

Security zSecure CICS Toolkit  
Version 2.2.1

*User Guide*





Security zSecure CICS Toolkit  
Version 2.2.1

*User Guide*



**Note**

Before using this information and the product it supports, read the information in "Notices" on page 129.

**December 2016**

This edition applies to version 2, release 2, modification 1 of IBM Security zSecure CICS Toolkit (product number 5655-N18) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 1988, 2016.

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

---

# Contents

<b>About this publication</b> . . . . .	<b>v</b>
zSecure documentation . . . . .	v
Obtain licensed documentation . . . . .	vi
IBM zSecure Suite library . . . . .	vi
IBM zSecure Manager for RACF z/VM library . . . . .	ix
Related documentation . . . . .	ix
Accessibility . . . . .	x
Technical training . . . . .	x
Support information . . . . .	x
Statement of Good Security Practices . . . . .	x

<b>Chapter 1. Introduction</b> . . . . .	<b>1</b>
Application interface . . . . .	1
Command interface . . . . .	1
RRSF concerns . . . . .	2
Dates before and after the year 2000 . . . . .	3

<b>Chapter 2. zSecure CICS Toolkit installation</b> . . . . .	<b>5</b>
Installation and post-installation checklist . . . . .	5
Sample JCL . . . . .	6
SMP/E zones creation and initialization . . . . .	7
Performing pre-installation steps . . . . .	8
TARGET and DLIB data sets allocation . . . . .	8
SMP/E DDDEFS update . . . . .	9
Receive the product . . . . .	9
zSecure CICS Toolkit code addition . . . . .	9
Integration of zSecure CICS Toolkit with your system . . . . .	9
Installing SVC . . . . .	10
Protecting the SVC . . . . .	10
Definition of SCQTLOAD as APF-authorized . . . . .	11
Updating the CICS startup JCL . . . . .	11
zSecure CICS Toolkit enablement in PARMLIB . . . . .	11
CQTPCNTL parameters definitions . . . . .	12
Definitions of programs, mapsets, and transactions to CICS . . . . .	12
Updating CICS tables . . . . .	13
Defining the RACF profiles . . . . .	14
Internal Security Resource Listing . . . . .	16
Automatic assignment of USS UIDs (OMVS AUTOUID) . . . . .	19
Automatic creation of home directories (OMVS MKDIR) . . . . .	19
zSecure CICS Toolkit restart . . . . .	20
RTST transaction definition . . . . .	21
Manual restart of the zSecure CICS Toolkit subtasks . . . . .	21
Use of the CICS Transaction Server with zSecure CICS Toolkit . . . . .	22
Globalization . . . . .	22

<b>Chapter 3. Parameters for zSecure CICS Toolkit</b> . . . . .	<b>23</b>
Parameter Descriptions . . . . .	23
CQTPCNTL parameter values verification . . . . .	26

<b>Chapter 4. Application security management</b> . . . . .	<b>29</b>
Operator ID or OPID check . . . . .	29
Application conversion . . . . .	29
Alias definitions . . . . .	31
Simple application security interface . . . . .	31
User information retrieval . . . . .	31
Resource access verification . . . . .	32

<b>Chapter 5. The zSecure CICS Toolkit command interface</b> . . . . .	<b>35</b>
Navigating the Main menu . . . . .	35
Adding, altering, or deleting a group (ADDGROUP, ALTGROUP, or DELGROUP command) . . . . .	36
Adding a user profile (ADDUSER command) . . . . .	37
Changing a profile (ALTUSER command) . . . . .	39
Altering the CICS segment for a user (ALTUSER CICS SEGMENT) . . . . .	41
Altering the TSO segment for a user (ALTUSER TSO SEGMENT) . . . . .	43
Altering the OMVS segment for a user (ALTUSER OMVS SEGMENT) . . . . .	44
Altering the WORKATTR segment for a user (ALTUSER WORKATTR SEGMENT) . . . . .	47
Connecting a user or group to a group (CONNECT command) . . . . .	48
Deleting a data set (DELETE DATASET command) . . . . .	49
Deleting a user profile . . . . .	50
Listing the profile for one or more data sets (LISTDSET command) . . . . .	51
LISTDSET Display Example . . . . .	54
Toggling the LISTDSET panel . . . . .	54
Viewing the users authorized, their access authority and access count (LISTDSET USERIDS) . . . . .	55
Viewing the program/userid combination (LISTDSET Programs) . . . . .	56
Listing the profile for one or more groups (LISTGROUP command) . . . . .	57
LISTGROUP Display Example . . . . .	59
Toggling the LISTGROUP panel . . . . .	60
Listing users for a group (LISTGROUP command, USERIDS option) . . . . .	60
Deleting user IDs from a LISTGROUP . . . . .	61
Listing the subgroups of a group . . . . .	62
Listing the profiles for a user ID (LISTUSER command) . . . . .	63
LISTUSER Display Example . . . . .	66
Toggling the LISTUSER panel . . . . .	67

Listing groups for a user ID (LISTUSER command, GROUPS option) . . . . .	68
Listing categories for a user ID (LISTUSER command, Categories option) . . . . .	69
Listing the TSO and CICS segments for a user ID (LISTUSER command, Segments option). . . . .	70
Granting or removing access to a resource (PERMIT command). . . . .	71
Maintaining associations (RACLINK command) . . . . .	73
Listing and maintaining profiles in a general resource class (RALTER / RDEFINE / RDELETE commands) . . . . .	75
Removing user IDs or groups from a group (REMOVE command) . . . . .	76
Listing the profiles for a general resource class (RLIST command) . . . . .	76
RLIST Display Example . . . . .	79
Listing the members in a profile (RLIST command, MEMBERS option) . . . . .	79
Listing user IDs in a profile and the access they have (RLIST command, USERS option) . . . . .	80
Listing users/groups in the conditional access list for a profile (RLIST command, CONDACC option) . . . . .	81
Listing, adding, updating, or removing the USRDATE fields from a profile (USRDATA command). . . . .	82

**Chapter 6. zSecure CICS Toolkit exit points specifications. . . . . 85**

**Chapter 7. Application programming interface (API) . . . . . 87**

Command requests using the COMMAREA . . . . .	87
Change the authorized user . . . . .	88
Perform a search . . . . .	89
Implementing field or record level security . . . . .	89
Access Authority Check function . . . . .	90
Access Authority Check (Extended) function . . . . .	91
Resource Profile List function . . . . .	92
TSQUEUE usage for profiles. . . . .	96
Return and Reason codes. . . . .	96
Access check and DATA retrieval (RSRD) . . . . .	97
Retrieval of USERDATA . . . . .	97
Definitions of USERDATA entries . . . . .	98
Additional considerations . . . . .	99
Use of RACLIST exits . . . . .	100
Restrictions . . . . .	100

API specification . . . . .	101
Return codes . . . . .	102
Installation considerations . . . . .	102
ADDGROUP / ALTGROUP / DELGROUP function (add, alter, or delete a group) . . . . .	103
ADDUSER function (add user profile) . . . . .	104
ALTUSER function (changing a profile). . . . .	105
ALTUSER (CICS SEGMENT) function (alter CICS segment) . . . . .	106
ALTUSER (TSO SEGMENT) function (change TSO segment) . . . . .	107
ALTUSER (OMVS SEGMENT) function (change OMVS segment) . . . . .	108
ALTUSER (WORKATTR SEGMENT) function (change WORKATTR segment) . . . . .	109
CONNECT function (connect a user or group to a group). . . . .	110
DELETE DATASET function (delete data set profile) . . . . .	111
DELETE USERID function (delete user profile) . . . . .	111
LISTDATASET function (list profile for one or more data sets) . . . . .	112
LISTGROUP function (list profile for a group) . . . . .	113
LISTUSER function (list profile for a user ID). . . . .	115
PASSWORD function (change password) . . . . .	117
PERMIT function (grant or remove access) . . . . .	118
PERMITX function (grant or remove access - any resource) . . . . .	118
RACLINK function (define, list, undefine, or approve user associations) . . . . .	119
REMOVE function (remove user IDs or groups from a group) . . . . .	120
RALTER/RDEFINE/RDELETE function (list and maintain profiles) . . . . .	121
RLIST function (list profiles for general resource class) . . . . .	121
USRDATA function (list and maintain users' USRDATA fields) . . . . .	123
VERIFY function (verify user ID and password) . . . . .	125
Sample programs . . . . .	125
Simple API interface . . . . .	126
Resource Profile List Interface . . . . .	126

**Notices . . . . . 129**  
**Trademarks . . . . . 131**

**Index . . . . . 133**

---

## About this publication

The IBM® Security zSecure™ CICS® Toolkit enhances CICS/RACF security features by enabling you to execute RACF® commands directly from CICS and eliminating the need to use TSO. Application programs can also use IBM Security zSecure CICS Toolkit for security features instead of relying on internal application security functions. In this manner, all security definitions can be maintained centrally, or distributed among security coordinators.

This publication describes the two components of the IBM Security zSecure CICS Toolkit: the applications programming interface (API) and the command interface. It explains how to install and use this product.

This publication is intended for the following people:

- Systems support personnel responsible for the installation of IBM Security zSecure CICS Toolkit
- CICS Security administrators responsible for implementing the additional RACF command controls provided by IBM Security zSecure CICS Toolkit

Readers must also be familiar with performing security and administration tasks in a CICS environment and with RACF concepts and commands.

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

---

## zSecure documentation

The IBM Security zSecure Suite and IBM Security zSecure Manager for RACF z/VM libraries consist of unlicensed and licensed publications. This section lists both libraries and instructions to access them.

Unlicensed zSecure publications are available at the IBM Knowledge Center for IBM zSecure Suite or IBM zSecure Manager for RACF z/VM. The IBM Knowledge Center is the home for IBM product documentation. You can customize IBM Knowledge Center, create your own collection of documents to design the experience that you want with the technology, products, and versions that you use. You can also interact with IBM and with your colleagues by adding comments to topics and by sharing through email, LinkedIn, or Twitter. For instructions to obtain the licensed publications, see “Obtain licensed documentation” on page vi.

IBM Knowledge Center for product	URL
IBM zSecure Suite	<a href="http://www.ibm.com/support/knowledgecenter/SS2RWS/welcome">http://www.ibm.com/support/knowledgecenter/SS2RWS/welcome</a>
IBM zSecure Manager for RACF z/VM	<a href="http://www.ibm.com/support/knowledgecenter/SSQQGJ/welcome">http://www.ibm.com/support/knowledgecenter/SSQQGJ/welcome</a>

The IBM Terminology website consolidates terminology for product libraries in one location.

## Obtain licensed documentation

All licensed and unlicensed publications for IBM Security zSecure Suite 2.2.1 and IBM Security zSecure Manager for RACF z/VM 1.11.2, except the Program Directories, are included on the *IBM Security zSecure Documentation CD, LCD7-5373*. Instructions for downloading the disk image (.iso) file for the zSecure *Documentation CD* directly are included with the product materials.

To obtain an extra copy of the .iso file of the *Documentation CD* or PDF files of individual publications, complete the following steps:

1. Go to the IBM Publications Center.
2. Select your country or region and click the **Go** icon.
3. On the **Welcome to the IBM Publications Center web** page, click **Customer Support** in the left navigation menu.
4. Complete the support form with the following information: your contact details, your customer number, and the numbers of the licensed publications you want to order.
5. Click **Submit** to send the form. The form is forwarded to an IBM Publications Center Customer Support representative who sends you details to fulfill your order.

Alternatively, you can send an email to [tivzos@us.ibm.com](mailto:tivzos@us.ibm.com) requesting access to the .iso file of the *zSecure Documentation CD*. Include your company's IBM customer number and your preferred contact information. You will receive details to fulfill your order.

## IBM zSecure Suite library

The IBM Security zSecure Suite library consists of unlicensed and licensed publications.

Unlicensed publications are available at the IBM Knowledge Center for IBM zSecure Suite. Unlicensed publications are available to clients only. To obtain the licensed publications, see *Obtaining licensed publications*. Licensed publications have a form number that starts with L; for example, LCD7-5373.

The IBM Security zSecure Suite library consists of the following publications:

- *About This Release* includes release-specific information as well as some more general information that is not zSecure-specific. The release-specific information includes the following:
  - *What's New*: Lists the new features and enhancements in zSecure V2.2.1.
  - *Release Notes*: For each product release, the release notes provide important installation information, incompatibility warnings, limitations, and known problems for the IBM Security zSecure products.
  - *Documentation*: Lists and briefly describes the zSecure Suite and zSecure Manager for RACF z/VM libraries and includes instructions for obtaining the licensed publications.
- *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 CA-ACF2
- IBM Security zSecure Visual
- IBM Security zSecure Adapters for QRadar SIEM for RACF, CA-ACF2, and CA-Top Secret
- *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 admin and audit features from ISPF panels. This manual also provides troubleshooting resources and instructions for installing the zSecure Collect for z/OS® component. This publication is available to licensed users only.
- *IBM Security zSecure Admin and Audit for RACF Line Commands and Primary Commands Summary*, SC27-6581  
Lists the line commands and primary (ISPF) commands with very brief explanations.
- *IBM Security zSecure Audit for ACF2 Getting Started*, GI13-2325  
Describes the zSecure Audit for CA-ACF2 product features and provides user instructions for performing standard tasks and procedures such as analyzing Logon IDs, Rules, Global System Options, and running reports. The manual also includes a list of common terms for those not familiar with ACF2 terminology.
- *IBM Security zSecure Audit for ACF2 User Reference Manual*, LC27-5640  
Explains how to use zSecure Audit for CA-ACF2 for mainframe security and monitoring. For new users, the guide provides an overview and conceptual information about using CA-ACF2 and accessing functionality from the ISPF panels. For advanced users, the manual provides detailed reference information, troubleshooting tips, information about using zSecure Collect for z/OS, and details about user interface setup. This publication is available to licensed users only.
- *IBM Security zSecure Audit for Top Secret User Reference Manual*, LC27-5641  
Describes the zSecure Audit for CA-Top Secret product features and provides user instructions for performing standard tasks and procedures. This publication is available to licensed users only.
- *IBM Security zSecure CARLa Command Reference*, LC27-6533  
Provides both general and advanced user reference information about the CARLa Auditing and Reporting Language (CARLa). CARLa is a programming language that is used to create security administrative and audit reports with zSecure. The *CARLa Command Reference* also provides detailed information about the NEWLIST types and fields for selecting data and creating zSecure reports. This publication is available to licensed users only.
- *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 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 *Documentation CD* is available as a downloadable .iso file; see Obtaining licensed publications to obtain this file.

Program directories are provided with the product tapes. You can also download the latest copies from Program Directories.

- *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 Security zSecure Adapters for QRadar SIEM.
- *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 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 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 for the zSecure Administration, Auditing, and Compliance solutions:
  - 5655-N23: *Program Directory for IBM Security zSecure Administration, GI13-2292*
  - 5655-N24: *Program Directory for IBM Security zSecure Compliance and Auditing, GI13-2294*
  - 5655-N25: *Program Directory for IBM Security zSecure Compliance and Administration, GI13-2296*

## IBM zSecure Manager for RACF z/VM library

The IBM Security zSecure Manager for RACF z/VM library consists of unlicensed and licensed publications.

Unlicensed publications are available at the IBM Knowledge Center for IBM zSecure Manager for RACF z/VM. Licensed publications have a form number that starts with L; for example, LCD7-5373.

The IBM Security zSecure Manager for RACF z/VM library consists of the following publications:

- *IBM Security zSecure Manager for RACF z/VM Release Information*  
For each product release, the Release Information topics provide information about new features and enhancements, incompatibility warnings, and documentation update information. You can obtain the most current version of the release information from the zSecure for z/VM<sup>®</sup> documentation website at the IBM Knowledge Center for IBM zSecure Manager for RACF z/VM.
- *IBM Security zSecure Manager for RACF z/VM: Installation and Deployment Guide, SC27-4363*  
Provides information about installing, configuring, and deploying the product.
- *IBM Security zSecure Manager for RACF z/VM User Reference Manual, LC27-4364*  
Describes how to use the product interface and the RACF administration and audit functions. The manual provides reference information for the CARLa command language and the SELECT/LIST fields. It also provides troubleshooting resources and instructions for using the zSecure Collect component. This publication is available to licensed users only.
- *IBM Security zSecure CARLa Command Reference, LC27-6533*  
Provides both general and advanced user reference information about the CARLa Auditing and Reporting Language (CARLa). CARLa is a programming language that is used to create security administrative and audit reports with zSecure. The *zSecure CARLa Command Reference* also provides detailed information about the NEWLIST types and fields for selecting data and creating zSecure reports. This publication is available to licensed users only.
- *IBM Security zSecure Documentation CD, LCD7-5373*  
Supplies the IBM Security zSecure Manager for RACF z/VM documentation, which contains the licensed and unlicensed product documentation.
- *Program Directory for IBM zSecure Manager for RACF z/VM, GI11-7865*  
To use the information in this publication effectively, you must have some prerequisite knowledge that you can obtain from the program directory. The *Program Directory for IBM zSecure Manager for RACF z/VM* is intended for the systems programmer responsible for installing, configuring, and deploying the product. It contains information about the materials and procedures associated with installing the software. The Program Directory is provided with the product tape. You can also download the latest copies from the IBM Knowledge Center for IBM zSecure Manager for RACF z/VM.

---

### Related documentation

If you are using IBM Security zSecure Suite components 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*, while information about auditing RACF can be found in the *RACF Auditor's Guide*.

For information about incompatibilities, see the "Release notes" section of "About This Release" on the IBM Security zSecure documentation website at IBM Knowledge Center for IBM zSecure Suite.

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

Manual	Order Number
z/OS Security Server RACF Command Language Reference	SA23-2292
z/OS Security Server RACF System Administrator's Guide	SA23-2289
z/OS Security Server RACF Auditor's Guide	SA23-2290
z/OS Security Server RACF System Programmer's Guide	SA23-2287
z/OS MVS™ System Commands	SA38-0666

---

## Accessibility

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

---

## Technical training

For technical training information, see the IBM Training and Skills website at <http://www.ibm.com/training>.

See the zSecure Training page in the zSecure public Wiki for information about available training for zSecure.

---

## 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. Introduction

The zSecure CICS Toolkit enhances the CICS/RACF security features used by many installations. With zSecure CICS Toolkit, CICS Security Administrators can run RACF commands directly from CICS, eliminating the need to use TSO. Application programs can use zSecure CICS Toolkit for security features instead of having to rely on internal application security functions. In this manner, all security definitions can be maintained centrally, or distributed among security coordinators.

zSecure CICS Toolkit is divided into two parts: the application programming interface (API) and the command interface.

Applications can use zSecure CICS Toolkit for security instead of internal application security. You can maintain all security definitions centrally or distribute them among security coordinators.

---

### Application interface

CICS systems programmers can use the zSecure CICS Toolkit API to customize the RACF commands and panels to their installation requirements.

In addition to having a central focal point for security definitions, the CICS systems programmer no longer needs to maintain the sign-on table; this table is no longer required for CICS version 2.x and later versions. As soon as transaction security checking is being performed by RACF, the only definition required in the sign-on table is the DEFAULT entry. Applications programmers no longer need to update their security tables and files because all the required information is defined to and maintained by RACF. Eliminating these tasks saves time that can be used for other work.

Another unique aspect of zSecure CICS Toolkit is the ability to RACF protect all application transactions. zSecure CICS Toolkit protects even those transactions that have not been protected previously. Examples of this protection are: a transaction that prints a report on a printer device, and an ATM transaction. Because printers and ATMs do not perform sign-ons, no security checking can be performed; therefore, transactions running on these devices could not be protected. zSecure CICS Toolkit removes this restriction so that all such transactions are protected. zSecure CICS Toolkit supports MRO environments and is easily installed, used, and maintained.

---

### Command interface

CICS systems programmers can use the command interface to run certain RACF commands directly from CICS instead of TSO. The command interface provides greater distribution of certain capabilities and responsibilities to CICS users. It eliminates the need to use the TSO application and reduces CPU overhead.

When you use the command interface for searches:

- You can use more extensive search masks to retrieve information about profiles.
- You have greater flexibility in the criteria that are used.

**Note:** zSecure CICS Toolkit uses a different concept for enabling the user authority to run commands or portions of them. For example, with GROUPSPECIAL, a user without any more authority can reset a user password or resume the user. This user does not require GROUPSPECIAL and does not have to be connected to the group of the user that is being reset.

**Note:** zSecure CICS Toolkit uses a different concept for enabling the user authority to run commands or portions of them. For example, a user can be permitted to reset the password of another user and/or resume a user without needing the broader authority to perform other tasks that are available through GROUPSPECIAL authority. The user does not require the GROUPSPECIAL attribute and does not need to be connected to the group of the user that is being reset.

Under this methodology, the responsibility of resetting user passwords can be decentralized to any other area where there is access to a CICS terminal. Data Security personnel can then use their time and efforts in other areas.

RACF protects both the transaction that is used to start zSecure CICS Toolkit for the command interface and the commands themselves. Even if there is no security on the zSecure CICS Toolkit transaction, the user must be *permitted* to the commands in RACF to start them. A user must also be *permitted* to have the authority to reset user IDs. The only exception is a user with the SPECIAL attribute or a user with access to the TOOLKIT.SPEC definition. For information about this definition, see Chapter 2, “zSecure CICS Toolkit installation,” on page 5.

All changes to RACF profiles produce SMF records.

---

## RRSF concerns

There is a difference between the zSecure CICS Toolkit actions and the TSO RACF commands.

All additions, updates, and deletions of RACF profiles that zSecure CICS Toolkit does are done with either the RACROUTE or ICHEINTY interface. In the remainder of this manual, the function is called by the equivalent RACF command processor name. For example, when the manual talks about the API interface that connects a user to a group, it is called the CONNECT command. The actual processing is **not** the CONNECT command. It is a similar set of actions that cause the same result as if the RACF CONNECT command was used. Therefore, the updates are considered application updates. The difference between the TSO RACF command and the zSecure CICS Toolkit actions is only apparent in the required RACF Remote Sharing Facility (RRSF) definitions if you need RRSF propagation of updates.

If you want to propagate changes through RRSF, activate this function by the operator RACF SET AUTOAPPL command either from the console, or by PARMLIB. You probably have already set the equivalent SET AUTODIRECT option to propagate RACF commands. For the automatic propagation of application updates, you must also define appropriate profiles in the RRSFDATA resource



class. The profiles must allow the RACF updates that the CICS region makes to be propagated. The user that needs access to the RRSFDATA profiles is the CICS region user, and **not** the CICS terminal user or the CICS default user. For more information about the definition of appropriate RRSFDATA profiles, see the "Automatic Direction" section in the RACF Remote Sharing Facility chapter in the *RACF Security Administrator's Guide*.

---

## Dates before and after the year 2000

When a date is specified for the REVOKEDATE and RESUMEDATE parameters, zSecure CICS Toolkit treats the dates according to IBM rules.

The date format is *YYDDD*, where *YY* is the year and *DDD* is the day number.

- If the specification for *YY* is 71 or greater, the date is treated as being in the 20th century; for example, 1971.
- If the specification for *YY* is 70 or less, the date is treated as being in the 21st century; for example, 2070.

These rules also provide compatibility with previous releases.



## Chapter 2. zSecure CICS Toolkit installation

Systems support personnel can use the information in this chapter to install zSecure CICS Toolkit. SMP/E is used for the installation process.

Before you begin the installation, see the *Program Directory: IBM Security zSecure CICS Toolkit* for information about the prerequisites. Then, see “Installation and post-installation checklist” for pointers to instructions for installation and post-installation tasks.

### Installation and post-installation checklist

Systems support personnel responsible for installing zSecure CICS Toolkit can use the checklist below to perform installation and post-installation tasks.

Table 2. Installation checklist

Step	Description	Instructions	Job name	Status
1	Load installation JCL.	“Sample JCL” on page 6		
2a	Create and initialize SMP/E zones.	“SMP/E zones creation and initialization” on page 7	CQTJSMPx (SCQTINST)	
2b	Allocate TARGET and DLIB data sets.	“TARGET and DLIB data sets allocation” on page 8	CQTJALL (SCQTINST)	
3	Update SMP/E DDEFs.	“SMP/E DDDEFs update” on page 9	CQTJDDD (SCQTINST)	
4a	RECEIVE zSecure CICS Toolkit.	“Receive the product” on page 9	CQTJREC (SCQTINST)	
4b	APPLY zSecure CICS Toolkit.	“zSecure CICS Toolkit code addition” on page 9	CQTJAPP (SCQTINST)	
4c	ACCEPT zSecure CICS Toolkit.	“Integration of zSecure CICS Toolkit with your system” on page 9	CQTJACC (SCQTINST)	
5	Install the SVC.	“Installing SVC” on page 10		
6	Protect the SVC.	“Protecting the SVC” on page 10	CQTJRDEF (SCQTSAMP)	
7	Define data sets as APF-authorized.	“Definition of SCQTLOAD as APF-authorized” on page 11		
8	Update the CICS startup JCL.	“Updating the CICS startup JCL” on page 11		
9	Check product enablement through <b>IFAPRDxx</b> in PARMLIB.	“zSecure CICS Toolkit enablement in PARMLIB” on page 11		
10	Update the <b>CQTPCNTL</b> parameters.	“CQTPCNTL parameters definitions” on page 12	CQTJCNTL (SCQTINST)	
11	Define programs, mapsets, and transactions to CICS.	“Definitions of programs, mapsets, and transactions to CICS” on page 12	CQTJRDO (SCQTSAMP)	
12	Update the CICS tables.	“Updating CICS tables” on page 13	CQTJPLT CQTJSRT (SCQTSAMP)	

Table 2. Installation checklist (continued)

Step	Description	Instructions	Job name	Status
13	Define RACF profiles to control access to zSecure CICS Toolkit functions.	"Defining the RACF profiles" on page 14	CQTJRDEF (SCQTSAMP)	
14	Grant more authorizations to the CICS Started Task to perform USS functions.	"Automatic assignment of USS UIDs (OMVS AUTOUID)" on page 19 and "Automatic creation of home directories (OMVS MKDIR)" on page 19		
15	Restart zSecure CICS Toolkit.	"zSecure CICS Toolkit restart" on page 20		

## Sample JCL

Systems support personnel responsible for installing zSecure CICS Toolkit can access the installation jobs from *relfile 2* and the SCQTSAMP data set or copy the jobs from the tape or product files.

To access the sample installation jobs, perform an SMP/E RECEIVE. Then copy the jobs from the *relfiles* to a work data set for editing and submission. The sample JCL used for the SMP/E steps is contained in *relfile 2*. The remaining sample JCL, used during the configuration steps, is contained in the SCQTSAMP data set that is created during the earlier installation steps.

To copy the jobs from the tape or product files, submit the job that follows this paragraph. Use either the **//TAPEIN** or the **//FILEIN DD** statement, depending on your distribution medium, and comment out or delete the other statement. Add a job card and change the lowercase parameters to uppercase values to meet your site requirements before you submit.

```
//STEP1 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//TAPEIN DD DSN=IBM.HCQT221.F2,UNIT=tunit,
//          VOL=SER=volser,LABEL=(x,SL),
//          DISP=(OLD,KEEP)
//FILEIN DD DSN=IBM.HCQT221.F2,UNIT=SYSALLDA,DISP=SHR,
//          VOL=SER=filevol
//OUT DD DSNAME=jcl-library-name,
//       DISP=(NEW,CATLG,DELETE),
//       VOL=SER=dasdvol,UNIT=SYSALLDA,
//       SPACE=(TRK,(5,5,5))
//SYSUT3 DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN DD *
          COPY INDD=xxxxIN,OUTDD=OUT
/*
```

In the previous sample, update the statements as follows:

- If you use TAPEIN:

***tunit*** The unit value that matches the product tape.

***volser*** The volume serial that matches the product tape.

***x*** The tape file number where the data set name is on the tape.

See the documentation that is provided by CBPDO to see where IBM.HCQT221.F2 is on the tape.

If you install from a non-CBPDO product tape, the tape *volser* is CQT221, and the file number for *relfile 2* is 3 on the tape.

- If you use FILEIN:
  - filevol* The volume serial of the DASD device where the downloaded files is located.
- In the OUT statement:
  - jcl-library-name*  
The name of the output data set where the sample jobs are stored.
  - dasdvol*  
The volume serial of the DASD device where the output data set is located.
- In the SYSIN statement:
  - xxxxIN*  
Either TAPEIN or FILEIN, depending on your input DD statement.

## SMP/E zones creation and initialization

Before starting the SMP/E installation, systems support personnel responsible for installing zSecure CICS Toolkit must decide on the SMP/E zones to use.

You can choose one of the following options:

- Install zSecure CICS Toolkit in new (dedicated) zones in a new CSI.  
This option is the only one for which sample jobs are provided.
- Install zSecure CICS Toolkit in new (dedicated) zones in an existing CSI.
- Install zSecure CICS Toolkit in existing zones in an existing CSI.

The sample jobs that are provided do not accommodate all possible combinations of CSIs and zones. The only option for which sample jobs are provided is the first option. The jobs can be used to set up a dedicated GLOBAL and PRODUCT CSI with dedicated TARGET and DLIB zones. For information, see “Performing pre-installation steps” on page 8.

The example jobs are provided in SCQTINST. Adapt the JCL and submit the jobs to complete the creation and initialization of the SMP/E environment. The jobs all use lowercase strings for the values that must be adapted to fit your installation standards. The following values are currently used.

### *Your-Global*

The data set prefix that you want to use for the GLOBAL SMP/E data sets. This prefix is used for the name of the GLOBAL CSI and for the SMP/E data sets shared between all SMP/E zones.

### *Your-Product*

The data set prefix that you want to use for the zSecure CICS Toolkit data sets. This data set prefix is also the prefix for the SMP/E data sets specific to zSecure CICS Toolkit.

### *sysallda*

The unit name that is used for all data set allocations.

*volser* The name of the DASD volume in your system where you want to create the zSecure CICS Toolkit data sets.

**Note:** In an SMS environment, the ACS routines might assign another volume than the one specified from the *volser*.

*tape* The unit name of the tape-unit where the zSecure CICS Toolkit distribution tape can be mounted.

**Note:** The value for *Your-Global* cannot be the same as that for *Your-Product*. If you want to use similar prefixes, add an extra qualifier for the GLOBAL zone. For example, you might use the following two values:

- SMPE.TOOLKIT.GLOBAL as the value for *Your-Global*
- SMPE.TOOLKIT as the value for *Your-Product*

Use the following table to record the values for the installation variables suitable for your environment.

Table 3. Pre-installation steps to define SMP/E zones

Variable	Your Value
<i>Your-Global</i>	
<i>Your-Product</i>	
<i>sysallda</i>	
<i>volser</i>	
<i>tape</i>	

## Performing pre-installation steps

Before beginning the installation, systems support personnel responsible for installing zSecure CICS Toolkit must define the required SMP/E zones.

### Procedure

1. Create a global CSI: The sample job for creation of a new GLOBAL CSI is provided in member CQTJSMPA. This job also defines a GLOBAL zone in this new CSI.
  - ▶ Adapt and submit job CQTJSMPA.
2. Create a product CSI: The sample job for creation of a new product CSI is provided in member CQTJSMPB. This job also defines a TARGET and DLIB zone in this new CSI.
  - ▶ Adapt and submit job CQTJSMPB.
3. Define an options entry for zSecure CICS Toolkit: The sample job for defining an options entry specific for zSecure CICS Toolkit is provided in member CQTJSMPC. It specifies the utilities and the data set prefix to be used during the remaining SMP/E installation steps.
  - ▶ Adapt and submit job CQTJSMPC.

## TARGET and DLIB data sets allocation

Systems support personnel responsible for installing zSecure CICS Toolkit must allocate the TARGET and DLIB data sets.

zSecure CICS Toolkit adds six target data sets and six distribution data sets to your SMP/E environment. Example JOB CQTJALL contains the necessary JCL to allocate the required TARGET and DLIB data sets.

- ▶ Submit CQTJALL

## SMP/E DDDEFs update

Systems support personnel responsible for installing zSecure CICS Toolkit perform this step to define to SMP/E the data sets that you allocated in the previous step. If you choose to include appropriate DD-statements in all of your SMP/E jobs, you can omit this step. If you want to use the suggested setup through dynamic allocation, this step is required. The example job CQTJDDD contains the JCL needed for this step.

The installation process also uses SMP/E CALLLIBS processing. This function is used to resolve external references during installation. When zSecure CICS Toolkit is installed, ensure that DDDEFs exist for the CSSLIB library.

**Note:** The DDDEFs are used only to resolve the link edit for zSecure CICS Toolkit with CALLLIBS. These data sets are not updated during the installation of zSecure CICS Toolkit. The provided sample job includes the required DDDEF.

- ▶ Submit job CQTJDDD.

## Receive the product

Systems support personnel need the SMP/E Modification Control Statements to correctly install zSecure CICS Toolkit.

If you are installing from the zSecure CICS Toolkit product tape, the first file on that tape is the SMPMCS data set. It contains the SMP/E Modification Control Statements that are needed for correct installation of zSecure CICS Toolkit.

- ▶ Submit job CQTJREC.

## zSecure CICS Toolkit code addition

Systems support personnel responsible for installing zSecure CICS Toolkit must add the code, examples, and documentation to the system.

The SMP/E statement that is needed is  
`APPLY GROUPEXTEND SELECT(HCQT221).`

Because of the use of a SELECT for the product FMID, SMP/E does not require the use of the FUNCTIONS keyword. An example job is included in member CQTJAPP. Before you run this job, specify the data set name of your GLOBAL CSI.

- ▶ Submit the job CQTJAPP.

## Integration of zSecure CICS Toolkit with your system

If systems support personnel responsible for the installation of zSecure CICS Toolkit are satisfied with its implementation, they can ACCEPT the product. After you ACCEPT, the product is a part of your system.

zSecure CICS Toolkit does not normally cause any further systems programming work. An example ACCEPT job is provided in CQTJACC.

