIT GROUP(....) LOGONID(C2EAUDIT) STCID(C2E-)
```

The group parameter is optional and, if specified, overwrites the group field in the logonid record.

– In Top Secret:

```
TSS ADD(STC) PROCNAME(C2E*) ACID(C2EAUD)
```

These commands assign a single userid, logonid, and acid to all of the started tasks that the Actuator consists of, as required in "Preparing the Agent-running userid and its workspace" on page 167. If your installation standards do not allow for generic or masked profiles, records or definitions, you must issue specific commands for each of the started tasks you want to use. In that case, make sure that they all assign the same userid, logonid, and acid.

See your RACF, ACF2, or Top Secret documentation for more information about how to create these profiles, records, or definitions.

- For processes that you want to run as batch jobs, copy the appropriate SCKSAMP member from Table 18 on page 170 into a data set of your own. Consider copying to a data set that contains jobs for your job-scheduling software. If you like, you can rename the jobs.

  Make sure that the jobs run under the userid you have selected by doing both of the following actions:

  – Specify the userid in the `JOB` statement.
  – Create a SURROGAT profile for RACF, the ACF2 equivalent, or a cross-acid permit for Top Secret. When you create one of these profiles or permits, your job-scheduling software can submit under the specified userid.

CARLa members that support adding CARLa statements (optional)

You can supply your own CARLa statements for processing at specific points within the z/OS Agent. To add your own CARLa, store it in the data set that is identified by the C2ECUST configuration parameter. See Appendix D, "Configuration parameters," on page 213. Concatenations are not supported.

The supported members are:

C2EI0ES

CARLa that is to be processed at the start of ES-Collect. You can use this member, for example, when IBM or vendor software writes badly formatted SMF records that cause errors during ES-collect. In this case, IBM Software Support can supply a set of CARLa statements to use until the problems with the OEM-vendor are solved, or a more permanent solution is built.

You can also exploit this member to specify the right CCSID when translating, for example, double byte character set (DBCS) characters to UTF-8. For instance, for Japanese Latin extended Unicode, you can include a CARLa statement:

```
OPTION MY_CCSID=1399
```

If you use this option:

- Be sure that you also specify the option in C2EI0UIS for the User Information Source.
- Be sure that the fix for Tivoli Security Information and Event Manager APAR IZ84905 is installed on your server when using this option.

C2EI0UIS
CARLa that is to be processed at the start of UIS-Collect. You can use this member, for example, for `SIMULATE SENSITIVE` statements, or to add installation-specific groups to the UIS output.

You can also exploit this member to specify the right CCSID when translating, for example, DBCS characters to UTF-8. For instance, for Japanese Latin extended Unicode, you can include a CARLa statement:

```
OPTION MY_CCSID=1399
```

If you use this option:

- Be sure that you also specify the option in C2EI0ES for the Event Source.
- Be sure that the fix for Tivoli Security Information and Event Manager APAR IZ84905 is installed on your server when using this option.

C2EIXES
CARLa EXCLUDEs for the SMF selection during ES-collect.

C2EICES
CARLa to customize the event source to also map installation-identified events. For example, you might have a product in place with its own SMF records.

Additional members can be used with `INCLUDE DD=C2ESAMP MEMBER=`*`xxxx`*. Changes in the C2ESAMP library are effective for subsequent ES- or UIS-collects; there is no need to restart the Agent.

Strategies for collecting SMF event data

An event source (ES) can be configured with an INTERCEPT, LIVE, POLL, or WAIT strategy. These configuration parameters are described in "Event Source and User Information Source properties" on page 186. Under the POLL and WAIT strategy, the actuator reads the SMF data from the specified data set. Usually, this data is your accumulated SMF data that is already in place.

Under the LIVE strategy, SMF data is collected from the active data set where SMF writes its records. This data set is usually a `MAN`*`x`* data set. However, if the `MAN`*`x`* data set is offloaded before the ES collection is run, SMF records are no longer available in that `MAN`*`x`* data set.

To prevent the ES actuator from missing those SMF records, the SMF Switch Intercept process is required. As part of the SMF Switch Intercept process, the offloaded records are saved in data sets dedicated to a particular Agent. During the ES collection process, the ES actuator automatically collects SMF records from these data sets in addition to the active `MAN`*`x`*data set. After successful completion of the ES collection process, the dedicated data sets are deleted.

Under the INTERCEPT strategy, ES-collect reads **only** the data sets that are created by the SMF intercept. The active SMF is not read; these records become available after they are offloaded (on the next SMF switch).

Starting with z/OS 1.9, it is possible to write SMF records to a log stream. Because the log stream easily accommodates several days of SMF records, it is no longer

necessary to offload SMF messages several times a day. If the ES actuator uses the LIVE strategy, you do not need the SMF Switch Intercept if *all* of the following conditions are true:

- The ES collection schedule is set for a frequency of at least once every 24 hours.
- SMF is directed to a log stream.
- The SMF log stream has a retention period of at least one day.
- The LOGR subsystem has been activated.

These requirements are different from the requirements when using `MANx` data sets. However, if these conditions are not fulfilled, you must implement an SMF offload process to save older SMF records using IFASMFDL, which is similar to the SMF Switch Intercept. You might have discarded the old offload process used for the `MANx` data sets. A similar process is now needed for the SMF log stream.

For an Event Source under POLL and WAIT strategy:

- The ES actuator uses the specified SMF data set. The easiest implementation is to use your already-in-place accumulated SMF data. On the Management Console, specify this data set as part of the Event Source properties.
- No SMF intercept is required. However, the intercept does no harm, so you need not remove the intercept when switching from LIVE to POLL or WAIT.
- There is no guaranteed way to prevent some SMF records from being read multiple times, especially when your SMF accumulation process uses DISP=MOD. The ES actuator reads and discards SMF intervals that have already been processed. Depending on the schedules for collection on the Tivoli Security Information and Event Manager server and for SMF offload on the z/OS system, there is also no guarantee against discarding some out-of-sequence SMF records.

For an Event Source under LIVE strategy:

- The ES actuator reads the active SMF data sets or the SMF log stream. This process is done under control of the collect schedule as configured on the Management Console.
- An SMF Switch intercept or SMF Offload process is required if:
  - The SMF records are written to `MANx` data sets.
  - The SMF records are written to a log stream, and your environment does not fulfill the requirements specified in the list.

Be sure that the SMF Switch Intercept is embedded in the SMF accumulation process that you already have in place. Normally this accumulation process is run in response to message IEE391A, or triggered by exit IEFU29. Embedding exploits the capability of the SMF dump program (IFASMFDP) to write the SMF records to multiple output files. Depending on how your SMF offload processing is currently set up, several methods exist to implement the SMF Switch Intercept.

The essential part of the SMF Switch Intercept is saving the offloaded SMF records. Saving is done in new data sets that are created for a particular Agent. At successful completion of the ES collection process, these data sets are deleted. Then, at the next SMF switch, another new data set is created. Because of this repeated creating and deleting, be sure that these data sets have a name that uniquely identifies them as belonging to a particular Agent. The following naming scheme is required:

```
&C2ELVFPX..SMF.**
```

The first qualifier must be the value specified for the configuration parameter &C2ELVPFX. The preferred method is to use JCL variables to ensure that the SMF Switch Intercept uses the same value as the ES actuator process. The second qualifier must be SMF. The remaining qualifiers must be chosen such that the names of the data set are created in ascending alphabetic order.

Note: Violating the required alphabetic order can result in loss of data when more than 100 intercept data sets are created between successive runs of ES-collect. If you use the standard System Symbols &YYMMDD and &HHMMSS, the order is automatically correct.

The variable &C2ELVLLQ can be used as the last qualifier to satisfy possible SMS allocation requirements. The ES actuator retrieves and subsequently deletes all data sets matching &C2ELPFX..SMF.**.

If your SMF offload runs as a started task, you can exploit system symbols in JCL. This implementation is the easiest. See "SMF intercept - allocation in JCL."

If system symbols, or other symbols to generate unique alphabetically ascending data set names, are unavailable, you can use JCL procedure C2ECLSMF, which uses REXX to allocate a data set with a unique name. This implementation is more complex. See "SMF intercept - creating a data set without system symbols" on page 175.

SMF intercept - allocation in JCL

If your SMF offload runs as a started task, you can exploit system symbols in JCL. In this case, you need only to add DD statements and control statements. For example, if your current offload process JCL is:

```
//OFFLOAD EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//DUMPIN   DD DISP=SHR,DSN=<SMF data set to be offloaded>
//DUMPOUT  DD DISP=MOD,DSN=<your existing offload data set>
```

you can change it to:

```
//OFFLOAD EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//DUMPIN   DD DISP=SHR,DSN=<SMF data set to be offloaded>
//DUMPOUT  DD DISP=MOD,DSN=<your existing offload data set>
//   INCLUDE  MEMBER=<your Agent's zSecure configuration>
//C2EA001  DD DISP=(MOD,CATLG),UNIT=SYSDA,
//            DSN=&C2ELVPFX..SMF.D&YYMMDD..T&HHMMSS.&C2ELVLLQ,
//            SPACE=(32760,(30000,30000))
//ENQ001   DD DISP=OLD,DSN=&C2ELVPFX..ENQ&C2ELVLLQ
//SYSIN    DD DISP=SHR,DSN=<file with control statements>
```

The required control statements are:

```
OUTDD(DUMPOUT,TYPE(000:255))
OUTDD(C2EA001,TYPE(000:255))
```

Note: For a started procedure, your zSecure configuration must reside in a standard procedure library of your JES, because a started procedure does not support JCLLIB. For a started job, you can use JCLLIB; be sure that it points to the data set that contains the zSecure configuration for your Agent.

This scheme also applies if your job scheduling system allows you to use variables in the JCL of batch jobs. System symbols or other variables are required to ensure unique data set names are created in ascending alphabetical order. Use of a

Generation Data Group (GDG) is possible, but do not use a GDG because GDG serialization problems can result in allocation failures and JCL errors for your SMF offload processing.

You must specify sufficient space for the intercept data sets. For IFASMFDP, specify space that is approximately equal to the size of the SMF data set that is being offloaded. For the CBT program SMFDUMP, the required space might exceed the size of the combined SMF data sets, because SMFDUMP will cause an additional SMF switch in order to capture the latest SMF records. See http://www.cbttape.org for information about CBT programs. Note that SMFDUMP requires DISP=MOD because it invokes IFASMFDP multiple times, and each invocation opens the output data sets. For the same reason, do not use RLSE if you use SMFDUMP.

The `ENQ001` DD statement, as shown in the example, is required to serialize the offload process with respect to the ES actuator. Without this DD statement, there is a slight chance that the actuator will fail to allocate a data set that the intercept just created but has not yet deallocated. However, under a LIVE strategy the Agent also retrieves SMF records that have not yet been offloaded; this retrieval causes the data in the just-created intercept data set to be ignored on the next collect.

If you want to use multiple Agents within a single z/OS image, complete the following steps:

- Provide an INCLUDE for the zSecure configuration for the corresponding Agent, in order to use the correct values for &C2ELVPFX and C2ELVLLQ.
- Supply a `C2EAnnn` output DD statement and an `IFASMFDP` control statement for each Agent.
- Supply an `ENQnnn` DD statement for each Agent as shown in the example.

Be sure that each set of `C2EAnnn` and `ENQnnn` DD statements is immediately preceded by the corresponding `INCLUDE` statement, becauseJCL uses the `INCLUDE` statement to substitute the required parameters into each DD statement.

SMF intercept - creating a data set without system symbols

If system symbols, or other symbols to generate unique alphabetically ascending data set names, are not available, you can use JCL procedure C2ECLSMF. This procedure uses REXX to allocate a data set with a unique name. A fixed alias is created because IFASMFDP must refer to the data set through JCL. The fixed alias poses no problem: it is only used during SMF Switch Intercept, and only one SMF Switch Intercept is active at a time. Using C2ECLSMF has the additional advantage that it saves space and processing when it detects that the ES actuator does not run under an INTERCEPT or LIVE strategy. With C2ECLSMF, your SMF offload looks similar to the following example:

```
//*  SMF intercept for Tivoli Security Information and Event Manager.
//*
//*  Make data sets with configuration(s) and procedure available
//    JCLLIB   ORDER=(your.prefix.CKRPARM,CKR.SCKRPROC)
//*
//*  Create unique data set for Agent. Also create alias for the
//*  INTCEPT jobstep and, if LIVE, write control statement.
//    INCLUDE  MEMBER=<your Agent's zSecure configuration>
//ACTUATR1 EXEC C2ECLSMF,OUTDD=C2EA001,SP='10 100'
//*
//*  In case of (allocation) failures, revert to offload-only in
//*  order to prevent full SMF data sets.
//OFFLOAD EXEC PGM=IFASMFDP,COND=((0,EQ),EVEN)
//SYSPRINT DD  SYSOUT=*
//DUMPIN   DD DISP=SHR,DSN=<SMF data set to be offloaded>
//DUMPOUT  DD DISP=MOD,DSN=<your existing offload data set>
```

