



CICS Essentials

Auditing **CICS** – A Beginner's Guide

Julie-Ann Williams
Mike Cairns
Martin Underwood
Craig Warren



Foreword

by **Brian Cummings**

A thorough Audit Guide for CICS is something that is long overdue. This document provides a wealth of information about CICS, its operations, and its various resources and capabilities along with audit guidelines and recommendations. Various documents on AuditNet and other sources have taken a stab at parts or all of CICS, but are likely not up to date or sufficiently complete. CICS largely remains an environment that holds its mysteries against auditors and security officers alike. The results of poor understanding can lead to dangerous levels of unidentified risk to the applications and sensitive information of entities that use the power of CICS for critical business applications.

Unlike any other environment, CICS security implementations fail in the first place because all of the security control is often only focused on transactions. Transactions are many levels of resources removed from the data files and data bases they query or update. In the end, we see the greatest level of security established for the least sophisticated technical users – end business users, and the least security facing the most technically sophisticated – the CICS sub-system programmer and the CICS Application programmer. For example, it is typical to leave FCT resources unsecured and to allow the CICS regions to have total rights to the data sets they access. This condition gives sub-system and application programmers full-reign to use CICS utilities to inherit the CICS regions' authorities and gain full access to freely browse and update data. Worse, such activity would take place well beneath the business and process internal controls established to assure the integrity of the data.

There are many other security failures prevalent in CICS security implementations such as: empowering the CICS region default userid; running all CICS sub-systems and regions under the same user account or group, thus failing to achieve a separation of function across business applications; and inadequate protection of high-risk CICS system supplied transactions.


I learned a great deal by reading this document, and will value it as a handy reference for my CICS security implementation and audit activities. I'm certain that you will find it equally useful, and possibly disturbing. As a peer professional so well said: When I realize that I don't know something that is important to me, job one becomes to learn what I need to know. This document is a great start.

Brian V. Cummings
Practice Lead, IRM Advisory Services
Tata Consultancy Services North America

by **Mike Cairns**

I was invited into this project late in its development, and asked to contribute some of my previously published articles on the subject of CICS security. When I was publishing online articles about CICS, the writing was limited to well under 2000 words to fit inside publishing limitations. With this book though, we see at last a larger format where subjects can be explained in more depth and detail than I could in my earlier work.

It's been a delight to be able to help a dedicated team of writers complete this



comprehensive introduction to auditing CICS. My contributions have been small, some old articles, and a bit of editing. The chance to re-write my old articles, and try to clarify the parts I now considered weak, was the best part of this project personally for me.

But for the group, I have to congratulate Julie-Ann, Martin and Craig for creating the first detailed work on CICS audit that I know of. It's a complex topic, and needs a book of this length to do it justice.

We hope that all auditors when faced with a z/OS audit will find our contribution useful, and we look forward to providing future assistance with similar publications.

Mike Cairns – August 2009

Table of Contents

About this Book	1
About the Book's Sponsor	1
About the Author(s)	1
About You	2
Icons Used in this Book	2
More Detailed Technical Information	3
Introduction to CICS Audit requirements	5
What is CICS?	5
How is CICS used?	6
Databases and CICS	7
Networks and CICS	7
External Security control and CICS	8
What types of risk need to be considered when auditing CICS? ..	11
z/OS elements	11
DB2 elements	12
Networking elements	13
Auditing CICS 101	14
Auditing CICS - A Beginners Guide	15
Where to look and what to look for	15
Job Control	15
Associated Userid	17
Datasets	17
STEPLIB/STEPCL	18
Journals and Logs	18
Dynamic transaction backout	19
Recovery after a system abnormally terminates	19
CSD	19
System Initialization Parameters	20
Override Parameter Settings	20
SIT Settings	20
CMDSEC	21
CONFDATA	21
CONFTXT	21
DFLTUSER	22
EJBROLEPREFIX	22
ENCRYPTION	22
ESMEXITS	22
GMTRAN	22
KEYRING	23
PLTPIUSR	23
PLTPISEC	23
PSBCHK	24
RESSEC	24
SEC	24
SECPRFX	24
SECPREFIXID	25
SNSCOPE	25

TCPIP	26
USRDELAY	26
XAPPC	26
XCMD	27
XDB2	29
XDCT	29
XEJB	30
XFCT	31
XHFS	31
XJCT	32
XPCT	33
XPPT	33
XPSB	34
XRES	34
XTRAN	36
XTST	36
XUSER	37
External Security	37
Userids	38
RACF classes	39
RACF Grouping Classes	40
Differences When Using Other External Security Managers	40
CA ACF2	41
CA Top Secret	43
CICS System Definition - CSD	46
History	49
Future (CPSM)	49
IBM Supplied CICS transactions	50
Category 1	50
Category 2	51
Category 3	52
Securing CSD Transactions	53
Glossary of Terms	55
Index	59
Future Publications	60

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

About this Book

This book is designed to work in the same sort of way as the famous “for Dummies” books. It’s not one of the official sequences but the style works so well for explaining complex functionality that it seems the best approach for this subject.