- ▶ Submit job CQTJACC.

## Installing SVC

Systems support personnel must install a type 3 SVC to be used for zSecure CICS Toolkit.

### Procedure

1. Include the SCQTLPA data set in your LPALSTxx member in PARMLIB. You can also copy SCQTSVC00 from the supplied SCQTLPA to an existing LPALIST data set. For example, SYS1.LPALIB.
2. Determine the SVC number to be used for the zSecure CICS Toolkit SVC. In this example, use 222.
3. Perform the following steps:
  - a. Update the IEASVCxx member in PARMLIB with an entry to define the zSecure CICS Toolkit SVC. An example is  

```
SVC Parm 222,REPLACE,TYPE(3),APF(NO),EPNAME(CQTSVC00)
```
  - b. Perform an IPL to install the SVC into the system.

**Note:** Some software packages provide the capability of updating the SVC table without performing an IPL. If you are using such a software package, you might not need an IPL to start using the SVC.

4. Add the DFHSRT entry with the following format:  

```
DFHSRT TYPE=SYSTEM,ABCODE=FX,ROUTINE=DFHSRTRR
```

where *XX* is the hexadecimal number of the SVC. For example, if the number assigned to the SVC is 222, the entry is FDE. For more information about updating the DFHSRT table, see “Updating CICS tables” on page 13.

► Update the LPALSTxx and IEASVCxx members and IPL your system.

## Protecting the SVC

To prevent unauthorized use of the zSecure CICS Toolkit SVC, a RACHECK is issued from the SVC to ensure that the caller is authorized.

### About this task

Before CICS can use zSecure CICS Toolkit, it must be given access to through RACF.

### Procedure

1. Define the SVC to RACF with the **RDEFINE** command. You must use resource class FACILITY.  

```
RDEFINE FACILITY TOOLKIT.SVC UACC(NONE)
```
2. Grant SVC access to each CICS region that will have an installation of the toolkit. Use the following command:  

```
PERMIT TOOLKIT.SVC CLASS(FACILITY) ID(userid) ACCESS(READ)
```

where *userid* is the ID of the CICS region.

A sample of the preceding RACF definitions is included in member CQTJRDEF in SCQTSAMP.



## Definition of SCQTLOAD as APF-authorized

The zSecure CICS Toolkit subtask programs perform the retrieval and updates to the profiles in the RACF database. Use of these RACF functions requires APF authorization.

The data set containing these subtask programs is SCQTLOAD. This data set must be APF-authorized. To define SCQTLOAD as APF-authorized, you must update the IEAAPFxx or PROGxx member in PARMLIB with the name of the SCQTLOAD data set.

- Update PROGxx and activate by operator command SET PROD=xx.

## Updating the CICS startup JCL

Several changes must be made to the CICS startup JCL.

### Procedure

1. Add the SCQTRPL data set that contains the zSecure CICS Toolkit programs and maps to the DFHRPL concatenation. For example:

```
//DFHRPL DD DISP=SHR,DSN=APPL1.LOADLIB
//      DD DISP=SHR,DSN=APPL2.LOADLIB
//      DD DISP=SHR,DSN=APPL3.LOADLIB
//      DD DISP=SHR,DSN=APPL4.LOADLIB
//      DD DISP=SHR,DSN=CICS.TOOLKIT.SCQTRPL
```

2. The SCQTLOAD data set contains the zSecure CICS Toolkit programs that are used as MVS subtasks. Add this data set to the CICS STEPLIB.
3. If you decide to add a zSecure CICS Toolkit loadlib to the STEPLIB concatenation, rather than copying the modules to an existing loadlib, you must also make that loadlib APF-authorized because *all* loadlibs in the CICS STEPLIB must be APF-authorized.

To define the zSecure CICS Toolkit loadlib as APF-authorized, you must update the IEAAPFxx or PROGxx member in SYS1.PARMLIB with the name of the zSecure CICS Toolkit loadlib. An example of updating the JCL STEPLIB parameter:

```
//STEPLIB DD DISP=SHR,DSN=LOADLIB1
//      DD DISP=SHR,DSN=LOADLIB2
//      DD DISP=SHR,DSN=CICS.TOOLKIT.SCQTLOAD
```

## zSecure CICS Toolkit enablement in PARMLIB

At startup of the zSecure CICS Toolkit tasks, zSecure CICS Toolkit verifies whether the product is enabled or disabled by IFAPRDxx in PARMLIB.

If the product is enabled or not defined in IFAPRDxx, initialization of zSecure CICS Toolkit continues normally.

If the product is disabled, a message (CQT907) is issued, and zSecure CICS Toolkit initialization is terminated. Disabling the product does not affect the remainder of CICS initialization.

To explicitly enable zSecure CICS Toolkit, include an entry such as the following one in an active IFAPRDxx member.

```
PRODUCT OWNER('IBM CORP')
        NAME('zSecure Toolkit')
        ID(5655-N18)
        VERSION(*) RELEASE(*) MOD(*)
        STATE(ENABLED)
```

If you want to disable zSecure CICS Toolkit, create an entry like this one, and replace the parameter STATE(ENABLED) by the parameter STATE(DISABLED).

- ▶ Update IFAPRDxx and activate by operator command SET PROD=xx.

## CQTPCNTL parameters definitions

CQTPCNTL defines certain parameters that zSecure CICS Toolkit uses.

These parameters include the SVC number and the RACF resource class that is used. See Chapter 3, “Parameters for zSecure CICS Toolkit,” on page 23 for CQTPCNTL parameter definitions and information about customizing CQTPCNTL.

- ▶ Adapt CQTPCNTL and Submit CQTJCNL.

After the definitions are made, transaction RTCK can be run to check the parameters in CQTPCNTL.

## Definitions of programs, mapsets, and transactions to CICS

You must define the programs, mapsets, and transactions that zSecure CICS Toolkit uses. Although you can still use CICS tables for this definition, the preferred method is through CICS Resource Definition Online (RDO).

An example job to make these definitions is provided in member CQTJRDO in SCQTSAMP. You must define the following mapsets:

```
CQTBST0 CQTBCH0 CQTB000 CQTB100 CQTB200 CQTB300
CQTB400 CQTB500 CQTB550 CQTB560 CQTB580 CQTB590
CQTB600 CQTB700 CQTB800 CQTB860 CQTB900 CQTBAA00
CQTB800 CQTB800 CQTB800 CQTB860 CQTB900 CQTBAA00
CQTB800 CQTB800 CQTB800 CQTB860 CQTB900 CQTBAA00
```

You must define the following programs:

```
CQTPAPI0 CQTPAPPL CQTPSNP0 CQTPATCH CQTPCHEK CQTPDTCB
CQTPPLT00 CQTPSTRT CQTP0000 CQTP0010 CQTP0020 CQTP0030
CQTP0031 CQTP0040 CQTP0041 CQTP0042 CQTP0043 CQTP0044
CQTP0050 CQTP0055 CQTP0056 CQTP0058 CQTP0059 CQTP0060
CQTP0070 CQTP0080 CQTP0081 CQTP0082 CQTP0083 CQTP0084
CQTP0086 CQTP0090 CQTP0091 CQTP0100 CQTP0110 CQTP0111
CQTP0112 CQTP0113 CQTP0114 CQTP0120 CQTP0130 CQTP0131
CQTP0132 CQTP0133 CQTP0134 CQTP0135 CQTP0136 CQTP0140
```

In addition to the regular executable programs, some modules contain data that is used by all zSecure CICS Toolkit programs. This data must be permanently available to all zSecure CICS Toolkit programs. You can make the data permanently available by defining these programs as *resident*. The following programs must be defined as resident programs:

- CQTPAPRM
- CQTPMSGE
- CQTPCNTL

All the zSecure CICS Toolkit programs (both the regular programs and the resident programs) must be defined with EXECKEY(CICS).

You can define the following programs for compatibility reasons. Your application programs can refer to these programs by the name that is used in the precursor product Consul zToolkit. These programs were renamed in Tivoli® zSecure CICS Toolkit version 1.8.1. For best results, use the following names:

- CQTPAPI0
- CQTPSNP0
- CQTPAPPL

If you did not adapt your existing applications to reflect the new names, you must also define the following programs. These programs are alias names of the new modules:

```
CRTKAPI CRTKSNP CRTKAPPL
```

To use the required functions of zSecure CICS Toolkit, you might also need to define the online transactions. If you use only the functions from the API interface, definition of these transactions is not required. The following transaction must be defined for the following programs:

```
RTCK --> PROG(CQTPCHEK)
RTST --> PROG(CQTPSTRT)
RTMM --> PROG(CQTP0000)
```

- Update and submit CQ TJRDO.

If you want these definitions to be active at startup, you must also include the *group* TOOLKIT in a *list* that is used to activate CICS resources.

## Updating CICS tables

Systems support personnel must update the CICS tables.

### Procedure

- Add the following entry to DFHPLTPI:

```
DFHPLT TYPE=ENTRY,PROGRAM=CQTPLT00
```

**Note:** CQTPLT00 must be placed after the DFHDELIM entry in the PLT.

- Add the following entry to DFHPLTSD:

```
DFHPLT TYPE=ENTRY,PROGRAM=CQTPDTCH
```

Without the shutdown entry in the PLT, CICS is likely to abend with system abend code A03 during regular shutdown processing.

- Add the following entry to DFHSRT:

```
DFHSRT TYPE=SYSTEM,ABCODE=Fxx,RECOVER=YES
```

For the exact definition of the parameter ABCODE=Fxx, see “Installing SVC” on page 10.

The previous tables must be translated with the CICS table update procedure in use at your installation. Typically, the procedure is called DFHAUPLE, and it is in a data set named similarly to CICSTS31.XDFHINST. An example is provided in job CQTJPLT and CQTJSRT.

- ► Adapt and submit job CQTJPLT.
- ► Adapt and submit CQTJSRT.

**Note:** The PLT program must be found during CICS startup. That means that the CQTPLT00 program must already be defined to CICS. The RDO example defines the resources, but does not automatically activate them. You must either include the TOOLKIT *group* in a *list*, or ensure that the resource definition is active in some other way.

## Defining the RACF profiles

Follow these steps to use the zSecure CICS Toolkit RACF command interface to define RACF profiles.

### Procedure

1. Define the zSecure CICS Toolkit commands to RACF.

Table 4 contains a list of the zSecure CICS Toolkit commands and the first and subsequent levels of authority that is required for each:

Table 4. zSecure CICS Toolkit commands: Required authorization levels

COMMAND	LEVEL	SUBSEQUENT LEVELS
<b>ADDGROUP</b>	TOOLKIT.ADGR	ADGR. <i>grpname</i>
<b>ADDUSER</b>	TOOLKIT.ADUS	ADUS. <i>dfltgrp</i>
<b>ALTGROUP</b>	TOOLKIT.ALGR	ALGR. <i>grpname</i>
<b>ALTUSER</b>	TOOLKIT.AUSR  When using the subfunctions to manage segments, access to TOOLKIT.ACIC, TOOLKIT.ATSO, TOOLKIT.AOMV or TOOLKIT.AWRK might also be required.	AUSR. <i>dfltgrp</i> When you assign a shared UID in an OMVS segment, you also need either System-SPECIAL or access to SHARED.IDS in the UNIXPRIV class.
<b>CONNECT</b>	TOOLKIT.CONN	CONN. <i>grpname</i>
<b>DELDSD</b>	TOOLKIT.DELD	DELD. <i>hlq</i>
<b>DELGROUP</b>	TOOLKIT.DELG	DELG. <i>grpname</i>
<b>DELUSER</b>	TOOLKIT.DELU	DELU. <i>dfltgrp</i>
<b>LISTDATASET</b>	TOOLKIT.LDSD	None
<b>LISTGROUP</b>	TOOLKIT.LGRP	LGRP. <i>grpname</i>
<b>LISTUSER</b>	TOOLKIT.LUSR	LUSR. <i>dfltgrp</i>
<b>PASSWORD</b>	None	PSWD. <i>grpname</i>  The <b>PASSWORD</b> command is available only through the API.
<b>PERMIT</b>	TOOLKIT.PEMT	PEMT. <i>grpname</i> / <i>grpname</i>  If the PERMIT is for a GROUP, PEMT. <i>grpname</i> is used. If it is for a USERID, then PEMT. <i>grpname</i> is used.  When issuing a PERMIT, the user also requires access to the resource that is being given access to.  If the resource is in a class that is not defined in the CICS SIT, the user also needs access to PEMX. <i>cdtclass</i> .
<b>RACLINK</b>	TOOLKIT.RACL	RACL. <i>dfltgrp</i>
<b>RALTER</b>	TOOLKIT.RALT	RALT. <i>cdtclass</i>
<b>RDEFINE</b>	TOOLKIT.RDEF	RDEF. <i>cdtclass</i>
<b>REMOVE</b>	TOOLKIT.REMV	REMV. <i>grpname</i>

Table 4. zSecure CICS Toolkit commands: Required authorization levels (continued)

COMMAND	LEVEL	SUBSEQUENT LEVELS
<b>RDELETE</b>	TOOLKIT.RDEL	RDEL.cdtclass
<b>RLIST</b>	TOOLKIT.RLST	RLST.cdtclass
<b>USRDATA</b>	TOOLKIT.USRL  When using the subfunctions to Add/Update/Delete, or when accessing these functions directly from the API, access to TOOLKIT.USRA or TOOLKIT.USRD is required.	USRU.grpname USRN.usrdata-name
<b>VERIFY</b>	None  VERIFY is available only through the API and allows applications to verify a user's ID and password without the need for the user to sign on.	None

In all of the preceding definitions:

- *grpname* is the GROUP name.
- *dfltgrp* is the DEFAULT GROUP name.
- *hlq* is the high-level-qualifier of the data set name.
- *cdtclass* is the GENERAL RESOURCE CLASS name that is defined in the CDT.

Each command is defined to RACF as a resource in a manner similar to the one in which transactions are defined.

It is best to define the following generic names first, with a UACC of NONE. This ensures that users are not given access to commands through other generic definitions within the resource class.

Assuming that the RSRCLASS parameter in CQTPCNTL is TCICSTRN:

```
RDEFINE TCICSTRN (TOOLKIT.* ADGR.* ADUS.* ALGR.* AUSR.*
CONN.* DELD.* DELG.* DELU.* LDSD.* LGRP.* LUSR.* PEMT.*
PSWD.* RACL.* RALT.* RDEF.* RDEL.* REMV.* RLST.*
USRU.* USRN.*) UACC(NONE)
```

Define the zSecure CICS Toolkit commands to RACF with the **RDEFINE** command. This example also assumes that TCICSTRN is the RSRCLASS parameter.

```
RDEFINE TCICSTRN (TOOLKIT.ADGR TOOLKIT.ADUS TOOLKIT.ALGR TOOLKIT.AUSR
TOOLKIT.CONN TOOLKIT.DELD TOOLKIT.DELG TOOLKIT.DELU TOOLKIT.LDSD TOOLKIT.LGRP
TOOLKIT.LUSR TOOLKIT.PEMT TOOLKIT.REMV TOOLKIT.RACL TOOLKIT.RALT TOOLKIT.RDEF
TOOLKIT.RDEL TOOLKIT.RLST TOOLKIT.USRL TOOLKIT.USRA TOOLKIT.USRD)
```

2. Give the users access to the commands that they can be authorized to use. For example, ADDUSER, ALTUSER, DELUSER.

When you use the software, use the authority of the user signed on at the terminal to determine which commands and functions can be run. If a transaction that is not running at a terminal uses the software to perform a RACF function, the authority of the DEFAULT USERID is used to determine which commands and functions can be executed. The authority of the default user is also used for terminals that are not explicitly signed-on.

Permit users to each command as required. Examples:

```

PERMIT TOOLKIT.LUSR CLASS(TCICSTRN) ID(USER01) ACCESS(READ)
PERMIT TOOLKIT.AUSR CLASS(TCICSTRN) ID(USER01) ACCESS(READ)
PERMIT TOOLKIT.LDSD CLASS(TCICSTRN) ID(USER01) ACCESS(READ)
PERMIT TOOLKIT.LGRP CLASS(TCICSTRN) ID(USER01) ACCESS(READ)
PERMIT TOOLKIT.CONN CLASS(TCICSTRN) ID(USER01) ACCESS(READ)
PERMIT TOOLKIT.REMV CLASS(TCICSTRN) ID(USER01) ACCESS(READ)
PERMIT TOOLKIT.ADUS CLASS(TCICSTRN) ID(USER01) ACCESS(READ)
PERMIT TOOLKIT.PEMT CLASS(TCICSTRN) ID(USER01) ACCESS(READ)

```

Only commands that a user has access to are displayed on the panel. For example, if a user has access to only the LUSR command, none of the other commands are displayed.

3. Define the subsequent levels of authority within those commands that the user can access. For example, after you give a user access to the ALTUSER command, you must specify which users they can alter.

**Note:** For a list of internal security resource classes for the commands, see “Internal Security Resource Listing.”

Most commands require one or more subsequent levels of authority to be given to a user. These next levels determine which users, groups, and resources the user can access or control after you have given the user access to a command.

For example, if a user is granted access to ALTUSER, the IDs to which the user has access must also be defined. This is done by defining the default group of users, prefixed with AUSR, as a resource. A user can then be granted access to this resource.

Table 5 shows examples of definitions that are required to give this type of capability.

*Table 5. Example resource definitions used within RACF commands*

USERID	DEFAULT GROUP	RSRCLASS
USER01	TECHSUPP	TCICSTRN
USER02	USERSUPP	TCICSTRN
USER03	QUALCNTL	TCICSTRN
USER04	AUDIT	TCICSTRN

The RACF command in this example:

```
RDEFINE TCICSTRN (AUSR.TECHSUPP AUSR.USERSUPP AUSR.QUALCNTL AUSR.AUDIT)
```

Then, permit users to each group:

```

PERMIT AUSR.TECHSUPP CLASS(TCICSTRN) ID(USER01) ACCESS(READ)
PERMIT AUSR.QUALCNTL CLASS(TCICSTRN) ID(USER01) ACCESS(READ)
PERMIT AUSR.AUDIT CLASS(TCICSTRN) ID(USER01) ACCESS(READ)

```

With these definitions, USER01 can perform an ALTUSER on all user IDs whose default group is TECHSUPP, QUALCNTL, or AUDIT, but not on user IDs whose default group is USERSUPP, or any other default group.

## Internal Security Resource Listing

zSecure CICS Toolkit uses resource names for its own internal security.

Table 6 lists the resource names that zSecure CICS Toolkit uses for its own internal security.

*Table 6. zSecure CICS Toolkit: Internal security resource list*

Resource name	What it allows if a user has read access
TOOLKIT.ADGR	Allows a user to run the ADDGROUP command of zSecure CICS Toolkit.

Table 6. zSecure CICS Toolkit: Internal security resource list (continued)

Resource name	What it allows if a user has read access
TOOLKIT.ADUS	Allows a user to run the ADDUSER command of zSecure CICS Toolkit.
TOOLKIT.ALGR	Allows a user to run the ALTGROUP command of zSecure CICS Toolkit.
TOOLKIT.AUSR	Allows a user to run the ALTUSER command of zSecure CICS Toolkit.
TOOLKIT.ACIC	Allows a user to manage CICS segments by the ALTUSER command of zSecure CICS Toolkit.
TOOLKIT.ATSO	Allows a user to manage TSO segments by the ALTUSER command of zSecure CICS Toolkit.
TOOLKIT.AOMV	Allows a user to manage OMVS segments by the ALTUSER command of zSecure CICS Toolkit.
TOOLKIT.AWRK	Allows a user to manage WORKATTR segments by the ALTUSER command of zSecure CICS Toolkit.
TOOLKIT.CONN	Allows a user to run the CONNECT command of zSecure CICS Toolkit.
TOOLKIT.DEKD	Allows a user to run the DELETE DATASET command of zSecure CICS Toolkit.
TOOLKIT.DELG	Allows a user to run the DELETE GROUP command of zSecure CICS Toolkit.
TOOLKIT.DELU	Allows a user to run the DELETE USER command of zSecure CICS Toolkit.
TOOLKIT.DUPE	Allows a user to sign on at a second terminal but forces a signoff at the original terminal. Any transaction that is currently attached at the original terminal is also purged.
TOOLKIT.GPID	Allows a userid to be used as a group-id so that more than one user can use it to sign on to CICS.
TOOLKIT.LDSD	Allows a user to run the LISTDATASET command of zSecure CICS Toolkit.
TOOLKIT.LGRP	Allows a user to run the LISTGROUP command of zSecure CICS Toolkit.
TOOLKIT.LUSR	Allows a user to run the LISTUSER command of zSecure CICS Toolkit.
TOOLKIT.PEMT	Allows a user to run the PERMIT command of zSecure CICS Toolkit.
TOOLKIT.RACL	Allows a user to run the RACLINK command of zSecure CICS Toolkit.
TOOLKIT.RALT	Allows a user to run the RALTER command of zSecure CICS Toolkit.
TOOLKIT.RDEF	Allows a user to run the RDEFINE command of zSecure CICS Toolkit.
TOOLKIT.RDEL	Allows a user to run the RDELETE command of zSecure CICS Toolkit.
TOOLKIT.REMV	Allows a user to run the REMOVE command of zSecure CICS Toolkit.
TOOLKIT.RLST	Allows a user to run the RLIST command of zSecure CICS Toolkit.

Table 6. zSecure CICS Toolkit: Internal security resource list (continued)

Resource name	What it allows if a user has read access
TOOLKIT.USRL	Allows a user to list usrdata fields as part of the USRDAT command of zSecure CICS Toolkit.
TOOLKIT.USRA	Allows a user to add and update usrdata fields as part of the USRDAT command of zSecure CICS Toolkit.
TOOLKIT.USRD	Allows a user to delete usrdata fields as part of the USRDAT command of zSecure CICS Toolkit.
TOOLKIT.SPEC	Gives a user the equivalent of SPECIAL, when running RACF commands from zSecure CICS Toolkit. Users who have access to TOOLKIT.SPEC are allowed access to all resources within a specific command. For instance, if users have access to the LISTUSER command and TOOLKIT.SPEC, they can list <i>any</i> user and are not restricted by the LUSR.dfltgrp definitions. While TOOLKIT.SPEC gives a user access to <i>all</i> resources within a command, it does not give the user access to the command. In this example, for the user to use the LISTUSER command, the user still requires access to TOOLKIT.LUSR. This applies only to RACF commands issued by the zSecure CICS Toolkit interface.
TOOLKIT.SVC	Allows a region to use the zSecure CICS Toolkit SVC.
ADGR.grpname	Defines the groups for which a user can issue an ADDGROUP.
ADUS.dfltgrp	Defines the default groups for which a user can issue an ADDUSER.
ALGR.grpname	Defines the default groups for which a user can issue an ALTGROUP.
AUSR.dfltgrp	Defines the groups for which a user can issue an ALTUSER.
CONN.grpname	Defines the groups for which a user can issue a CONNECT.
DELD.hlq	Defines the high-level-qualifier of data set names for which a user can issue a DELETE DATASET. See “Deleting a data set (DELETE DATASET command)” on page 49 for more details.
DELG.grpname	Defines the groups for which a user can issue a DELETE GROUP.
DELU.dfltgrp	Defines the default groups for which a user can issue a DELETE USER.
LGRP.grpname	Defines the groups for which a user can issue a LISTGROUP.
LUSR.dfltgrp	Defines the default groups for which a user can issue a LISTUSER.
PEMT.dfltgrp / grpname	Defines the default groups/groups for which a user can issue a PERMIT.
PEMX.cdtclass	Defines the general resource classes, including the DATASET class, for which a user can issue a PERMIT, if the resource is not in a class that is defined in the CICS SIT.
PSWD.dfltgrp	Gives authority to change a user's PASSWORD within the specified default groups.
RACL.dfltgrp	Defines the groups for which a user can issue a RACLINK.



Table 6. zSecure CICS Toolkit: Internal security resource list (continued)

Resource name	What it allows if a user has read access
RALT.cdtclass	Defines the general resource classes for which a user can issue an RALTER.
RDEF.cdtclass	Defines the general resource classes for which a user can issue an RDEFINE.
RDEL.cdtclass	Defines the general resource classes for which a user can issue an RDELETE.
REMV.grpname	Defines the groups for which a user can issue a REMOVE.
RLIST.cdtclass	Defines the general resource classes for which a user can issue an RLIST.
SECL.nnn	Specifies the SECLEVEL a user can specify for userids for which the user is authorized. The <i>nnn</i> is the SECLEVEL number from 001 to 254.
USRU.dfltgrp	Defines the default groups for which a user can display USRDATA fields.
USRN.usrdata-name	Defines the <i>usrdata-names</i> that can be displayed or added/updated/deleted by the user.

## Automatic assignment of USS UIDs (OMVS AUTOUID)

If you want to use the automatic assignment of unique USS UIDs, you must ensure that the following requirements are fulfilled.

- The RACF database must be enabled for Application Identity Mapping. The minimum stage that is required is stage 2.
- The profile BPX.NEXT.USER in the FACILITY class must be defined, with appropriate APPLDATA. The details are described in the *RACF Security Administrator's Guide*. See the chapter *RACF and z/OS UNIX*.
- RACF TSO command usage of AUTOUID also requires that profile SHARED.IDS is defined and that the UNIXPRIV resource class is active and RACLISTed.

## Automatic creation of home directories (OMVS MKDIR)

If you want to use the automatic creation of the home directory when you create an OMVS segment for a user, the following extra RACF requirements must be fulfilled.

- The CICS region userid must have an OMVS segment that assigns a UID (or you must have enabled the use of a DEFAULT UID).
- The current group of the CICS region userid must have an OMVS segment that assigns a GID. Or, you must have enabled the use of a DEFAULT GID.
- The CICS region userid must have sufficient access to create the home directories. It can be implemented by one of the following methods:

### UID=0

This option gives the CICS region full control over the entire USS environment. It is acceptable during initial testing, but is not suitable for a regular production environment.

### UNIXPRIV

You can also grant CONTROL access to the UNIXPRIV profile SUPERUSER.FILESYS because it gives the CICS region READ/WRITE access to all files in the file system. Do not use this option for production usage.

**WRITE access to the directory where user home directories must be created.**

This option gives the CICS region exactly the authorization that is required. It is the preferred option.

When you select not to grant UID=0 to the CICS region userid, you must also give the CICS region the authority to set the correct owner for the newly created home directories. Failure to do so might render the new home directories unusable to the intended users. Set the correct owner with the CHOWN command, which typically requires authorization:

**UNIXPRIV SUPERUSER.FILESYS.CHOWN**

Granting READ access to this profile allows changing the owner (userid = dfltgrp) for all files in the system. Because it is a rather powerful authorization, it is better not to use this approach.

**UNIXPRIV CHOWN.UNRESTRICTED**

This discrete profile enables all users to change the owner of files/directories that they own. It is like the way RACF behaves for conventional data set profiles. Because zSecure CICS Toolkit creates the home directories initially with the CICS region as owner, it is authorized to change the owner to the intended user.

---

## **zSecure CICS Toolkit restart**

Normally, after doing all the required updates and definitions for the installation, you must restart the CICS system to activate all the changes. For the initial installation, the restart is required to activate, for example, the SRT definitions. You can combine this activation with the system IPL that is required to pick up the definition of the zSecure CICS Toolkit SVC.

zSecure CICS Toolkit internally uses MVS subtasks. Normally, these subtasks are started during CICS initialization by a PLT program. These MVS subtasks are detached again at CICS termination by a second PLT program. Restarting CICS is the preferred method for this part of the installation process.

If all the required definitions are in place, but restarting CICS is not possible, you can use the following alternative method instead.

**Attention:** If you use this process to do the initial activation of zSecure CICS Toolkit, you must also use this process to deactivate it before you shut down the CICS system. Failure to do so results in A03 abends, probably followed by a system memory dump.

If the zSecure CICS Toolkit SVC is not installed, and the CICS SRT definition is not activated, attempts to install the software subtasks might result in a terminating abend of the CICS Started Task.

Sometimes, an error occurs during execution of the zSecure CICS Toolkit programs. Some of these errors result in the termination of one of the MVS subtasks that are used for zSecure CICS Toolkit processing. Current versions of the zSecure CICS Toolkit provide a transaction for stopping and restarting the MVS subtasks. Previous versions require a manual process. This manual process is described in “Manual restart of the zSecure CICS Toolkit subtasks” on page 21.

## RTST transaction definition

The RTST transaction can be defined to run the CRTKSTRT program. This program performs the necessary functions to stop and restart the zSecure CICS Toolkit subtasks. It also provides a function for refreshing the various zSecure CICS Toolkit programs, maps, and parameter modules.

When the RTST transaction is run, the following panel is displayed:

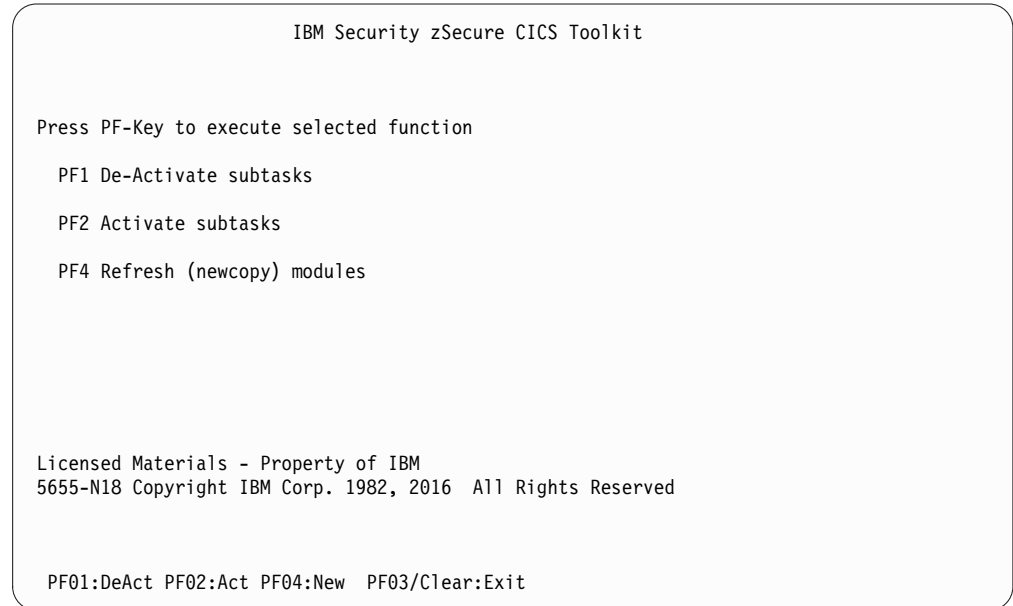


Figure 1. RTST Transaction panel

This panel provides the following functions:

- PF1** The currently active subtasks are terminated. Because this is done by a regular stop request to each task, it might take several seconds to run this function.
- PF2** Start the subtasks. Before you start the subtasks, the program verifies that the subtasks are currently not active.
- PF4** Obtain a new copy of the zSecure CICS Toolkit programs, maps, and parameter modules. This process first ensures that the modules are no longer in use. It involves releasing some permanently resident modules. Also, the MVS subtasks are verified not to be active now. If any other terminal user is concurrently using the zSecure CICS Toolkit interface, the user might experience unpredictable results.

Because the RTST transaction does not perform any internal authorization verification, access to the transaction must be controlled. Access must be restricted to those people who must be able to stop and start the zSecure CICS Toolkit subtasks, or refresh the modules.

## Manual restart of the zSecure CICS Toolkit subtasks

When the RTST transaction is not available, you might also use the manual process to stop and start the zSecure CICS Toolkit subtasks. This process involves the use of the CECI transaction.

Run the following two commands:

```
CECI LINK PROG(CQTPKDTCH)
CECI LINK PROG(CQTPLT00)
```

When running these transactions, the terminal user must have access to CSML in DCICSDCT. If you chose to configure a different DESTID in CQTPCNTL, CSML must be replaced by your selected DESTID.

---

## Use of the CICS Transaction Server with zSecure CICS Toolkit

There are several specifications to be aware of if you are running the CICS Transaction Server.

Be aware of the following specifications:

- The RSRCLASS parameter in CQTPCNTL must specify the correct RACF resource class. It does not default to the XTRAN parameter defined in the SIT.
- If you want to use the DUPEUSER capabilities of zSecure CICS Toolkit, it is best to start program CQTPSNP0. You can start this program in one of the following ways:
  - By doing an XCTL or LINK to the program
  - By starting the program as a transaction from your own signon program
  - From the signon exit point

The major purpose for using the DUPEUSER support is the ability to automatically cancel an existing session when you log on to a second terminal.

**Note:** Starting with CICS TS version 2.1, the behavior of the EXEC CICS SIGNON command changed. The new ID becomes effective only after termination of the current transaction. Module CQTPSNP0 in version 1.8.1 has more functionality to obtain the ID of the user that was signed on and verify the access to the TOOLKIT.DUPE and TOOLKIT.GPID resources. Other zSecure CICS Toolkit functions use the authorization of the user that started the signon transaction, and do not use the new ID.

---

## Globalization

The BMS mapsets that are used by zSecure CICS Toolkit are provided as samples in SCQTSAMP.

An installation might modify these BMS mapsets, if the new mapsets are compatible with the existing mapsets. The copybook (symbolic map) resulting from the modification must be identical to the unmodified copybook that is used by the zSecure CICS Toolkit program modules. It can be done by retaining all fields and their lengths. The only parts that must be changed are field attributes (for example to suppress display of a certain field), or field initial values. Although CICS might allow other parts of the BMS maps to be changed, it is not supported in zSecure CICS Toolkit. Changed mapsets must be translated with the standard DFHMAPS procedure.

---

## Chapter 3. Parameters for zSecure CICS Toolkit

The parameter module CQTPCNTL is used to define the parameters that zSecure CICS Toolkit uses.

After the parameters are set, use the RTCK transaction to check them. Errors are displayed with an error message. Resolve all errors before you implement zSecure CICS Toolkit.