```
//*
//*  Enforce abend in order to have Intercept failure noticed.
//ABEND   EXEC PGM=ABEND806,COND=((0,EQ),EVEN)
//*
//*  Offload SMF concurrently into cumulative data, and into
//*  the data set(s) for Agent(s).
//INTCEPT EXEC PGM=IFASMFDP,COND=(0,NE)
//SYSPRINT DD  SYSOUT=*
//DUMPIN   DD DISP=SHR,DSN=<SMF data set to be offloaded>
//DUMPOUT  DD DISP=MOD,DSN=<your existing offload data set>
//SYSIN    DD DISP=SHR,DSN=&CPREFIX..SCKRCARL(C2EILSMF)
//         DD DISP=(OLD,DELETE),DSN=&&C2EAUD
//C2EA001  DD DISP=MOD,DSN=&C2ELVPFX..OFFLODNG&C2ELVLLQ
//ENQ001   DD DISP=OLD,DSN=&C2ELVPFX..ENQ&C2ELVLLQ
```

In the SP parameter, you specify the space (the primary and secondary number of megabytes) that applies for the INTERCEPT or LIVE strategy. The same considerations apply as in the JCL allocation scheme. Under the POLL or WAIT strategy, the SP parameter is not used. A small data set is created because the INTCEPT job step references it, but no data is written. The ES actuator will eventually clean up the data set.

Member C2EILSMF in the SCKRCARL library contains the control statements that keep your cumulative offload in place. If you already have control statements in place for the SMF accumulation (for example, to split your output), change the SYSIN DD statement to point to your own control statements. Keep the concatenated &&C2EAUD in place, as it contains the control statements generated by C2ECLSMF.

For multiple Agents within a single z/OS image:

- Invoke C2ECLSMF once for each Agent, with a unique OUTDD parameter. Be sure that each invocation is immediately preceded by an INCLUDE for the zSecure configuration for the specific Agent, in order to use the correct values for &C2ELVPFX and &C2ELVLLQ.
- Specify DD statements for each output in the INTCEPT job step. Be sure that each DDNAME matches the OUTDD parameter from the corresponding C2ECLSMF call.
- Supply a C2EAnnn output DD statement and an IFASMFDP control statement for each Agent.
- Specify unique ENQ*nnn* DD statements for each Agent. These statements serialize the offload process with respect to the ES actuator. Without the DD statement, there is a slight chance that the actuator will successfully allocate and delete a data set that the intercept just created (because its name does not occur further down the JCL, the INTCEPT job step accesses it through the alias). Because deleting the data set also deletes the alias, job step INTCEPT would then fail with a JCL error.
- Before each pair of <OUTDD> and ENQ*nnn* DD statements, INCLUDE the corresponding zSecure configuration.

For JES3, this scheme does not apply, because JES3 allocates data sets for all the steps before the job is started. Therefore, under JES3 use a scheme that exploits System symbols or JCL variables. You can still profit from the processing-saving scheme by:

- Adding, on each invocation of C2ECLSMF, a DD statement that creates data set &C2ELVPFX..SMF.D&YYMMDD..T&HHMMSS..&C2ELVLLQ. Use a ddname that matches the OUTDD parameter, so that C2ECLSMF will detect that the data set already exists.

- Using these same data sets with DISP=MOD in the INTCEPT job step, instead of the aliases.

Do **not** split this JCL into multiple jobs. Doing so breaks the serialization, with the result that the Agent might find and delete an Intercept-created data set before it is filled. Offload failures can result.

Secure connection setup

To set up the secure connection for the Agent for Tivoli Security Information and Event Manager, review the background information and instructions in the following sections:

- "Setting up the configuration file"
- "Initial connection to the server"
- "Full-function startup of the Agent" on page 178

Setting up the configuration file

Procedure

1. Configure the z/OS image as a z/OS event source. Follow the instructions in the Tivoli Security Information and Event Manager documentation and in "Event Source and User Information Source properties" on page 186.
   - Make sure that you use, for the z/OS side of the connection, the port you selected and enabled for the Agent.
   - Be sure that you know the exact name of the configuration file (for example, IPO1-INSIGHTDOMAIN.cfg).
2. Transfer this file, in text mode, into the Agent root directory.

   If FTP is unavailable on your z/OS system, you must use a different transfer method. For example:
   - You can use IND$FILE (**text** mode) to transmit into an OS-dataset, and then use OPUT to copy it into the actuator subdirectory.
   - You can also copy and paste the configuration data from a Windows file into a USS file using an editor such as oedit.
3. Make sure that the Agent userid can read the file. For example, you can use the Agent's userid and password for the transfer. If you transferred or edited while working under a root account, you must **chown** and **chmod** the configuration file.

Initial connection to the server

Before you begin

Adapt and run job C2ECNNCT to first set up the secure connection. Before running this job, you must perform these steps:

- Make sure that the job runs under a user with the same UID as the user that is intended to run the Agent process. The easiest way to do this is by using SURROGAT submission: set the USER parameter in the JOB statement to the user that is intended to run the Agent process, and make sure that SURROGAT checking grants you access.

  You **can** run C2EAROOT under a root userid. However, if you do so, you must also run C2ECNNCT as root, and you must run job C2EFLXFR.
- Specify, in the JCLLIB and INCLUDE statements, the zSecure configuration that contains your C2EPATH and C2ESW parameters.
- Specify the name of the configuration file you just transferred.

Procedure

To establish the initial secure server connection, perform these steps:

1. Run the C2ECNNCT job.
2. Browse the log file <agent-root>/log/agent.log to see whether initialization is ready. If initialization is ready, the following message is present in the file:

   ```
   LCM: Initial certification completed successfully
   ```

   Note: A configuration file is valid for only 24 hours before it expires. If it is not used to initialize the Actuator within that time period, complete the following steps:

   a. If needed, stop the Agent with procedure C2ECSTOP or job C2EJSTOP, as described in "Full-function startup of the Agent."
   b. On the Management Console, generate a new Install Password. This process is documented in the *Tivoli Security Information and Event Manager User Guide*. See the section about viewing or changing machine properties.
   c. Edit this new password in the configuration file that you transferred earlier.
   d. Rerun job C2ECNNCT.

Full-function startup of the Agent

Job C2ECNNCT establishes communication with the Server by starting a first-time, limited mode run of the Agent, but does not terminate it.

You must stop the Agent with job C2EJSTOP or started task C2ECSTOP. Stop the Agent under the same userid that you use to run C2ECNNCT. Normally, both C2ECNNCT and C2ECSTOP/C2EJSTOP should be run under the userid that you prepared (in job C2EZAUSR) to run the Agent under. However, if you submitted job C2ECNNCT as root user, without using SURROGAT (or JOBFROM), do the same for job C2EJSTOP.

Also, if you run job C2ECNNCT as root, then you must stop the limited mode Agent and then run job C2EFLXFR, also as root. This corrects the file ownership so that the full function Agent, running under its regular userid, has access.

Next, you can start the Agent as a regular production process with started task C2EAUDIT or job C2EJSTRT. The Agent is normally started soon after each IPL. See "Starting and stopping the Agent" on page 169.

You are now ready to collect logs at the Tivoli Security Information and Event Manager Server. Using the Management Console, select the GEM database you want to hold the z/OS logs and start loading the mainframe events, indicating that new information must be collected. Make sure that you have used a database that is large enough to handle the amount of data you hold in the mainframe bin.

Set up a schedule in the Management Console to collect z/OS logs once daily, or according to a schedule that you have chosen.

Make sure that you run at least one UIS collect before you run any ES collect. Otherwise, the ES collect fails because there is no UIS information available for enrichment.

Guidelines for upgrading an existing Agent

Before upgrading an Agent, first make sure that the following requirements are satisfied:

- The new level of the Agent software is unpacked into a directory different from the previous level.
- The Agent userid has the required access to the new directory and the files in the directory.
- The Agent is stopped. If needed, run job C2EJSTOP or started task C2ECSTOP, depending on your previous setup. Do not restart the Agent until after finishing the upgrade process.

Also, carefully review the JCL for the Agent.

- The new JCL (procedure C2EAUDIT, as of zSecure 1.8) has additional job steps for preparation and uses a new configuration parameter `C2ESW`. See Appendix D, "Configuration parameters," on page 213. The `C2ESW` parameter specifies the directory where the `ibm.tsiem.actuator.pax.Z` file (or the `C2EPAX.Z` file if you are using the precursor product, Tivoli Compliance Insight Manager) is unpacked. This parameter is required.
- Before zSecure 1.8, DD statements `C2ESAMP`, `C2EI0ES`, `C2EI0UIS`, `C2EIXES`, and `C2EICES` in the C2EAUDIT JCL were optional. They were only needed when using installation-supplied CARLa. Starting with zSecure 1.8:
  - The zSecure configuration that is used by the Agent must contain the `C2ECUST` and `C2ESW` parameters. See Appendix D, "Configuration parameters," on page 213. The default is *your.prefix*`.CKRPARM`.
  - DD statement `C2ESAMP` is required in the C2EAUDIT JCL, even if you do not use zSecure-supplied CARLa. This DD statement is supplied in the SCKRPROC data set provided with zSecure 1.8. If you copied the C2EAUDIT procedure (to SYS1.PROCLIB to run as a started task, for example), you must add the `C2ESAMP` DD statement to the C2EAUDIT JCL, or copy the C2EAUDIT procedure again.
  - The data set identified by the `C2ECUST` parameter must exist, and it must contain members `C2EI0ES`, `C2EI0UIS`, `C2EIXES`, and `C2EICES`. By default, comment-only members, copied from the SCKRCARL library, are supplied.
  - DD statements `C2ESAMP`, `C2EI0ES`, `C2EI0UIS`, `C2EIXES`, and `C2EICES` are ignored. If you share the JCL with a zSecure 1.7 Agent, and have zSecure-supplied CARLa statements, the 1.7 Agent still uses them. See the 1.7 manual for information.

For a first-time installation, the listed defaults are set up by the installation process. However, to avoid overwriting your customized data, the installation process does not update an existing configuration data set. If you use an existing configuration data set, or if your `C2ECUST` parameter addresses a different data set, copy the required `SC2RSAMP` members yourself.

If your previous Agent runs with the z/OS Agent, available starting with version 6 of Consul InSight Security Manager, you can upgrade (or fall back) by changing the `C2ESW` parameter and restarting the Agent.

When upgrading from zSecure 1.8.0 or earlier, and using a LIVE User Information Source, also rename the IOCONFIG data set, as created by job C2EAROOT, into a CKFREEZE data set. That is, change the low-level qualifier. Do not use an alias to allow fallback without renaming back because of the different serialization behavior of aliases and true names.

If your old Agent runs with the OS/390 actuator, it has, in the directory identified by `C2EPATH`, a true subdirectory `bin` rather than a symlink. This `bin` subdirectory contains the old actuator software. To keep your fallback path available, this directory is not automatically removed during the upgrade. If you upgrade from the OS/390 actuator, you must remove or rename the `bin` directory yourself, in order for the Agent to establish a symlink.

In addition:

- In the z/OS Agent, the SMF filter and the Environment collector no longer run as independent batch jobs or started tasks. Instead, they run as USS processes, or (partly) as part of your regular SMF-accumulation jobs. In this way, processing is better synchronized. Remove procedures C2ECSMF and C2ECUTIL, and jobs C2EJSMF and C2EJUTIL from your production processes.
- The directory named `actuator` is no longer a mandatory part of the path names and has been removed from procedures C2EAUDIT, C2ECSMF, and C2ECUTIL. Instead, you can run multiple Agents under the same userid with each Agent using its own subdirectory in the home directory. Because your old Agent does contain the `actuator` directory, you must change the `C2EPATH` parameter of your configuration to include `actuator`. That is, change the parameter value from `'/u/c2eaudit'` to `'/u/c2eaudit/actuator'`.
- After starting the Agent, access the Tivoli Security Information and Event Manager Management Console:
  1. For all UIS Properties for the particular system (Point of Presence):
     a. Make any true update.
     b. Click OK.
     c. Open again.
     d. Back out the update.
  2. If any ES on the system runs under a LIVE strategy, enforce Collect for this ES.

  These actions refresh the Properties on the Agent side. The refresh is compatible; no particular action is required when fallback to OS/390 Actuator is required.

Reinstalling and uninstalling the Agent

About this task

You can reinstall or uninstall the Agent software and the Agent process and its data separately.

For the Agent software, reinstalling or upgrading can be done without first uninstalling. For best results, unpack the pax file into a new, clean directory. A dedicated file system is optional, depending on your policy. When installing into a new directory, you can activate it by stopping the Agent, changing the C2ESW parameter, and restarting the Agent. When not using a new directory, you must stop the Agent before unpacking the pax file.

Procedure

- You can reinstall an Agent process without reinstalling the Agent software. For instance, you can set up a new Agent in addition to the existing one (connected to a different Server). Or, reinstalling is required when the IP address of the Tivoli Security Information and Event Manager Server has been changed, or when the data in the Agent root directory is corrupted and cannot be recovered.

1. Clear and remove the `run` and `log` subdirectories, but not the `bin` subdirectory.
2. Rerun job C2EAROOT.
3. Create a new initial password on the Management Console. Edit this password into the configuration file that you used during initial connect, and rerun job C2ECNNCT.

   If you run C2ECNNCT as root, then also run job C2EFLXFR.

Alternatively, you can create a new configuration file on the Management Console.

- If you want to completely remove an Agent process, or the Agent software, follow these steps:
  1. Remove the SMF intercept. If the Agent no longer runs, the intercept data sets will no longer be removed.
  2. Stop the Agent processes that are no longer required. That is, run C2EJSTOP/C2ECSTOP for each Agent's zSecure configuration.
  3. Remove the Agent root directory (`C2EPATH`) and all files in the directory.
  4. If you also want to remove the software, remove the software directory (`C2ESW`) and all files in it.
  5. If you used dedicated file systems, you might want to remove these file systems also.
  6. Delete the started task and jobs created, and remove all references to them from your job-scheduling and automated operations processes.
  7. Remove any references to these processes in your security database (for example: users, permissions and `STARTED` profiles).
  8. Delete the platform instance in the Management Console.

Performance and multi-image considerations

For best results, use a setup such as the following one:

- Separate Agents on each z/OS image that you want to monitor.
- INTERCEPT strategy for the Event Source, with a schedule as frequent as needed, corresponding to your requirement to have events available on the Server in a reasonable amount of time.
- If you want to have events available in the Server as soon as possible, or if INTERCEPT is not available in your level of Tivoli Security Information and Event Manager, use LIVE, rather than INTERCEPT.
- If the Agent for Tivoli Security Information and Event Manager is the only component of IBM Security zSecure that you use, use a LIVE strategy for the User Information Source (UIS). A collect schedule of once a day is sufficient for most cases.
- If you use other components of IBM Security zSecure (for example, zSecure Admin or zSecure Audit for ACF2), use a POLL or WAIT strategy for the User Information Source. In the UIS Properties:
  - Specify the CKFREEZE data set that you already use for other zSecure components. See “Use of a fresh CKFREEZE and UNLOAD each day” on page 42.
  - Do not specify an UNLOAD data set. Specifying this data set causes the active database to be used, so that newly created users are immediately known during the next ES-collect.

- An Agent can process collected data only from a system that has the same security product (RACF, ACF2 or Top Secret) active as on the system where the Agent is running.

Considerations for this setup are explained in the sections that follow.

Single z/OS image

Actuator performance of an Event Source depends to a large extent on your existing SMF accumulation practice. SMF offload/accumulation is usually done by appending the output of IFASMFDP to an existing data set. As a result, the cumulation data set grows over time, until a new data set is taken into use, often once a week or once a month. Under a POLL or WAIT strategy, this cumulation data set is read during each Event Source collect run. Records that have already been processed are automatically filtered away, but only after being read.

Performance worsens when the cumulation is written as a multi-volume tape data set. When appending (`DISP=MOD`), z/OS needs to mount only the last volume. However, reading this volume results in all volumes being mounted, although normally only the last volume contains new SMF records. (If a volume switch occurred during the most recent SMF offload, other volumes might contain new SMF records.) Mounting all volumes wastes processing time, and with a frequent collect schedule the contention on your tape drives can become intolerable.

A LIVE or INTERCEPT strategy for your Event Source does not suffer from this effect, because it never reads the SMF accumulation tapes. An Event Source under a LIVE strategy directly reads the live SMF data sets. It also receives data from the SMF switch intercept; otherwise, the most recent SMF records (after ES-collect and before SMF-switch) would be missed.

However, note that all SMF records that are read from the live SMF data sets will also be read (after the next SMF switch) from the intercept data sets. Therefore, the INTERCEPT strategy saves you processing, at the cost of a somewhat longer delay, depending on the size of your SMF data sets, the frequency of SMF switch on your system, and your ES-collect schedule.

A Collect schedule that is far more frequent than is your SMF switch (combined with a LIVE strategy, in order to have very recent events in Tivoli Security Information and Event Manager) also reads data multiple times. You can reduce the amount of multiple reading by increasing your SMF switch frequency; for example, you can make several small `SYS1.MAN`*x* data sets. However, you must balance this against more tape-mounts for the accumulation process, as an example. Also, do not collect your Event Source more frequently than specified by your SMF interval.

Note: For best results, be sure that your z/OS system has SMF interval recording active, and that the Collect schedule (which you specify for each Event Source and UIS on the Management Console) allows for at least an entire SMF interval between two Collect runs. In this way, it is assured that events generated by long-running jobs can be tracked to the job they pertain to.

For the User Information Source, the main consideration is whether you use other zSecure components, such as zSecure Admin or zSecure Audit for ACF2. If you use any of these components, you probably have already set up for a daily refreshed

CKFREEZE and UNLOAD. See “Use of a fresh CKFREEZE and UNLOAD each day” on page 42. There is no reason to maintain another CKFREEZE and UNLOAD.

If the Agent for Tivoli Security Information and Event Manager is the only component of IBM Security zSecure that you use, the easiest setup is a LIVE UIS.

Multiple z/OS images

When processing multiple z/OS images, the preferred setup is as follows. Each image has its own (single-image) Agent, processing SMF, CKFREEZE, and the security database from that image. However, when most of the DASD is shared, a performance gain can be achieved by not writing all shared information to all CKFREEZEs:

- On one of the z/OS images, the "normal" full-size CKFREEZE is maintained. This can be done by the Agent itself (LIVE), or it can be done by the regular C2RJPREP job. See “Use of a fresh CKFREEZE and UNLOAD each day” on page 42.
- For the image where the "normal" CKFREEZE is created, specify this CKFREEZE dataset name on the Management Console. Alternatively, use a strategy of LIVE and do not specify a CKFREEZE.
- For each of the other images:
  - Create and periodically refresh a reduced CKFREEZE. This can be done with job C2RJPREP, by specifying `OPTCOLL='SHARED=NO'`.
  - On the Management Console, specify two User Information Sources. One UIS specifies the reduced CKFREEZE that pertains to that particular image. The other UIS species the "normal" CKFREEZE, which contains all information about the shared DASD.

You might need to specify system affinity or NJE routing to ensure that each job runs on the intended image.

In this way, each Agent has a CKFREEZE that was created under the same image as the SMF-records that it is to process. And most UIS information (that is, the part that pertains to the shared DASD) needs to be created only once.

If your current SMF offload from multiple z/OS images writes into a common accumulation data set, and you use this data set as input for your Event Source, you have effectively a multi-image Agent. Because UIS data (UNLOAD or active security database, and CKFREEZE) from all these images is required during ES collection, add multiple User Information Sources to this Agent through the Management Console:

- If the foreign images share the security database with the image where the Agent runs, you must specify each UIS without UNLOAD. In that case, the actuators will use the live database, even for a POLL or WAIT UIS.
- If the foreign images do not share their security databases with the image where the Agent runs, you should specify an UNLOAD on each of the corresponding UISes.
- For each foreign image, you must specify a CKFREEZE in the corresponding UIS.
- For each CKFREEZE or UNLOAD that you specify on the Management Console, you must make sure that it is refreshed periodically; for example, with job C2RJPREP. See “Use of a fresh CKFREEZE and UNLOAD each day” on page 42. Also, a CKFREEZE must be created on the image that it pertains to.

- All required data must be accessible from the image where the Agent runs. In practice, this means that shared DASD is required.

A multi-image Agent usually requires more processing than several single-image Agents. More processing is required because each ES collect references all UIS data. That is, CKFREEZEs, and possibly UNLOADs from all images are processes during each chunk of SMF-collection.

In general, do not use multi-image Agents. If you run with a common SMF accumulation data set and do not want to split it, consider setting up a single-image Agent on each z/OS image and use INTERCEPT or LIVE ES. In this way, each Agent processes only its own image's SMF. There is no restriction against combining the INTERCEPT or LIVE ES with a POLL or WAIT UIS.

An Agent can process collected data only from a system that has the same security product (RACF, ACF2 or Top Secret) active as on the system were the Agent is running.

Same Agent definition on multiple images

The userid assigned to the Agent started task can be used by more than one Agent. However, each Agent must have a unique root path (C2EPATH) and a unique data set prefix (C2ELVPFX) assigned to it. If your installation requires one physical home directory for each RACF userid, you can create unique versions of the "actuator" directory for each Agent in the Agent's (shared) home directory; for example, `/u/c2eaudit/actuator.IPO1` and `/u/c2eaudit/actuator.IPO2`. Similarly, the system ID can be used as the second qualifier of the MVS work dataset names.

Assignment of C2EPATH and C2ELVPFX is performed in the zSecure configuration member specified in the C2EAUDIT started task JCL. There are several ways to use system symbols to achieve unique qualifiers for each Agent. For example, you can build a separate zSecure configuration member for each image and use the &SYSCLONE or &SYSNAME symbol as part of the member name (for example, in C2EAUDIT). See the following example:

```
// SET CONFIG=C2EPRM&SYSCLONE
// INCLUDE MEMBER=&CONFIG
```

For image IPO1 with SYSCLONE value T1, build a member C2EPRMT1 that includes the standard member and sets overrides:

```
// INCLUDE MEMBER=C2R$PARM
// SET C2EPATH='/u/c2eaudit/actuator.IPO1'
// SET C2ELVPFX=C2EAUDIT.IPO1
// SET C2ELVLLQ=
```

This approach requires a separate zSecure configuration member for each image, which must be accessible in the PROCLIB concatenation.

It is also possible to code the logic in one CKRPARM member, using the SYSNAME symbol, and include it from C2EAUDIT:

```
// SET CONFIG=C2E$PARM
// INCLUDE MEMBER=&CONFIG
```

Member C2E$PARM uses the standard assignments from C2R$PARM and works around the limitations of quoted strings and system symbols:

```
// INCLUDE MEMBER=C2R$PARM
// SET QUOTE=''''
// SET C2EPATH=&QUOTE./u/c2eaudit/actuator.&SYSNAME.&QUOTE
// SET C2ELVPFX=C2EAUDIT.&SYSNAME
// SET C2ELVLLQ=
```

System symbols are unavailable in normal batch job JCL; therefore, in the installation jobs and other batch jobs, you must set the values yourself:

```
/*ROUTE XEQ IPO1
// SET SYSNAME=IPO1
// INCLUDE MEMBER=C2E$PARM
```

Also, be aware that most symbols resolve to uppercase. If your path names are lowercase, you must create additional symbolic links.

Switching between LIVE and POLL or WAIT

If you want to switch an Event Source from POLL or WAIT to LIVE, first make sure that the Agent has collected the latest SMF records from your accumulated SMF data set. Otherwise, collecting recent SMF causes older data from your accumulation to be discarded. If needed, start the collect process from the Management Console immediately before switching to LIVE.

When switching an Event Source from LIVE to POLL or WAIT, no special actions are required. The Server is expected to contain recent SMF data because the Event Source has been LIVE. The Agent removes any data sets that the SMF intercept created earlier. All subsequent SMF records are read from your SMF accumulation.

When switching a User Information Source to LIVE, verify that you did not remove the creation of your CKFREEZE data set from job C2EAROOT, and verify its size. See “Preparing the Agent root” on page 169.

After switching a User Information Source to POLL or WAIT, if you do not plan to switch to LIVE again, you can delete the CKFREEZE data set that C2EAROOT created.

Recovering a lost SMF interval

About this task

You can recover a lost interval from your accumulated SMF data. The same process can also be used to initially load a Tivoli Security Information and Event Manager Server with SMF data.

Procedure

1. The first step in recovery is preparing an extract of your accumulated SMF data. Be sure that the extract data set has the prefix used by the Tivoli Security Information and Event Manager Agent, &C2ELVPFX You can use a job step such as the following one:

```
//    JCLLIB   ORDER=(your.prefix.CKRPARM)
//    INCLUDE  MEMBER=<your Agent's zSecure configuration>
//PREPDATA EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//IFASMFDI DD DISP=SHR,DSN=<your SMF cumulation dataset>
//IFASMFDO DD DISP=(,CATLG),UNIT=SYSALLDA,SPACE=<specify>,
//            DSN=&C2ELVPFX.PREP.RECOVER,
//            DCB=(LRECL=32760,BLKSIZE=0,RECFM=VBS)
//SYSIN    DD *
```