**Note:** The contents of CQTPCNTL are revised and are not compatible with the contents of member CRTKCNTL used in the precursor product Consul zToolkit.

The following is an example of setting up the CQTPCNTL parameters:

```
CQTPCNTL CSECT
CQTPCNTL AMODE 31
CQTPCNTL RMODE ANY
*
EXITPGM DC CL8'      '
DESTID  DC CL4'CSML'
CICSAPPL DC CL8'IGNOREIT'
RSRCLASS DC CL8'TCICSTRN'
CMNDPFX DC CL8'TOOLKIT.'
SMFUID  DC CL8'      '
SVCNUM  DC CL3'222'
DUPEUSER DC CL1'2'
RACFCMND DC CL1'Y'
LOGGING DC CL1'Y'
PEMTALL DC CL1'Y'
LGDFLTU DC CL1'N'
END
```

Use the supplied CQTIJCNTL sample job to apply the update from SMP/E. If that is not possible, you can also use the IBM supplied procedures DFHASMVS and DFHLNKVS to assemble and link the module. The parameters must be in the order as shown in the example. The definitions in the example are the defaults in the installation modules. You can customize the parameters for your installation.

---

### Parameter Descriptions

The following list describes the parameters present in CQTPCNTL.

#### EXITPGM

The name of a program that is to receive control whenever the main zSecure CICS Toolkit transaction (normally RTMM) terminates. If you do not want to use an exit program, specify blanks. For a complete description of the exit program features, see Chapter 6, “zSecure CICS Toolkit exit points specifications,” on page 85.

#### DESTID

The destination zSecure CICS Toolkit uses for writing run time messages. The default is CSML. It can be changed to any other entry but must conform to the definitions for CSML as specified in the CICS Resource Definition manual.

#### CICSAPPL

This parameter is used as a prefix for the transaction names when you use the RSRC API interface for application security. If IGNOREIT is specified, or

the parameter is left blank, the parameter is ignored. If the parameter is coded, it can be up to eight bytes in length and must conform to RACF naming conventions. For more information, see Chapter 4, “Application security management,” on page 29.

### **RSRCLASS**

The RACF resource class that is used by zSecure CICS Toolkit. zSecure CICS Toolkit uses this class when it checks resources. It can be any of the classes that are defined in the SIT to CICS so that it is RACLISTed by CICS. It must be the name of the MEMBER class, not the group class (for example, TCICSTRN and *not* GCICSTRN). If you leave it blank or specify an invalid class, zSecure CICS Toolkit fails to initialize. The resource class to be used by the API when performing resource access checking (extended) can be defined as a parameter being passed to the API and overrides this definition for those resource checks. See the API documentation for further information.

### **CMNDPFX**

The zSecure CICS Toolkit commands that a user can use, and that are displayed when the user enters the zSecure CICS Toolkit transaction (RTMM), are determined by the RACF definitions as outlined in step 1 in “Defining the RACF profiles” on page 14. These definitions all have a prefix of T00LKIT.. Using the CMNDPFX parameter, it is possible to specify a different prefix. However, this prefix is only in effect when the zSecure CICS Toolkit transaction is being used. When zSecure CICS Toolkit is being accessed through the API a prefix of T00LKIT. is always used.

For example, if CMNDPFX is CICSONE and a user wants to do an ADDUSER, the user must have:

- Access to CICSONE.ADUS if they are using the zSecure CICS Toolkit transaction
- Access to T00LKIT.ADUS if they are doing the ADDUSER through the API

This provides more flexibility to do things with the API and restrict the options on the zSecure CICS Toolkit transaction display.

This parameter must be eight characters, must end with a period (.) and cannot contain any blanks. The default is T00LKIT.

### **SMFUID**

When zSecure CICS Toolkit creates an SMF record that reflects a change to the RACF database, the ID of the user logged on at the terminal is used in the SMF records. However, there might be situations where you want the ID in the SMF record to be different. For instance, if the SMF record is being shipped to another system to update another RACF database, the user might not have the required authority on that system to allow the change to occur. You can specify in SMFUID a different ID that is used in the SMF records created by zSecure CICS Toolkit.

If this parameter is left blank, the ID of the terminal user is used in the SMF records.

**Note:** This value is different from the SMF80UID field. The SMF80UID field is always set to the value T00LKIT\* to indicate that the record was produced as part of a zSecure CICS Toolkit function.

### **SVCNUM**

The SVC number that was assigned to the zSecure CICS Toolkit SVC.

## DUPEUSER

Use this parameter to control user signons.

- If this parameter is coded as 0, no checking takes place when a user signs-on.
- If it is a 1, zSecure CICS Toolkit checks to see whether the same RACF user ID is signed on at another terminal. If it is, the duplicate signon is disallowed and the user ID is signed off.
- Coding this parameter as 2 has the same effect as 1 with the exception that the terminal is logged-off as well (the equivalent of performing doing a CSSF LOGOFF).

Checking for duplicate user IDs is only done in the terminal that owns region when it is run in an MRO environment. If a user uses the CRTE transaction to route to an application region, no checking takes place in that region.

### Permitting users:

When the DUPEUSER parameter is specified as 1 or 2, you have the option of allowing specific user IDs to be used as group IDs, or you can specify that certain IDs can log on at a second terminal but will be logged off at the terminal at which they are currently signed on. To use this capability, define the following resources to RACF. (This example assumes that the XTRAN parameter in the SIT is CICSTRN):

```
RDEFINE TCICSTRN (TOOLKIT.GPID TOOLKIT.DUPE) UACC(NONE)
```

To allow a user ID to be used as a group ID (that is, it can be shared by multiple users), permit it access to TOOLKIT.GPID.

For example:

```
PERMIT TOOLKIT.GPID CLASS(TCICSTRN) ID(GROUP01) ACCESS(READ)
```

This allows GROUP01 to be used by more than one user when they sign on to CICS.

To allow a user to log on at a second terminal, but to be forced off the first terminal, permit it access to TOOLKIT.DUPE.

For example:

```
PERMIT TOOLKIT.DUPE CLASS(TCICSTRN) ID(USER01) ACCESS(READ)
```

This enables USER01 to sign on at a second terminal but forces a signoff at the first terminal at which the user was signed on.

If a user has access to both TOOLKIT.DUPE and TOOLKIT.GPID, TOOLKIT.GPID takes precedence.

## RACFCMND

If zSecure CICS Toolkit is only used by applications for internal security checking, controlling signons, or both, it does not require all of its subtasks in the region. These subtasks are only required if you use zSecure CICS Toolkit or the API to run RACF commands. It saves approximately 40K per region and results in an even greater savings if you have zSecure CICS Toolkit in multiple regions.

Specify Y if you use the commands or N if you do not.

## LOGGING

When an application program uses the API to check resource access, zSecure CICS Toolkit produces SMF records based on the AUDIT

parameters for the resource. In other words, when a user does not have access to the resource and auditing for failures is turned on or AUDIT is all, zSecure CICS Toolkit produces SMF records.

This can result in a large amount of unwanted SMF records being produced. If you want zSecure CICS Toolkit to produce these records, specify Y. If you want them to be suppressed, specify N. For the RSRC and RSRX functions, it is also possible to specify S. This results in suppression of possible access violation messages, while still creating SMF records about these violations. For all other functions, S is interpreted the same as Y.

This parameter setting does not apply to SMF records produced by zSecure CICS Toolkit when a user alters or updates a profile on the RACF database.

#### **PEMTAL**

This parameter is used to control the scope of the PERMIT command. The PERMIT command can be restricted to giving access only to resources that are in a class that is specified in the SIT (for example; XDCT, XFCT, XJCT, SPPT, XTRAN), and is RACLISTED by CICS. In this case specify N. To allow the PERMIT command to be used for ALL resource classes, including DATASET, specify Y. For more information, see the PEMX and PEMT definitions in the “Internal Security Resource Listing” on page 16 and “Granting or removing access to a resource (PERMIT command)” on page 71.

#### **LGDFLTU**

This parameter is used to control the display of USERIDs as a subfunction of the LISTGROUP function. If this value is set to N, all users that are connected to the specified GROUP are shown. If this value is set to Y, only users that are connected to the DFLTGRP of the terminal user are shown.

---

## **CQTPCNTL parameter values verification**

Each region can have its own version of CQTPCNTL. However, select the CICSAPPL value carefully to avoid errors.

After coding CQTPCNTL, use transaction **RTCK** to verify its parameters. Perform this action before zSecure CICS Toolkit is implemented.

Display the RTCK transaction panel. The following panel is displayed and shows any errors that might have occurred.



```

                                IBM Security zSecure CICS Toolkit
                                CICS level = 0690 Toolkit level = HCQT221

Exit program                    Has not been defined to CICS
Destination                     CSML      Destination id for messages
Prefix (Appl security)          IGNOREIT Application prefix for security
Resource class                   TCICSTRN Member class name for Toolkit

LG users in DFLTGRP only        N        N=All users, Y=Only matching DFLTGRP
Duplicate Signon                 2        0=Yes,1=No(Signoff),2=No(Logoff)
Toolkit SVC                       222     Required for RACF commands
RACF commands                     Y        Region may issue RACF commands
Logging                           Y        SMF Records if audit specified
PEMTALL                           Y        Allow permits for all classes

PF03/Clear=Quit      Check Highlighted fields for error messages

```

Figure 2. RTCK transaction panel

Select the function for the task you want to accomplish:

- Press PF01 from the main menu to view YOUR access to zSecure CICS Toolkit commands.
- Press PF02 from the main menu to view the zSecure CICS Toolkit programs, their status and PTF level.
- Press PF04 from the main menu to view the zSecure CICS Toolkit subtasks, their status and PTF level.
- Press CLEAR or PF03 to terminate the transaction. Any other key re-displays the main menu.

The following two panels show example output for the programs and subtasks.

**Sample partial panel showing the output for the program status:**

```

Program  PTFlevl ST Program  PTFlevl ST Program  PTFlevl ST Program  PTFlevl ST
CQTPAPRM      OK CQTPCNTL      OK CQTPMSGE      OK CQTPLT00 HCQT221 OK
CQTPATCH HCQT221 OK CQTPDTCH HCQT221 OK CQTPCHEK HCQT221 OK CQTPAPI0 HCQT221 OK
CQTP0000 HCQT221 OK CQTP0010 HCQT221 OK CQTP0020 HCQT221 OK CQTP0030 HCQT221 OK
CQTP0040 HCQT221 OK CQTP0041 HCQT221 OK CQTP0042 HCQT221 OK CQTP0043 HCQT221 OK
CQTP0044 HCQT221 OK CQTP0050 HCQT221 OK CQTP0055 HCQT221 OK CQTP0056 HCQT221 OK
CQTP0058 HCQT221 OK CQTP0059 HCQT221 OK CQTP0060 HCQT221 OK CQTP0070 HCQT221 OK
CQTP0080 HCQT221 OK CQTP0081 HCQT221 OK CQTP0082 HCQT221 OK CQTP0083 HCQT221 OK
CQTP0084 HCQT221 OK CQTP0086 HCQT221 OK CQTP0090 HCQT221 OK CQTP0091 HCQT221 OK
CQTP0100 HCQT221 OK CQTP0110 HCQT221 OK CQTP0111 HCQT221 OK CQTP0112 HCQT221 OK
CQTP0113 HCQT221 OK CQTP0114 HCQT221 OK CQTP0120 HCQT221 OK CQTP0130 HCQT221 OK
CQTP0131 HCQT221 OK CQTP0132 HCQT221 OK CQTP0133 HCQT221 OK CQTP0134 HCQT221 OK
CQTP0140 HCQT221 OK

```

Figure 3. zSecure CICS Toolkit: Program status output

**Sample partial panel showing the output for the subtask status:**

Program	PTFlevl	ST	Program	PTFlevl	ST	Program	PTFlevl	ST	Program	PTFlevl	ST
CQTSUBS	HCQT221	AV	CQTS000	HCQT221	AV	CQTS010	HCQT221	AV	CQTS020	HCQT221	AV
CQTS030	HCQT221	AV	CQTS041	HCQT221	AV	CQTS042	HCQT221	AV	CQTS043	HCQT221	AV
CQTS044	HCQT221	AV	CQTS050	HCQT221	AV	CQTS055	HCQT221	AV	CQTS056	HCQT221	AV
CQTS058	HCQT221	AV	CQTS059	HCQT221	AV	CQTS060	HCQT221	AV	CQTS070	HCQT221	AV
CQTS081	HCQT221	AV	CQTS082	HCQT221	AV	CQTS083	HCQT221	AV	CQTS084	HCQT221	AV
CQTS086	HCQT221	AV	CQTS090	HCQT221	AV	CQTS100	HCQT221	AV	CQTS111	HCQT221	AV
CQTS112	HCQT221	AV	CQTS113	HCQT221	AV	CQTS114	HCQT221	AV	CQTS120	HCQT221	AV
CQTS131	HCQT221	AV	CQTS132	HCQT221	AV	CQTS133	HCQT221	AV	CQTS134	HCQT221	AV
CQTS135	HCQT221	AV	CQTS136	HCQT221	AV	CQTS140	HCQT221	AV			

Figure 4. zSecure CICS Toolkit: Subtask status output

---

## Chapter 4. Application security management

zSecure CICS Toolkit lets you request access to multiple resources with a single system call.

Traditionally, applications that run under CICS use some form of their own internal security. Even though CICS and external security manager might control access to the transactions, access to subfunctions of those transactions was inevitably maintained by the application.

This strategy causes the following undesirable results:

- Each application has a different form of security. Any user that requires access to multiple applications must frequently request access from multiple control points.
- Delays occur in granting access, and there is a lack of control as to who has what.
- Workloads increase.
- A single user often needs multiple IDs.

Since several years, applications also have the option to issue an EXEC CICS QUERY SECURITY function. However, if an application must establish authorization to many resources (for example, for determining which options to display on a selection panel), considerable time might be involved in requesting the authorization. In that situation, zSecure CICS Toolkit provides advantages. With the zSecure CICS Toolkit RSRC or RSRX functions, you can request access to multiple resources with a single system call.

---

### Operator ID or OPID check

The traditional method that is employed by various applications to do security checking is based on the 3-byte operator ID of a user. This ID is checked against a table or a file, which contains an array or matrix of functions that this user, or operator ID, can perform.

Such internal security methods leads to numerous exposures. CICS provides no capability to ensure that operator IDs are unique. And the three characters that are used for the OPID are often not sufficient to accommodate all users. By using zSecure CICS Toolkit for application security checking, you can overcome these exposures. Other added advantages are the centralization of security definitions, and one security system (RACF) being used by all applications.

---

### Application conversion

To use zSecure CICS Toolkit to check the OPID, you must convert existing applications.

This requires a coding change to the application that now checks security. If the application is a package, contact the vendor and create exit point within the package, where the security checking takes place. Use the zSecure CICS Toolkit API for security checksto use RACF security facilities. With the API, more than 2000 resources can be checked with one call. For more information about using the API and its multiple functions and capabilities, see Chapter 7, "Application

programming interface (API),” on page 87. Application programs can link to CQTPAPI0, with a parameter that specifies the function that the user is attempting to access. CQTPAPI0 then checks to see whether the user has access to this function and return to the calling application with the appropriate return code. This example shows how an application would use CQTPAPI0 and the definitions that would be made to RACF and the possible return codes from CQTPAPI0.

If a transaction that is run by a user had multiple functions, it would probably provide the user with a menu panel. The user would then select one of those options. In the following example, the user ran transaction ABCD. The application then gave the user an option menu as follows:

```

OPTION DESCRIPTION
1      READ PAYROLL MASTER RECORDS
2      UPDATE PAYROLL MASTER RECORDS
3      ADD PAYROLL MASTER RECORDS
4      DELETE PAYROLL MASTER RECORDS

ENTER OPTION : _

```

Figure 5. Option menu example

The same transaction, ABCD, performs all the functions, but not all users must have access to all of them. To define the individual functions to RACF, an alias is assigned to each function. These functions can be READ for option one, UPDT for option two, ADDS for option three, and DELT for option four. After the user selected one of these options, the application program would LINK to CQTPAPI0 with a COMMAREA. This is described in the following example, which would contain one of the aliases (READ, UPDT, ADDS, or DELT). The return code would be set by CQTPAPI0.

```

*
*           MVC API_FUNC,=CL4'RSRC'           MOVE FUNCTION CODE FOR
*                                           A RESOURCE CHECK
*           MVC  API_RESOURCE_NAME,=CL13'READ' OPTION REQUESTED BY USER
*           MVI  API_END,X'FF'                END OF RESOURCE NAMES
*
*           EXEC CICS LINK PROGRAM('CQTPAPI0') COMMAREA(API-COMM)
*
*           CLI  API_RESOURCE_RC,X'00'        ACCESS ALLOWED ?
*           BE   ACCESSOK                     YES
*           B    ERROR                         NO
*
*
* API_COMM   DS 0CL99
* API_FUNC   DS CL4                           FUNCTION CODE
* API_RC     DS XL1                           RETURN CODE
* API_MSG    DS CL79                           MESSAGE AREA
*
* API_RESOURCE_NAME DS CL13                   RESOURCE NAME
* API_RESOURCE_RC   DS XL1                   RACF RETURN CODE
* API_END          DS XL1                   X'FF' END OF LIST
*

```

The return codes would be a one-byte hexadecimal code with the following meanings:

RETURN CODE MEANING:

X '00' Access is allowed to resource

X '94' The resource or classname is not defined to RACF

- X '08' The user or group is not authorized to use the resource
- X '0C' RACF is not active
- X '10' Fracheck installation exit error
- X '14' RACF is not installed or at the wrong level

If the COMMAREA was not large enough, the APU\_RC field would contain X '02'.

## Alias definitions

The aliases, or mnemonics, that are used for application security checking, are defined to RACF in the same way as transactions.

When you define application aliases, they must be prefixed with the value defined in CQTPCNTL for CICSAPPL. If CICSAPPL is IGNOREIT or blank, a 13-byte alias would then be defined to RACF without a prefix. It must be padded to the right with blanks. The RACF definitions must be entered with the same RACF class name as the one used for zSecure CICS Toolkit. It is as specified in the RSRCLASS parameter of CQTPCNTL.

If the API\_FUNCTION is RSRX, the resource name can be up to 246 characters in length. However, it must not be greater than the maximum defined for the resource class in the Class Descriptor Table.

Assuming the value that is coded for CICSAPPL was PRODAPPL, and the RSRCLASS value of TCICSTRN, the aliases would be defined as follows:

```
RDEFINE TCICSTRN (PRODAPPL.READ PRODAPPL.UPDT PRODAPPL.ADDS PRODAPPL.DELT)
```

Users would then be permitted to the resource as required:

```
PERMIT PRODAPPL.READ CLASS(TCICSTRN) ID(USER01) ACCESS (READ)
PERMIT PRODAPPL.UPDT CLASS(TCICSTRN) ID(USER02) ACCESS (READ)
PERMIT PRODAPPL.ADDS CLASS(TCICSTRN) ID(USER03) ACCESS (READ)
PERMIT PRODAPPL.DELT CLASS(TCICSTRN) ID(USER03) ACCESS (READ)
```

No definitions are required in any CICS tables. After the RACF profiles are created, zSecure CICS Toolkit is ready to check application security.

---

## Simple application security interface

zSecure CICS Toolkit provides a direct interface as alternative to the full API provided by CQTPAPI0 as described in Chapter 5.

The simple interface provides only two functions:

- Access the user profile, current group profile, and instdata of the currently signed on user
- Check access to a resource.

These two functions can also be accessed from modern CICS services like EXEC CICS ADDRESS ACEE and EXEC CICS QUERY SECURITY. To assist existing customers in their migration to these standard CICS services, the CQTPAPPL interface is provided.

## User information retrieval

To use the user information retrieval function, you must provide a commarea of at least 22 bytes. If the commarea is larger than 286 bytes, the DFTLGRP and the INSTDATA for the user are returned as well.

To use this function, the first 4 bytes of the commarea must contain the value '????' (that is, four question marks). Here is an example of how to use this function:

```

MVI   COMMAREA,' '           Clear commarea
MVC   COMMAREA+1(L'COMMAREA-1),COMMAREA
MVC   COMM_FUNC,=CL4'????'   Move function code
EXEC  CICS_LINK PROGRAM('CQTPAPPL')
      COMMAREA(COMMAREA)
      LENGTH(COMALEN)
CLI   COMM_RC,X'00'           Ok?
*
*   Further processing
*
COMMAREA DS 0CL286           Space for COMMAREA
COMM_FUNC DS CL4             Function code ???
COMM_RSVD DS CL9            Unused
COMM_RC   DS XL1            Return code
COMM_USER DS CL8            Userid
COMM_GRP  DS CL8            Group
COMM_IDL  DS XL1            Length of instdata
COMM_IDA  DS CL255          Instdata
COMALEN   DC AL2(*-COMMAREA) Length of COMMAREA

```

The return code has two possible values:

- X'00' user information is returned
- X'10' Invalid or no ACEE present.

If the commarea is too small, the value '\*' (asterisk) is returned in the first byte of COMM\_RESN. If the commarea has length zero, or is absent, no information is returned.

## Resource access verification

To use the resource access verification function, you must provide a commarea of at least 14 bytes, containing the name of a resource.

The access of the current user is verified to be at least READ and the COMM\_RC is set accordingly. The resource name must be left-aligned and padded with blanks. When processing the request, the name of the resource is prefixed with the value specified in CICSAPPL in CQTPCNTL. If the value of CICSAPPL is IGNOREIT or blank, no prefix is applied. The resource name as used in the access verification request is returned in the field COMM\_RESN. Here is an example of how to use this function:

```

MVI   COMMAREA,' '           Clear commarea
MVC   COMMAREA+1(L'COMMAREA-1),COMMAREA
MVC   COMM_RESN,=CL13'PAYROLL' Move resource name
EXEC  CICS_LINK PROGRAM('CQTPAPPL')
      COMMAREA(COMMAREA)
      LENGTH(COMALEN)
CLI   COMM_RC,X'00'           Access?
*
*   Further processing
*
COMMAREA DS 0CL14           Space for COMMAREA
COMM_RESN DS CL13          Resource name
COMM_RC   DS XL1            Return code
COMALEN   DC AL2(*-COMMAREA) Length of COMMAREA

```

The return code has three possible values:

- X'00' Access is allowed to resource
- X'04' The resource or classname is not defined to RACF

- X'08' The user is not authorized to use the resource.

If the commarea is too small, the value '\*' (asterisk) is returned in the first byte of COMM\_RESN. If the commarea has length zero, or is absent, no information is returned.





---

## Chapter 5. The zSecure CICS Toolkit command interface

zSecure CICS Toolkit provides the capability of issuing RACF commands from CICS.

The commands that can be issued are:

**ADDGROUP (ADGRP), ADDUSER (ADUSER), ALTGROUP (ALTGRP), ALTUSER (ALUSER), CONNECT (CONNCT), DELETE GROUP (DELGRP), DELETE USER (DELUSER), LISTDSD (LDSO), LISTGRP (LGRP), LISTUSER (LUSER), PASSWORD, PERMIT, REMOVE, RALTER, RDEFINE, RDELETE, REMOVE, RLIST, RACLINK, USRDATA, and VERIFY.**

For any option used to alter profiles, SMF records are produced, indicating the changes and who changed. These records show up in your normal RACF reports.

The SMF records all show the value TOOLKIT\* in the SMF80UID field. This special value is used to indicate that the record was produced as part of a zSecure CICS Toolkit function.

The **PASSWORD** and **VERIFY** commands are only available through the API. You can use the zSecure CICS Toolkit API to customize the panels for your installation. For more information, see the API documentation.

Before users can use zSecure CICS Toolkit, they must be given access to one or more zSecure CICS Toolkit commands, as described in Chapter 2, "zSecure CICS Toolkit installation," on page 5.

---

### Navigating the Main menu

To view the command interface main menu, you must have access to the transaction to execute it. If you have not completed a signon to CICS with a valid RACF user ID, the transaction automatically terminates with message CQT006.

#### Procedure

1. To view the command interface main menu, enter RTMM at a clear panel.

```

Termid = CP24          IBM Security zSecure CICS Toolkit   Date = 2007/094
Userid = BCSCGB1      MAIN MENU                          Time = 11:09:18
Name   = John Smith

PF01 ADGRP  PF02 ADUSER  PF03 ALTGRP  PF04 ALUSER  PF05 CONNCT  PF06 DELDSD
PF07 DELGRP  PF08 DELUSR  PF09 LDS   PF10 LGRP    PF11 LUSER   PF12 PERMIT
PF13 RALTER  PF14 RACLNK  PF15 RDEFNE  PF16 RDELTE  PF17 REMOVE  PF18 RLIST
PF19 USRDAT

Number ==> __

Licensed Materials - Property of IBM
5655-N18 Copyright IBM Corp. 1982, 2016. All Rights Reserved.

Use PF key or enter NUMBER for desired command. Press CLEAR to exit

```

Figure 6. Main Menu

- To make a selection, press the PF key, or type in the number of the command you want to use and press **Enter**.

When displaying a field within a profile (for example, the groups a user is connected to), you can use PF08 to page down if there is more than one panel of entries.

---

## Adding, altering, or deleting a group (ADDGROUP, ALTGROUP, or DELGROUP command)

Use the **ADDGROUP**, **ALTGROUP**, and **DELGROUP** commands to add a new group to the system, or to alter or delete an existing group.

### About this task

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.ADGR / TOOLKIT.ALGR / TOOLKIT.DELG / TOOLKIT.LGRP, depending on the command that is performed) and the group (ADGR.grpname / ALGR.grpname / DELG.grpname / LGRP.grpname).

### Procedure

- Access the **ADDGROUP**, **ALTGROUP**, and **DELGROUP** commands by pressing the designated **PF** key, as shown on the main menu.

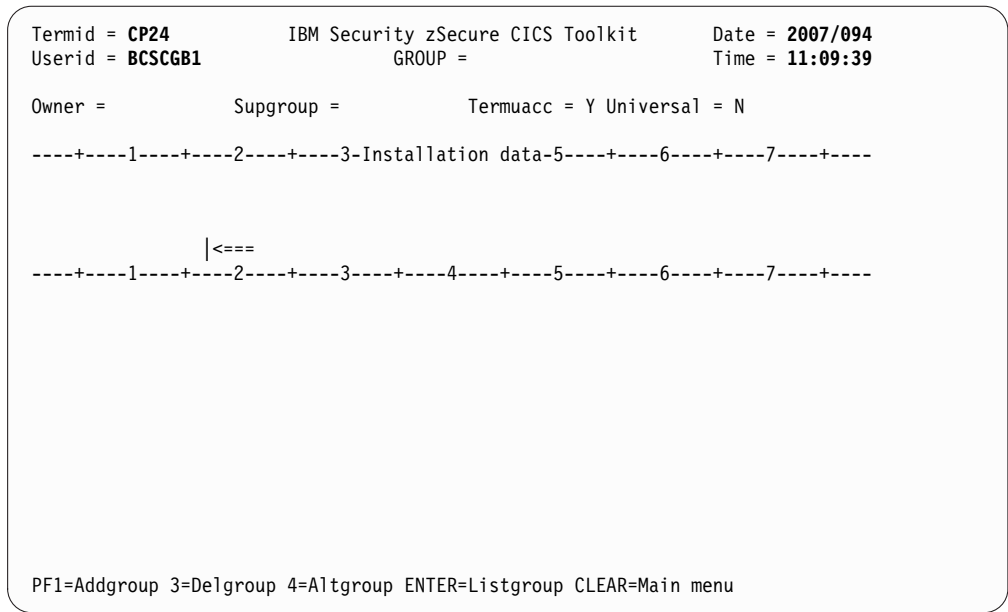


Figure 7. ADDGROUP / ALTGROUP / DELGROUP panel

2. Specify the values that are needed to perform the selected task. The different fields and their meanings are described under the LISTGROUP command. The fields that are required when you perform the different commands are as follows:

**ADDGROUP**

To define a new group, the GROUP name must be entered. This GROUP name must be unique and not currently exist as a group or user name. The OWNER, if not entered, defaults to your user ID. If a group name is entered as the owner, it must be the same name as the superior group (SUPGROUP). If you do not enter a SUPGROUP, it defaults to your current connect group. TERMUACC must be Y or N. UNIVERSAL must also be Y or N. The INSTALLATION DATA field is optional. After you entered the required information press PF01 to add the new group.

**ALTGROUP**

To alter the GROUP profile. You can change any of the fields except the GROUP. After you entered the data that you want to change, press PF04 to update the profile.

**DELGROUP**

To delete the GROUP profile. Enter the GROUP name that you want to delete, and press PF03. The group must not have any subgroups, users that are connected to it, or any group data sets. zSecure CICS Toolkit has no way to find all the users that might be connected to the group. Therefore, the group might not be a Universal group. zSecure CICS Toolkit checks for subgroups and users but not for group data sets.

---

**Adding a user profile (ADDUSER command)**

Use the **ADDUSER** command to add a user profile to the system.

## About this task

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.ADUS) and the default group of the user profile that is added (ADUS.dfltgrp).

## Procedure

1. Access the **ADDUSER** command by pressing the designated **PF** key, as shown on the main menu.

```
Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      Adduser =                          Time = 11:09:56

Name =                Dfltgrp =                Authority = U
Seclevel =            SMTWTFS FROM TILL
                    YYYYYY 0000 0000          Password =                Owner =

Password Phrase =

                    |<===

-----1-----2-----3-Installation data-5-----6-----7-----

                    |<===

-----1-----2-----3-----4-----5-----6-----7-----

CQT020 -Enter details of user to be added
PF5=AddUser ENTER=Redisplay CLEAR=Main menu
```

Figure 8. ADDUSER panel

2. Enter information in these fields to add a user:

### USERID

In the ADDUSER= field. USERID can be 1 - 8 characters.

### NAME

Is normally the users name, 1- 20 characters.

### DFLTGRP

The user default group. It must be a valid group name.

### AUTHORITY

The authority of the user within the group. The default is U (use), but might be changed to C (create).

### SECLEVEL

Specify the SECLEVEL for the user or leave blank. Available SECLEVELs can be displayed with the RLIST command and displaying class SECDATA, resource name SECLEVEL and then selecting the MEMBERS display.

### LOGON DAYS

Enter a Y or N to indicate which days the user can access the system. Specifying N for any particular day prevents the user from accessing the system on that day.

### LOGON TIME

LOGON TIME specifies the time of day the user might log on. Leave

the FROM and TILL fields zero to allow the user to log on at any time. If a time is specified, it must be in the range 0001 through 2359.

**PASSWORD**

The initial password for the user. The initial password is always set as expired.

**OWNER**

The owner of the profile.

**PASSWORD PHRASE**

The initial password phrase for the user. The initial password phrase is always set as expired. Trailing blanks are removed.

**Note:** Password phrases are only supported on z/OS 1.8 and higher. Attempts to set a password phrase on z/OS levels that do not support them result in message CQT184. The user is defined as specified but without a password phrase.

**INSTALLATION DATA**

Information about the user can be entered in this field. If data is entered, the EOF key must be pressed after the last character was entered. This field is optional.

3. Press PF05 to add the user to the system.

The initial PASSWORD for the user, if not specified, can be the same as the name of the DEFAULT GROUP. Users must enter a new password the first time they log on.

---

## Changing a profile (ALTUSER command)

Use the **ALTUSER** command to change the profile for a specific user.

### About this task

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.AUSR) and the users default group (AUSR.*dfltgrp*). SMF records are created any time that a profile is altered by any user.

### Procedure

1. Access the **ALTUSER** command by pressing the designated **PF** key, as shown on the main menu.

```

Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      ALTUSER =                          Time = 11:10:09

Password = ***** Resume user? (Y/N) = N Expire PW? (Y/N) = Y

Name = *****      Revokedt = ***** Resumedt = *****

Password Phrase = *****
***** |<===

----+----1----+----2----+----3-Installation data-5----+----6----+----7----+----
*****
*****
***** |<===
+++++
CLAuth =              NoCLAuth =
Special = * Operations = * Auditor = * Restr = * Grpacc = * Adsp = *
Protected = * Uaudit = * Dfltgrp = ***** Owner = *****

SMTWTF5 From Till
***** *** Mode1 = *****
CQT009 -Enter userid to be updated
PF5=Update 6=CICS 7=TSO 8=OMVS 9=WORK ENTER=Redisplay CLEAR=Main menu

```

Figure 9. ALTUSER panel

2. Specify the user for whom you want to change the profile.  
When you display a user, you are given the option to resume the user or not.
3. To reset a user password to the name of the DEFAULT GROUP, clear the PASSWORD field: place the cursor in the first position of the field, press EOF, then press PF05.
4. To set a specific PASSWORD, clear the field, place the cursor in the first position of the field, enter the new password and press PF05. By default the new password is set to expired. If you have the special attribute, you can indicate that the new password does not need to be changed immediately (set **Expire PW** to N).  
Starting with z/OS 1.8, users might also be issued a password phrase. If you must change a password phrase, specify the new value. Ensure that you remove excess characters at the end. zSecure CICS Toolkit strips trailing blanks, but trailing question marks (?) or asterisks (\*) are included as part of the password phrase. If you want to remove an existing password phrase, blank out or erase the entire password phrase field. If you want to retain the current value, leave all the question marks as shown.
5. You can REVOKE a user one of two ways by specifying REVOKEDT:
  - Set REVOKEDT to the current date, the equivalent of specifying ALTUSER *userid* REVOKE. The revoke flag is set and the REVOKE and RESUME dates are cleared.
  - Specify a REVOKEDT date other than current date. The user is revoked on that date.
6. Use RESUMEDT to resume a user. You can use the special value 00000 (five zeros) to remove any REVOKE or RESUME date. Setting the field to blanks or empty, or leaving the value as shown, results in retaining the current value.  
Specify **Y** for **Resume User** to resume a user immediately.
7. To update one of the supported segments, press the indicated PF key.
  - To modify CICS information for the user, press PF06.
  - To update the TSO segment, press PF07.
  - For the OMVS and WORKATTR segments, use PF08 and PF09.

- Unless you have SPECIAL, or access to TOOLKIT.SPEC, you are not permitted to change any field below the delimiter line of ++++++.

There are exceptions for the DFLTGRP value the RESTRICTED attribute. Regular administrators can change these two fields without the need for SPECIAL. Another exception is the EXPIRED setting for a new password. The user needs system special or access to TOOLKIT.SPEC to set non-expired passwords. If you do have SPECIAL, you can alter and of the fields on the panel and press PF05 to update.

- Specify an entry in the CLAUTH or NOCLAUTH to give or remove class authority in the specified class.  
zSecure CICS Toolkit does not verify that the specified class is valid.

## Altering the CICS segment for a user (ALTUSER CICS SEGMENT)

Use the **ALTUSER** command with the CICS SEGMENT option to alter the CICS segment for a specific user.

### About this task

The user must have the following authorizations:

- The user must have access to the zSecure CICS Toolkit command (TOOLKIT.AUSR) and the user default group (AUSR.dfltgrp).
- To manage the CICS segment, the user must have access to TOOLKIT.ACIC.

### Procedure

- To access the ALTUSER CICS SEGMENT command, press the PF06 key in the main ALTUSER panel.

```
Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      ALTUSER = BCSCGB2                   Time = 11:10:47

OPIdent = 123
OPPrty  = 123
Timeout = 000
XRFSoff = NOFORCE
OPClass =

RSLKey  =
TSLKey  =

CQT074 -Command completed successfully
PF5=Update 6=User 11=Delete  Enter=Redisplay CLEAR=Main menu
```

Figure 10. ALTUSER (CICS SEGMENT) panel

#### OPIDENT

The one-to-three character operator identification to assign to this user.

#### OPPRTY

The operator priority for this user. It can be in the range 000 - 255.

## TIMEOUT

The number of minutes that must elapse since the user last used the terminal before CICS timeouts the terminal.

For releases of RACF before 2.2, it must be a value in the range 000 - 255. For later releases of RACF, the range is 000 - 999. In both cases, a value of zero means the terminal is not timed out.

## XRFSOFF

The CICS extended recovery facility sign-off option. Specify FORCE to sign off the operator in the event of XRF takeover, or NOFORCE to leave the operator signed on.

## OPCLASS

Operator classes are used by CICS when it routes basic mapping support messages. The valid classes are in the range 01 - 24. When you specify the operator classes, they must be separated by a comma (for example: 01,04,05,16,24).

## RSLKEY

The RSL keys are used by CICS on distributed platforms. Each CICS resource has one RSL key that is assigned to it. In order for a user to access a resource, the user must have the same RSL key as the RSL key assigned to the resource. The valid keys are in the range 01 - 24. The values 00 and 99 have special meaning. When you specify RSLKEYs, they must be separated by a comma, for example, 01,04,05,16,24.

**Note:** In the current release, zSecure CICS Toolkit provides space for only 22 RSLKEY values.

## TSLKEY

The TSL keys are used by CICS on distributed platforms. Each CICS transaction has one TSL key that is assigned to it. In order for a user to run a transaction, the user must have the same TSL key as the TSL key assigned to the transaction. The valid keys are in the range 01 - 64. The values 00 and 99 have special meaning. When you specify TSLKEYs, they must be separated by a comma, for example, 01,04,05,16,24.

**Note:** In the current release, zSecure CICS Toolkit provides space for only 22 TSLKEY values.

2. To display the current information in the CICS SEGMENT of a user, enter the userid, and press Enter.
3. To change any or all of the information, enter the new data and press PF5. If there are any errors, an error message displays, indicating the problem.
4. To remove the CICS SEGMENT, press PF11.

**Note:** Removing a CICS SEGMENT does not prevent a user from accessing CICS services. Access to CICS services might be controlled by profiles in the APPL resource class.

Without a CICS segment, CICS users inherit certain values from the CICS SEGMENT of the CICS Default User.

Whenever the OPCLASS, RSLKEY or TSLKEY parameter is updated, it completely replaces the prior values. For example, if a user has OPCLASS 01,02,03 defined, and you update the profile by specifying 02,05,06, the user only has 02, 05, and 06 defined. The prior values of 01,02, and 03 are deleted.



## Altering the TSO segment for a user (ALTUSER TSO SEGMENT)

Use the **ALTUSER** command with the TSO SEGMENT option to alter the TSO segment for a specific user.

### About this task

The user needs the following authorizations:

- The user must have access to the zSecure CICS Toolkit command (TOOLKIT.AUSR) and the user default group (AUSR.dfltgrp).
- To manage the TSO segment, the user must have access to TOOLKIT.ATSO.

### FUNCTION

### AUTHORITY:

### Procedure

1. To access the ALTUSER TSO SEGMENT command, press the PF07 key in the main ALTUSER panel.

```
Termid = CP24          IBM Security zSecure CICS Toolkit   Date = 2007/094
Userid = BCSCGB1      ALTUSER = BCSCGB2                 Time = 11:11:05

Acctnum = *
Destid =
HClass =              JClass =              MsgClass=              SClass =
Size = 00000000      Maxsize = 00000000   Seclabl =
Proc = ISPFPROC      Unit =              Udata = 0000

CQT074 -Command completed successfully
PF5=Update 7=User 11=Delete  Enter=Redisplay CLEAR=Main menu
```

Figure 11. ALTUSER (TSO SEGMENT) panel

#### ACCTNUM

The users default TSO account number.

#### DESTID

The default destination for dynamically allocated SYSOUT data sets.

#### HCLASS

The default hold class of the user.

#### JCLASS

The default job class of the user.

#### MSGCLASS

The default message class of the user.

**SCLASS**

The default sysout class of the user.

**SIZE** The minimum region size if the user does not request one at logon time.

**MAXSIZE**

The maximum region size the user can request at logon time.

**SECLABL**

The users security label.

**PROC** The name of the user default logon procedure.

**UNIT** Default name of a device or group of devices that a procedure uses for allocations.

**UDATA**

Installation data for the user.

2. To display the current information in the TSO SEGMENT of a user, enter the userid, and press Enter.
3. To change any or all of the information, enter the new data and press PF5.  
If there are any errors, an error message displays, indicating the problem. If a field is set to blanks, that parameter is deleted from the user TSO segment.  
When the parameters ACCTNUM, PROC and SECLABL are specified, the user must have access to these definitions in the appropriate RACF resource class.  
Refer to the RACF Command Language Reference for complete information and the other TSO segment fields.
4. To remove the TSO SEGMENT, press PF11.

**Note:** Removing a TSO SEGMENT prevents the user from accessing TSO interactive services.

## Altering the OMVS segment for a user (ALTUSER OMVS SEGMENT)

Use the **ALTUSER** command with the OMVS SEGMENT option to alter the OMVS segment for a specific user.

### About this task

The user needs the following authorizations:

- The user must have access to the zSecure CICS Toolkit command (TOOLKIT.AUSR) and the user default group (AUSR.*dfltgrp*).
- For managing the OMVS segment, the user must have access to TOOLKIT.AOMV.

### Procedure

1. To access the **ALTUSER OMVS SEGMENT** command, press the **PF08** key in the main ALTUSER panel.

```

Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      ALTUSER = BCSCGB2                    Time = 11:11:22

UID      = 0000002009 (# or AUTOUID) Shared = N   MKDIR  = N
Home     =
Program  =

ASSizeMax =
CPUTimeMax =
FileProcMax =
MMapAreaMax =
ProcUserMax =
ThreadsMax =
MemLimit =
SHMemMax =

CQT074 -Command completed successfully
PF5=Update 8=User 11=Delete  Enter=Redisplay CLEAR=Main menu

```

Figure 12. ALTUSER OMVS segment panel

**UID** The OMVS UID of the user. When you change the UID to another value, it is possible to enter AUTOUID in this field. If the required profiles are defined, zSecure CICS Toolkit assigns the next available UID. When you change the UID into a value that is already assigned to another user, the command is rejected. For authorized users, it can be overridden by usage of the SHARED parameter.

**Shared** When you assign a UID to a user, the UID must be unique. When the terminal user has the System-SPECIAL attribute or has access to SHARED.IDS in the UNIXPRIV class, the user might request that the UID value can be shared between multiple users. In other words, the UID value then does not need to be unique. If you want to assign a shared UID, enter Y in the SHARED field.

**MKDIR** When you assign a HOME directory to a user, the directory must exist in the file system. By selecting option Y, the zSecure CICS Toolkit task attempts to create the directory, and set the owner to the userid and dfltgrp. The authority to run the necessary USS commands is based on the CICS region user, and not on the authority of the CICS terminal user. If this function is not enabled in your installation, leave the value for this field as N.

**Home** The home directory of the user. When you change the OMVS segment of the user, ensure that the case of this field is correct. Either do not update the OMVS segment at all, or ensure that your terminal uses mixed case. Or, verify that the actual home directory of the user is defined in ALL UPPERCASE. If the HOME directory (case sensitive) cannot be located, use of UNIX System Services might fail.

**Program** The initial program (shell program) for the user. When you change the OMVS segment of the user, ensure that the case of this field is correct. Either do not update the OMVS segment at all, or ensure that your

terminal uses mixed case. Or, verify that the program actually exists in ALL UPPERCASE. If you leave the field blank, USS normally uses the value */bin/sh* as default value.

**ASSizeMax**

The address-space-size that you define is a numeric value 10 485 760 - 2 147 483 647. The value that is specified overrides any value that is provided by the MAXASSIZE parameter of BPXPRMxx. If the system value is adequate, you must leave this field blank.

**CPUTimeMax**

The processor time that you define is a numeric value 7 - 2 147 483 647. The value that is specified overrides any value that is provided by the MAXCPU TIME parameter of BPXPRMxx. If the system value is adequate, you must leave this field blank.

**FileProcMax**

The files-per-process that you define is a numeric value 3 - 524287. Regular users can use the value 256. The value that is specified overrides any value that is provided by the MAXFILEPROC parameter of BPXPRMxx. If the system value is adequate, you must leave this field blank.

**MMapAreaMax**

The memory-map-size that you define is a numeric value 1 - 16 777 216. The value that is specified overrides any value that is provided by the MAXMMAPAREA parameter of BPXPRMxx. If the system value is adequate, you must leave this field blank.

**ProcUserMax**

The processes-per-UID that you define is a numeric value 3 - 32 767. The value that is specified overrides any value that is provided by the MAXPROCUSER parameter of BPXPRMxx. If the system value is adequate, you must leave this field blank.

**ThreadsMax**

The threads-per-process that you define is a numeric value 0 - 100 000. Specifying a value of 0 prevents applications that are run by this user from using the pthread\_create service. The value that is specified overrides any value that is provided by the MAXTHREADS parameter of BPXPRMxx. If the system value is adequate, you must leave this field blank.

**MemLimit**

The nonshared-memory-size that you define to RACF is a numeric value 0 - 16777215, followed by the letter M, G, T, or P.

**SHMemMax**

The shared-memory-size that you define to RACF is a numeric value 1 - 16777215, followed by the letter M, G, T, or P. The value that is specified for SHMEMMAX overrides any value that is provided by the IPCSHMNSEGS parameter of BPXPRMxx. If the system value is adequate, you must leave this field blank.

2. To display the current information in the OMVS SEGMENT of a user, enter the userid, and press Enter.
3. To change any or all of the information, enter the new data and press PF5. If there are any errors, an error message displays, indicating the problem.

You can delete a field from the user's OMVS segment by setting the value to all blanks. For setting the UID to the value zero (0), the terminal user must have the RACF System-SPECIAL attribute. Access to TOOLKIT.SPEC is not applicable to this particular function.

Use the value AUTOUID whenever possible. The AUTOUID function is only available in z/OS 1.4 and higher. It requires definition of the BPX.NEXT.USER profile in the facility class. For more information, see Chapter 2, "zSecure CICS Toolkit installation," on page 5.

4. To remove the OMVS SEGMENT, press PF11.

**Note:** Removing an OMVS SEGMENT prevents the user from accessing any UNIX System Services. Access to USS might also be provided by the default UID designated by BPX.DEFAULT.USER in the FACILITY class.

## Altering the WORKATTR segment for a user (ALTUSER WORKATTR SEGMENT)

Use the **ALTUSER** command with the WORKATTR SEGMENT option to alter the WORKATTR segment for a specific user.

### About this task

The user needs the following authorizations:

- The user must have access to the zSecure CICS Toolkit command (TOOLKIT.AUSR) and the user default group (AUSR.dfltgrp).
- For managing the WORKATTR segment, the user must have access to TOOLKIT.AWRK.

### Procedure

1. To access the ALTUSER WORKATTR SEGMENT command, press the PF09 key in the main ALTUSER panel.

```
Termid = CP24          IBM Security zSecure CICS Toolkit   Date = 2007/094
Userid = BCSCGB1      ALTUSER = BCSCGB2                 Time = 11:11:32

Name   = John Smith
Account =
Bldg   =
Dept   = CICS Toolkit Development
Room   = Annex-1
Addr1  = 't Zandt Labs
Addr2  = The Netherlands
Addr3  =
Addr4  =

CQT074 -Command completed successfully
PF5=Update 9=User 11=Delete  Enter=Redisplay CLEAR=Main menu
```

Figure 13. ALTUSER (WORKATTR SEGMENT) panel

**Name** Specifies the name of the user SYSOUT information is to be delivered to.

**Account**

Specifies an account number for APPC/MVS processing. Although RACF accepts any string of up to 255 characters, the zSecure CICS Toolkit interface allows up to 60 characters.

**Bldg** Specifies the building that SYSOUT information is to be delivered to.

**Dept** Specifies the department that SYSOUT information is to be delivered to.

**Room** Specifies the room SYSOUT information is to be delivered to.

**Addr1** Address-line-1 specifies other address line for SYSOUT delivery.

**Addr2** Address-line-2 specifies other address line for SYSOUT delivery.

**Addr3** Address-line-3 specifies other address line for SYSOUT delivery.

**Addr4** Address-line-4 specifies other address line for SYSOUT delivery.

2. To display the information in WORKATTR SEGMENT of a user, enter the userid, and press Enter.
3. To change any or all of the information, enter the new data and press PF5. If there are any errors, an error message displays, indicating the problem. If a field is set to blanks, that parameter is deleted from the users WORKATTR segment.

When you change the WORKATTR segment of the user, ensure that the case of these fields is correct. If your installation requires mixed case values in these fields, you might refrain from updating the WORKATTR segment at all, or ensure that your terminal uses mixed case.

4. To remove the WORKATTR SEGMENT, press PF11.

**Note:** Removing a WORKATTR SEGMENT normally does not affect the users of their authorization to use any system services.

---

## Connecting a user or group to a group (CONNECT command)

Use the **CONNECT** command to connect a user or group to a group.

### About this task

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.CONN) and to the target group (CONN.*grpname*).

### Procedure

1. To access the CONNECT command, press the designated PF key on the main menu.
2. To connect a user to a group, enter the user and group name as indicated and press PF05. If there are any errors, such as an invalid user or group name, an error message indicates the problem.

```

Termid = CP24          IBM Security zSecure CICS Toolkit   Date = 2007/094
Userid = BCSCGB1      CONNECT                          Time = 11:14:03

Connect =             Userid          Group          Authority = U   Owner = BCSCGB1
Special = N   Operations = N   Revokedt =      Resumedt =

CQT016 -Enter userid and group name
PF5=Update ENTER=Redisplay CLEAR=Main menu

```

Figure 14. CONNECT panel

**AUTHORITY**

Defaults to U (use) but might be set to C (create), N (connect), or J (join).

**SPECIAL**

Specify Y if the user must have the group-special attribute.

**OPERATIONS**

Specify Y if the user must have the group-operations attribute.

**OWNER**

Defaults to the ID of the person who is issuing the command, but any valid user ID or group can be entered.

**REVOKEDT**

The date (YYDDD) the user is prevented from connecting to the group. If today's date is specified, the connection is revoked immediately. In this case, the value of RESUMEDT is ignored, and both the RESUMEDT and the REVOKEDT are reset. The special value 00000 (five zeros) can be used to remove an existing REVOKEDT. Setting the field to blanks or empty, results in leaving the current value unchanged.

**RESUMEDT**

The date (YYDDD) the user can connect to the group. If today's date is specified, the connection is resumed immediately. In this case, the value of REVOKEDT is ignored, and both the RESUMEDT and the REVOKEDT are reset. The special value 00000 (five zeros) can be used to remove an existing RESUMEDT. Setting the field to blanks or empty, results in leaving the current value unchanged.

---

**Deleting a data set (DELETE DATASET command)**

Use the **DELETE DATASET** command to delete a data set profile from the system.

## About this task

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.DELD) and the high-level-qualifier of the data set profile name (DELD.hlq). If the user does not have access to the DELD.hlq, standard RACF authority checking is used. Refer to the RACF Command Language Reference manual for information about which data set profiles a user is authorized to delete.

## Procedure

To access the **DELETE DATASET** command, press the designated PF key on the main menu.

```
Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1                                           Time = 11:14:18

          Dsname
Delete   =                                               Generic = Y

CQT144 -Enter dataset profile to be deleted. Specify Y if Generic, N if not
PF5=Update ENTER=Redisplay CLEAR=Main menu
```

Figure 15. DELETE DATASET panel

---

## Deleting a user profile

Use the **DELETE** command to delete a user profile from the system.

### About this task

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.DELU) and the default group of the user (DELU.dfltgrp).

Before the user profile is deleted, it must be **REMOVED** from any groups it is connected to except its default group. No data set profiles that have this userid as a high-level qualifier can exist.

zSecure CICS Toolkit checks for group connections but not for data set profiles.

### Procedure

To access the **DELETE** command, press the designated PF key on the main menu.



```
Termid = CP24          IBM Security zSecure CICS Toolkit    Date = 2007/094
Userid = BCSCGB1      Time = 11:14:35

Delete      =      Userid

PF5=Update ENTER=Redisplay CLEAR=Main menu
```

Figure 16. DELETE USER panel

---

## Listing the profile for one or more data sets (LISTDSET command)

Use the **LISTDSET** command to list the profile for a specific data set or multiple data sets.

### About this task

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.LDSD).

### Procedure

1. To access the **LISTDSET** command, press the designated PF key on the main menu.

```

Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      Listdsset =                          Time = 11:14:47
G , D OR * *
-----
Owner = ***** Cre = ***** Last ref = ***** Last chg = ***** Uacc = *****
Alter acc = ***** Cntrl acc = ***** Updte acc = ***** Read acc = *****
Group ds = * WARN = * Cre grp = ***** Dataset type = **** Level = ***
Audit = * Aud Succ = * Aud Fail = * Glbl audit = * Gaud Succ = * Gaud Fail = *
Sec1 = *** Numctgy = **** NumPgms = **** NumUsrs = ****

-----1-----2-----3-Installation data-5-----6-----7-----
*****
*****
*****
***** |<===
-----1-----2-----3-----4-----5-----6-----7-----
PF1=Toggle 3=Chgopts 5=Userids 7=Programs 11=Search CLEAR=Main menu

```

Figure 17. LISTDSET panel

**LISTDSET**

The ID of the data set to be displayed (if you perform a **listuser**). If a search is being done, this field can be bypassed, or any characters can be entered in any position as part of the search criteria.

**Note:** If you are doing a normal **listdsset**, the following fields are not used for entry. These other fields are only used as entries if you are doing a search (PF11).

G,D, or \* Allows you to search for generic (G), discrete (D), or both (\*) types of profiles.

**OWNER**

A user or group that was defined as the owner of the data set.

**CRE**

The date the data set was created. The format is YYDDD.

**LAST REF**

The date the data set was last referenced. The format is YYDDD.

**LAST CHG**

The date the data set was last updated. The format is YYDDD.

**UACC**

The universal access for the data set. This field can be *ALTER*, *CONTROL*, *UPDATE*, *READ*, or *NONE*.

**ALTER ACC**

The number of times the data set was accessed with *ALTER*.

**CNTRL ACC**

The number of times the data set was accessed with *CONTROL*.

**UPDTE ACC**

The number of times the data set was accessed with *UPDATE*.

**READ ACC**

The number of times the data set was accessed with *READ*.

**GROUP DS**

This field can be Y or N.

**WARN**

Indicates whether the data set is in warning mode. This field can be *Y* or *N*.

**CRE GROUP**

The current connect group of the user that created this data set.

**DATASET TYPE**

This field identifies the data set type. These first two characters in this field indicate a VSAM (*VS*), or Non-VSAM (*NV*) data set profile. The third character indicates whether the profile is a model profile (*M*) or not (*N*). Finally, the fourth character indicates whether the profile is for a tape data set (*T*) or not (*N*).

**LEVEL**

The level indicator for the data set. This field is a numeric field.

**AUDIT**

Indicates the audit flag for the data set. The settings can be: *A* to audit all accesses, *S* to audit successful accesses, *F* to audit failures, or *N* for no auditing.

**AUD SUCC**

The audit *SUCCESS* flag. The settings can be: *R* to audit successful reads, *U* to audit successful updates, *C* to audit successful control accesses, or *A* to audit successful alter accesses.

**AUD FAIL**

The audit *FAILURE* flag. The settings can be: *R* to audit unsuccessful reads, *U* to audit unsuccessful updates, *C* to audit unsuccessful control accesses, or *A* to audit unsuccessful alter accesses.

**GLBL AUDIT**

The Global audit options as specified by a user with the *AUDITOR* attribute. The settings can be: *A* to audit all accesses, *S* to audit successful accesses, *F* to audit failures, or *N* for no auditing.

**GAUD SUCC**

The GLOBAL audit *SUCCESS* flag. The setting can be: *R* to audit successful reads, *U* to audit successful updates, *C* to audit successful control accesses, or *A* to audit successful alter accesses.

**GAUD FAIL**

The GLOBAL audit *FAILURE* flag. The settings can be: *R* to audit unsuccessful reads, *U* to audit unsuccessful updates, *C* to audit unsuccessful control accesses, or *A* to audit unsuccessful alter accesses

**SECL** The security level of the data set. This field is a numeric field.

**NUMCTGY**

The number of security categories to which the data set belongs.

**NUMPGMS**

The number of programs that are authorized to access the data set.

**NUMUSRS**

The number of users and groups authorized to access the data set.

**INSTALLATION DATA**

The information that is contained in the data sets *DATA* field. This installation data can be up to 255 characters.

2. Press PF05 to display the access list of this data set profile (the users and groups).

3. Press PF07 to display the programs in the conditional access list.
4. Press PF01 to toggle the display (if you are doing a search) and display all the data sets that match the criteria.
5. Press PF03 to clear the fields and enter new criteria for a search or LISTDSET.

## LISTDSET Display Example

You can view the programs in the conditional access list.

```

Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      Listdset =                          Time = 11:15:07
G , D OR * G      SYS1.**

Owner = SYS1      Cre = 05033 Last ref = 05033 Last chg = 05033 Uacc = READ
Alter acc = 000000 Cntrl acc = 000000 Updte acc = 000000 Read acc = 000000
Group ds = Y WARN = N Cre grp = SYS1      Dataset type = NVNN Level = 000
Audit = F Aud Succ = R Aud Fail = R G1b1 audit = N Gaud Succ = R Gaud Fail = R
Sec1 = *** Numctgy = 0000 NumPgms = 0000 NumUsrs = 0003

-----1-----2-----3-Installation data-5-----6-----7-----
                                     |<===
-----1-----2-----3-----4-----5-----6-----7-----
PF1=Toggle 3=Chgopts 5=Userids 7=Programs 11=Search CLEAR=Main menu

```

Figure 18. LISTDSET Display panel

Now you can choose to do one of the following:

- Display access list entries (PF05, "userids").
- Display conditional access list entries (PF07, "programs").
- Change the search or list options (PF03).
- Return to the main menu (CLEAR).
- If you are doing a search, display all data sets that meet the criteria (PF01).

## Toggling the LISTDSET panel

If you are doing a search, you can toggle the panel and display all the data sets that match the criteria.

### Procedure

- To do a search, press PF01. All data sets that match the criteria are displayed.

```

Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      Listdset =                          Time = 11:21:14
                               SYS1.ZTKTEST
G SYSAPPL.**
G SYS1.BROADCAST
G SYS1.MAN**
G SYS1.RACF**
G SYS1.ZTKTEST
G SYS1.**
D SYS1.ZTKTEST

```

```

CQT064 -End of entries matching this criteria
CQT015 -PF1=Toggle 3=Chgopts ENTER=Next CLEAR=Main Menu

```

Figure 19. LISTDSET Toggle panel

- Choose a PF key for the task you want to to:
  - Change the search or list options (PF03).
  - Return to the main menu (CLEAR).
  - If the panel is full with data sets, display the next panel (ENTER).
  - To perform a LISTDSET on any ID, enter the ID in the LISTDSET field and press PF01.

## Viewing the users authorized, their access authority and access count (LISTDSET USERIDS)

Use the LISTDSET user IDs option to display the users authorized to access the data set, the access authority of the user, and their access count.

### Procedure

- Press PF05 from a LISTDSET panel.

```
Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      Listdset = (USERIDS)                 Time = 11:22:41
                               SYSAPPL.**
STCUSER /A/00000      C2POLICE/U/00000      BCSCGB1 /A/00000

PF3=Chgopts 5=Userids 7=Programs 8=Down 9=Datasets ENTER=Next CLR=Main menu
```

Figure 20. LISTDSET USERIDS panel

- Choose a PF key for the task you want to do:
  - Change the search or list options (PF03).
  - Display the programs that can access this data set (PF07).
  - Return to the LISTDSET panel (PF09).
  - Display the next data set if you are doing a search (ENTER).
  - Return to the main menu (CLEAR).

## Viewing the program/userid combination (LISTDSET Programs)

Use the LISTDSET panel to display the program/userid combination that is authorized to access the data set and to display the access authority.

### Procedure

- Press PF07 from a LISTDSET panel.

```
Termid = CP24          IBM Security zSecure CICS Toolkit   Date = 2007/094
Userid = BCSCGB1      Listset = (PROGRAMS)              Time = 11:23:23
                      SYS1.RACF*.**
C2RCARLA/*           /R
```

PF3=Chgopts 5=Userids 7=Programs 8=Down 9=Datasets ENTER=Next CLR=Main menu

Figure 21. LISTDSET Programs

- Choose a PF key for the task you want to do:
  - Change the search or list options (PF03).
  - Display the access list entries (PF05).
  - Return to the LISTDSET panel (PF09).
  - Display the next data set if you are doing a search (Enter).
  - Return to the main menu (CLEAR).

---

## Listing the profile for one or more groups (LISTGROUP command)

Use the **LISTGROUP** command to list the profile for a specific group or multiple groups.

### About this task

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.LGRP) and the target group (LGRP.grpname).

### Procedure

1. To access the **LISTGROUP** command, press the designated PF key on the main menu.

```

Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1     LISTGROUP = *****              Time = 11:23:38

Supgroup = ***** Owner = ***** Univ = * Cre = ***** Uacc = *****

Termuacc = * Number of subgroups = ***** Number of users = *****

Model = *****

-----1-----2-----3-Installation data-5-----6-----7-----+-----
*****
*****
***** |<===
-----1-----2-----3-----4-----5-----6-----7-----+-----

PF1=Toggle 3=Chgopts 4=UserD 5=Users 6=Dfltu 7=Subgrps 11=Search CLR=Main menu

```

Figure 22. LISTGROUP panel

**LISTGROUP**

The ID of the group to be displayed (if performing a listgroup). If a search is being performed, this field can be bypassed, or any characters can be entered in any position as part of the search criteria.

**Note:** If you are performing a normal listgroup, the fields are not used for entries. These other fields are only used as entries if you are doing a search (PF11).

**SUPGROUP**

The superior group to this group.

**OWNER**

A user or group that has been defined as the owner of this group.

**UNIV** An indicator if it is a Universal group. The list of users connected to a Universal group only shows those users that have non-standard authorizations within the group.

**CRE** The date this profile was created. The format is YYDDD.

**UACC** The authority of a user to the group if the user is not connected to the group. This field can be JOIN, CONNECT, CREATE, USE or NONE. This field cannot be set using any RACF command or zSecure CICS Toolkit. It must have a value NONE for all groups, except for the fixed group VSAMDSET.

**TERMUACC**

Indicates if a group or user must be explicitly authorized to access a terminal. This field can be Y or N.

**NUMBER OF SUBGROUPS**

The number of subgroups to this group. This field is a numeric field.

**NUMBER OF USERS**

The number of users connected to this group. This field is a numeric field. For a Universal group, it only reflects the number of users that have non-standard authorizations within the group.



## MODEL

The name of a discrete data set profile to be used as a model for a new *groupname* data sets. This field is an alphanumeric field.

## INSTALLATION DATA

The information contained in the data sets DATA field. This information can be up to 255 characters.

2. You can either enter a specific group to perform a listgroup on that ID, or enter any character in the field as the search criteria.
  - After entering the search criteria, press **PF11** to start the search.
  - A normal listgroup is performed by entering the group name and pressing **Enter** to open the LISTGROUP panel.
3. Press **PF04** to display the users connected to this group with a DELETEUSER option.
4. Press **PF05** to display the users connected to this group.
5. Press **PF07** to display the subgroups.
6. Use **PF01** to toggle the display (if you are performing a search) and display all the groups that match the criteria.
7. Press **PF03** to clear the fields and enter new criteria for a search or LISTGROUP.

## LISTGROUP Display Example

Enter a group to be listed or initiate a search to display the panel.

```
Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      LISTGROUP = SYSPROG              Time = 11:24:02

Supgroup = SYS1      Owner = SYS1      Univ = N Cre = 05033 Uacc = NONE
Termuacc = Y Number of subgroups = 00000 Number of users = 00003

Model =

-----1-----2-----3-Installation data-5-----6-----7-----
SYSTEM PROGRAMMERS

          |<===
-----1-----2-----3-----4-----5-----6-----7-----

PF1=Toggle 3=Chgopts 4=UserD 5=Users 6=Dfltu 7=Subgrps 11=Search CLR=Main menu
```

Figure 23. LISTGROUP Display panel

Now you can choose to:

- Display users (PF05).
- Display only users that are also connected to your default group.
- Display subgroups (PF07).
- Change the search or list options (PF03).
- Return to the main menu (CLEAR).
- If you are performing a search, display all groups that meet the criteria (PF01).

## Toggling the LISTGROUP panel

If you are doing a search, you can toggle the LISTGROUP panel and displays all groups that match the criteria.

### Procedure

- To perform a search, press **PF01**. All groups that match the criteria are displayed.

```
Termid = CP24          IBM Security zSecure CICS Toolkit   Date = 2007/094
Userid = BCSCGB1      LISTGROUP = SYSADMA              Time = 11:24:21

SYSADMA  SYSAPPL  SYSAUDIT  SYSCTLG  SYSOPRA  SYSPROG  SYS1

CQT064 -End of entries matching this criteria
CQT015 -PF1=Toggle 3=Chgopts ENTER=Next CLEAR=Main Menu
```

Figure 24. LISTGROUP Toggle panel

- Choose a PF key for the task you want to perform:
  - Change the search or list options (PF03).
  - Return to the main menu (CLEAR).
  - If the panel is full with group names, display the next panel (press **Enter**).To perform a listgroup on any name, enter the name in the LISTGROUP field and press **PF01**.

## Listing users for a group (LISTGROUP command, USERIDS option)

You can use the **LISTGROUP** command with the **USERIDS** option to list all the users connected to a group.

### Procedure

- Press **PF05** from a LISTGROUP panel to display all the users connected to the group.

```
Termid = CP24          IBM Security zSecure CICS Toolkit   Date = 2007/094
Userid = BCSCGB1      LISTGROUP = SYSPROG (USERIDS)         Time = 11:24:47
BCSCGB1 BCSCWN1 BCSCWN2
```

```
PF3=Chgopts 4=UserD 5=Users 6=Dfltu 8=Down 9=Grps ENTER=Next CLEAR=Main menu
```

Figure 25. LISTGROUP USERIDS panel

- Choose a PF key for the task you want to perform:
  - Change the search or list options (PF03).
  - Display the alternate users display (PF04).
  - Display the subgroups (PF07).
  - Return to the LISTGROUP panel (PF09).
  - Display the next group if performing a search (ENTER).
  - Return to the main menu (CLEAR).

## Deleting user IDs from a LISTGROUP

You can use delete one or more user IDs from the list of user IDs connected to a group.

### Procedure

- Press **PF04** from a LISTGROUP panel to display all the users connected to the group and the option to delete one or more of them.

```

Termid = CP24          IBM Security zSecure CICS Toolkit    Date = 2007/094
Userid = BCSCGB1      LISTGROUP = SYSPROG (USERS LIST)    Time = 11:25:05

S  Userid  Name                Created  Last signon
-  BCSCGB1  JOHN SMITH           05033   07094
-  BCSCWN1  #####              05033   03126
-  BCSCWN2  #####              05033   02338
-
-
-
-
-
-
-
-
-
-
-
-

PF3=Chgopts 5=Users 8=Down 9=Grps 11=Deluser ENTER=Next CLEAR=Main menu

```

Figure 26. LISTGROUP Userids Delete panel

- Choose a PF key for the task you want to perform:
  - Change the search or list options (PF03).
  - Display the users display (PF05).
  - Display the subgroups (PF07).
  - Return to the LISTGROUP panel (PF09).
  - Display the next group if performing a search (ENTER).
  - Return to the main menu ( CLEAR).
- To delete selected users, enter a D next to the user profile you want to delete and then press **PF11**. Authority to delete the user profile is controlled by the standard zSecure CICS Toolkit DELUSER authorization.

## Listing the subgroups of a group

You can list all the subgroups of a group.

### Procedure

- Press **PF07** from a LISTGROUP panel to display all the subgroups of the group.