```
INDD(IFASMFDI,OPTIONS(DUMP))
OUTDD(IFASMFDO,TYPE(0:255))
DATE(2007171,2007171)
START(1530),END(1540)
```

You must adapt this job to specify your SMF selection, data set names, and zSecure configuration as match your system. Make sure that, in this stage of the process, your output data set does not match &C2ELVPFX..SMF.**. These data sets are removed by the Agent.

2. The next actions depend on the strategy that was specified for the Event Source:
   - For a LIVE strategy:
     a. On the Management Console, switch the **Collect Schedule** of your Event Source to **Never**.
     b. Rename the output you just prepared so that it matches &C2ELVPFX..SMF.**.
     c. On the Management Console, open the Properties of your Event Source and set **Collect past data** to **YES**.
     d. Trigger an ES-collect (that is, set the **Collect Schedule** to **Once** and specify a time in the near future).
     e. After the Collect has started, set the **Collect past data** switch back to **NO** and reinstate your original schedule.

     In this way, the **Collect past data** switch is enabled no longer than necessary, because it might incur some duplicate data from the live SMF.
   - For a POLL or WAIT strategy:
     a. On the Management Console, open the Properties of your Event Source, specify the data set you just prepared as **SMF Data Set name**.
     b. On the same Properties panel, set **Collect past data** to **YES**.
     c. Trigger an ES-collect (that is, set the **Collect Schedule** to **Once** and specify a time in the near future).
     d. After the Collect has started, set **Collect past data** back to **NO** and reinstate your original **SMF Data Set name** and your original schedule.

Event Source and User Information Source properties

The following sections contain information about the z/OS Event Source and User Information Source properties.

Properties for the z/OS Event Source

Collect strategy

You can use an INTERCEPT, LIVE, POLL, or WAIT collect strategy.

Under the INTERCEPT strategy, the Agent reads data sets that are created by the SMF intercept during your regular SMF offload. See “Strategies for collecting SMF event data” on page 172. SMF-offload is usually triggered by an SMF switch (when an SMF data set fills up). It is the only process that is certain not to miss any SMF records, if the SMF data sets are offloaded in the correct order. Using the INTERCEPT (or LIVE) strategy requires you to implement the SMF switch intercept; see “Strategies for collecting SMF event data” on page 172.

Note: INTERCEPT requires that the Agent software be at a sufficient maintenance level. See the documentation provided with your currently installed level of Tivoli Security Information and Event Manager. You

might need to apply a fix pack to the z/OS-resident part of Tivoli Security Information and Event Manager. If INTERCEPT is not available on your level of Tivoli Security Information and Event Manager, use LIVE instead. At a later time, you can switch to INTERCEPT.

The LIVE strategy is equivalent to INTERCEPT, except that it also reads the active SMF data set or logstream. LIVE does not collect more events than INTERCEPT collects. The only difference is that, with LIVE, SMF records are read earlier. After the next SMF switch, all SMF records from the active SMF data set or logstream are available in the intercept data sets (and filtered away when already processed).

Under a single Agent, only one Event Source is allowed with either the LIVE or the INTERCEPT strategy. Normally, you would not need multiple Event Sources under a single Agent; it is better to set up separate Agents under each z/OS image. But if you do need multiple Event Sources under a single Agent, all of these Event Sources, except the one for the z/OS image where the Agent runs, must have the POLL or WAIT strategy.

Attention: Do not use the LIVE strategy without setting up the SMF switch intercept unless you comply with all conditions specified in "Strategies for collecting SMF event data" on page 172. If you do not comply with all the conditions, all SMF records that were written between the last scheduled ES-collect and a subsequent SMF switch will be discarded.

Note:

1. With IFASMFDP, you must ensure that the SMF data sets are offloaded in the correct order. With SMFDUMP, the correct order happens automatically, because SMFDUMP processes all SMF data sets that qualify for offloading.
2. SMF records can be lost by the z/OS system itself if either of the following events occur:
   - The last SMF data set is full; messages IEE351I and IEE979W are displayed.
   - The SMF intercept fails, for example, because no disk space is available. In that case, the C2ECLSMF procedure reverts to your standard offload and raises an error condition. See "Strategies for collecting SMF event data" on page 172.

Under the POLL strategy, data is collected from the data set that you specify under SMF Data Set Name. If this data set is in use at the moment that ES Collect starts (or, for instance, when the data set resides on tape and all tape drives are in use), this particular ES Collect is canceled. The Schedule remains active, so in due time a new attempt is made.

The WAIT strategy acts the same as POLL strategy, with the difference that if the data set is in use, the ES Collect waits (up to half an hour) until the data set is available.

Error retention

The number of days that message log files are kept. Older log files are deleted at the next event source collect.

SMF Data Set Name

The data set from which data is collected when the collect strategy is POLL or WAIT. For normal production, specify your SMF accumulation data set here. Often, installations offload their active SMF into a data set that is a

member of a Generation Data Group (GDG); for instance, offload into SYS2.WEEKLY.SMF(0), and once a week create SYS2.WEEKLY.SMF(+1). If your installation uses a GDG, you can specify SYS2.WEEKLY.SMF(0), which represents the most recent member of the GDG.

An Agent can process collected data only from a system that has the same active security product (RACF, ACF2 or Top Secret) as the system where the Agent is running.

This field must be empty when the collect strategy is LIVE.

Store raw files
Reserved for future use.

Collect past data (discouraged)
Collect SMF records regardless of their timestamps. This option is intended for recovery of lost SMF intervals, and for initially loading a Tivoli Security Information and Event Manager Server with SMF data. For normal production, set it to NO, so that SMF intervals that have already been collected are not collected again.

Properties for the z/OS User Information Source

Collect strategy
You can use a LIVE, POLL, or WAIT collect strategy.

Under the LIVE strategy, User Information Source (UIS) information is obtained from the active security database where the Agent is running (primary for RACF, backup for ACF2), and from a CKFREEZE data set that the Agent itself refreshes during UIS collection, under control of the Collect schedule. For Top Secret, UIS information is obtained only from a CKFREEZE data set. Do not run more than one ES for each Agent under the LIVE strategy.

Under the POLL strategy, data is collected from the data set that you specify under CKFREEZE Data Set Name and (optionally) UNLOAD Data Set Name. If either of these data sets is in use at the moment that UIS Collect starts (for instance, because the periodical refresh is active), this particular UIS Collect is canceled. The Schedule remains active, so in due time a new attempt is made.

The WAIT strategy acts the same as POLL strategy, with the difference that if either data set is in use, the UIS Collect waits (up to half an hour) until the data set is available.

Complex name
The name of a complex.

You can view a complex as a group of z/OS images that share a security database. For a single-image Agent, specify the SMF system ID. A single-image Agent is the normal case. See “Performance and multi-image considerations” on page 181,

When setting up a multi-image Agent, be sure that all images that share a particular security database have the same complex name.

Error retention
The number of days that message log files are to be kept. Older log files are deleted at the next event source collect.

CKFREEZE Data Set Name
When the collect strategy is POLL or WAIT, this data set name is required;

ensure that it is refreshed regularly. Specify the data set that is periodically refreshed by job C2RJPREP; see “Use of a fresh CKFREEZE and UNLOAD each day” on page 42.

An Agent can process collected data only from a system that has the same active security product (RACF, ACF2, or TopSecret) as the system where the Agent is running.

This field must be empty when the collect strategy is LIVE. In that case, the Agent maintains its own data set; see “Preparing the Agent root” on page 169.

Store raw files
Reserved for future use.

System Policy Type
Production or Test.

The system policy type can be used to specify whether specific events generate policy exceptions. For instance, a system programmer might update the SYS1.PARMLIB. In the policies provided with zSecure, this update generates an exception on a production image, while on a test system it does not. On the Management Console, you can update policies or create your own. The qualifier Production or Test is included in the automatically maintained policy grouping files for this system.

UNLOAD Data Set Name
In most cases, leave this field blank. If it is blank, the Agent uses information from the active security database where the Agent is running. For a LIVE UIS, the UNLOAD Data Set Name must be blank.

A nonblank UNLOAD Data Set Name is required only when you access the security database of a foreign image, and that database is not shared with the image where the Agent is running. Only in that case, specify the UNLOAD that is periodically refreshed by job C2RJPREP. See “Use of a fresh CKFREEZE and UNLOAD each day” on page 42.

When several z/OS images share their security database, be sure to specify the same UNLOAD for all these images. As described in “Use of a fresh CKFREEZE and UNLOAD each day” on page 42, only a single UNLOAD is needed for a shared database. If the system where the Agent is running uses that same database, you can leave the UNLOAD Data Set Name blank for all these images.

Problem determination

This section describes installation and startup issues you might encounter. This section contains the following troubleshooting topics:

Master file

Data set &C2ELVPFX..MASTER&C2ELVLLQ is primarily used to communicate between the Agent and the SMF intercept. Do not edit this data set, and enqueue it no longer than is needed for diagnosis.

Note: The data set is rewritten during each ES Collect or UIS Collect. When the data set is enqueued and writing fails, the Collect continues, but changes are not reflected until the next ES or UIS Collect.

The master file looks like the following example:

```
[Agent]
Id=12.1.100
Server=192.168.45.3:5992
ConfigFile=IPO1-INSIGHTDOMAIN.cfg

[EventSource]
Id=18.1.118
Strategy=LIVE
Last_Collect=2012-03-25;14:06:12
Last_Successful_Collect=2012-03-25;14:06:12
Error_Retention=3
Collect_Past_Data=NO