Termid = **CP24** IBM Security zSecure CICS Toolkit Date = **2007/094**  
 Userid = **BCSCGB1** LISTGROUP = SYS1 (GROUPS) Time = **11:25:21**

SYSCTLG	VSAMDSET	TEST	OMVSRP	IMWEB	EXTERNAL	EMPLOYEE	SPECIAL
DB2	DSN710	UUCPG	TTY	ADB210	ADCD	APS330	ASU
BPO110	CATALOG	CBC	CEE	CICSTS22	CICSTS23	CRMB	CSQ520
CSQ530	CSQ531	C2RSRVG	DCF140	DIT130	DSNA	EOY	EUV
FAN130	FAN140	FMN410	FON210	GDDM	GIM	GLD	GLDGRP
HFS	HLA	ICQ	IGY310	IOE	ISF	ISP	P390
QMFA	QMF710	REVOKE	SCPTST	SMTP	STCGRP	SYSADMA	SYSAPPL
SYSAUDIT	SYSOPRA	SYSPROG	USER	USERCAT	#EMPLOY	#READ	AUT220
NETV	CDS	CIM	CMX	CSF	ECN	EPH	EUVF
GSK	ICA	IMO	IMW	ING	NFS	BIP210	BIP501
HPJ200	IEL330	IGY330	IXM140	JVA130	JVA140	AUT230	IXM160
NETV510	IXGLOGR	FMN510	IOA	EQA510	IPT110	FFST	AOP
IDI510	ITP110	OMVS	BCSC	ZTKQA	SLDMVSS	CRMA	

PF3=Chgopts 4=UserD 5=Users 6=Dflt 8=Down 9=Grps ENTER=Next CLEAR=Main menu

Figure 27. LISTGROUP (Subgroups) panel

- Choose a PF key for the task you want to perform:
  - Change the search or list options (PF03).
  - Display the alternate users display (PF04).
  - Display the users (PF05).
  - Display the users also connected to your default group (PF06).
  - Return to the LISTGROUP panel (PF09).
  - Display the next group if performing a search (ENTER).
  - Return to the main menu (CLEAR).

## Listing the profiles for a user ID(LISTUSER command)

You can use the **LISTUSER** command to list the profile for a specific user ID.

### About this task

#### FUNCTION

#### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.LUSR) and the users default group (LUSR.dfltgrp).

### Procedure

1. To access the LISTUSER command, press the designated PF key on the main menu.

```

Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      LISTUSER = *****                Time = 11:25:41

Name = ***** Owner = ***** Password = ***** Cre = *****
Dfltgrp = ***** Authority = ***** Uacc = ***** Classcnt = ****
Special = * Operations = * Auditor = * Restr = * Grpacc = * Adsp = *
Protected = * Uaudit = * Revoke = * Revokedt = ***** Resumedt = *****
Lastacc = ***** Passdate = ***** Passint = *** PwTry = ** Sec1 = ***
SMTWTFS From Till Pwdgen = *** Pwdcnt = *** NumCtgy = **** NumGrp = ****
***** **** **** Model = *****

-----1-----2-----3-Installation data-5-----6-----7-----
*****
*****
***** |<===
-----1-----2-----3-----4-----5-----6-----7-----

PF1=Toggle 3=Chgopts 5=Ctgy 6=Segments 7=Groups 11=Search CLEAR=Main menu

```

Figure 28. LISTUSER panel

**LISTUSER**

The ID of the user to be displayed (if performing a listuser). If a search is being performed, this field can be bypassed, or any characters can be entered in any position as part of the search criteria.

**Note:** If you are performing a normal listuser, the fields are not used for entries. These other fields are only used as entries if you are doing a search (PF11).

**NAME**

The user name. A maximum of 20 alphanumeric characters.  
 If you want to search for a name, or part of a name, anywhere within the name field, use the following format:

To search for *SMITH* anywhere in the name field enter <>*SMITH*. It returns all profiles that have the characters *SMITH* anywhere in the name field. The <> indicates to zSecure CICS Toolkit that the criterion is a different search criterion for this field than if using the wild-card characters of "\*".

**OWNER**

A user or group that has been defined as the owner of the user.

**PASSWORD**

Not used.

**CRE**

The date this profile was created. The format is YYDDD.

**DFLTGRP**

The name of the default group for the user.

**AUTHORITY**

This is the users authority within the default group. The possible entries for this field are *ASORGGAT*. The meanings of the subfields are: *A* indicates ADSP, *GA* indicates GROUP AUDITOR, and *T* indicates that terminal access is required.

**UACC** The universal access of the user for the default group. This field can be *ALTER, CONTROL, UPDATE, READ* or *NONE*.

**CLASSCNT**

The number of classes in which the user is allowed to define profiles.

**SPECIAL**

Indicates if the user has the SPECIAL attribute. This field can be Y or N.

**OPERATIONS**

Indicates if the user has the OPERATIONS attribute. This field can be Y or N.

**AUDITOR**

Indicates if the user has the AUDITOR attribute. This field can be Y or N.

**RESTR**

This field indicates if the UACC, GAC and ID(\*) do apply for this user. This field can be Y or N.

**GRPACC**

Specifies that group data sets created by this user are accessible to other users in the group. This field can be Y or N.

**ADSP** Indicates that new data sets created by this user are automatically protected by discrete profiles. This field can be c.

**PROTECTED**

This field indicates if the userid can be used by specification of the password. PROTECTED userids can only be propagated, started, or used through surrogate. This field can be Y or N.

**UAUDIT**

Indicates if all RACHECKs and RACDEFs issued for the user can be logged. This field can be Y or N.

**REVOKE**

Indicates if the REVOKE attribute is set for the user. This field can be Y or N.

**REVOKEDT**

The date that the user is revoked. The format is *YYDDD*.

**RESUMEDT**

The date the user is resumed. The format is *YYDDD*.

**LASTACC**

The date and time the user last accessed the system by using RACINIT. The format is *YYDDD/HH:MM:SS*. If the user has never logged on, this field contains ? in the first position.

**PASSDATE**

The date the users password was last changed. The format is *YYDDD* or the field is zero if it has been reset.

**PASSINT**

The interval that the users password is in effect. This field is a numeric field.

**PWTRY**

The number of unsuccessful password attempts by this user. This field is a numeric field.

**SECL** The security level of the user. This field is a numeric field.

**SMTWTFS**

The days of the week that the user can logon. A 'Y' indicates the user can logon for that day, N indicates the user is restricted for that day.

**FROM**

If the user is restricted by time, FROM is the starting time that the user might log on at. The format is *HHMM*. If there are no time restrictions, both FROM and TILL are *0000*.

**TILL** The latest time the user can logon to the system. The format is *HHMM*.

**PWDGEN**

The current password generation number for the user. This field is a numeric field.

**PWDCNT**

The number of old passwords present for this user. This field is a numeric field.

**NUMCTGY**

The number of security categories the user has access to. This field is a numeric field.

**MODEL**

The data set profile model for this user. This field is an alphanumeric field.

**INSTALLATION DATA**

The information contained in the users DATA field. This information can be up to 255 characters.

2. You can either enter a specific userid to perform a LISTUSER on that user, or enter any character in the field as the search criteria.
  - After entering the search criteria, press PF11 to start the search.
  - A normal LISTUSER is performed by entering the userid and pressing **Enter**.
3. Press PF05 to display the categories for this user.
4. Press PF07 to display the groups.
5. Use PF01 to toggle the display (if you are performing a search) and display all the users that match the criteria.
6. Press PF03 to clear the fields and enter new criteria for a search or LISTUSER.

## LISTUSER Display Example

This example shows what might be displayed if you specify a user ID and press **Enter**, or enter different criteria and press **PF11** to initiate a search.

**Note:** If you enter a full user ID and click **PF11**, you obtain the same results as if you clicked **Enter** because there is only one profile with the specified user ID.



```

Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      LISTUSER = BCSCGB2                  Time = 11:25:55

Name = GUUS 2ND       Owner = BCSC      Password = ???????? Cre = 05033

Dfltgrp = BCSC      Authority =          Uacc = NONE      Classcnt = 0001

Special = N Operations = Y Auditor = N Restr = N Grpacc = N Adsp = N

Protected = N Uaudit = N Revoke = N Revokedt = ***** Resumedt = *****

Lastacc = 07089/09:17:57 Passdate = 07068 Passint = 180 PwTry = 00 Sec1 = ***

SMTWTFS From Till Pwdgen = 006 Pwdcnt = 006 NumCtgy = 0000 NumGrp = 0005
YYYYYYY 0000 0000 Model =

-----1-----2-----3-Installation data-5-----6-----7-----

                                |<===
-----1-----2-----3-----4-----5-----6-----7-----

PF1=Toggle 3=Chgopts 5=Ctgy 6=Segments 7=Groups 11=Search CLEAR=Main menu

```

Figure 29. LISTUSER Display panel

From this point, you can choose to:

- Display the categories (PF05).
- Display the groups (PF11).
- Change the search or list option (PF03).
- Return to the main menu (CLEAR).
- If you are performing a search, display all users that meet the criteria (PF01).

### Toggling the LISTUSER panel

You can use the LISTUSER panel to list all the users that match the criteria you enter.

#### Procedure

- To perform a search, press PF01. All users that match the criteria are displayed.

```

Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      LISTUSER = B8FU0142                  Time = 11:32:50

B8FTEST B8FU0000 B8FU0001 B8FU0002 B8FU0003 B8FU0004 B8FU0005 B8FU0006
B8FU0007 B8FU0008 B8FU0009 B8FU0010 B8FU0011 B8FU0012 B8FU0013 B8FU0014
B8FU0015 B8FU0016 B8FU0017 B8FU0018 B8FU0019 B8FU0020 B8FU0021 B8FU0022
B8FU0023 B8FU0024 B8FU0025 B8FU0026 B8FU0027 B8FU0028 B8FU0029 B8FU0030
B8FU0031 B8FU0032 B8FU0033 B8FU0034 B8FU0035 B8FU0036 B8FU0037 B8FU0038
B8FU0039 B8FU0040 B8FU0041 B8FU0042 B8FU0043 B8FU0044 B8FU0045 B8FU0046
B8FU0047 B8FU0048 B8FU0049 B8FU0050 B8FU0051 B8FU0052 B8FU0053 B8FU0054
B8FU0055 B8FU0056 B8FU0057 B8FU0058 B8FU0059 B8FU0060 B8FU0061 B8FU0062
B8FU0063 B8FU0064 B8FU0065 B8FU0066 B8FU0067 B8FU0068 B8FU0069 B8FU0070
B8FU0071 B8FU0072 B8FU0073 B8FU0074 B8FU0075 B8FU0076 B8FU0077 B8FU0078
B8FU0079 B8FU0080 B8FU0081 B8FU0082 B8FU0083 B8FU0084 B8FU0085 B8FU0086
B8FU0087 B8FU0088 B8FU0089 B8FU0090 B8FU0091 B8FU0092 B8FU0093 B8FU0094
B8FU0095 B8FU0096 B8FU0097 B8FU0098 B8FU0099 B8FU0100 B8FU0101 B8FU0102
B8FU0103 B8FU0104 B8FU0105 B8FU0106 B8FU0107 B8FU0108 B8FU0109 B8FU0110
B8FU0111 B8FU0112 B8FU0113 B8FU0114 B8FU0115 B8FU0116 B8FU0117 B8FU0118
B8FU0119 B8FU0120 B8FU0121 B8FU0122 B8FU0123 B8FU0124 B8FU0125 B8FU0126
B8FU0127 B8FU0128 B8FU0129 B8FU0130 B8FU0131 B8FU0132 B8FU0133 B8FU0134
B8FU0135 B8FU0136 B8FU0137 B8FU0138 B8FU0139 B8FU0140 B8FU0141 B8FU0142

CQT015 -PF1=Toggle 3=Chgopts ENTER=Next CLEAR=Main Menu

```

Figure 30. LISTUSER Toggle panel

- Choose a PF key for the task you want to perform:
    - Change the search or list options (PF03).
    - Return to the main menu (CLEAR).
    - If the panel is full with userids, display the next panel (ENTER).
- To perform a LISTUSER on any ID, enter the ID in the LISTUSER field and press PF01.

### Listing groups for a user ID (LISTUSER command, GROUPS option)

You can use the LISTUSER command with the Groups option to display the groups that a user ID is connected to.

#### Procedure

- Press PF07 from a LISTUSER panel to display the groups that a user is connected to.

```
Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      LISTUSER = BCSCGB1 (Groups)          Time = 11:33:30

BCSC   #READ   P390   SYSAUDIT SYSPROG CRMA   CRMB

PF3=Chgopts 5=Ctgy 6=Segment 7=Group 8=Down 9=User ENTER=Next CLEAR=Main menu
```

Figure 31. LISTUSER GROUPS panel

- Choose a PF key for the task you want to perform:
  - Change the search or list options (PF03).
  - Display the categories for this user (PF05).
  - Return to the LISTUSER panel (PF09).
  - Display the next user if performing a search.
  - Return to the main menu (CLEAR).

### Listing categories for a user ID (LISTUSER command, Categories option)

You can use the LISTUSER command with the Categories option to list the categories that a user ID is connected to.

#### Procedure

- Press PF05 from a LISTUSER panel to display the categories that a user is connected to.

```
Termid = CP24          IBM Security zSecure CICS Toolkit    Date = 2007/094
Userid = BCSCGB1      LISTUSER = BCSCGB1 (Categories)      Time = 11:34:40

00000001

PF3=Chgopts 5=Ctgy 6=Segment 7=Group 8=Down 9=User ENTER=Next CLEAR=Main menu
```

Figure 32. LISTUSER (Categories) panel

- Choose a PF key for the task you want to perform:
  - Change the search or list options (PF03).
  - Display the groups for this user (PF07).
  - Return to the LISTUSER panel (PF09).
  - Display the next user if performing a search.
  - Return to the main menu (CLEAR).

## Listing the TSO and CICS segments for a user ID (LISTUSER command, Segments option)

You can use the **LISTUSER** command with the Segments option to list the TSO and CICS segments for a user ID.

### Procedure

- Press **PF06** from a LISTUSER panel to display the TSO and CICS segments for the user.

```

Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1     LISTUSER = BCSCGB2 (Segments-1)      Time = 11:34:54

TSO
Acctnum = *
Destid =
HClass =              JClass =              MsgClass=              SClass =
Size = 0000000      Maxsize = 0000000      Sec1ab1 =
Proc = ISPFPROC     Unit =              Udata = 0000

CICS
OPIdent = 123
OPPrtty = 123
Timeout = 0000
XRFSoff = NOFORCE
OPClass =

RSLKey =
TSLKey =

PF3=Chgopts 5=Ctgy 6=Segment 7=Group 8=Down 9=User ENTER=Next CLR=Main menu

```

Figure 33. LISTUSER (Segments)

- Press **PF08** to display the OMVS and WORKATTR segments. Pressing **PF08** again scrolls up to the previous segment display.

```

Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1     LISTUSER = BCSCGB2 (Segments-2)      Time = 11:35:00

OMVS
UID = 0000002009
Home =
Program =
ASSizeMax =          CPUTimeMax =
FileProcMax =       MMapAreaMax =
ProcUserMax =       ThreadsMax =
MemLimit =          SHMemMax =

WORKATTR
Name = JOHN SMITH
Account =
Bldg =
Dept = CICS TOOLKIT DEVELOPMENT
Room = ANNEX-1
Addr1 = 'T ZANDT LABS
Addr2 = THE NETHERLANDS
Addr3 =
Addr4 =

PF3=Chgopts 5=Ctgy 6=Segment 7=Group 8=Up 9=User ENTER=Next CLR=Main menu

```

Figure 34. LISTUSER: OMVS and WORKATTR panel

- Choose a PF key for the task you want to perform:
  - Change the search or list options (PF03).
  - Display the groups for this user (PF07).
  - Return to the LISTUSER panel (PF09).
  - Display the next user if performing a search.
  - Return to the main menu (CLEAR).

## Granting or removing access to a resource (PERMIT command)

You can use the **PERMIT** command to grant or remove access to a resource.

## About this task

The resource might be in:

1. One of the resource classes defined in the SIT for this run of CICS or
2. Any other general resource class or the DATASET class.

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.PEMT) and the default group of the user ID or group (PEMT.dfltgrp)

and

If the resource is in a class defined in the SIT, the user must also have access to the resource, at a level equal to or higher than the level of access that is being given. After the PERMIT has been completed, the resource classes must be recreated in order to have immediate effect. zSecure CICS Toolkit does not provide a way for issuing the required **SETROPTS REFRESH** command.

If the resource is in any other class, the user must also have authority to issue **PERMIT** commands in that class (PEMX.classname), and also to the resource, at a level equal to or higher than the level of access that is being given.

## Procedure

1. To access the **PERMIT** command, press the designated **PF** key on the main menu.

```
Termid = CP24          IBM Security zSecure CICS Toolkit    Date = 2007/094
Userid = BCSCGB1      Permit                               Time = 11:35:10

User/Grp =            Rsrcclass =

Resource =

<====

Delete = N    Access = R    (R=Read,N=None,U=Update,A=Alter,C=Control)
Specify "Delete = Y" to remove the user from the access list

CQT029 -Enter userid/group name and resource
PF5=Update ENTER=Redisplay CLEAR=Main menu
```

Figure 35. PERMIT panel

### USERID

The name of the user or group to be granted access (or removed).

### RESOURCE

The name of the resource (for example, CEMT if it was the transaction CEMT).

### RSRCLASS

The resource class name. If blank, the value of the XTRAN parameter specified in the SIT is used.

### **DELETE**

Specify Y in this field to remove a user or group from the access list for this resource.

### **ACCESS**

Specify R for READ access or N for NONE, U for update, A for alter or C for control. If N is specified, the user must have a minimum of READ authority to issue the command.

2. Update or specify the information and press **PF5**. The user executing the **PERMIT** command must have access to the resource that is being altered. For example, if access is being given to CEMT, the user must have access to CEMT. If DELETE is specified as Y, the user executing the command must still have access to the resource. The level of access required is whatever specified in the **ACCESS** field.

---

## **Maintaining associations (RACLINK command)**

You can use the **RACLINK** command to define, list, undefine, or approve user associations.

### **About this task**

The **RACLINK** command only works with local nodes. Independent of the value of the **NODE** specified on the panel, zSecure CICS Toolkit assumes it to be the name of the local node.

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.RACL). To issue a **RACLINK** for a different user, the user must have RACF Special, TOOLKIT.SPEC or access to the default group of the user (RACL.dfltgrp).

### **Procedure**

1. To access the **RACLINK** command, press the designated PF key on the main menu.

```

Termid = CP24          IBM Security zSecure CICS Toolkit    Date = 2007/094
Userid = BCSCGB1     RACLINK = BCSCGB1                Time = 11:37:03

      (               ) <= Password(optional)
- ---TYPE--- --NODE-- -USERID- PWD          Status      Created by YYYY/MM/DD
                     Sync
- PEER OF   IDFX   IBMUSER NO   ESTABLISHED   BCSCGB1    2007/04/04
- PEER OF   IDFX   BCSCGB2 NO   ESTABLISHED   BCSCGB1    2007/04/04
- PEER OF   OBLX   BCSCGB2 NO   ESTABLISHED   BCSCGB1    2007/04/04
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-

PF5=Update 8=Down ENTER=List CLEAR=Main menu

```

Figure 36. RACLINK panel

**TYPE** The type of association. It can be PEER or MANAGED.

**NODE**  
 The name of the node where the association is defined.

**USERID**  
 The user that the association is being defined for.

**PWD SYNC**  
 Specifies if the association has password synchronization or not (enter YES or NO)

**PASSWORD**  
 Optional parameter. The password for the user specified.

2. In the field RACLINK =, enter the user for which the command is being issued. Press **Enter** to list the associations for that user.
3. Enter a U to the left of an association and then press PF05 to cause it to be undefined.
  - The association needs to be for your own ID, or you need system special, TOOLKIT.SPEC or RACL.dfltgrp.
  - If the association is awaiting approval and it is your own ID (or you have system special or zSecure CICS Toolkit special), you can approve it by entering A to the left of it and pressing PF05.
  - In the preceding display, a new association is about to be defined. After the user presses the PF5 key, the new association is defined. The D must be present to indicate a DEFINE.
  - The association is implicitly approved if a valid PASSWORD was provided or the issuer of the command has RACF SPECIAL, TOOLKIT.SPEC or RACL.dfltgrp.
  - To define a new user association, you need access to the normal RACLINK authorizations as defined for the TSO RACLINK command. These authorizations are RACLINK.DEFINE.nodename and RACLINK.PWSYNC.nodename.



## Listing and maintaining profiles in a general resource class (RALTER / RDEFINE / RDELETE commands)

You can use the **RALTER**, **RDEFINE**, and **RDELETE** commands to list and maintain profiles in a general resource class defined in the CDT.

### About this task

The user must have access to the zSecure CICS Toolkit command (**TOOLKIT.RALT** / **TOOLKIT.RDEF** / **TOOLKIT.RDEL**, depending on the command being performed) and to the general resource class (RLST.cdtclass in addition to RALT.cdtclass / RDEF.cdtclass / RDEL.cdtclass).

### Procedure

1. To access the **RALTER** / **RDEFINE** / **RDELETE** command, press the designated **PF** key on the main menu.

```
Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      RES Class =                          Type =          Time = 11:37:16
Profile

Member

Owner =              Notify =              Uacc = None      Warn = N Level = 000
Audit = F Aud succ = R Aud fail = R

-----1-----2-----3-Installation data-5-----6-----7-----

|<===
-----1-----2-----3-----4-----5-----6-----7-----

PF1=Rdef PF2=Addmem PF3=Rdel PF4=Delmem PF5=Updprof ENTER=R1st CLEAR=Main menu
```

Figure 37. **RALTER** / **RDEFINE** / **RDELETE** panel

#### RDEFINE

To define a new profile/resource, all fields are required, except for the installation data. After entering these information, press **PF01** to perform the **RDEFINE**.

#### RDELETE

To delete a profile/resource, enter the CLASS and PROFILE names and press **PF03**.

#### RALTER

Allows you to add a new member to a group (ADDMEM), delete a member from a group (DELMEM), or update the profile information (UPDPROF). The information required depends on the subcommand being performed.

#### ADDMEM

Requires the CLASS, PROFILE and MEMBER to be added. Press **PF02** to complete the ADDMEM.

### DELMEM

Requires the CLASS, PROFILE and MEMBER to be deleted. Press **PF04** to delete the member.

### UPDPROF

Requires all fields, except for the installation data. By entering the **CLASS** and **PROFILE** fields and pressing **Enter**, all the current entries for each field display. These entries can then be over stepped. Press **PF05** to complete the update.

2. Specify the values for the action you want to perform using the field descriptions in step 1, then press the corresponding **PF** key to initiate the change.

---

## Removing user IDs or groups from a group (REMOVE command)

You can use the **REMOVE** command to remove user IDs or groups from a group. User IDs cannot be removed from their default group.

### About this task

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.REMV) and the target group (REMV.*grpname*).

### Procedure

1. To access the REMOVE command, press the designated PF key on the main menu.

```
Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1                                           Time = 11:37:31

Remove = Userid      Group =

CQT016 -Enter userid and group name
PF5=Update ENTER=Redisplay CLEAR=Main menu
```

Figure 38. REMOVE panel

2. To remove a user from a group, enter the user and group name as indicated and press **PF05**. Users cannot be removed from their default group.

---

## Listing the profiles for a general resource class (RLIST command)

You can use the **RLIST** command to list the profiles for a general resource class defined in the CDT.

## About this task

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.RLST) and the general resource class (RLST.cdtclass).

## Procedure

1. To access the **RLIST** command, press the designated **PF** key on the main menu.

```
Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      Rlist class =                      Type =
                               Profile
-----
-----
-----
Owner = ***** Dte = ***** Last ref = ***** Last chg = ***** Uacc = *****
Audit = * Aud succ = * Aud fail = * Notify = ***** Warn = * Level = ***
Glbl Audit = * Gaud Succ = * Gaud Fail = * Sec1 = ***
Members = **** NumUsrs = **** Condacc = ****

-----1-----2-----3-Installation data-5-----6-----7-----
*****
*****
*****
***** |<===
-----1-----2-----3-----4-----5-----6-----7-----
PF1=Toggle 3=Chgopts 5=Members 7=Users 9=Condacc 11=Search CLEAR=Main menu
```

Figure 39. RLIST panel

### RLIST CLASS

This is the name of the resource class to be displayed. It must be a valid entry in the RACF Class Descriptor Table.

**TYPE** This defines the class as being a member (TYPE=M) or group (TYPE=G) class. It is provided as information only and is not used as input for a list or search.

### PROFILE

The name of the profile to be displayed. Enter the name of the profile to be displayed in this field. If a search is being performed, this field can be bypassed, or any characters can be entered in any position as part of the search criteria. This field can be up to 246 characters in length.

**Note:** If you are only performing an **RLIST**, the rest of the fields are ignored. The remaining fields are only used as input when doing a search (PF11).

### OWNER

A user or group that has been defined as the owner of the profile.

**DTE** The date this profile was created. The format is YYDDD.

### LAST REF

The date the data set was last referenced. The format is YYDDD.

### LAST CHG

The date the profile was last updated. The format is YYDDD.

**UACC** The universal access for the profile. This field can be ALTER, CONTROL, UPDATE, READ or NONE.

**AUDIT**

Indicates the audit flag for the profile. The settings can be: A to audit all accesses, S to audit successful accesses, F to audit failures, or N for no auditing.

**AUD SUCC**

This is the audit SUCCESS flag. The settings can be: R to audit successful reads, U to audit successful updates, C to audit control accesses, or A to audit successful alter accesses.

**AUD FAIL**

This is the audit FAILURE flag. The settings can be: R to audit unsuccessful reads, U to audit unsuccessful updates, C to audit unsuccessful control accesses, or A to audit unsuccessful alter accesses.

**NOTIFY**

The user to be notified when access is denied to this profile.

**WARN**

Indicates if the profile is in warning mode. This field can be Y or N.

**LEVEL**

The level indicator for the data set. This field is a numeric field.

**GLBL AUDIT**

The Global audit options as specified as by a user with the AUDITOR attribute. The settings can be: A to audit all accessed, S to audit successful accesses, F to audit failures, or N for no auditing.

**GAUD SUCC**

This is the GLOBAL audit SUCCESS flag. The settings can be: R to audit successful reads, U to audit successful updates, C to audit successful control accesses, or A to audit successful alter accesses.

**GAUD FAIL**

This is the GLOBAL audit FAILURE flag. The settings can be: R to audit unsuccessful reads, U to audit unsuccessful updates, C to audit unsuccessful control accesses, or A to audit unsuccessful alter accesses.

**SECL** The security level of the profile. This field is a numeric field.

**MEMBERS**

The number of members in this profile, if it is a group profile.

**NUMUSRS**

The number of users and groups authorized to access the profile. This field is a numeric field.

**CONDACC**

The number of user/groups on the conditional access list. This field is a numeric field.

**INSTALLATION DATA**

The information contained in the DATA field of the profile. It can be up to 255 characters.

2. You can choose to:

- Display the users/groups with access to the profile (PF07).
- Display users/groups on the conditional access list (PF09).
- If the resource class is a group class, as indicated by the **TYPE=** field, display the members in the profile (PF05).

When displaying members, users, or the conditional access list, you can use **PF08** page down if there is more than one panel to be displayed.

## RLIST Display Example

This example shows what is displayed when you enter a profile to be listed.

If you enter a profile to be listed, or initiate a search, the next panel is displayed as follows.

```

Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      Rlist class = GCICSTRN Type = G       Time = 11:38:02
                                     Profile
CICSA.CAT1

Owner = SYS1      Dte = 05033 Last ref = 05033 Last chg = 05033 Uacc = NONE
Audit = F Aud succ = R Aud fail = R Notify = ***** Warn = N Level = 000
Glbl Audit = N Gaud Succ = R Gaud Fail = R Sec1 = ***
Members = 0051 NumUsrs = 0003 Condacc = 0000

-----1-----2-----3-Installation data-5-----6-----7-----

                                     |<===
-----1-----2-----3-----4-----5-----6-----7-----

PF1=Toggle 3=Chgopts 5=Members 7=Users 9=Condacc 11=Search CLEAR=Main menu

```

Figure 40. RLIST Display panel

From this point, you can choose to:

- Display the members (PF05).
- Display the users (PF07).
- Display the conditional access list (PF09).
- Change the search or list options (PF03).
- Return to the main menu (CLEAR).
- If you are performing a search, display all profiles that meet the criteria (PF01).

## Listing the members in a profile (RLIST command, MEMBERS option)

You can use the **RLIST** command with the **Members** option to list the members in a profile.

### Procedure

- Press **PF05** from an RLIST panel to display the members in profile.

```

Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      Rlist = (MEMBERS)                    Time = 11:38:12
CICSA.CAT1

CICSA.CRTP
CICSA.CPIR
CICSA.CATA
CICSA.CATD
CICSA.CDBD
CICSA.CDBF
CICSA.CDBO
CICSA.CDBQ
CICSA.CDTS
CICSA.CESC
CICSA.CESD
CICSA.CEX2
CICSA.CFCL
CICSA.CFOR
CICSA.CFQR
CICSA.CFQS
CICSA.CFTL
CICSA.CFTS

PF1=Toggle 3=Chgopts 5=Memb 7=User 8=Down 9=Condacc ENTER=Next CLEAR=Main menu

```

Figure 41. RLIST Members panel

- From this point, you can choose to:
  - Display the users (PF07).
  - Display the conditional access list (PF09).
  - Change the search or list options (PF03).
  - Return to the main menu (CLEAR).
  - If you are performing a search, display the next profile (ENTER).

## Listing user IDs in a profile and the access they have (RLIST command, USERS option)

You can use the **RLIST** command with the Users option to list the user IDs in the profile and the access they have.

### Procedure

- Press **PF07** from an RLIST panel to display the users in the profile and the access they have.

```
Termid = CP24          IBM Security zSecure CICS Toolkit    Date = 2007/094
Userid = BCSCGB1      Rlist = (USERS)                   Time = 11:38:23
CICSA.CAT1
```

```
IBMUSER /A CICSA /R CICSASTC/R
```

```
PF1=Toggle 3=Chgopts 5=Memb 7=User 8=Down 9=Condacc ENTER=Next CLEAR=Main menu
```

Figure 42. RLIST Users panel

- From this point, you can choose to:
  - Display the members (PF05).
  - Display the conditional access list (PF09).
  - Change the search or list options (PF03).
  - Return to the main menu (CLEAR).
  - If you are performing a search, display the next profile (ENTER).

## Listing users/groups in the conditional access list for a profile (RLIST command, CONDACC option)

You can use the **RLIST** command with the Conditional Access option to list the users/groups in the conditional access list for the profile.

### Procedure

- Press **PF09** from an RLIST panel to display the user/groups in the conditional access list for the profile.

```
Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      Rlist = (CONDACC)                   Time = 11:39:04
CICSA.CAT1

BCSC /R-TERMINAL=D20AK021

PF1=Toggle 3=Chgopts 5=Memb 7=User 8=Down 9=Condacc ENTER=Next CLEAR=Main menu
```

Figure 43. RLIST Conditional Access panel

- From this point, you can choose to:
  - Display the members (PF05).
  - Display the users (PF07).
  - Change the search or list options (PF03).
  - Return to the main menu (CLEAR).
  - If you are performing a search, display the next profile (Enter).

## Listing, adding, updating, or removing the USRDATE fields from a profile (USRDATA command)

You can use the **USRDATA** command to list, add, update, or remove the USRDATA fields from a user profile.

### About this task

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.USRL), the userid (USRU.dfltgrp), and the USRDATA name (USRN.usrdata-name). For the **ADD**, **UPDATE** and **DELETE** subfunctions, access to the corresponding command profile is required (TOOLKIT.USRA for **ADD** and **UPDATE** or TOOLKIT.USRD for **DELETE**).

### Procedure

1. To access the **USRDATA** command, press the designated **PF** key on the main menu.



```

Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      USRDATA                               Time = 11:39:16

Fill in Profile and ENTER. For Add, fill in fields, select A and PF5
_Class = USER      Profile =
USRDATA
  Name =            Value =

                                     |<===
  Name      Value  (Use S/L and ENTER for details, or D and PF5 for delete)
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-

CQT018 -Enter userid
PF5=Update 8=Down ENTER=List CLEAR=Main menu

```

Figure 44. USRDATA panel

2. To display USRDATA of a user, enter the user ID and press **Enter**.
  - The USRDATA names and the first 64 characters of the corresponding values display on the bottom part of the panel.
  - If more USRDATA names are present than fit on the panel, press **PF8** to scroll down.
  - For an untruncated display of one USRDATA value, use the **S** (or **L**) command in front of the wanted USRDATA name/value and press **Enter**.

```

Termid = CP24          IBM Security zSecure CICS Toolkit      Date = 2007/094
Userid = BCSCGB1      USRDATA                               Time = 11:41:00

  Class = USER      Profile = BCSCGB2
USRDATA
  Name =  PHONE      Value = +1 123-456-7890

                                     |<===

PF5=Update 11=Delete ENTER=Refresh CLEAR=Back

```

Figure 45. USRDATA Display panel

3. To add USRDATA for a user, enter the name and value of the USRDATA, then enter an A in the field in front of the **CLASS** and press **PF05**. You can also use

this same method to delete or update USRDATA fields from selection of a D or U in this command field. This latter method is an alternative to the preferred methods described here.

4. To delete one of the displayed USRDATA name/value pairs, use either of the following methods:
  - Use the **D** command in front of the USRDATA you want, and press **PF05**.
  - Use the **S** or **L** line command to display the USRDATA value, followed by **PF11** on the detail panel.
5. To update existing USRDATA values, go to the detail panel obtained by using **S** (or **L**) in front of the listed USRDATA name. On the detail panel, type over the value with the new value and press **PF05**.

---

## Chapter 6. zSecure CICS Toolkit exit points specifications

In the EXITPGM parameter in CQTPCNTL, you can specify a program that is to receive control whenever the main zSecure CICS Toolkit transaction (normally RTMM) terminates and returns control to CICS.

Control is passed to the EXITPGM through the XCTL command. zSecure CICS Toolkit does not receive control again after the EXITPGM, which includes a return code and the data, if any, that was sent to the user's panel. The format of the COMMAREA is as follows:

EXITRC	DC CL1	Return Code
		0 = CICS Toolkit transaction has terminated
		1 = Signon transaction failed or was terminated with the clear key
		3 = Signon completed. User was signed-on at a second terminal but was not authorized (did not have access to TOOLKIT.DUPE and DUPEUSER checking is in effect).
		4 = Signon completed. Same as 3, but terminal logged off CICS.
		5 = Signon completed. No CICS segment found for user.
		6 = Signon completed. CICS segment was found for user.
		7 = Signon completed. Error in installation data being used for operator information.
*		
EXITDATA	DC CL335	Data sent to users screen from signon or CICS Toolkit transaction termination. If the return code is 0, this field is only 79 bytes in length. For any other return code, this field will contain the data, if any, that was sent to the terminal user.

A sample EXITPG is provided in the SCQTSAMP pds as member CQTXSNEX.



---

## Chapter 7. Application programming interface (API)

The zSecure CICS Toolkit Applications Programming Interface allows users to access the RACF database directly from a CICS application program. No special knowledge of RACF or its database format is required and the applications do not need to run authorized.

Using the API, an installation can tailor the zSecure CICS Toolkit panels to suit their own requirements, or produce different types or reports. zSecure CICS Toolkit ensures that only authorized users access the RACF database. The same rules apply when using the API as for using zSecure CICS Toolkit. The user executing a transaction that is using the API must be authorized to the zSecure CICS Toolkit commands used. If requesting user information, the user must have access to the default group of the user profile being displayed. For more information about this subject, see Chapter 5, “The zSecure CICS Toolkit command interface,” on page 35.

Using the API is a procedure that only requires the CICS application program to call the interface module, CQTPAPI0, and pass certain parameters to it in a COMMAREA. These parameters inform zSecure CICS Toolkit of the command being requested and also provide the storage area where the requested information is returned.

For compatibility reasons, the CQTPAPI0 program has an alias CRTKAPI. Although both names refer to the same module, use the new name, CQTPAIP0, in all applications.

For any option used to alter profiles, SMF records are produced, indicating the changes and who changed them. These records show up in your normal RACF reports.

The SMF records all show the value TOOLKIT\* in the SMF80UID field. This special value is used to indicate that the record was produced as part of a zSecure CICS Toolkit function.

zSecure CICS Toolkit does not cause the CICS main task to wait while it is reading or updating the RACF database. All these commands are processed by the zSecure CICS Toolkit subtasks, leaving CICS free to continue with normal transaction processing. This CICS region does not run in an authorized state at any time, in keeping with the IBM statement of integrity.

---

### Command requests using the COMMAREA

The COMMAREA is the way zSecure CICS Toolkit is informed of the command being requested. The size of the COMMAREA varies depending upon the command being executed. In all cases, there is a common header, followed by specific information for the relevant command.

The format of the header is as follows:

API_FUNCT	DC CL4	This field specifies the command being requested.
*		
*		
API_RC	DC XL1	A one byte hexadecimal return code.
*		

```

*
API_MSG          DC  CL79          The message that would normally
*                be displayed on the terminal if
*                the user was using the standard
*                Toolkit transaction is returned
*                in this field.

```

The application program invokes the API using a standard CICS LINK command:

```
EXEC CICS LINK PROGRAM('CQTPAPI0') COMMAREA(APICOMM) LENGTH(APILEN)
```

If a COMMAREA is not passed, CTKAPI just returns to the caller without any further processing. If an error is detected in the COMMAREA layout, either the length or the command requested is invalid. Or, the user is not authorized to the zSecure CICS Toolkit COMMAND and the API sets API\_RC with a return code, indicating the nature of the error. The possible return codes are:

#### RETURN CODE

- X'00'** COMMAREA is correct
- X'01'** An invalid command was requested. Check the parameter specified in API\_FUNC and verify that it is correct.
- X'02'** The length of the COMMAREA is too small for the requested command.
- X'03'** The user is not authorized to use this zSecure CICS Toolkit COMMAND or is not signed on at the terminal.
- X'04'** There was no profile protecting the TOOLKIT,function resource. An authorization decision could not be made, and the API function was not executed.
- X'05'** Internal error, contact Technical Support

If the commarea passed to the API is correct, but the function fails for some other reason, the API\_RC contains the value x'00'. The function-specific return code field (typically called *API\_function\_RC*) contains an error indicator. Most API-functions use the value -1 as indicator that an error occurred. The field API\_MSG contains the error message describing the failure. Here are the examples of these messages:

- CQT039 is for the **ALTUSER** command if the specified ID does not exist.
- CQT080 is for the **LIST** commands if the requested profile cannot be found.

For the complete text of the error messages, see the *IBM Security zSecure: Messages Guide*.

All fields in the COMMAREA must be padded with blanks unless indicated otherwise in the documentation.

## Change the authorized user

To execute a command through the API, the userid associated with the task must be authorized to the zSecure CICS Toolkit command. This userid is normally the one for the user logged on at the terminal or the CICS default user. You might specify a different userid to be the authorized user.

To change the authorized user, make the following definition in the first 24 bytes of the API-MSG area before calling CQTPAPI0:

Table 7. API-MSG definition to change the authorized userid

API Message Variable	Value	Description
API_MSG_USERAUTH	DC CL8'USERAUTH'	A constant of USERAUTH
API_MSG_USERID	DC CL8'USERAUTH'	The USERID to be used as the authorized user.
API_MSG_PASSWORD	DC CL8' <i>password</i> '	The PASSWORD for the user.

If the password is incorrect for the user, the command fails and the appropriate message is returned to the calling program.

---

## Perform a search

A search of the RACF database can be performed using the **LISTUSER**, **LISTGROUP**, **LISTDATASET**, and **RLIST** commands.

The COMMAREA for each of these commands contains a one-byte code field. It indicates whether a search is being performed, if the next profile must be retrieved, or if other information about the profile is required.

When a search is requested, the profile attribute fields in the COMMAREA must be padded with asterisks. The search mask can be any combination of valid characters or letters in any combination of fields. When zSecure CICS Toolkit finds a match, it returns the profile information into the COMMAREA. To retrieve the next profile that matches the search criteria, set the code field in the COMMAREA to N (next) and call the API. The profile itself must not be padded with asterisks, but instead must be padded with blanks (x'40'), nulls (x'00') or underscores (x'6D'). The reason for this exception to the general masking rule, is to allow a specific search for profiles containing generic characters.

As zSecure CICS Toolkit returns the profile information into the COMMAREA, you must build the search mask in working storage and move it to the COMMAREA before each call to the API. There is also an API\_RESERVED field used by the API during a search and the contents of this field must be preserved between calls.

To retrieve all profiles (for example, all user profiles), the search mask must be all asterisks. Initiate the search by specifying S in the code field. You can then retrieve the rest of the profiles by setting the code field to N. Continue to call the API until a non-zero return code is returned. The API\_MSG field also contains a message indicating that there are no more profiles matching the criteria or that you have reached the end of the RACF database. Remember to recreate the search mask before each call.

---

## Implementing field or record level security

Field or record level security can be implemented by using the API resource authorization checking capabilities, especially when using resource names of up to 246 bytes.

### About this task

By defining resource names that represent a particular field or record within a file, access to those records/fields can be restricted. An application program can call the

API to verify the access authority of a user and determine what action must be taken. The action includes updating the record, displaying the record, updating the field, blanking out the field, and so on.

An example might be for a file with a DDNAME of PAYFILE, whose keys are social security numbers.

### Procedure

1. Define the DDNAME and social security numbers to RACF:  
RDEFINE RSRCLASS PAYFILE.999-99-999 UACC(NONE)
2. Permit the user to the record:  
PERMIT PAYFILE.999-99-999 CLASS(RSRCLASS) ID(USERIDA) ACC(READ)

### What to do next

The application can now call the API to perform a resource access check and determine the users' level of access to the record/field.

## Access Authority Check function

You can use the Access Authority Check function to determine if a user has access to one or more resources. No authority is required.

### COMMAREA

Minimum size 99 bytes.

In your application, use the CQTMPIA or CQTMPIIC mapping macros (copybooks) provided in the SCQTMAC library. Example:

```

API_FUNC      DC  CL4'RSRC'  FUNCTION code for access check
API_RC        DC  XL01'00'   Return code
API_MSG       DC  CL79' '    Message area
*
API_RSRC_NAMES DC  XCL14     This is a list of resources for
*                           which the access authority of the user
*                           signed on at the terminal is to be checked.
*                           The size of this field depends on the
*                           number of resources being checked. Each
*                           resource name requires thirteen bytes,
*                           padded with blanks, followed by a one byte
*                           return code field.
*
*                           The return code field may also specify the
*                           level access to be checked. This may be
*                           'R' (read), 'U' (update), 'C' (control) or
*                           'A' (alter). Read is the default.
*
*                           For example to check the users access to
*                           AUDIT and PAYROLL the following
*                           entries could be coded.
*
*                           DC  CL13'AUDIT'   First resource name
*                           DC  XL1'00'      Return Code
*                           DC  CL13'PAYROLL' Next resource name
*                           DC  XL1'00'      Return Code
*
*                           DC  XL1'FF'      The last field in the COMMAREA
*                           must be a one byte field containing X'FF',
*                           indicating the end of the list of
*                           resource names.

```



The resource class used when making the access check is that specified in the RSRCLASS parameter in CQTPCNTL.

If a prefix has been specified for application resource names (see the CICSAPPL parameter of CQTPCNTL), it is used to prefix the resource names passed to CQTPAPI0. Refer to Chapter 4, "Application security management," on page 29 for more information about application security and defining resources to RACF.

SMF records are produced depending on the AUDIT parameters specified for the resources. If you want to suppress the ICH408I messages on the system console and the CICS log, you can specify the value S in the API\_RC field. Specifying the value results in suppression of possible access violation messages, while still creating SMF records about these violations.

The return codes are a 1-byte hexadecimal field with the following meanings:

#### RETURN CODE

- X'00' Access allowed to resource.
- X'04' The resource is not defined to RACF.
- X'08' The user is not authorized to use the resource.
- X'0C' RACF is not active.
- X'10' FRACHECK installation exit error.
- X'14' RACF is not installed or at the wrong level.

---

## Access Authority Check (Extended) function

You can use the Access Authority Check (Extended) function to check if a user has access to one or more resources. No authority is required.

#### COMMAREA

The minimum size is 348 bytes.

In your application, use the CQTMPIA or CQTMPIB mapping macros (copybooks) provided in the SCQTMAC library. Example:

```
API_FUNC      DC  CL4'RSRX'  Function code for access check
API_RC        DC  XL01'00'   Return code
API_MSG       DC  CL79' '    Message area
*
API_RSRX_USERID  DC  CL8      Specify the USERID to be used to perform
*                          third party authorization checking.
*                          If blank the ACEE of the user Signed on
*                          at the terminal is used.
*
API_RSRX_CLASS  DC  CL8      This field may be used to specify the
*                          resource class to be used
*                          the access checks. If used, it overrides
*                          the definition x for RSRCLASS in CQTPCNTL.
*                          It must be x a valid class defined in the
*                          SIT unless a userid has been specified
*                          in API_RSRX_USERID. In this case
*                          any class may be specified.
*
API_RSRX_NAMES  DC  XCL247   This is a list of resources for which
*                          the access authority of the user signed
*                          on at the terminal is to be checked.
*                          The size of this field depends on the number
*                          of resources being checked. Each
*                          resource name requires 246 bytes, padded with
*                          blanks, followed by a one byte return code field.
*
*                          The return code field may also specify
*                          the level of access to be checked.
*
```

```

*                               This may be: 'R' (read), 'U' (update),
*                               'C' (control) or 'A' (alter).
*                               Read is the default.
*
API_RSRX_ACC      EQU API_RSRX_NAMES+246,1
*                               The access level
*
*                               For example to check the users access to
*                               AUDIT and PAYROLL the following entries
*                               could be coded.
*                               DC CL246'AUDIT'   First resource name
*                               DC XL1'00'       Return Code
*                               DC CL246'PAYROLL' Next resource name
*                               DC XL1'00'       Return Code
*
*                               DC XL1'FF'       The last field in the COMMAREA must be a
*                               one byte field containing X'FF',
*                               indicating the end of the list of
*                               resource names.

```

The resource class used when making the access check is that specified in the RSRCLASS parameter in CQTPCNTL. If specified, the value in API\_RSRX\_RSRCLASS overrides that in CQTPCNTL.

If a prefix has been specified for application resource names (see the CICSAPPL parameter of CQTPCNTL), it is used to prefix the resource names passed to CQTPAPI0. Refer to Chapter 4, “Application security management,” on page 29 for more information about application security and defining resources to RACF.

SMF records are produced depending on the AUDIT parameters specified for the resources. If you want to suppress the ICH408I messages on the system console and the CICS log, you can specify the value N in the API\_RC field. Specifying the value results in Suppression of possible access violation messages, while still creating SMF records about these violations.

The return codes are a 1-byte hexadecimal field with the following meanings:

**RETURN CODE:**

- X'00' Access allowed to resource.
- X'04' The resource is not defined to RACF.
- X'08' The user is not authorized to use the resource.
- X'0C' RACF is not active.
- X'10' FRACHECK installation exit error.
- X'14' RACF is not installed or at the wrong level.

---

## Resource Profile List function

You can use the Resource Profile List function to provide a list of the profiles accessible to a user. No authorization is required.

You can use a function provided through the zSecure CICS Toolkit API for high-performance listing of all authorized profiles in a specified resource class. This API provides an alternative to the combined SEARCH and RLIST interfaces. You can use it to list all profiles in the specified resource class to which a user has access at a certain level. Internally, the API is based on high-performance RACF functions like the Profile Name List function (IRRPNL00) and the fast authorization checking function (RACROUTE REQUEST=FASTAUTH). This function is available only through the API.

## COMMAREA

Minimum size 146 bytes.

In your application, use the APICOMMA or APICOMMC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'RSRL'	FUNCTION code for Resource List
API_RC	DC	XL1'00'	API Return code
API_MSG	DC	CL79' '	Message area
*			
API_RSRL_RET	DS	XL1	Return code
API_RSRL_REAS	DS	XL1	Reason code
API_RSRL_CLASS	DS	CL8	CLASS
API_RSRL_USERID	DS	CL8	USERID or blank
API_RSRL_GROUP	DS	CL8	GROUPID or blank
API_RSRL_TSQUEUE	DS	CL16	TSQUEUE name or blank
API_RSRL_Prefix	DS	CL16	Filter prefix or blank
API_RSRL_Retflag	DS	CL1	Processing flags
*			"C" Return data in Commarea
*			"T" Return data in TSQUEUE
*			"B" Return data in Comm/TSQ
*			"N" Don't return any data
API_RSRL_FL_PR	DS	CL1	Use filter prefix Y/N
API_RSRL_FL_AC	DS	CL1	Return all profiles Y/N
API_RSRL_ACCESS	DS	CL1	Requested Access
*			"R" Read
*			"U" Update
*			"C" Control
*			"A" Alter
API_RSRL_PROFCNT	DS	XL4	Returned number of entries
API_RSRL_PROFLST	DS	XL2	Length of the profile below (CLx)
	DS	CL1	Flag byte
*			Possible values for the flag byte
*			"A" Access to discrete profile
*			"N" READ to discrete profile
*			"B" Access to generic profile
*			"G" READ to generic profile
	DS	CLx	Profile

Output of this function consists of a list of profile names. The list is returned in the provided commarea or in the specified Temporary Storage Queue (TSQUEUE). The return option set in API\_RSRL\_RETFLAG determines which of the return areas is used. The list of profiles contains either all profiles to which the user has at least READ access or only those profiles to which the user has at least the access specified in API\_RSRL\_ACCESS. The API\_RSRL\_FL\_AC flag byte controls what the list of profiles contains. You can filter the list of profiles based on the first characters of the profile name. Specify the filter pattern in the field API\_RSRL\_Prefix. Filtering is activated by the API\_RSRL\_FL\_PR flag byte.

When profiles are to be returned in a CICS TSQUEUE, a non-recoverable MAIN-storage TSQUEUE must be specified. When using a recoverable TSQUEUE, the program might terminate with an ATSP ABEND. Using an AUXILIARY storage queue might involve I/O to auxiliary storage, impacting the application response time. If you want the associated system resources freed, the calling program must delete the TSQUEUE after processing the data.

When profiles are to be returned in the provided COMMAREA, be sure that the commarea is sufficiently large. If the output data does not fit in the provided space, the data is truncated and a message is issued. This RSRL API does not allow retrieval of the remainder of the truncated data. If additional data is required, the request must be re-issued providing a larger commarea, or requesting that output is returned in a TSQUEUE.

The specified resource class must have been RACLISTed using the RACF SETROPTS command. Globally RACLISTed resource classes, like the CICS resource class TCICSTRN, are not supported.

The following list describes the fields in the API commarea.

**API\_FUNC**

Describes the function being called. For the List Authorized Resources function, it must contain 'RSRL'.

**API\_RC**

The API return code. This is the return code for the API interface. The return code from the RSRL function is provided in the API\_RSRL\_RETC field. Possible values for the API\_RETC are described in "Command requests using the COMMAREA" on page 87.

**API\_MSG**

A warning or error message.

**API\_RSRL\_RETC**

The return code from the RSRL function. See "Return and Reason codes" on page 96 for a description of the possible return codes.

**API\_RSRL\_REAS**

The reason code from the RSRL function. See "Return and Reason codes" on page 96 for a description of the possible reason codes.

**API\_RSRL\_CLASS**

The resource class for which the authorized profiles are required. This resource class must be RACLISTed using the SETROPTS RACLIST command. Globally RACLISTed resource classes, like the CICS resource class TCICSTRN, are not supported. If the resource class is not SETROPTS RACLIST, an error message is issued and execution stops. The class name must be specified in full. Abbreviations and generic characters are not supported.

**API\_RSRL\_USERID**

The user for which the list of authorized profiles is to be determined. If this field is empty, the list of authorized profiles is determined for the logged-on terminal user. The field is considered empty if the first position contains either a blank or a hexadecimal null.

**API\_RSRL\_GROUP**

The RACF group to be used for the user specified in API\_RSRL\_USERID. This field must either be empty or contain a valid non-revoked group connect for the specified user. This field is ignored if the API\_RSRL\_USERID field is empty.

**API\_RSRL\_TSQUEUE**

Specifies the CICS Temporary Storage Queue (TSQUEUE) to be used for the output data. The TSQUEUE name is up to 16 characters long, and must be padded with blanks or nulls. The name must be that of a non-recoverable MAIN-storage TSQUEUE. Recoverable TSQUEUEs are not supported. Using an AUXILIARY storage queue might involve I/O to auxiliary storage, impacting the application response time. If you want the associated system resources freed, the calling program must delete the TSQUEUE after processing the data. This field is ignored if the API\_RSRL\_RETFLAG has any value other than T or B.

**API\_RSRL\_PREFIX**

Specifies the prefix filter to be used for filtering the authorized profiles.

The prefix is up to 16 characters long, and must be padded with blanks or nulls. The prefix is used to compare against the profiles, similar to the process used for the TSO SEARCH MASK keyword. The first characters of the profile name must match the characters specified for the prefix filter. If these characters do not match, the profile is skipped and is excluded from the list of profiles.

#### **API\_RSRL\_RETFLAG**

Specifies processing flags. The following values for the processing flags can be used:

- C** The output data is returned in the COMMAREA. The number of entries located is shown in the API\_RSRL\_PROFCNT field. If the COMMAREA is too small, this number can be higher than the actual number of profiles in the COMMAREA.
- T** The output data is returned in the specified TSQUEUE. The number of entries located is shown in the API\_RSRL\_PROFCNT field.
- B** The output data is returned in the COMMAREA and the specified TSQUEUE. The number of entries located is shown in the API\_RSRL\_PROFCNT field.
- N** No output data is returned in either the COMMAREA or the specified TSQUEUE. The number of entries that might have been returned is still shown in the API\_RSRL\_PROFCNT field.

#### **API\_RSRL\_FL\_PR**

The prefix filter is used to limit the profiles listed. Possible values are Y or N. Any other value is treated as if the value is N.

#### **API\_RSRL\_FL\_AC**

The output data always includes only those profiles to which the user has at least READ access. Profiles to which the user has no access are never shown. The API\_RSRL\_FL\_AC flag can be used to reduce the profiles returned. Possible values for the flag are Y and N. Any other value is treated as if the value is N.

If this flag has the value N, only the profiles to which the user has at least the requested access are returned. If this flag has the value Y, all profiles to which the user has at least READ access are returned, and the profile flag shows whether the user has only READ access or the requested access.

#### **API\_RSRL\_ACCESS**

Specifies the minimum required access of the user. Possible values are R for READ, U for UPDATE, C for CONTROL, or A for ALTER. If this field is empty or contains any other value, access is determined as if the value R was specified. If the user has access at the requested access level or higher, the profile flag has the value A or B. If the user does not have the requested access, the profile flag has the value N or G.

#### **API\_RSRL\_PROFCNT**

This output field shows the number of entries listed. If the response area is large enough, it contains the number of profiles returned. If the response area is too small, it contains the number of profiles that would have been returned if the response area were large enough.

#### **API\_RSRL\_PROFLST**

This part of the commarea contains the list of authorized profiles. It consists of an array of profiles using the following format:

**NAME LENGTH**

The 2-byte length of the profile name. This value does not include the length of the length field itself or the length of the FLAG byte.

**FLAG** A 1-byte flag field. Possible values for this return flag are:

- A** The user has access to the discrete profile at the requested access level or higher.
- B** The user has access to the generic profile at the requested access level or higher.
- N** The user does not have the requested access to the discrete profile. The user has READ access to the profile.
- G** The user does not have the requested access to the generic profile. The user has READ access to the profile.

**PROFILE NAME**

A variable-length profile name

If the application requests to return those profiles to which the user has at least READ access, the profile flag for the returned profiles has the value A or B. In this situation the values N and G are not used. If the requested access is one of the other values, all four values for the profile flag are used.

**TSQUEUE usage for profiles**

If the application specifies the value **T** or **B** in the **API\_RSRL\_RETFLAG** field, the profiles are returned in the TSQUEUE specified in the **API\_RSRL\_TSQUEUE** field.

The TSQUEUE must be a non-recoverable MAIN-storage TSQUEUE. Recoverable TSQUEUES are not supported. Using an AUXILIARY storage queue is discouraged because it might involve I/O to auxiliary storage, impacting the application response time. The API program clears the entire TSQUEUE before writing any records. If you want the associated system resources freed, the calling program must delete the TSQUEUE after processing the data.

Each of the requested profiles is written in a separate record. The layout of the record is identical to that of the **API\_RSRL\_PROFLST** shown in the preceding list.

**Return and Reason codes**

Some of the return and reason codes returned by this API are specific ones for this function, and some are the ones used by RACF for the IRRPNL00 function.

The following list summarizes the specific return and reason codes. See *z/OS Security Server RACF Macros and Interfaces* for information about the return and reason codes for the IRRPNL00 function.

**RC=00** No error occurred. The requested profiles are provided in the specified areas.

**RC=04** See IRRPNL00 return and reason codes.

**RC=08** See IRRPNL00 return and reason codes.

**RC=0C**

**REAS=00** The internal work area used to process the authorized profiles is too small. Only profile list requests that need less than 128Kbyte of data can be processed.

REAS=04 Profile return in a TSQUEUE was requested, but no TSQUEUE name is given.

REAS=08 The terminal user does not have access to the specified TSQUEUE name.

REAS=0C The COMMAREA provided is not large enough to contain all authorized profiles.

**RC=14 to RC=24**

See IRRPNL00 return and reason codes.

**RC=32** RACROUTE REQUEST=VERIFY for the specified user failed. See message CQT030 for the RACF return and reason codes.

---

## Access check and DATA retrieval (RSRD)

The RSRD function can be used to retrieve the USERDATA associated with the access specification for a user ID.

The user access can be granted through an individual permit, a group connection, access to ID(\*), or through the UACC. If matching entries are defined in the USERDATA fields in the appropriate profile, the associated DATA is returned to the caller of the API function.

### Retrieval of USERDATA

The USERDATA is retrieved for the alphabetically highest, best fitting ACL entry from the most specific or alphabetically highest member or grouping class profile that was used during the RACLIST processing of the resource profiles for the class.

In sequence, the following items are checked to determine where to retrieve the USERDATA:

1. A direct permit to the user ID.
2. An indirect permit through a group that the user ID is connected to.
3. If access is granted through multiple groups, the alphabetically highest group.
4. Access granted through ID(\*)
5. Access granted through the UACC.
6. If access is granted through a member-class profile and one or more grouping-class profiles, the member-class profile.
7. If access is granted through multiple grouping-class profiles, the alphabetically highest grouping-class profile.

This strategy is probably best illustrated by using an example. The purpose of this first example is mainly to show the profile from which the USERDATA is retrieved. Subsequent examples focus on determining which USERDATA entry is used to locate the requested DATA.

As an example, assume that the following profiles are defined in the resource classes \$GROUP and \$MEMBER:

```
$GROUP GRPA Addmem(MEMA,MEMB) READ(USER1,USER2,GROUP1)
$GROUP GRPB Addmem(MEMA,MEMC) READ(USER1,USER3,GROUP2)
$MEMBER MEMA READ(USER4)
USER1 CONNECT(GROUP1,GROUP2)
USER2 CONNECT(GROUP1)
USER3 CONNECT(GROUP2)
USER4 CONNECT(GROUP4)
USER5 CONNECT(GROUP1,GROUP2)
```

When the RSRD API is called for resource MEMA and user USER2, the API function checks that USER2 has access. Because the user has a direct permit, the USERDATA entry for USER2 is retrieved. Also, USER2 has access to only one profile. Therefore, the USERDATA is retrieved from that single profile \$GROUP GRPA.

When the API is called for MEMA and user USER1, the API function detects that USER1 has access through profiles \$GROUP GRPA and \$GROUP GRPB. The relevant ACL entry for both profiles is the same (USER1). In that case, the highest alphabetical profile is used. Therefore, USERDATA is retrieved from profile \$GROUP GRPB.

For USER5 accessing MEMA, access is granted through groups GROUP1 and GROUP2. Two different grouping-class profiles are involved and access is granted through two different GROUPs. When different GROUPs grant access, the alphabetically highest group (GROUP2) is used.

For the access of USER4 to MEMA, only one profile is relevant: \$MEMBER MEMA.

The following table shows the profiles that are used to retrieve the applicable USERDATA.

*Table 8. Profiles used to retrieve USERDATA*

User	MEMA	MEMB	MEMC
USER1	GRPB/USER1	GRPA/USER1	GRPB/USER1
USER2	GRPA/USER2	GRPA/USER2	None
USER3	GRPB/USER3	None	GRPB/USER3
USER4	MEMA/USER4	None	None
USER5	GRPB/GROUP2	GRPA/GROUP1	GRPB/GROUP2

It is possible to specify in the API parameter list that access is to be checked at a certain level. This access level is used for the access verification process, but is not used to determine the most applicable USERDATA entry. So, if USER5 has UPDATE access through GROUP1 and READ access through GROUP2, the USERDATA is always retrieved for GROUP2, even though the actual access is granted through GROUP1.

## Definitions of USERDATA entries

USERDATA entries consist of two parts: the USRNM and the USRDATA. The USRNM is used as an index to locate the associated USRDATA.

In the remainder of this information, the term USRDATA is used to refer to the data-value field in the RACF database, and the term USERDATA is used to refer to the combination of the two fields.

The USERDATA can be entered into the RACF profiles, for example, by using the CKGRACF function of zSecure Admin. The correct implementation for the RSRD function requires that each ACL entry is mirrored by a USERDATA entry. Additional USERDATA entries can be defined for the UACC and access granted through ID(\*). Entries defined for the UACC are represented by a USRNM of -UACC-, and entries for access granted through ID(\*) are represented by a USRNM of -STAR-. ACL entries for users or groups that have ACCESS=NONE must *not* be represented in the USERDATA entries.



The USRDATA can contain any character that is supported by the tool that is used to add these values to the profile. The zSecure CICS Toolkit RSRD function does not impose any restrictions on the characters used. Embedded blanks are allowed. The maximum length of the USRDATA returned by RSRD to the application is 64 characters.

**Note:** As stated earlier, the correct implementation for the RSRD function *requires* that each ACL entry with access other than NONE is mirrored by a USERDATA entry. The RSRD function detects inconsistencies where information is missing. However, some types of inconsistencies are not detected, and might lead to unexpected results. For example, if a USERDATA entry is missing on the member-class profile that granted access, but is present on one of the applicable grouping-class profiles, the absence of the correct USERDATA entry might be undetected. Another example might be that DATA for ID(\*) is returned when data for a GROUP was expected.

## Additional considerations

When using the RSRD function, also consider the information in this topic.

Although the resource name can be up to 246 characters, the profiles used to define access to the resource have a maximum length of 40 characters. If longer profile names are used, they are truncated at 40 positions. If profile truncation occurs, the value returned for the USERDATA is undefined.

If grouping-class profiles are used, the RSRD function process must determine the name of that grouping-class profile. However, during the SETROPTS RACLIST processing, the name of the grouping-class profile is dropped and it is no longer available in memory. As a substitute for the grouping-class profile name, the RSRD function uses the APPLDATA of the profile. The APPLDATA is retained in the in-memory profiles built during SETROPTS RACLIST processing. To provide the profile name to the RSRD function, the name of the profile must be specified in the APPLDATA of the profile itself. Although not needed for member-class profiles, adding the profile name to the APPLDATA of member-class profiles is also supported. Examples of such definitions are:

```
RDEFINE $GROUP GRP1 ADDMEM(RES1,RES2) APPLDATA('GRP1')
RDEFINE $GROUP GRP2 ADDMEM(RES3,RES4) APPLDATA('GRP2')
RDEFINE $MEMBER RES5 APPLDATA('RES5')
```

As a result of these definitions, the following in-memory logical profiles are built during RACLIST processing:

```
$MEMBER RES1 APPLDATA(GRP1)
$MEMBER RES2 APPLDATA(GRP1)
$MEMBER RES3 APPLDATA(GRP2)
$MEMBER RES4 APPLDATA(GRP2)
$MEMBER RES5 APPLDATA(RES5)
```

Using this approach, the RSRD function can use the APPLDATA to locate the correct grouping-class profile. For example, the data for resource RES1 can be retrieved from profile GRP1 in class \$GROUP.

Using the APPLDATA is insufficient if the same resource is defined as a member in multiple grouping-class profiles. During RACLIST processing, profiles are combined into in-memory (logical) profiles. However, only one value for the APPLDATA is retained. Using the example profiles introduced in “Retrieval of USERDATA” on page 97, the following in-memory profiles are built:

```

$MEMBER MEMA APPLDATA(MEMA) READ(USER1,USER2,USER3,USER4,GROUP1,GROUP2)
$MEMBER MEMB APPLDATA(GRPA) READ(USER1,USER2,GROUP1)
$MEMBER MEMC APPLDATA(GRPB) READ(USER1,USER3,GROUP2)

```

During the RACLIST processing, only one APPLDATA value is retained. For example, the in-memory logical profile for MEMA only has the APPLDATA value for MEMA, and the information from GRPA and GRPB is no longer available. This situation can be remedied by implementing the RACLIST exits. See “Use of RACLIST exits.”

## Use of RACLIST exits

To accommodate members that are defined in multiple grouping-class profiles, RACLIST exits can be exploited.

(For an example of accommodating members that are defined in multiple grouping-class profiles, see “Retrieval of USERDATA” on page 97.)

zSecure CICS Toolkit provides two RACLIST exits for this purpose.

### ICHRLX01

The RACLIST preprocessing and postprocessing exit. It is used to determine whether special processing is needed for the resource class being RACLISTed. A resource class is eligible for special RACLIST processing if the class name is included in the APPLDATA of profile ICHRLX02.PROCESS.CLASS in the XFACILIT resource class. An example of the profile is:

```
RDEFINE XFACILIT ICHRLX02.PROCESS.CLASS APPLDATA('$MEMBER')
```

### ICHRLX02

The RACLIST selection or processing exit. In this implementation, the APPLDATA of the in-memory profile is updated to contain a list of all profiles that contributed to the resulting in-memory logical profile. Profiles in the RACF database itself are unaffected by this exit.

If both exits are active at the time of the RACLIST or the SETROPTS REFRESH RACLIST command, the following in-memory logical profiles are created:

```

$MEMBER MEMA APPLDATA(MEMA GRPB GRPA) READ(USER1,USER2,USER3,USER4,GROUP1,GROUP2)
$MEMBER MEMB APPLDATA(GRPA) READ(USER1,USER2,GROUP1)
$MEMBER MEMC APPLDATA(GRPB) READ(USER1,USER3,GROUP2)

```

Notice that the only difference is the value of the in-memory APPLDATA. It now has a list of *all* the contributing profiles.

If these RACLIST exits are used, it is no longer necessary to specify any APPLDATA value on the profiles in the RACF database. The exit will use the profile names directly to build the in-memory list that the RSRD function uses.

## Restrictions

When using the provided RACLIST exits to support definition of the same member as part of multiple grouping or member class profiles, the following limitations apply:

- The total length of the profile names that contribute to the in-memory profile for a resource cannot exceed 255 bytes. If more characters are needed, some profile names are truncated.

- A maximum of 16 grouping and member class profiles can be used to define the effective protection of a single resource. If more profiles are used, some profile names are ignored.
- Each profile name can have a maximum length of 40 characters. If profile names have more characters, the profile name is truncated.

## API specification

This section describes the COMMAREA as required for the RSRD API function.