[UserInformationSource]
Id=18.1.122
Complex=IPO1
Strategy=POLL
CKFREEZE=C2R.C2R2.1.0.IPO1.CKFREEZE
UNLOAD=
Last_Collect=2012-03-25;12:14:33
Last_Successful_Collect=2012-01-05;19:22.56
Error_Retention=3
```

Logs and related data

Messages that are important for the system operator, or for automated operations (for example, starting and stopping an ES or UIS Collect), are available in the MVS syslog.

The logs of the Agent are stored in the `log` subdirectory within the Agent root directory.

This directory also contains subdirectories *xx*`.`*yy*`.`*zz*`.props` for each Tivoli Compliance Insight Manager Enabler for z/OS, where *xx*, *yy* and *zz* are numbers that match the ID parameter in the corresponding section of the Master file. The numbers are assigned during the creation, on the Management Console, of the ES or UIS. In these directories, you can find the `stdout` and `stderr` files of the collect scripts, and message files from zSecure and zSecure Collect. The Actuators themselves delete old files from these directories, as determined by the Error retention field in the Properties dialog.

See "Preparing the Agent root" on page 169 for information about the Agent root directory.

Removing old run and log data

For Event Sources and User Information Sources that are no longer used (for instance when they have been deleted on the Management Console), run and log data are no longer maintained. However, this data might still affect the working of the Agent in the following ways:

- During each ES Collect, CKFREEZEs and UNLOADs from all UISes are processed. This processing is required because there is not always a one-to-one relation between ES and UIS.
- An Agent is considered to run under a LIVE strategy as long as it contains an ES that specifies LIVE. As a result, the SMF switch intercept writes its data, and ES Collect for other ESses (under a POLL or WAIT strategy) does not remove these data sets.

Therefore, be sure that you remove data for Actuators that are no longer used. This data is in the `ES_properties` and `UIS_properties` subdirectories of `run`. For each Tivoli Compliance Insight Manager Enabler for z/OS, there is a file `xx.yy.zz.props`, reflecting its Properties. For Event sources, there also exists a file *`xx.yy.zz`*`.props_cutoff`, which is used to prevent already collected SMF from being collected again. The *`xx.yy.zz`* in the names of these files match the ID parameter in the corresponding section of the Master file.

You can simply remove files (in `ES_properties` or `UIS_properties`) of Actuators that are no longer used. Similarly, you can delete old data in the `log` directory. The files and directories can be related to their actuators by numbers *`xx.yy.zz`* (or only *`zz`*) and part of their file name. Otherwise, do not update or serialize these files, or any other file in the `run` or `log` directory.

Setup error messages

You might see any of the following error messages during setup:

- **IEF642I** When running job C2EUNPAC, C2EAROOT, or C2ECNNCT, you might see the following message:

  ```
  IEF642I EXCESSIVE PARAMETER LENGTH IN THE PARM FIELD
  ```

  The maximum length of the PARM field in the JCL is 100 characters. When, through substitution, the parameter gets longer, this error occurs. In that case, the command must be moved out of the JCL into the SYSTSIN input:
  - Remove the PARM field and the comma that precedes it from the EXEC statement.
  - Change the SYSTSIN DD-statement to specify instream input (DD *).
  - Supply the command under //SYSTSIN DD *

  You must perform parameter substitution yourself.

  When the command exceeds the length of a line, you must split it, using a plus sign (+) or a minus sign (-), preceded by a blank, as the last character of each line that you want to continue.
- **FSUM2078**

  This message is displayed if you did not create a home directory for the Tivoli Security Information and Event Manager Server userid.
- **FSUM6241 Unknown option "-S"**

  This message is displayed when running the script on OS/390 releases before Version 2 Release 6; it is an OMVS bug. To avoid this error, temporarily set the following variable to NO before invoking the shell script:

  ```
  export _BPX_SPAWN_SCRIPT="NO"
  ```

  Generally, this variable is set to `YES` for performance reasons. Another circumvention is to explicitly prefix all shell script invocations with "sh".
- **FSUM7351 not found**

This message is displayed if you do not prefix a command with "./" and you do not have the current directory (".") in your PATH environment variable. For security reasons, it is unwise to include "." in your PATH variable.

- Directory not removed/renamed in job C2EAROOT

  1. This problem occurs when C2EAROOT finds a subdirectory `bin` in your AgentRoot. For the old OS/390 Agent, this was the usual directory layout. The z/OS Agent, however, requires its software and data to be in separate directories. This practice makes it easier to upgrade the software while leaving all data in place. Also, you can install the software once and run multiple Agents.

  If this message occurs during an upgrade installation (from the old OS/390 Agent), rename the directory. Later, when you are certain that no fallback will be needed, you can remove it.

  If this message occurs during a fresh installation, create a directory layout that separates software and data. Preferably, the directories also have different owners, and the userid that runs the Agent process does not have update access to the software.

Error messages when the Agent is running

The following errors can occur when the Agent is running:

- Insufficient authority to dub

```
ICH408I USER (C2EAUDIT) GROUP (C2E) NAME(CEA/ACTUATOR)
 CL(FSOBJ    )
INSUFFICIENT AUTHORITY TO DUB
```

  This message is displayed if you attempt to start the Agent for z/OS before OMVS initialization is finished (that is, too soon after IPL). The task runs, but not as an OMVS process, which makes it useless. To fix this problem, cancel the task, wait for the BPX1004I message, and then start the task again.

- Insufficient authority to lookup

  This error can occur when using a newly prepared file system, as preferred for the Agent data, and optionally for the Agent software. In a newly created file system, the only file that exists is the root directory. This directory is owned by the user who allocated the zFS data set, and has permission bits 700, so that no other user can access files in this directory. Therefore, the initial owner (or the superuser) must transfer ownership. That is:

```
<mount new file system>
cd <mountpoint>
chown <intended-user:intended-group> .
chmod g+rX .
```

- Abend EC6/FF09 during ES- or UIS-collect

  The most likely cause of this abend is a TIME limit in the JCL of C2EJSTRT/C2EAUDIT. Although starting the Agent is itself a short-running process, its TIME limit is inherited by USS processes that run under the Agent. Depending on the amount of data, this time can be insufficient.

- TCPIP Conn: can't bind to socket (errno 111)

  This message is displayed when the Agent is not able to bind to the TCP port that was specified (on the Management Console). The most likely cause is that the Agent is not allowed to use TCP/IP, or is not allowed access to this socket. In addition, you might see messages in the MVS syslog such as:

```
ICH408I USER(C2EAUDIT ) GROUP(STCGROUP) NAME(AGENT FOR INSIGHT)
  EZB.STACKACCESS.IPO1.TCPIP CL(SERVAUTH)
  INSUFFICIENT ACCESS AUTHORITY
  FROM EZB.STACKACCESS.*.* (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
```

See "TCP/IP security" on page 163 for information about the required access.

- TCPIP Conn: Socket error 112

  This message is displayed if you attempt to run the Tivoli Compliance Insight Manager Enabler for z/OS when TCP/IP is not running.

- GenTool: Error deleting temporary file ./temp###_0.gz created by ../bin/zOS_collect_events.sh

  This message can be ignored.

- GenTool: Error creating chunk log using ../bin/zOS_collect_events.sh, returncode = 8, error code = 8

  This message points out that an error occurred. The error might or might not be in the C2RCARLA program. To get more information, go to the directory `actuator/log/xx.yy.zz.props`, where *zz* is the actuator number. Look for a file *yyyy-mm-dd.hh:mm:ss*`.C2RCARLA.SYSPRINT` with the most recent time. This file contains the CNR messages that can explain the reason for the error.

c2ediag command

About this task

The **c2ediag** command collects diagnostic information about to the z/OS Agent and its environment. You can provide the information to IBM Software Support.

Note: The **c2ediag** tool must be in the directory into which you unpacked the `ibm.tsiem.actuator.pax.Z` file (or the `C2EPAX.Z` file if you are using the precursor product, Tivoli Compliance Insight Manager). If the command fails, verify that it is in that directory. For old releases of Tivoli Security Information and Event Manager, you can download the command from the IBM Support Web site at http://www.ibm.com/software/support/probsub.html. Search for a technote with **C2EDIAG** in the title.

The **c2ediag** command does not require the Agent to be running.

Procedure

To run the command and send the output to IBM Software Support, use the following steps:

1. Log on with a userid with root authority (uid=0). This authority is required because the command retrieves full information about all active processes in the system. Running as root also ensures the following permissions:
   - READ and WRITE permission in the *AgentRoot* directory (the directory identified by the C2EPATH parameter in the zSecure configuration) and in the subdirectories
   - READ and EXECUTE permission to the software in the &C2ESW directory
2. Go to an OMVS command shell and `cd` into the *AgentRoot* directory.
3. Type the following command:

   ```
   ./bin/c2ediag
   ```

Note: In some cases, it is necessary to collect diagnostic data on a very specific point in time; for instance, during the start of an ES- or UIS-collect. In this case, run at least two terminal sessions at the same time:

- One session to monitor the system (for example, with an MCS console or with SDSF)
- One terminal session with the command already typed, so that you can issue the command by only pressing the Enter key.

In this way, you can capture diagnostic data at the very moment that a suspected process is running.

4. Use binary mode to transfer the **C2Ediag_dump_***xxxxx***.pax** files to IBM Software Support. Each time the **c2ediag** command is issued, it generates a file named **C2Ediag_dump_***xxxxx***.pax** in the *AgentRoot* directory. In the filename, *xxxxx* represents a timestamp.

   System log output (SDSF) is not captured by **c2ediag**. If this information is considered relevant, also supply an extract of the system log around the time of the suspected events.

   Attention: Never unpack the pax files in the directory where they were created; doing so overwrites the Agent's data.
5. After you transfer the files and receive confirmation by IBM, delete the files to prevent the disk space from filling up. The Agent itself cannot delete the **c2ediag**-generated files, because they are root-owned.

Checklists for configuring a z/OS Agent

Preparing the base z/OS system

Use the following checklists to configure the z/OS Agents.

Note: All tasks are organized by administrative function to help you consolidate requests for help from different departments within your organization. These checklists do not reflect the order of the installation actions.

Table 19. Checklist for preparing the base z/OS system

Task	Multiplicity	Details	Check
Parmlib members IEASYSxx, BPXPRMxx, CLOCKxx, CUNIMGxx, CUNUNIxx, IFAPRDxx, SMFPRMxx.	Image (or shared)	• CLOCKxx: "Setting z/OS UNIX time zones" on page 162 • BPXPRMxx: "Owners, directories, and file systems" on page 164, "Preparation of the owner and location for the software" on page 165 • IEASYSxx, CUNIMGxx, CUNUNIxx: "Unicode requirement" on page 162 • IFAPRDxx: "Enablement of license features" on page 18 • SMFPRMxx: "Overview" on page 159	
Proclib members C2ECSTRT, C2ECSTOP, zSecure configuration	Image (or shared)	• "Starting and stopping the Agent" on page 169 • "Parameters for the z/OS Agent" on page 167	

Table 19. Checklist for preparing the base z/OS system (continued)

Task	Multiplicity	Details	Check
SMF intercept	Agent	"Strategies for collecting SMF event data" on page 172	

Storage management

Table 20. Checklist for configuring storage management

Task	Multiplicity	Details	Check
File system for software	Install	• "Owners, directories, and file systems" on page 164 • "Preparation of the owner and location for the software" on page 165	
File systems for data	Agent (or shared)	• "Owners, directories, and file systems" on page 164 • "Preparing the Agent-running userid and its workspace" on page 167	
Dedicated CKFREEZE (permanent)	LIVE UIS	• "Preparing the Agent root" on page 169 • "Event Source and User Information Source properties" on page 186	
Planning space for SMF Intercept (short-lived)	LIVE ES	"Strategies for collecting SMF event data" on page 172	

Production control

Table 21. Checklist for configuring production control

Task	Multiplicity	Details	Check
Set up batch jobs or started tasks	Agent	• "Parameters for the z/OS Agent" on page 167 • "Starting and stopping the Agent" on page 169	
Ensure fresh CKFREEZE (and optionally UNLOAD)	Shared for ES and POLL/WAIT UIS	"Event Source and User Information Source properties" on page 186	
Start Agent after each IPL	Agent	"Starting and stopping the Agent" on page 169	

User and group administration

Table 22. Checklist for configuring users and groups

Task	Multiplicity	Details	Check
User and Group to own the software files	Install	• "Owners, directories, and file systems" on page 164 • "Preparation of the owner and location for the software" on page 165	
User and Group to run each Agent	Shared across Agents	• "Owners, directories, and file systems" on page 164 • "Preparing the Agent-running userid and its workspace" on page 167	
OMVS segments	Install + Agent	• "Owners, directories, and file systems" on page 164 • "Preparing the Agent-running userid and its workspace" on page 167	
Aliases in master catalogs	Image	• "Owners, directories, and file systems" on page 164 • "Preparing the Agent-running userid and its workspace" on page 167	

Permissions

Table 23. Checklist for configuring permissions

Task	Multiplicity	Details	Check
Installer needs SURROGAT, or must chown/chmod	Complex (that is, for each image/Agent) unless security is shared.	• "Owners, directories, and file systems" on page 164 • "Unpacking the software" on page 166 • "Preparing the Agent root" on page 169	
Installer must mount file system	Complex	"Owners, directories, and file systems" on page 164	
Allow Agent-running user access to software by group permissions	Complex	• "Owners, directories, and file systems" on page 164 • "Preparation of the owner and location for the software" on page 165	
Agent might need to run zSecure Collect	Complex, LIVE UIS only	"Preparing the Agent-running userid and its workspace" on page 167	
Agent must read active RACF/ACF2 database, or UNLOAD **Note:** This task does not apply to Top Secret systems.	Complex	"Preparing the Agent-running userid and its workspace" on page 167	

Table 23. Checklist for configuring permissions (continued)

Task	Multiplicity	Details	Check
Agent must read SMF (accumulated or live)	Complex	"Preparing the Agent-running userid and its workspace" on page 167	

Network management

Table 24. Checklist for configuring network management

Task	Multiplicity	Details	Check
Ensure connectivity	Agent	-	
Assign IP ports	Agent	"TCP/IP security" on page 163	

Management Console

Table 25. Checklist for configuring the Management Console

Task	Multiplicity	Details	Check
Configure Point of Presence	Agent	Tivoli Security Information and Event Manager	
Save configuration file	Agent	Tivoli Security Information and Event Manager	
Configure UIS	Organization-specific	• Tivoli Security Information and Event Manager • ES and UIS properties	
Configure ES	Organization-specific	• Tivoli Security Information and Event Manager • ES and UIS properties	
Collect schedule	Organization-specific	Tivoli Security Information and Event Manager	

Setup tasks

Table 26. Checklist for configuring setup tasks

Task	Multiplicity	Details	Check
Create user and group to own software	Install	"Owners, directories, and file systems" on page 164	
Create/mount software-only zFS	Install	"Preparation of the owner and location for the software" on page 165	
Upload *x*:\zOS\ ibm.tsiem.actuator.pax.Z (or C2EPAX.Z if you are using Tivoli Compliance Insight Manager)	Install	"Uploading the pax file" on page 166	
Unpack	Install	"Unpacking the software" on page 166	

Table 26. Checklist for configuring setup tasks (continued)

Task	Multiplicity	Details	Check
Create user and group to run Agent	Complex	"Preparing the Agent-running userid and its workspace" on page 167	
Create/mount zFS for data	Agent (or shared)	"Preparing the Agent-running userid and its workspace" on page 167	
Set up production JCL for Agent	Agent (or shared)	"Starting and stopping the Agent" on page 169	
Set up SMF intercept	Agent (live ES)	"Strategies for collecting SMF event data" on page 172	
Set up secure connection (first-time Agent start)	Agent	"Secure connection setup" on page 177	
Stop limited mode Agent	Agent	"Secure connection setup" on page 177	
Start full mode Agent	Agent	"Secure connection setup" on page 177	
Run UIS- and ES-collect	Organization-specific	"Secure connection setup" on page 177	

Appendix A. Site module

Note: If your z/OS system is older than release 1.8, consider applying the fix for APAR OA10774, rather than using CKRSITE to fall back to the FACILITY class. This APAR enables you to use the XFACILIT class which saves you a conversion at a later point in time if you want to exploit longer profile names.

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 27 list the zSecure options that can be customized in the Site module CKRSITE:

Table 27. Site module options

Parameter	Allowed values	Description	Default
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 199 for instructions).

zSecure uses SAF to configure menus, and to determine what data (profiles, rules, SMF records) users are allowed to see. At a bare minimum, create catchall profiles `CKF.**, CKG.**, CKR.**, C2R.**, and C2X.**` with `UACC=NONE`, or the equivalent resource rules, and grant READ access to the people entitled to use all functions of zSecure. For further refining, see "Security resources specific to zSecure" on page 208. In particular, the profile CKG.** (in the CKRSITE-configured class) is required; a wider generic profile like *.** is not sufficient. Also, zSecure requires that the IRR.** profile in the FACILITY class is present. This profile is not subject to configuration by CKRSITE.

Data presentation 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. The ISPF user interface menus are determined through resources CKR.OPTION.* and the available line commands are controlled using resources CKR.ACTION.*. Users need at least READ access.

If no matching profile is found for CKR.READALL, access to the RACF source determines scoping of resources and profiles. See Appendix C, "Restricted mode," on page 209. If no matching profiles are found for the CKR.OPTION.* and CKR.ACTION.* resources, all menu items are included and all action commands are allowed. The default return code of the XFACILIT resource class is not used when determining control of these items. To avoid use of the default access, you can define backstop generic profiles that set the appropriate access.

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.AU for AUDIT
CKR.OPTION.CO for COMMANDS etc.
```

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 in RACF or ACF2, 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
AI.I: ACF2_INFOLINE
AK.F: ACF2_RULELINE
AL.L: ACF2_LID
AR.I: ACF2_INFORULE
CH.C: Action commands on Change tracking exceptions overview
CL.$: RACF CLASS
CP.$: CICS_PROGRAM
CR.$: CICS_REGION
CS.$: CICS_TRANSACTION
CT.C: Action commands on Change tracking systems overview
DK.$: DB2_PACKAGE
DN.$: DB2_PLAN
DR.$: DB2_REGION
DS.$: DSN
DT.$: DB2_TABLE
FL.$: FIELD
IC.$: RACF SETROPTS_CLASS
MB.$: MEMBER
MC.$: IMS_REGION
MP.$: IMS_PSB
MT.$: IMS_TRANSACTION
RA.$: RACF_ACCESS
R1.$: REPORT_AC1
RC.D: RACF (DATASET entities)
```