### FUNCTION

Check whether a user has access to a resource, and retrieve the associated value of the USERDATA.

### AUTHORITY

None required.

### COMMAREA

Minimum size 412 bytes.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTSAMP library.

API_FUNC	DC	CL4'RSRD'	Function code for access check
API_RC	DC	XL01'00'	Return code
*			On input, the following values are supported
*			"S" Suppress ICH408I messages for violations
*			"N" Suppress ICH408I messages and SMF Audit
API_MSG	DC	CL79' '	Message area
*			
API_RSRD_USERID	DC	CL8	Userid for which the access must be
*			checked and for which the associated
*			USRDATA must be retrieved.
*			If blank, the ACEE of the userid signed
*			on at the terminal is used. On return,
*			this field contains the id that was used
*			to retrieve the associated USRDATA.
*			
API_RSRD_ACC	DC	CL1	The required access level. Valid values are
*			'R' (read), 'U' (update), 'C' (control) or
*			'A' (alter). Read is the default.
*			On return, this field contains the
*			return code from the function.
*			
API_RSRD_CLASS	DC	CL8	This field specifies the resource class
*			for the resource. This class must be
*			RACLISTed through SETROPTS RACLIST. On
*			return this field contains the resource
*			class of the matching profile.
*			
API_RSRD_NAMEL	DS	XL1	Length of the resource name. The specified
*			length can be greater than the actual
*			resource name, provided it is padded with
*			blanks up to the specified length.
*			
API_RSRD_NAME	DS	CL246	Name of the resource. On return, this field
*			contains the name of the profile used.
*			
API_RSRD_DATA	DS	CL64	On return, this field contains the USRDATA
*			associated with the userid.

Most fields in the COMMAREA are used for input and output. On output, the API\_RSRD\_CLASS and API\_RSRD\_NAME reflect the profile used for the retrieval of the API\_RSRD\_DATA. Also, the API\_RSRD\_USERID contains the value of the USRNM index in

the USERDATA used to retrieve the API\_RSRD\_DATA. Because most fields are updated as part of the process, you must reinitialize the entire COMMAREA on each call.

Auditing the access verification through SMF records is done based on the AUDIT settings of the RACF profiles or through RACF SETROPTS LOGOPTIONS. If you want to suppress ICH408I resource access violation messages on the system console and in the CICS log, you can specify the value "S" in the API\_RC field. Specifying this value results in suppression of possible access violation messages, while still creating SMF records about these violations. It is also possible to specify the value "N" in the API\_RC field. In that case, both ICH408I resource access violation messages and SMF auditing are suppressed. If you do not specify an "S" or an "N" character, messages and SMF records can be suppressed according to the settings in CQTPCNTL.

### Return codes

The return codes are returned in the field API\_RSRD\_ACC.

In the following list, the return codes are shown as 1-byte hexadecimal fields with the following meanings:

- X'00' Access to the resource is allowed. The API COMMAREA is updated with the information retrieved for the associated USRDATA.
- X'04' The resource is not defined to RACF.
- X'08' The user is not authorized to use the resource at the specified access level.
- X'0C' USRDATA specification error. The user has access to the specified resource, but no associated USRDATA value could be found.
- X'10' USERID specification error. The user ID specified in the API COMMAREA could not be used. Check the system log for the corresponding ICH408I message for additional information about why setting up the security environment for this user ID failed.
- X'14' Profile consistency error. The RSRD function uses the information in the in-storage APPLDATA field to determine the name of grouping class profiles that contribute to the RACLISTed in-memory profile. The APPLDATA information is incorrect and contains profile names that do not exist.
- X'18' CLASS specification error. The resource class specified in the API COMMAREA could not be found in the system.

## Installation considerations

When using the RACLIST exits, these exits must be active at the time that an initial RACLIST or a SETROPTS RACLIST REFRESH is done.

To make these exits active at the correct time, either install the provided exits in a SYSTEM library using the RACF names ICHRLXnn, or use the zSecure Exit Activator function (program C2XACTV), followed by a SETROPTS RACLIST REFRESH command. The C2XACTV program is provided as part of the zSecure Admin, zSecure Audit, and zSecure Alert products.

If the same resource class is used in multiple LPARs and RACF sysplex communication is enabled, the RACLIST exits must be installed and active on all systems in the sysplex.

The ICHRLX01 exit uses the word at offset 48 (X'30') in the exit parameter list to communicate to the ICHRLX02 exit if the current resource class must be processed. If you have your own RACLIST exits in place, they cannot use that same communication area.

Using the provided exits as regular RACF exits is only supported if you currently do not have any RACLIST exits active. If you have, your exits cannot use the communication area mentioned before and you must modify your exits to call the exits provided with zSecure CICS Toolkit. Also, your ICHRLX01 exit cannot specify non-default RACLIST merge rules for the UACC, UADIT, GLOBALAUDIT, INSTDATA, and APPLDATA fields.

To install as a regular RACF exit, follow these steps:

1. Rename the supplied exit routines from CQTRLX01 and CQTRLX02 into ICHRLX01 and ICHRLX02. The C2XRLZxx routines are not used.
2. Copy exit routines ICHRLX01 and ICHRLX02 to an LPALIST data set; for example, zSecure CICS Toolkit SCQTLPA.
3. IPL your system with CLPA.
4. Ensure that the required resource classes are set up as RACLISTed.

To install the exits using the zSecure Exit Activator program, follow these steps:

1. Run a job similar to the following one, using a concatenation of the zSecure Admin and zSecure CICS Toolkit load libraries as steplib.

```
//C2XACTV EXEC PGM=C2XACTV
//STEPLIB DD DISP=SHR,DSN=<ZSECURE.SCQTLOAD>
//          DD DISP=SHR,DSN=<ZSECURE.SCKRLOAD>
//SYSTSPRT DD SYSOUT=*
//C2XPRINT DD SYSOUT=*
//C2XIN DD *
DYNEXIT DEACTIVATE ICHRLX01 DIRECT
DYNEXIT RECOVER ICHRLX01 DIRECT
DYNEXIT ACTIVATE ICHRLX01 DIRECT
DYNEXIT DEACTIVATE ICHRLX02 DIRECT
DYNEXIT RECOVER ICHRLX02 DIRECT
DYNEXIT ACTIVATE ICHRLX02 DIRECT
```

The user ID running this job must have UPDATE access to the following profiles:

```
XFACILIT C2X.ICHLRX01
XFACILIT C2X.ICHLRX02
```

2. Issue a SETROPTS RACLIST or SETROPTS RACLIST REFRESH for the required resource classes.

The provided RACLIST exits perform no processing unless a profile has been defined in the XFACILIT resource class. The APPLDATA of the profile specifies for which resources classes the exit must create in-memory APPLDATA values for use by the zSecure CICS Toolkit RSRD function. An example of the profile follows:

```
RDEFINE XFACILIT ICHRLX02.PROCESS.CLASS APPLDATA('$MEMBER')
```

---

## **ADDGROUP / ALTGROUP / DELGROUP function (add, alter, or delete a group)**

Use the ADDGROUP, ALTGROUP, and DELGROUP function to add a new group to the system or to alter or delete an existing group.

### **AUTHORITY**

The user must have access to the zSecure CICS Toolkit command

(**TOOLKIT.ADGR** / **TOOLKIT.ALGR** / **TOOLKIT.DELG** / **TOOLKIT.LGRP**, depending in the command being performed) and the group (**ADGR.grpname** / **ALGR.grpname** / **DELG.grpname** / **LGRP.grpname**).

#### COMMAREA

Minimum size for this function is 370 bytes in order to support Universal groups.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'ADGR'	Function code
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
*			
API_AGRP_RC	DC	XL1	Return code from requested command
*			If non-zero the command failed.
*			API_MSG will give the reason for the failure.
*			
API_AGRP_CODE1	DC	CL4"xxxx"	'ADGR' for ADDGROUP
*			'ALGR' for ALTGROUP
*			'DELG' for DELGROUP
*			'LGRP' for LISTGROUP
*API_AGRP_GROUP	DC	CL8	Group name.
*			
API_AGRP_OWNER	DC	CL8	Owner name
*			
API_AGRP_SUPGRUP	DC	CL8	Superior Group name
*			
API_AGRP_TERMUAC	DC	CL1	Terminal UACC ('Y' or 'N')
*			
API_AGRP_INSTDATA	DC	CL255	Installation data
*			
API_AGRP_UNIVERS	DC	CL1	Universal Group ('Y' or 'N')
*			

To retrieve information about the group, enter LGRP in the **API\_AGRP\_CODE1** field. When altering data, blank fields are ignored and are not updated. To DELETE the installation data field, set the first byte to binary zeros (X'00').

## ADDUSER function (add user profile)

Use the ADDUSER function to add a new user profile to the system.

#### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (**TOOLKIT.ADUS**) and the default group of the user being added (**ADUS.dfltgrp**).

#### COMMAREA

Minimum size 408 bytes.

If your application reserves space for the no longer supported automatic create of a CICS segment, the required size would be 495 bytes.

If you need to specify a password phrase, the required size is 595 bytes.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'ADUS'	Function code for ADDUSER
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
*			

API_ADUS_RC	DC	XL1	Return code from ADDUSER. If non-zero the command failed. API_MSG will give the reason for the failure.
*			
*			
*			
API_ADUS_USERID	DC	CL8	Userid being added.
API_ADUS_PGMNAME	DC	CL20	Users name.
*			
API_ADUS_DFLTGRP	DC	CL8	The users default group.
*			
API_ADUS_AUTHRTY	DC	CL1	Authority in the default group. Must be 'U' (use) or 'C' (create).
*			
*			
API_ADUS_SMTWTF5	DC	CL7	The days of the week the user can logon. Specify 'Y' for each day the user may logon and 'N' for the days they may not.
*			
*			
*			
API_ADUS_FROM	DC	CL4	The time of day the user can logon from (24 hour clock).
*			
*			
API_ADUS_TILL	DC	CL4	The time of day the user can logon till (24 hour clock).
*			
*			
API_ADUS_INSTDATA	DC	CL255	Installation data field.
*			
API_ADUS_PASSWORD	DC	CL8	Initial password for the user. If it is omitted, the password defaults to the users default group.
*			
*			
API_ADUS_OWNER	DC	CL8	The owner of the profile.
*			
*			
*			
*			
*			
*			
API_ADUS_OPIDENT	DC	CL3	Retained for compatibility
API_ADUS_OPPTY	DC	CL3	Retained for compatibility
API_ADUS_TIMEOUT	DC	CL3	Retained for compatibility
API_ADUS_XRFSOFF	DC	CL7	Retained for compatibility
API_ADUS_OPCLASS	DC	CL71	Retained for compatibility
*			
API_ADUS_PHRASE	DC	CL100	The password phrase of the userid.
*			

If you do not specify a value when creating the user profile, the initial PASSWORD for the user is set to the value of the DEFAULT GROUP.

The first time users log on, they are required to enter a new password.

---

## ALTUSER function (changing a profile)

Use the ALTUSER function to change the profile for a specific user.

### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.AUSR) and the users default group (AUSR.dfltgrp).

For several fields, the user must also have system special, or access to TOOLKIT.SPEC. These fields are flagged with an asterisk (\*).

### COMMAREA

Minimum size 487 bytes.

If you need to specify a password phrase, the required size is 587 bytes.

In your application, use the CQTMPIA or CQTMPIB mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'AUSR'	Function code for ALTUSER
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
*			
API_ALUS_RC	DC	XL1	Return code from ALTUSER
*			If non-zero the command failed.
*			API_MSG will give the reason for
*			the failure.
*			
API_ALUS_USERID	DC	CL8	The userid to be altered.
API_ALUS_PASSWRD	DC	CL8	Password
API_ALUS_RESUME	DC	CL1	Resume the userid (Y or N)
API_ALUS_PGMNAME	DC	CL20	Name
API_ALUS_INSTDATA	DC	CL255	Installation data field.
API_ALUS_DFLTGRP	DC	CL8	Default group.
API_ALUS_REVOKED	DC	CL5	Revoke Date(YYDD).
API_ALUS_RESUMED	DC	CL5	Resume Date(YYDD).
API_ALUS_AUTHOR	DS	CL8	* OWNER
API_ALUS_GRPACC	DC	CL1	* Group access.
API_ALUS_ADSP	DC	CL1	* ADSP.
API_ALUS_SPEC	DC	CL1	* Special
API_ALUS_OPER	DC	CL1	* Operations
API_ALUS_AUDITOR	DC	CL1	* Auditor
API_ALUS_RESTR	DC	CL1	* UACC and similar not used.
API_ALUS_PROTECT	DC	CL1	* Password cannot be used.
API_ALUS_UAUDIT	DC	CL1	* Audit all RACHECK's/RACDEF's.
API_ALUS_LOGDAY	DC	CL7	* Days user can logon.
API_ALUS_LOGFROM	DC	CL4	* Starting time for logon.
API_ALUS_LOGTILL	DC	CL4	* Latest time for logon.
API_ALUS_MODEL	DC	CL44	* Dataset profile model.
API_ALUS_CLAETH	DC	CL8	* Give class authority.
*API-ALUS-AUTH			Name used in COBOL copybook
API_ALUS_NOCLAETH	DC	CL8	* Remove class authority.
*API-ALUS-NAUTH			Name used in COBOL copybook
API_ALUS_PASSEXP	DC	CL1	* New password is expired (Y or N).
API_ALUS_PHRASE	DC	CL100	The password phrase of the userid.
*			

The fields in the commarea must be initialized to BINARY ZEROES. Only fields that are to be altered need to contain data. Refer to Chapter 5, "The zSecure CICS Toolkit command interface," on page 35 for a description of the fields and the restrictions on who can update which fields. After linking to CQTPAPI0, the users profile will be updated with the information contained in any field that was not binary zeros.

To specify that the password phrase must be removed, specify a value of 100 blanks. Any other value results in the password phrase being changed into the specified value, or be retained at its current value.

You can use the special date zeros (c'00000' = x'F0F0F0F0F0') to remove the Revoke/Resume date. Using this special value, you might implement a function like the NOREVOKE and NORESUME keywords of the z/OS 1.7 RACF **ALTUSER** command.

## **ALTUSER (CICS SEGMENT) function (alter CICS segment)**

Use the ALTUSER (CICS SEGMENT) function to change the CICS segment for a specific user.



## AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.AUSR) and the default group of the user (AUSR.dfltgrp). In addition, for managing the CICS segment, the user must have access to TOOLKIT.ACIC. In version 1.4 of Consul zToolkit, this requirement was only enforced if the TOOLKIT.ACIC profile has been defined or is covered by a generic profile. In version 1.8.1 and higher of zSecure CICS Toolkit, access to resource TOOLKIT.ACIC is required.

## COMMAREA

Minimum size is 184 bytes. If your application requires access to the TSLKEY and RSLKEY, the required minimum size is 316 bytes.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'ACIC'	Function code for CICS segment.
*			For compatibility reasons, ACSG is also accepted
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
*			
API_ACSG_RC	DC	XL1	Return code from ALTUSER. If non-zero the command failed.
*			API_MSG will give the reason for the failure.
*			
*			
API_ACSG_CODE	DC	CL4	Specify LIST to retrieve the current specifications for the user. See LISTUSER (TSO/CICS)
*			
*			UPDT to update the CICS segment with the values in the COMMAREA.
*			DELT to delete the CICS segment.
API_ACSG_USERID	DC	CL8	The userid to be listed/updated.
API_ACSG_OPIDENT	DC	CL3	Three character opident
API_ACSG_OPPRTY	DC	CL3	Operator priority (000-255)
API_ACSG_TIMEOUT	DC	CL3	See ALTUSER command
API_ACSG_XRFSOFF	DC	CL7	FORCE or NOFORCE
API_ACSG_OPCLASS	DC	CL71	Operator classes (01-24, separated by a comma, e.g.; 01,03,04)
*			
API_ACSG_TSLKEY	DS	CL66	TSL KEYS (00, 99, or 01-64, separated by a comma, e.g.; 01,03,04)
*			
API_ACSG_RSLKEY	DS	CL66	RSL KEYS (00, 99, or 01-24, separated by a comma, e.g.; 01,03,04)
*			

When the CICS segment is updated, all fields in the CICS segment are replaced. For this reason, valid data must be supplied for all parameters.

**Note:** In the current release, zSecure CICS Toolkit does not verify that each TSLKEY and RSLKEY value only provides space for 22 TSLKEY and RSLKEY values.

## ALTUSER (TSO SEGMENT) function (change TSO segment)

Use the ALTUSER (TSO SEGMENT) function to change the TSO segment for a specific user.

## AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.AUSR) and the default group of the user (AUSR.dfltgrp). In addition, for managing the TSO segment, the user must have access to TOOLKIT.ATSO. In version 1.4 of Consul zToolkit, this requirement was only enforced if the TOOLKIT.ATSO profile has been defined (or is covered

by a generic profile). In version 1.8.1 and higher of zSecure CICS Toolkit, access to resource TOOLKIT.ATSO is required.

#### COMMAREA

Minimum size 191 bytes.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC CL4'ATSO'	Function code for TSO segment.
*		For compatibility reasons, ATSG is also accepted
API_RC	DC XL01'00'	Return code
API_MSG	DC CL79' '	Message area
*		
API_ATSG_RC	DC XL1	Return code from ALTUSER. If non-zero the command failed.
*		API_MSG will give the reason for the failure.
*		
*		
API_ATSG_CODE	DC CL4	Specify LIST to retrieve the current specifications for the user.
*		See LISTUSER (TSO/CICS)
*		UPDT to update the TSO segment
*		with the values in the COMMAREA.
*		DELT to delete the TSO segment.
API_ATSG_USERID	DC CL8	The userid to be listed/updated.
API_ATSG_ACCTNUM	DC CL40	The account number
API_ATSG_DESTID	DC CL8	The destination id.
API_ATSG_HCLASS	DC CL1	The hold class.
API_ATSG_JCLASS	DC CL1	The job class.
API_ATSG_MSGCLASS	DC CL1	The message class.
API_ATSG_SCLASS	DC CL1	The sysout class.
API_ATSG_SECLABL	DC CL8	The security label.
API_ATSG_SIZE	DC CL7	The region size.
API_ATSG_MAXSIZE	DC CL7	The maximum region size.
API_ATSG_PROC	DC CL8	The logon proc.
API_ATSG_UNIT	DC CL8	The allocation device.
API_ATSG_UDATA	DC CL4	The installation data.

When the TSO segment is updated, all fields in the TSO segment are replaced. For this reason, valid data must be supplied for all parameters. Leaving a field empty (blanks or nulls) results in the corresponding field in the TSO segment to be deleted.

## ALTUSER (OMVS SEGMENT) function (change OMVS segment)

Use the ALTUSER (OMVS SEGMENT) function to change the OMVS segment for a specific user.

#### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.AUSR) and the default group of the user (AUSR.*dfltgrp*). In addition, for managing the OMVS segment, the user must have access to TOOLKIT.AOMV.

#### COMMAREA

Minimum size 273 bytes. If your applications access to the MEMLIM and SHMMAX fields, the minimum size is 291 bytes.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'AOMV'	Function code for OMVS segment.
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
*			
API_AOMV_RC	DS	XL01	Return code from command
*			
API_AOMV_CODE	DS	CL04	UPDT = Update segment
*			LIST = Retrieve segment
*			DELT = Delete segment
API_AOMV_USERID	DS	CL08	The userid
API_AOMV_UID	DS	CL10	UID (1-10# Left align, AUTOUID)
API_AOMV_SHARED	DS	CL01	Shared UID N/Y
API_AOMV_MKDIR	DS	CL01	MKDIR N/Y
API_AOMV_HOME	DS	CL60	HOME (1-60 CHARS) mixed case
API_AOMV_PROGRAM	DS	CL60	SHELL PROGRAM (1-60 CHARS)
API_AOMV_ASSIZE	DS	CL10	ASSIZEMAX (1-10 Digits left al)
API_AOMV_CPUTIME	DS	CL10	CPUTIMEMAX (1-10 Idem)
API_AOMV_FILEPROC	DS	CL06	FILEPROCMAX (1-6 Idem)
API_AOMV_MMAPAREA	DS	CL08	MMAPAREA (1-8 Idem)
API_AOMV_PROCUSE	DS	CL05	PROCUSE (1-5 Idem)
API_AOMV_THREADS	DS	CL06	THREADS (1-6 Idem)
API_AOMV_MEMLIM	DS	CL09	MEMLIMIT (1-8 DIGITS + M/G/T/P)
API_AOMV_SHMMAX	DS	CL09	SHMEMMAX (1-8 DIGITS + M/G/T/P)

When the OMVS segment is updated, all fields in the OMVS segment are replaced. For this reason, valid data should be supplied for all parameters. Leaving a field empty (blanks or nulls) results in the corresponding field in the OMVS segment to be deleted.

## ALTUSER (WORKATTR SEGMENT) function (change WORKATTR segment)

Use the ALTUSER (WORKATTR SEGMENT) function to change the WORKATTR segment for a specific user.

### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.AUSR) and the default group of the user (AUSR.dfltgrp). In addition, for managing the OMVS segment, the user must have access to TOOLKIT.AWRK.

### COMMAREA

Minimum size 637 bytes.

In your application, use the CQTMPIA or CQTMPIB mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'AWRK'	Function code for WORKATTR segment.
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
*			
API_AWRK_RC	DS	XL01	Return code from command
*			
API_AWRK_CODE	DS	CL04	UPDT = Update segment
*			LIST = Retrieve segment
*			DELT = Delete segment
API_AWRK_USERID	DS	CL08	The userid
API_AWRK_NAME	DS	CL60	WANAME (1-60 Chars)
API_AWRK_ACCNT	DS	CL60	WAACCNT (1-60 Chars)
API_AWRK_BLDG	DS	CL60	WABLDG (1-60 Chars)
API_AWRK_DEPT	DS	CL60	WADEPT (1-60 Chars)
API_AWRK_ROOM	DS	CL60	WAROOM (1-60 Chars)
API_AWRK_ADDR1	DS	CL60	WAADDR1 (1-60 Chars)

API_AWRK_ADDR2	DS	CL60	WAADDR2 (1-60 Chars)
API_AWRK_ADDR3	DS	CL60	WAADDR3 (1-60 Chars)
API_AWRK_ADDR4	DS	CL60	WAADDR4 (1-60 Chars)

When the WORKATTR segment is updated, all fields in the WORKATTR segment are replaced. For this reason, valid data must be supplied for all parameters. Leaving a field empty (blanks or nulls) results in the corresponding field in the WORKATTR segment to be deleted.

---

## CONNECT function (connect a user or group to a group)

Use the CONNECT function to connect a user or group to a group.

### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.CONN) and the target group (CONN.dfltgrp)

### COMMAREA

Minimum size 112 bytes. If your application requires access to the REVOKE and RESUME dates, the minimum length is 122 bytes.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'CONN'	Function code for CONNECT
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
API_CONN_RC	DC	XL1	Return code from CONNECT. If non-zero the command failed.
*			API_MSG will give the reason for the failure.
*			
*			
API_CONN_USERID	DC	CL8	Userid/group being connected.
*			
API_CONN_GROUP	DC	CL8	Group being connected to.
*			
API_CONN_AUTH	DC	CL1	Connect authority
*			Must be 'U' (use),
*			'C' (create)
*			'N' (connect) or
*			'J' (join).
*			
API_CONN_OWNER	DC	CL8	Owner of the connect profile. It must be a valid userid or group.
*			
*			
API_CONN_SPEC	DC	CL1	Specify 'Y' if the user should have the group-special attribute otherwise specify 'N'.
*			
*			
API_CONN_OPER	DC	CL1	Specify 'Y' if the user should have the group-operations attribute otherwise specify 'N:'.
*			
*			
API_CONN_REVOKE	DC	CL5	The date (YYDDD) the user is to be REVOKED
*			
API_CONN_RESUME	DC	CL5	The date (YYDDD) the user is to be RESUMED

The **API\_CONN\_REVOKE** and **API\_CONN\_RESUME** fields can be used to set or remove the REVOKE and RESUME dates for the connection. If you use the value blanks (x'4040404040'), the revoke and resume dates are left at their current value. If you use the special date zeros (c'00000' = x'F0F0F0F0F0'), the current Revoke/Resume date is removed. Using this last special value, you can implement a function like the NOREVOKE and NORESUME keywords of the z/OS 1.7 RACF **CONNECT** command. If you specify today's date for either the REVOKEDT or the

RESUMEDT, the revoke-status for the user is updated immediately and the other date value is ignored. Both the RESUMEDT and the REVOKEDT are reset.

---

## DELETE DATASET function (delete data set profile)

Use the DELETE DATASET function to delete a data set profile from the system.

### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.DELD) and the high-level-qualifier of the data set profile name (DELD.hlq). If the user does not have access to the DELD.hlq, standard RACF authority checking is used. Refer to the RACF Command Language Reference manual for information about which data set profiles a user is authorized to delete.

### COMMAREA

Minimum size 130 bytes.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'DELD'	Function code for DELDSD
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
API_DELD_RC	DC	XL01'00'	Return code from DELDSD.
*			If non-zero the command
*			failed. API_MSG will give the reason for
*			the failure.
*			
API_DELD_DSNAME	DC	CL44'DS-Profile'	The dataset profile to be deleted.
API_DELD_GENERIC	DC	CL01'Y'	Specify 'Y' if the profile is Generic
*			or 'N' if it is not.
*			

---

## DELETE USERID function (delete user profile)

Use the DELETE USERID function to delete a user ID from the system.

### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.DELU) and the default group of the user ID (DELU.dfltgrp)

### COMMAREA

Minimum size 93 bytes.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'DELU'	Function code for DELUSER
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
API_DELU_RC	DC	XL01'00'	Return code from DELUSER.
*			If non-zero the command
*			failed. API_MSG will give the
*			reason for the failure.
*			
API_DELU_USERID	DC	CL8'USERID'	The userid to be deleted.
*			

The user ID must be REMOVED from all groups, except the default group, and no data set profiles using this user ID as a high-level qualifier must exist, before the DELETE is issued.

zSecure CICS Toolkit checks for group connections but not for data set profiles.

---

## LISTDATASET function (list profile for one or more data sets)

Use the LISTDATASET function to list the profile for a specific data set or data sets.

### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.LDSD)

### COMMAREA

Minimum size 524 bytes to display the data set profile or 2374 if requesting the users or programs.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

**Note:** If you want to perform a SEARCH, initialize all fields according to your search pattern. Use asterisks for all field padding except for the DSETID field, which must be padded with nulls, blanks, or underscores.

API_FUNC	DC CL4	'LDSD'	Function code for LISTDATASET
API_RC	DC XL01	'00'	Return code
API_MSGDC	DC CL79	' '	Message area
*			
API_LDSD_RC	DC XL1		Return code from LISTDATASET.
*			If non-zero the command failed.
*			API_MSG will give the reason for
*			the failure.
*			
API_LDSD_CODE1	DC CL1		Request code.
*			'S' = start search
*			'N' = get next profile
*			'L' = retrieve this profile
*			'U' = retrieve users
*			'P' = retrieve programs
*			
API_LDSD_RESERVED	DC CL46		This field is reserved for the
*			API and must be preserved between
*			calls.
*			
API_LDSD_DTYPE	DC CL1		Profile type (generic or discrete).
*			
API_LDSD_DSETID	DC CL44		The dataset to be retrieved.
*			Only required when the CODE
*			field is L, U or P; otherwise it is
*			used as part of the search criteria.
*			
API_LDSD_AUTHOR	DC CL8		Owner of the profile.
API_LDSD_CREADAT	DC CL5		Creation date.
API_LDSD_LREFDAT	DC CL5		Last reference date.
API_LDSD_LCHGDAT	DC CL5		Last update date.
API_LDSD_ACSALTR	DC CL6		# of alter accesses.
API_LDSD_ACSCNTL	DC CL6		# of control accesses.
API_LDSD_ACSUPDT	DC CL6		# of update accesses.
API_LDSD_ACSREAD	DC CL6		# of read accesses.
API_LDSD_UACC	DC CL7		Universal access to the dataset.
API_LDSD_GRPDST	DC CL1		Group dataset.
API_LDSD_AUDIT	DC CL1		Audit flag.
API_LDSD_GROUPNM	DC CL8		Current connect group.
API_LDSD_DSTYPE	DC CL4		Dataset type.
API_LDSD_LEVEL	DC CL3		Level indicator.
API_LDSD_GAUDIT	DC CL1		Global audit option.
API_LDSD_AUDITQS	DC CL1		Audit success flag.

API_LDSD_AUDITQF	DC CL1	Audit failure flag.
API_LDSD_GAUDQS	DC CL1	Global audit success flag.
API_LDSD_GAUDQF	DC CL1	Global audit failure flag.
API_LDSD_WARNING	DC CL1	Warning mode.
API_LDSD_SECLEVL	DC CL3	Security level.
API_LDSD_NUMCTGY	DC CL4	Number of categories.
API_LDSD_NUMPGMS	DC CL4	Number of programs.
API_LDSD_NUMUSER	DC CL4	Number of users/groups.
API_LDSD_INSTDATA	DC CL255	Installation data field.
	ORG API_LDSD_RESERVED	
API_LDSD_USERPGMS	DC CL???	When the users or programs are requested they will be returned into this area.
*		
*		

When the list of programs is returned, the format of the output is as follows:

Description	Length
Length of program name	4 bytes
Program name	8 bytes
Length of userid	4 bytes
Userid	8 bytes
Length of access field	4 bytes
Access	1 byte
	X'80' Alter access
	X'40' Control access
	X'20' Update access
	X'10' Read access
	X'08' Execute access
	X'01' None

When the list of users is returned, the format of the output is as follows:

Description	Length
Length of Userid	4 bytes
Userid	8 bytes
Length of access field	4 bytes
Access	1 byte
	X'80' Alter access
	X'40' Control access
	X'20' Update access
	X'10' Read access
	X'08' Execute access
	X'01' None
Length of access count	4 bytes
Access count	2 bytes (binary)

In all cases, when the first field is zero (x'00000000'), it indicates the end of the data.

---

## LISTGROUP function (list profile for a group)

Use the LISTGROUP function to list the profile for a specific group or groups.

### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.LGRP) and the group name (LGRP:*grpname*)

### COMMAREA

Minimum size 441 bytes to display the group profile or 2374 if requesting the users of subgroups. To enable Universal group support, the minimum size is 442 bytes.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

**Note:** If you want to perform a SEARCH, remember to initialize all fields according to your search pattern. Use asterisks for all field padding except for the **GROUP** field, which must be padded with nulls, blanks, or underscores.

API_FUNC	DC	CL4'LGRP'	Function code for LISTGROUP
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
*			
API_LGRP_RC	DC	XL1	Return code from LISTGROUP.
*			If non-zero the command failed.
*			API_MSG will give the reason for
*			the failure.
*			
API_LGRP_CODE1	DC	CL1	Request code.
*			'S' = start search
*			'N' = get next profile
*			'L' = retrieve this profile
*			'G' = retrieve subgroups
*			'U' = retrieve users
*			
API_LGRP_RESERVED	DC	CL9	This field is reserved for the
*			API and must be preserved between
*			calls.
*			
API_LGRP_GROUP	DC	CL8	The group to be retrieved.
*			Only required when the CODE field
*			is L, G or U otherwise it is used
*			as part of the search criteria.
*			
API_LGRP_SUPGRP	DC	CL8	This groups superior group.
API_LGRP_OWNER	DC	CL8	Owner of this group.
API_LGRP_DTE	DC	CL5	Date this profile was created.
API_LGRP_UACC	DC	CL7	Authority of a user to the group
*			if the user is not connected to
*			the group.
*			
API_LGRP_TERMACC	DC	CL1	Authority to access a terminal* required.
*			
API_LGRP_SUBGRPS	DC	CL5	Number of subgroups.
*			
API_LGRP_USERS	DC	CL5	Number of users.
*			
API_LGRP_MODEL	DC	CL44	Name of a profile to be used as
*			model for new group-name
*			datasets.
*			
API_LGRP_INSTDATA	DC	CL255	Installation data field.
*			
API_LGRP_UNIVERS	DC	CL1	Universal Group ('Y' or 'N')
*			
		ORG API_LGRP_RESERVED	
API_LGRP_USERSUBG	DC	CL????	When the users or subgroups are
*			requested they will be returned
*			into this area.

When the list of users or subgroups is returned, the format of the output is as follows:

Description	Length
Length of member	4 bytes
User/Subgroup name	8 bytes

In all cases, when the first length field is zero (x'00000000'), this indicates the end of the data.



---

## LISTUSER function (list profile for a user ID)

Use the LISTUSER function to list the profile for a specific user or user IDs.

### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.LUSR) and the users default group (LUSR.*dfltgrp*).

### COMMAREA

Minimum size 544 bytes to display the group profile or 2374 if requesting the groups or categories.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

**Note:** If you want to perform a SEARCH, you need to initialize all fields according to your search pattern. Use asterisks for all field padding except for the **USERID** field, which must be padded with nulls, blanks, or underscores.