```
RC.G: RACF (GROUP entities)
RC.R: RACF (RESOURCE entities)
RC.U: RACF (USER entities)
RD.$: REPORT_NONDEFAULT
RO.$: REPORT_OUTOFGROUP
RN.$: REPORT_REDUNDANCY
RP.$: REPORT_PADS
RR.$: REPORT_PROFILE
RS.$: REPORT_SENSITIVE
SC.D: RACF REPORT_SCOPE (DATASET entities)
SC.R: RACF REPORT_SCOPE (RESOURCE entities)
SD.$: SENSDSN
SM.D: SMF (DATASET entities)
SM.G: SMF (GROUP entities)
SM.L: SMF (LOGONID entities)
SM.R: SMF (RESOURCE entities)
SM.U: SMF (USER entities)
SP.$: SPT
ST.$: REPORT_STC
TR.$: TRUSTED
UN.$: UNIX
ZA.B: zAlert action commands on alert categories display
ZA.C: zAlert action commands on alert configurations display
ZA.D: zAlert action commands on e-mail destination sets display
ZA.R: zAlert action commands on alerts display
```

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*.

Access to the security database

The userids using 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 209 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 206.

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 199), 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 a value 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.

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 199)
- 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 199.)

- 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," on page 213.
- 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 for running the daily CKGRACF job" on page 43.

  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 RACF new password exit is C2X.ICHPWX01.

Resources that specify which data can be seen or updated

For the READALL resource, see Appendix C, "Restricted mode," on page 209. 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.

Security resources specific to zSecure

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 199.)

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 primarily determined by the access that the live system grants the user to the CKR.READALL resource (in the XFACILIT class, unless you changed this. See Appendix A, “Site module,” on page 199):

- NO access enforces restricted mode.
- READ access grants unrestricted mode.

When access to CKR.READALL is undecided, restricted mode is in effect when at least one of the following is true:

- When 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 210.
- The Site module (see Appendix A, “Site module,” on page 199) has been configured to specify restricted mode.

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.

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 57.

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* (SC23-3726 for OS/390 RACF (also known as RACF 2.1), and SA22-7683 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('TCPIP.SEZALINK'//NOPADCHK)
   RALT PROGRAM * ADDMEM('TCPIP.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'//NOPADDCHK)    /* IBM Security zSecure */
   RALT PROGRAM * ADDMEM('FAN130.SEAGLMD//NOPADCHK)           /* REXX     */
   RALT PROGRAM * ADDMEM('ISP.SISPLOAD//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.)

If you are installing on z/OS 1.2 or later, you can also use the RACF ICH420I messages to determine the cause for the dirty environment.

Appendix D. Configuration parameters

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, Tivoli Compliance Insight Manager Enabler for z/OS, Change Tracking, 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

This parameter specifies the data set name of the 'compliant authorized ID population' members for zSecure Audit option AU.R - Rule-based compliance evaluation. Users can concatenate their own CKACUST library in front of the library specified in the configuration member with option CO.1 or SE.8. See the topic "CO.1 LIBRARIES - Data set selection" in the *IBM Security zSecure Admin and Audit for RACF: User Reference Manual*.

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.

C2ECUST

A partitioned data set for installation-supplied CARLa for the z/OS Agent for Tivoli Security Information and Event Manager. See “CARLa members that support adding CARLa statements (optional)” on page 171 on how to use this data set.

C2ELVLLQ

This parameter provides the low-level qualifier for the same data sets referenced by the C2ELVPFX parameters. You can use this optional parameter to make the names for your SMF data sets comply with your data set naming convention. You might want to do this if you have ACS-routines that make processing decisions based on the low-level qualifier. If you specify a non-blank C2ELVLLQ, specify a value with a period as the first character, `.daily_smfdata` for example.

C2ELVPFX

One or more qualifiers, to be used as prefix for SMF data sets to be used by the Agent for Tivoli Security Information and Event Manager. This parameter is only required when you use this component. Data sets with this prefix are created in your SMF accumulation processing, and then read and removed by the Agent. Be sure that only Agent-owned data sets exist under the prefix C2ELVPFX. It is a good idea to use the user ID of the Agent as the high-level qualifier.

C2EPATH

Path to the agent root directory for the Agent for Tivoli Security Information and Event Manager. By default, it is the home directory of the user that runs the Tivoli Compliance Insight Manager Enabler for z/OS agent, but you might choose to use a subdirectory. Only required when you use this component. If you run multiple Agents, each one needs its own agent root.

C2EQCUST

This parameter applies to the data preparation for QRadar SIEM. See “Setup of the collection process” on page 153.

C2EQPATH
This parameter applies to the data preparation for QRadar SIEM. See "Setup of the collection process" on page 153.

C2ESW
Path to the directory where you unpacked the z/OS Agent for Tivoli Security Information and Event Manager software. Multiple z/OS Agents can share this directory. This parameter is required when you use this component.

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 59.

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. The method is similar to the one for the Tivoli Security Information and Event Manager Agent; see "Same Agent definition on multiple images" on page 184.

This parameter applies only to the zSecure Visual component. For additional information, see Chapter 13, "Setup and use of the zSecure Visual Server," on page 113.

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 embedded during a client-initiated transaction

C2RWEXR1
CARLa to be embedded 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 128 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. When upgrading a Server that currently runs with zSecure 1.11 or older, copy these members from the SCKRCARL library, and add the C2RWCUST parameter to your zSecure configuration. Also, 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 45.

This parameter applies only to the zSecure Visual component. For additional information, see Chapter 13, "Setup and use of the zSecure Visual Server," on page 113.

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 120.

When this parameter is OFF, older encryption algorithms are accepted. Use this setting for:

- Upgrading the run directory of a server previously on version 1.8.1 to version 1.13.1.
- Clients that connect using a zSecure Visual 1.8.1 to 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 120.

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 13, "Setup and use of the zSecure Visual Server," on page 113.

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 120.

This parameter applies only to the zSecure Visual component. For additional information, see Chapter 13, "Setup and use of the zSecure Visual Server," on page 113.

C2XEXITS = [ACTIV | INACT]

This parameter controls the activation of the zSecure-supplied ICHPWX01 exit C2XPWX01 by zSecure and Tivoli Security Information and Event Manager.

ACTIV

When zSecure and Tivoli Security Information and Event Manager start under control of a zSecure configuration that specifies `ACTIVE` and the ICHPWX01 exit is not yet active, the exit is activated. Activating the exits is only useful on z/OS release 1.6 or older, with RACF active. On more recent z/OS levels, and on systems with ACF2 or Top Secret, the exits have no function (and they do no harm).

INACT

No activation or deactivation is done on a start of either IBM Security zSecure or Tivoli Security Information and Event Manager. If you want to deactivate an already established exit, use the C2XACTV program, as described in the *User Reference Manual*.

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 45.

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. 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 tee 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 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 223.

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)

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 201)

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 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 226 and “Customizing individual menu options” on page 226.

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 overtypeable 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 embedding 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            Action on language items
1  1. English                 1  1. No action (only use specified language)
   2. Dutch                     2. Reset all items to company default
   3. French                    3. Reset all items to IBM Security zSecure default
   4. German                    4. Customise menu items
   5. Italian                   5. Customise action line commands
   6. Portuguese
   7. Spanish
   8. Japanese
   9. Other
   0. zSecure
Language used: User English
```

Figure 11. SETUP - NLS panel

Selecting a different language: From the panel shown in Figure 11 on page 225, 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.

Customizing individual menu options: For all languages except Japanese, selecting Option **SE.N** Action **4** opens the panel shown in Figure 12 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 language.

Note: For information about double-byte character set (DBCS) support in zSecure, see "Setup (default) National Language Support (SE.D.N)" on page 224.

```
 Menu   Options   Info   Commands
------------------------------------------------------------------------------
                        zSecure Suite - Setup - NLS    Row 1 to 17 of 171
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 IOCO
_  SE 3          3  Preamble        Commands run before every query
_  SE 4          4  Confirm         Specify command generation options
_  SE 5          5  View            Specify view options
_  SE 7          7  Output          Specify output options
_  SE 8          8  Command files   Select and maintain command library
_  SE U          U  User defined    User defined input sources
_  SE C          C  Change Track    Maintain Change Tracking parameters
_  SE C  M       M  Site msgs       Site defined message table
_  SE C  C       C  zSecure msgs    zSecure defined message table
_  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 configuration
```

Figure 12. 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 13. 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. You are allowed to specify the menu items with the same option; the first one on the ultimate menu panel is used. This allows you to be more flexible with the profiles. For example, one group of users can use the first menu item, the other the second, while on the menu the options are the same. User changes to default menu options are *not* propagated during an NLS upgrade. For user options, the Menu option must match the Official option.

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
_  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 14. 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 28 shows the relationship between product (license) and function:

Table 28. Relationship of products to licensed functions in NLS

Product	RACF Admin	RACF Audit	RACF Report	ACF2 Admin	ACF2 Audit	ACF2 Report	TSS Audit	CKGRACF
zSecure Admin	/							/
zSecure Audit for RACF		/	/					

Table 28. Relationship of products to licensed functions in NLS (continued)

Product	RACF Admin	RACF Audit	RACF Report	ACF2 Admin	ACF2 Audit	ACF2 Report	TSS Audit	CKGRACF
Security zSecure Admin and Audit for RACF	/	/	/					/
zSecure Audit for ACF2					/	/		
zSecure Audit for Top Secret							/	
IBM Security zSecure	/	/	/	/	/	/	/	/
zSecure Audit		/	/		/	/	/	

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 15. 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 16. 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 14 on page 228 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 16 on page 231), 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 17. 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 16 on page 231), 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 18. 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 29. 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

Table 29. Entity types versus allowed selects / excludes (continued)

Entity type	for	Allows to select / exclude
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 . SYS1.JSXLOG.**_____________________________ (EGN mask)
```

Figure 19. 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)

This 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 20. 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 30. 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 **not** 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 up the member list with the **E** line command, or option **CO.2**.

Retaining your Setup default data when upgrading zSecure

When upgrading zSecure, you can retain your Setup default data by continuing to use your existing PROFDSN data set. The `PROFDSN` data set in use is specified in the `PROFDSN` parameter in your configuration. For additional information, see Appendix D, "Configuration parameters," on page 213.

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 can be used to create your own modified versions of the zSecure panels or to add new functionality to the product. This 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 224). 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," on page 213). 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 REXX program CKRERUN can be used to run either zSecure or a TSO command from the ISPF menu of Security zSecure. The 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           ===>_UUSER   +

+Usercatalog      ===>_UCAT                       +(no quotes)

)PROC
  VPUT (UUSER UCAT) SHARED
  &CKRNEXT = &Z              /* no continuation panel */
)END
```

Figure 21. 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.

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
2Z4A/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 are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, Acrobat, 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 trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

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.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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 trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Index

Special characters

A

B

C

D

E

F

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

Z

Printed in USA

SC27-5638-00