API_FUNC	DC	CL4'LUSR'	Function code for LISTUSER.
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
*			
API_LUSR_RC	DC	XL1	Return code from LISTUSER.
*			If non-zero the command failed.
*			API_MSG will give the reason for
*			the failure.
*			
API_LUSR_CODE1	DC	CL1	Request code.
*			'S' = start search
*			'N' = get next profile
*			'L' = retrieve this profile
*			'G' = retrieve groups
*			'A' = retrieve groups and authority to groups
*			'C' = retrieve categories
*			
API_LUSR_RESERVED	DC	CL9	This field is reserved for the
*			API and must be preserved between
*			calls.
API_LUSR_USERID	DC	CL8	The userid to be retrieved.
*			Only required when the CODE field
*			is L, G or C otherwise it is used
*			as part of the search criteria.
*			
API_LUSR_PGMNAME	DC	CL20	Users name.
API_LUSR_AUTHOR	DC	CL8	Owner of the profile.
API_LUSR_PASSWRD	DC	CL8	Password field (will contain ?)
API_LUSR_AUTHDTE	DC	CL5	Creation date.
API_LUSR_DFLTGRP	DC	CL8	Default group.
API_LUSR_AUTHRTY	DC	CL7	Authority.
API_LUSR_UACC	DC	CL7	Universal access.
API_LUSR_CLASCNT	DC	CL5	Number of classes.
API_LUSR_ADSP	DC	CL1	ADSP.
API_LUSR_SPEC	DC	CL1	Special.
API_LUSR_OPER	DC	CL1	Operations.
API_LUSR_REVOKE	DC	CL1	Revoke.
API_LUSR_GRPACC	DC	CL1	GRPACC.
API_LUSR_AUDITOR	DC	CL1	Auditor.
API_LUSR_PROTECT	DC	CL1	Password cannot be used.
API_LUSR_RESTR	DC	CL1	UACC and similar not used.
API_LUSR_UAUDIT	DC	CL1	Audit all RACHECK's/RACDEF's.
API_LUSR_REVOKEC	DC	CL2	# unsuccessful pwd attempts.
API_LUSR_REVOKED	DC	CL5	Date user will be revoked.
API_LUSR_SECL	DC	CL2	Security level. This is a binary field that
*			represents the security level. Eg.:

```

* X'00FE' would be a security level of 254.
API_LUSR_RESUMED DC CL5 Date user will be resumed.
API_LUSR_LASTACC DC CL14 Last access date and time.
API_LUSR_PASSDTE DC CL5 Date password last changed.
API_LUSR_PASSINT DC CL3 Password interval.
API_LUSR_PWDGEN DC CL3 Current password generation #.
API_LUSR_PWDCNT DC CL3 Number of old passwords.
API_LUSR_NUMCTGY DC CL4 Number of categories.
API_LUSR_NUMGRP DC CL4 Number of groups.
API_LUSR_LOGDAY DC CL7 Days user can logon.
API_LUSR_LOGFROM DC CL4 Starting time for logon.
API_LUSR_LOGTILL DC CL4 Latest time for logon.
API_LUSR_MODEL DC CL44 Dataset profile model.
API_LUSR_INSTDATA DC CL255 Installation data field.
ORG API_LUSR_RESERVED
API_LUSR_GRPCTGY DC CL???? When the groups or categories are
* requested they will be returned
* into this area.

```

When the list of groups is returned, the format of the output is as follows:

Description	Length
Length of group name	4 bytes
Group name	8 bytes

When the list of groups and authority is returned, the format of the output is as follows:

Description	Length
Combined length of the following 'group length/names'	4 bytes
Length of group name	4 bytes
Group name	8 bytes

For each of the following fields, there is a one-to-one relationship to the groups. If the user was connected to two groups, there are two ADSP flags, two SPECIAL flags, two OPERATIONS flags, two REVOKE flags, two GRPACC flags, and two GROUP AUDITOR flags. If bit 0 in the flag is turned on (X'80'), the user has that attribute in the group.

Combined length of the ADSP lengths/flags	4 bytes
Length of the ADSP flag	4 bytes
ADSP flag	1 byte

Combined length of the SPECIAL lengths/flags	4 bytes
Length of SPECIAL flag	4 bytes
SPECIAL flag	1 bytes

Combined length of the OPERATIONS lengths/flags	4 bytes
Length of OPERATIONS flag	4 bytes
OPERATIONS flag	1 byte

Combined length of the REVOKE lengths/flags	4 bytes
Length of the REVOKE flag	4 bytes
REVOKE REVOKE flag	1 byte

Combined length of the GRPACC lengths/flags	4 bytes
Length of GRPACC flag	4 bytes
GRPACC flag	1 byte

Combined length of the GROUP AUDITOR lengths/flags	4 bytes
Length of GROUP AUDITOR flag	4 bytes
GROUP AUDITOR flag	1 byte

When the list of categories is returned, the format of the output is as follows:

Description	Length
Length of category	4 bytes
Category number	2 bytes (binary)

In all cases, when the first length field is zero (x'00000000'), it indicates the end of the data.

---

## PASSWORD function (change password)

Use the PASSWORD function to change the password of a user.

### AUTHORITY

NONE, unless you are specifying a value of 255 for the interval value (which corresponds to NOINTERVAL), or changing the INTERVAL value for a user ID other than your own. It requires you to have SPECIAL, or to have access to TOOLKIT.SPEC or PSWD.*dfltgrp* (*dfltgrp* is the default group of the user ID that is being altered). If you are using the **PASSWORD** command on a user ID different from your own, you might only change the INTERVAL value. To change another user's password, use the **ALTUSER** command.

### COMMAREA

Minimum size 112 bytes.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'PSWD'	Function code for PASSWORD.
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
*			
API_PSWD_RC	DC	XL01'00'	Return code from PASSWORD.
*			If non-zero the command failed.
*			API MSG will give the reason for the failure.
*			
API_PSWD_USERID	DC	CL08'USERID '	The userid being altered.
*			
API_PSWD_PASSWORD	DC	CL08'PASSWORD'	The password for this userid.
*			
API_PSWD_NEWPASS	DC	CL08'NEWPSWD'	The new password for this userid.
API_PSWD_PASSINT	DC	CL03'060'	The new password interval for this userid.

PASSWORD does not perform a signon for the user. It only verifies that the password entered for this user ID is correct and then changes the password to the new specification and changes the password interval.

Multiple failures to change the password of a user might result in the user ID being revoked, depending on your system parameters.

The password interval of the user might also be changed. The new interval might be 001 - 254, but might not exceed the global maximum specified. If it does, it can be set to the maximum allowed. If the value specified for the interval is invalid, the parameter is ignored. When only changing the password interval, it is not necessary to provide the password. However, if the interval is invalid and ignored,

zSecure CICS Toolkit treats the operation as if it is a request to change passwords and it checks for a password and new password. If a new password has not been supplied, the error message returned reflect it as the error, rather than the password interval being incorrect.

This command is only available through the API.

---

## PERMIT function (grant or remove access)

Use the PERMIT function to grant access to or remove access from a CICS resource. The resource must be in one of the resource classes defined in the SIT for this run of CICS.

### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.PEMT), the default group of the user ID or group that is being granted access (PEMT.*dfltgrp*), and must also have access to the resource. If the access is granted to a group, the resource used is PEMT.*group*. After the PERMIT has been completed, the resource classes must be refreshed in order to have immediate effect. Use the RACF **SETROPTS REFRESH** command to accomplish it.

### COMMAREA

Minimum size 116 bytes.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC CL4'PEMT'	Function code for PERMIT.
API_RC	DC XL01'00'	Return code
API_MSG	DC CL79' '	Message area
*		
API_PEMT_RC	DC XL01'00'	Return code from PERMIT
*		If non-zero the command failed
*		API_MSG will give the reason for the failure
*		
API_PEMT_USERID	DC CL08'USERID'	The userid or group.
*		
API_PEMT_RSRC	DC CL13'CEMT '	The name of the CICS resource.
*		
API_PEMT_CLASS	DC CL08'TCICSTRN'	
*		The resource classname.
*		If blank the value of the XTRAN parameter
*		specified in the SIT is used.
*		
API_PEMT_DELT	DC CL01'Y'	Specify 'Y' in this field to remove
*		a person from the access list for
*		this resource. The user or group
*		will no longer have access to the resource.
*		
API_PEMT_ACC	DC CL01'R'	Access allowed to the resource
*		Specify 'R' for read,
*		'N' for none,
*		'U' for update,
*		'A' for alter or
*		'C' for control.
*		Read is the default

---

## PERMITX function (grant or remove access - any resource)

Use the PERMITX function to grant access to or remove access from any resource. It might also be used to grant access to DATASET profiles.

#### AUTHORITY:

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.PEMT. The default group of the user ID or group that is being granted access (PEMT.*dfltgrp*), and must also have access to the resource. If the access is granted to a group, the resource used is PEMT.*group*. The user must also be authorized to the PEMX.*classname* profile. After the PERMIT has been completed, the resource classes must be refreshed in order to have immediate effect. Use the RACF **SETROPTS REFRESH** command to accomplish it.

In order for the PERMITX function to be available, it must have been enabled in CQTPCNTL by specifying PENTALL=Y.

#### COMMAREA

Minimum size 349 bytes.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'PEMX'	Function code for PERMITX
API_RC	DC	XL01'00	Return code
API_MSG	DC	CL79' '	Message area
*			
API_PEMX_RC	DC	XL01'00'	Return code from PERMIT
*			If non-zero the command failed
*			API_MSG will give the reason for the failure.
*			
API_PEMX_USERID	DC	CL08'USERID'	The userid or group.
*			
API_PEMX_RSRC	DC	CL246'CEMT '	The name of the resource.
*			
API_PEMX_CLASS	DC	CL08'TCICSTRN'	The resource class name.
*			If blank the value of the
*			XTRAN parameter.
*			specified in the SIT is used
API_PEMX_DELT	DC	CL01'Y'	Specify 'Y' in this field
*			to remove a person from the access
*			list for this resource. The user
*			or group will no longer have access.
*			access to the resource.
API_PEMX_ACC	DC	CL01'R'	Access allowed to the resource.
*			Specify 'R' for read,
*			'N' for none,
*			'U' for update,
*			'A' for alter,
*			'C' for control.
*			Read is the default

---

## RACLINK function (define, list, undefine, or approve user associations)

Use the RACLINK function to list, define, approve, and undefine RRSF user ID associations on the local system.

#### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.RACL) and the default group of the user ID (PEMT.*dfltgrp*) and for the DEFINE function must also have access to the RACLINK.DEFINE.*nodename* and RACLINK.PWSYNC.*nodename* profiles in the RRSFDATA resource class.

#### COMMAREA

Minimum size 1150 bytes.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'RACL'	Function code for RACLINK
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
*			
API_RACL_RC	DS	XL01	Return code from command
API_RACL_CODE1	DS	XL01	'D' DEFINE
*			'U' UNDEFINE
*			'A' APPROVE
*			'L' LIST ASSOCIATIONS
*			
API_RACL_USERID	DC	CL8'userid '	Userid on whose behalf
API_RACL_ATYPE	DC	CL8'PEER '	Assoc. type (PEER/MANAGED
)			
API_RACL_ANODE	DC	CL8'nodename'	Assoc. node
API_RACL_AUSERID	DC	CL8'ibmuser '	Assoc. Userid
API_RACL_PWSYNC	DC	CL4'yes '	yes/no
API_RACL_APSWD	DC	CL8'sys1 '	Assoc. Userid Password
API_RACL ASSOCLST	DS	15CL68	List of 15 associations
*			

The list of associations for the user has the following format:

API_RACL ASSOCTYPE	DC	CL10	PEER/MANAG
API_RACL ASSOCCNODE	DC	CL8	node
API_RACL ASSOCCUSER	DC	CL8	USER
API_RACL ASSOCCPWSYNC	DC	CL4	pwsync
API_RACL ASSOCCSTAT	DC	CL20	status
API_RACL ASSOCCREAT	DC	CL8	creator
API_RACL ASSOCCDATE	DC	CL10	date

The end of the list of associations is indicated by an entry consisting of blanks.

---

## REMOVE function (remove user IDs or groups from a group)

Use the REMOVE function to remove a user or group from group.

### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.REMV) and the target group (REMV.grpname)

### COMMAREA

Minimum size 101 bytes.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'REMV'	Function code for REMOVE
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
*			
API_REMV_RC	DC	XL1	Return code from REMOVE.
*			If non-zero the command failed.
*			API_MSG will give the reason for the failure.
*			
API_REMV_USERID	DC	CL8	Userid/group being removed.
*			
API_REMV_GROUP	DC	CL8	Group being removed from.

Users might not be removed from their default group.

---

## RALTER/RDEFINE/RDELETE function (list and maintain profiles)

Use the RALTER, RDEFINE, and RDELETE function to list and maintain profiles in a general resource class defined in the CDT.

### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.RALT / TOOLKIT.RDEF / TOOLKIT.RDEL depending on the command being performed) and the general resource class (RALT.cdtclass / RDEF.cdtclass / RDEL.cdtclass / RLST.cdtclass)

### COMMAREA

Minimum size 875 bytes.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'Rxxx'	Function code:
*			RALT for RALTER
*			RDEF for RDEFINE
*			RDEL for RDELETE
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
*			
API_RUPD_RC	DC	XL1	Return code.
*			If non-zero the command failed.
*			API_MSG will give the reason for* the failure.
*			
API_RUPD_CODE1	DC	CL4	Type of command being performed
*			'RDEF' to define a profile
*			'RDEL' to delete a profile
*			'AMEM' to add a member
*			'DMEM' to delete a member
*			'UPDP' to update fields in the profile
*			
API_RUPD_CLASS	DC	CL8	The class containing the profile
API_RUPD_CTYPE	DC	CL1	Profile type (not used as input to the API).
API_RUPD_ENTRY	DC	CL246	The profile name
API_RUPD_MEMBER	DC	CL246	The member name
*			
API_RUPD_OWNER	DC	CL8	Owner of the profile.
API_RUPD_NOTIFY	DC	CL8	User to be notified.
API_RUPD_UNIVACS	DC	CL7	Universal access to the dataset.
API_RUPD_WARNING	DC	CL1	Warning mode.
API_RUPD_LEVEL	DC	CL3	Level indicator.
API_RUPD_AUDIT	DC	CL1	Audit flag.
API_RUPD_AUDITQS	DC	CL1	Audit success flag.
API_RUPD_AUDITQF	DC	CL1	Audit failure flag.
API_RUPD_INSTDATA	DC	CL255	Installation data field.

---

## RLIST function (list profiles for general resource class)

Use the RLIST function to list the profile details for a general resource class defined in the CDT.

### AUTHORITY

The user must have access to the zSecure CICS Toolkit command (TOOLKIT.RLST) and the general resource class (RLIST.cdtclass).

### COMMAREA

Minimum size 907 bytes to display the profile. If the request is for the members, users or condacc, the commarea must be large enough to hold all the data returned. If it is not, the **API\_RLST\_RC** is non-zero and the message indicates it as the error.

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

**Note:** If you want to perform a SEARCH, you need to initialize all fields according to your search pattern. Use asterisks for all field padding except for the ENTRY field, which must be padded with nulls, blanks, or underscores.

API_FUNC	DC	CL4'RLST'	Function code for RLIST
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
*			
API_RLST_RC	DC	XL1	Return code from RLIST.
*			If non-zero the command failed.
*			API_MSG will give the reason for
*			the failure.
*			
API_RLST_CODE1	DC	CL1	Request code.
*			'S' = start search
*			'N' = get next profile
*			'L' = retrieve this profile
*			'C' = retrieve conditional access list
*			'M' = retrieve members
*			'U' = retrieve users
*			
API_RLST_RESERVED	DC	CL248	This field is reserved for the
*			API and must be preserved between calls.
*			
API_RLST_CLASS	DC	CL8	The class containing the profile to be
*			retrieved. This field is <i>always</i> required
*			
API_RLST_CTYPE	DC	CL1	Profile type (not used as input to the API).
*			
API_RLST_ENTRY	DC	CL246	The profile to be retrieved. Only
*			required when the CODE field is L, U, C
*			or M otherwise it is used as part of the
*			search criteria.
*			
API_RLST_OWNER	DC	CL8	Owner of the profile.
API_RLST_DEFDAT	DC	CL5	Creation date.
API_RLST_LREFDAT	DC	CL5	Last reference date.
API_RLST_LCHGDAT	DC	CL5	Last update date.
API_RLST_UACC	DC	CL7	Universal access to the dataset.
API_RLST_AUDIT	DC	CL1	Audit flag.
API_RLST_AUDITQS	DC	CL1	Audit success flag.
API_RLST_AUDITQF	DC	CL1	Audit failure flag.
API_RLST_NOTIFY	DC	CL8	User to be notified.
API_RLST_WARNING	DC	CL1	Warning mode.
API_RLST_LEVEL	DC	CL3	Level indicator.
API_RLST_GAUDIT	DC	CL1	Global audit option.
API_RLST_GAUDQS	DC	CL1	Global audit success flag.
API_RLST_GAUDQF	DC	CL1	Global audit failure flag.
API_RLST_SECLEVL	DC	CL3	Security level.
API_RLST_NUMMEM	DC	CL4	Number of members.
API_RLST_NUMUSER	DC	CL4	Number of users/groups.
API_RLST_NUMPGMS	DC	CL4	Number of programs.
API_RLST_INSTDATA	DC	CL255	Installation data field.
		ORG API_RLST_RESERVED	
API_RLST_MEMBUSRS	DC	???XL1	When the members, users or conditional
*			access list is requested the data will be
*			returned into this area.

When a profile or member name is returned, it might be generic. It cannot be converted to a displayable format. To do so, you must be aware of the RACF naming conventions for generic characters.



Generic Character	Converted To
The first '.'	X'02'
Ending double asterisks	X'FD'
Ending single asterisks	X'FC'
Internal double asterisks	X'FBFC90' (when a general resource class) X'FCFC' (when not a general resource class)
Internal single asterisk	X'FBFC80'
Percent sign	X'FB'
Ampersand	X'FA70'

When the list of members is returned, the format of the output is as follows:

Description	Length
Length of member	4 bytes
Member name	? bytes (length determined by length field)

When the list of users is returned, the format of the output is as follows:

Description	Length
Length of userid	4 bytes
Userid	8 bytes
Length of access field	4 bytes
Access	1 byte X'80' Alter access X'40' Control access X'20' Update access X'10' Read access X'01' None

When the conditional access list is returned, the format of the output is as follows:

Description	Length
Length	14 bytes
Filler	8 bytes
Length of userid	4 bytes
Userid	8 bytes
Length of access field	4 bytes
Access	1 byte X'80' Alter access X'40' Control access X'20' Update access X'10' Read access X'01' None
Device type length	3 bytes
Device type	8 bytes
Device name length	3 bytes
Device name	? bytes (length determined by length field)

In all cases, when the first length field is zero (x'00000000'), it indicates the end of the data.

---

## USRDATA function (list and maintain users' USRDATA fields)

Use the USRDATA function to list, add, update, or remove the USRDATA fields of a user profile.

### AUTHORITY

The user must have access to the zSecure CICS Toolkit command. Depending on the function requested, the profile is TOOLKIT.USRL for the list function, TOOLKIT.USRA for the ADD and UPDATE function, and TOOLKIT.USRD for the DELETE function. The user must also have access

to the *USRN.usrdata-name* profile. The affected **USERID** must be within scope for **USRDATA** management functions. This means that the user must have access to *USRU.dfltgrp*.

## COMMAREA

Minimum size 365 bytes.

In your application, use the **CQTMAPIA** or **CQTMAPIC** mapping macros (copybooks) provided in the **SCQTMAC** library.

API_FUNC	DC	CL4'USRD'	Function code for USRDATA management
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
*			
API_USRD_RC	DC	XL1	Return code.
*			01 Invalid function in CODE1 below
*			03 Data from L(ist) function does
*			not fit in commarea
*			
API_USRD_CODE1	DC	CL1	Type of command being performed
*			'L' to list all or one USRDATA field
*			'A' to add a USRDATA field
*			'U' to update a USRDATA field
*			'D' to delete a USRDATA field
*			
API_USRD_CLASS	DC	CL8	Must be 'USER' followed by four blanks
API_USRD_PROF	DC	CL8	Profile (=USERID)
API_USRD_USRN	DC	CL8	The name of the USRDATA field
API_USRD_USRV	DC	CL255	The value for the USRDATA field
*			
API_USRD_USRDLS	DC	CL8	Space for returned USRDATA
	DC	CL255	names and values

The API provides three methods of listing the **USRDATA** of the indicated **USER**. If you select **L** for **CODE1** and specify a value only for **PROF**, all **USRDATA** fields are returned in **USRDLST**. If you also specify a value for **USRN**, the value for the specified **USRDATA** name is returned in **USRV**. If a value is present in the **USRV** field, it is used as the first few characters for the requested **USRDATA** value. For this last function, zSecure CICS Toolkit supplements **USRV** with additional characters from the first matching **USRDATA** Name/Value pair.

Except for the **LIST** and **DELETE** functions, zSecure CICS Toolkit does not support duplicate **USRDATA** names. If you have multiple **USRDATA** name/value pairs with the same name, you might only inspect and delete them, until the name becomes unique.

When using the **LIST** function, only those **USRDATA** names or values are returned for which you are authorized by *USRN.usrdata-name*. If you request all **USRDATA** names/values, the returned list excludes those items for which you are not authorized.

If you request a **LIST** of all **USRDATA** names/values, only those names and values that fit completely within the supplied commarea are provided. In addition, the **API\_USRD\_RC** is set to indicate the overflow condition. If you need all values, you must provide a sufficiently large commarea.

**Note:** The space reserved for each **USRDATA** value is 255 bytes, irrespective of the actual length of the data.

---

## VERIFY function (verify user ID and password)

Use the VERIFY function to verify a user ID and password.

### AUTHORITY

NONE

### COMMAREA

Minimum size 101 bytes (125 bytes if using Newpass, Termid, or APPL parameters).

In your application, use the CQTMAPIA or CQTMAPIC mapping macros (copybooks) provided in the SCQTMAC library.

API_FUNC	DC	CL4'VERF'	Function code for VERIFY.
API_RC	DC	XL01'00'	Return code
API_MSG	DC	CL79' '	Message area
*			
API_VERIFY_RC	DC	XL01'00'	Return code from VERIFY.
*			If non-zero the command failed.
*			API_MSG will give the reason
*			for the failure.
*			
API_VERIFY_USERID	DC	CL08'USERID '	The userid being verified.
*			
API_VERIFY_PASSWORD	DC	CL08'PASSWORD'	The password for this userid.
*			
API_VERIFY_NEWPASS	DC	CL08'NEWPASS'	The new password for this userid.
*			
API_VERIFY_TERMID	DC	CL08'TERMINAL'	A terminal id.
*			
API_VERIFY_APPL	DC	CL08'APPLNAME'	An application name.

The VERIFY does not perform a signon for the user. It only verifies that the password entered for this user ID is correct.

Multiple failures to verify a user ID and password might result in the user ID being revoked, depending on your system parameters.

You might also specify the following optional parameters:

### API\_VERIFY\_NEWPASS

Specifying a new password changes the users password to the specified new password. The **API\_VERIFY\_PASSWORD** field has to contain the valid current password for the user before the new password works.

### API\_VERIFY\_TERMID

If a terminal ID is present and TERMINAL checking is turned on in RACF, it verifies the users authority to use this terminal at the current time and date.

### API\_VERIFY\_APPL

If an application name is present and APPL checking is turned on in RACF, this will verify the users authority to use this application.

This command is only available through the API.

---

## Sample programs

Several example programs are provided for your use as part of the product.

The example described in “Simple API interface” is a general example showing how to use the API interface. The example in “Resource Profile List Interface” is a simple program showing the use of the Resource Profile List API.

## Simple API interface

A sample program that demonstrates how the API might be used is provided in the SCQTSAMP data set. This program shows how a resource check might be performed using the RSRX API interface.

The user interface of these programs is simple, and does not perform any validation or additional processing. The examples are intended only to demonstrate the use of the CQTMPIA area for passing information back and forth between the API and your application program.

To install the sample programs, translate and compile the mapset and the program, and define the resources to CICS. See the following example resource definitions:

```
DEFINE PROG(CQTXAPIR) L(ASSEMBLER) EXECKEY(USER) DA(ANY) GROUP(CQTSAMP)
DEFINE TRANSACTION(XAPI) PROG(CQTXAPIR)
      PROFILE(CQTSAMP) GROUP(CQTSAMP) TASKDATALOC(ANY)
DEFINE MAPSET(CQTXAMP) GROUP(CQTSAMP)
```

Program CQTXAPIR allows you to enter resource names and resource classes. It then checks to see if you have access to the resource. Alternatively, you might enter a user ID, other than your own, against which to perform the access check.

You can tailor this example to suit your own environment or to do specific editing on any of the fields.

## Resource Profile List Interface

An example program that demonstrates how to use the Resource Profile List interface is provided in members CQTXAPIL, CQTXAML, CQTXCPIL, and CQTXCML in SCQTSAMP. The same program is provided in both assembler and COBOL form.

### About this task

The programs use a BMS map to display an initial panel, where the user can fill in some filters and options for the RSRL (Resource Profile List) API interface. The source for the BMS map is provided twice to generate different include members, but the two source members are otherwise identical. The programs call the API and show some relevant parts of the output. The programs serve no practical purpose, aside from illustrating how the API can be used, and verifying that the installation has completed successfully.

### Procedure

To install the samples programs:

1. Translate, compile and linkedit the mapset and the program.
2. Define the resulting modules to CICS.

### Results

See the following example resource definitions:

```

DEFINE PROG(CQTXCPIL)    L(COBOL)          EXECKEY(USER) DA(ANY) GROUP(CQTSAMP)
DEFINE TRANSACTION(RSRC) PROG(CQTXCPIL)
                        PROFILE(CQTSAMP) GROUP(CQTSAMP) TASKDATALOC(ANY)
DEFINE PROG(CQTXAPIL)    L(ASSEMBLER)      EXECKEY(USER) DA(ANY) GROUP(CQTSAMP)
DEFINE TRANSACTION(RSRA) PROG(CQTXAPIL)
                        PROFILE(CQTSAMP) GROUP(CQTSAMP) TASKDATALOC(ANY)
DEFINE MAPSET(CQTXAML)  GROUP(CQTSAMP)

```

**Note:**

- To use the example programs, install the group CQTSAMP as shown, and run the transaction RSRA or RSRC.
- To view the output, you might need to use CEBR to view the entire TSQUEUE that is created as part of this program.
- It is the responsibility of the calling program to remove the TSQUEUE after it has been created.
- The example programs do not delete the TSQUEUE after usage, so that you can inspect the data after the transaction has ended.
- After completing your testing, manually delete the TSQUEUE.



---

## 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](http://www.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](http://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

## A

- accessibility x
- ADDGROUP command 36
- ADDUSER command
  - panel 38
- ALTGROUP command 36
- ALTUSER command 39
  - CICS segment option 41
  - OMVS segment option 44
  - panel 39
  - TSO segment option 43
  - WORKATTR segment option 47
- API 87
  - Access Authority Check (Extended) function 91
  - Access Authority Check function 90
  - ADDGROUP / ALTGROUP / DELGROUP function 103
  - ADDUSER function 104
  - ALTUSER (CICS SEGMENT) function 107
  - ALTUSER (OMVS SEGMENT) function 108
  - ALTUSER (TSO SEGMENT) function 107
  - ALTUSER (WORKATTR SEGMENT) function 109
  - ALTUSER function 105
  - CONNECT function 110
  - DELETE DATASET function 111
  - DELETE USERID function 111
  - implementing field or record level security 89
  - LISTDATASET function 112
  - LISTGROUP function 113
  - LISTUSER function 115
  - PASSWORD function 117
  - PERMIT function 118
  - PERMITX function 119
  - RACLINK function 119
  - RALTER/RDEFINE/RDELETE function 121
  - REMOVE function 120
  - Resource Profile List function 92
  - RLIST function 121
  - Sample programs 126
  - searching the RACF database 89
  - simple interface 126
  - USRDATA function 123
  - VERIFY function 125
- API functions
  - Changing the authorized user 88
  - using the COMMAREA 87
- Application interface
  - overview 1
- Application security check
  - application conversion 29
- Application security checking
  - alias definitions 31
  - checking the OP-ID 29
  - with zSecure CICS Toolkit 29

## B

- BMS mapsets 22

## C

- checklist
  - installation 5
  - post-installation 5
- CICS
  - application security checking for defining programs, mapsets and transactions to 12
  - Transaction Server 22
  - updating tables 13
- Command interface
  - overview 1
- Command interface overview 35
- commands
  - ADDGROUP 36
  - ADDUSER 38
  - ALTUSER 39
    - CICS segment option 41
    - OMVS SEGMENT option 44
    - TSO segment option 43
    - WORKATTR segment option 47
  - Command interface overview 35
  - CONNECT 48
  - DELETE 50
  - handling API requests 87
  - LISTDSET 51
    - panel 54
    - toggle option 54
  - LISTGROUP 57
  - LISTUSER 63
  - Main menu 35
  - PERMIT 72
  - RACLINK 73
  - RALTER 75
  - RDEFINE 75
  - RDELETE 75
  - REMOVE 76
  - RLIST 77
  - USRDATA 82
- CONNECT command 48
- Control 85
- CQTJACC 9
- CQTJALL 8
- CQTJAPP 9
- CQTJDDD 9
- CQTJRDO 12
- CQTJREC 9
- CQTJSMFA 8
- CQTJSMPB 8
- CQTJSMPC 8
- CQTJPCNTL
  - parameter definitions 12
  - parameter descriptions 23
  - parameters for zSecure CICS Toolkit 23
  - parameters verification 26

## CSI

- defining Global 8
- defining Product 8

## D

- data set profile
  - listing 112
- DATASET profiles
  - granting access to 119
  - removing access to 119
- date formatting 3
- DELETE command 50
- DELGROUP command 36

## E

- education x
- Exit points
  - transferring control from zSecure CICS Toolkit 85

## F

- Field
  - implementing security for 89

## G

- Group
  - adding 103
  - altering 103
  - connecting a user or group to 110
  - deleting 103
  - listing profile 113
  - removing a user or group from 120

## I

- IBM
  - Software Support x
  - Support Assistant x
- IEASVCxx
  - update 10
- information retrieval 32
- installation
  - allocating TARGET and DLIB data sets 8
  - applying the product 9
  - checklist 6
  - creating and initializing SMP/E zones 7
  - defining options 8
  - protecting the SVC 10
  - receiving the product 9
  - SMP/E 5
  - updating SMP/E DDDEFs 9
- Installation
  - automatically assigning USS UIDs 19

Installation (*continued*)  
 automatically creating home directories 19  
 defining programs, mapsets and transactions to CICS 12  
 defining SCQTLOAD as APF-authorized 11  
 Enabling/Disabling zSecure CICS Toolkit 11  
 installing the SVC 10  
 making the RACF definitions 14  
 updating CICS tables 13  
 updating the CICS startup JCL 11  
 installation checklist 5  
 introduction  
 installation 5  
 IPL  
 update 10  
 IRRPNL00 function 96

**J**  
 JCL  
 for installation 6

**L**  
 LISTDSET command 51  
 display option 54  
 Programs option 56  
 panel 56  
 toggle option 54  
 Userids option 55  
 panel 55  
 LISTGROU command 57  
 Display option 59  
 panel 59  
 panel 57  
 Subgroups option 62  
 panel 62  
 Toggle option 60  
 panel 60  
 USERIDS Delete option 61  
 panel 61  
 USERIDS option 60  
 panel 60  
 LISTUSER command 63  
 Categories option  
 panel 69  
 Display option 66  
 panel 66  
 Groups option 68  
 panel 68  
 OMVS option  
 panel 70  
 panel 63  
 Segments option 70  
 panel 70  
 Toggle option  
 panel 67  
 WORKATTR option  
 panel 70  
 LPALSTxx  
 update 10

**N**  
 National Language Support 22

**O**  
 OMVS  
 assigning UIDs automatically 19  
 automatically creating home directories 19  
 online  
 publications v, vi, ix  
 terminology v  
 Operator ID check 29  
 application conversion 29

**P**  
 parameters  
 CQTPCNTL values verification 26  
 Parameters  
 descriptions of CQTPCNTL parameters for zSecure CICS Toolkit 23  
 Parameters for 23  
 PARMLIB 11  
 Password  
 changing 117  
 verifying 125  
 PERMIT command 72  
 panel 72  
 post-installation checklist 5  
 problem-determination x  
 Profile  
 delete 111  
 listing and maintaining 121  
 view details of 121  
 publications  
 accessing online v, vi, ix  
 list of for this product v, vi, ix  
 obtaining licensed v  
 obtaining licensed publications vi

**R**  
 RACF  
 defining zSecure CICS Toolkit commands to 14  
 RACF database  
 searching 89  
 RACLINK command 73  
 panel 73  
 RALTER command 75  
 RDEFINE command 75  
 RDELETE command 75  
 reason codes, IRRPNL00 function 96  
 Record  
 implementing security for 89  
 REMOVE command 76  
 Resource  
 checking user access 90, 91  
 granting access to 118, 119  
 removing access to 118, 119  
 resource access verification 32  
 Resource Profile List 126  
 Restart 20

Restart (*continued*)  
 manually 21  
 return codes, IRRPNL00 function 96  
 RLIST command 77  
 Conditional access option 81  
 panel 81  
 Display option 79  
 panel 79  
 Members option 79  
 panel 79  
 Users  
 panel 80  
 Users option 80  
 RRSF 2  
 RRSF userid associations  
 approving 119  
 defining 119  
 listing 119  
 removing 119  
 RSRC / RSRX functions 29  
 RTCK transaction  
 verifying CQTPCNTL parameters 26  
 RTST transaction 21

**S**  
 SCQTLOAD 11  
 Security checking  
 single point 29  
 simple  
 application security interface 31  
 simple application security interface 31  
 SMP/E Modification Control  
 Statements 9  
 SVC  
 unauthorized use 10

**T**  
 terminology v  
 toggle option, LISTDSET command 54  
 training x  
 Translating BMS mapsets 22  
 troubleshooting x  
 TSQUEUE for profiles 96

**U**  
 User  
 adding a profile for 104  
 auhorization for RACF  
 commands 14  
 changing password 117  
 changing the authorized user 88  
 changing the CICS segment 107  
 changing the OMVS segment 108  
 changing the profile for 105  
 changing the TSO segment 107  
 changing the WORKATTR  
 segment 109  
 check access to resources 90, 91  
 delete id 111  
 listing and maintaining  
 USRDATA 123  
 listing profile 115  
 user information retrieval 32

USRDATA command 82

## **V**

verifying resource access 32

## **Z**

zSecure CICS Toolkit  
  manual restart 21  
  Security resources 16







Printed in USA

SC27-5649-03