You should never try to audit anything using just a “for Dummies” book! The aim here is to make the whole process slightly less intimidating and more accessible to people who have already been around the audit industry for a while.

There’s nothing to memorize. There will be no tests at the end.

What you will find are de-jargon-ified explanations of concepts and specific parameters. It is a distillation of a number of people’s personal experiences in the field written in “Clear English”.

About the Book’s Sponsor

For most of my career I have been a Trainer. I like being able to make it easier for other people to understand a subject than I found it when I first learnt. I’ve been wanting to write a book like this about CICS Audit for a long time. CICS is a complex topic and auditing it can be a real challenge even if you do understand the basics - I still hesitate before accepting a CICS audit assignment. When I heard that NewEra wanted to commission this book I jumped at the opportunity. NewEra Software is one of several providers of z/OS integrity solutions and when they asked me to undertake the assignment I made it clear that I would not show bias towards any product. There are a number of different choices a customer can make about vendor solutions and this book needs to stand in any CICS environment. I feel that to offer auditing solutions suggestions would distract from the underlying message - that CICS, whilst complicated, is understandable.

NewEra, along with a number of other vendors, offer solutions which can greatly enhance the compliance of the CICS environment. For details on these products please look out for future white papers from the author(s).

About the Author(s)

Julie-Ann Williams has been messing around with IBM mainframe computers for most of the last 30 years. She has been helping Customers to get ready for external audits since 1987.

As well as being something of a RACF geek, Julie-Ann has extensive experience with web enabling mainframe applications for large IBM Customers and was one of the first people in Europe to implement Domino (Lotus Notes) on a mainframe. She has an unusual blend of skills encompassing detailed mainframe knowledge as well as “newer” technologies like WebSphere, TCP/IP and Unix combined with communications abilities and Mentoring.

Julie-Ann took the lead in writing this book ably assisted by a number of Industry Experts including very significant contributions from:

Mike Cairns started in mainframes in the mid 1980s. After some years as a developer he discovered his calling in a RACF support role involving lots of assembler and ISPF dialog programming - he never looked back.

A regular participant in the RACF public forums for over ten years now, and writer for IBM Systems Magazine (google ‘mike cairns RACF’), he is now employed by IBM working with RACF customers throughout the Asia Pacific region.

Martin Underwood is an enthusiastic z Evangelist with almost 25 years experience. His most recent specialism has been in helping z Customers to prepare for all sorts of IT audits. He says he never wants to stop learning and that teaching is the best way to learn. Deep.

Craig Warren has been in the z Industry for a quarter of a century. He laughs at those who say the mainframe is dead and his past couple of decades working on bleeding edge z projects gives him good reason!

About You

You are either:

- a CICS Systems Programmer who is approaching an external audit, possibly for the first time, and wants to know what might be looked at.
- an IT Auditor who finds yourself auditing CICS at a z Series installation, possibly for the first time, and wants to know how to ask questions that will actually get useful answers.

Icons Used in this Book

You've seen other "for Dummies" books. You know how this works. Icons are used as short hand ways of saying the same important things.

The following two icons indicate expert knowledge that you will need in order to understand how to audit a CICS system:



Don't forget to remember these important points – or at least remember where you read about them! They will help you to understand the background of the system that you are auditing. CICS has been around for 40 years. There are a few quirks which you should know about.



This icon alerts you to a juicy piece of information that will make auditing CICS easier. It may be a technical tip or advice to talk to a specific group of people to save time when finding the right information or other gems of hand acquired wisdom.



Other icons in the margin indicate a specific software product or auditing standard which is relevant to the point being made:

IBM's DB2 subsystem. They say: "DB2 offers industry leading performance, scale, and reliability on your choice of platform from Linux to z/OS." DB2 is one of the most common products to find in conjunction with CICS. Auditing the CICS to DB2 connection does not constitute a full DB2 audit.



IBM's IMS subsystem. They say: "Information Management System (IMS) is IBM's premier transaction & hierarchical database management system.". IMS is less commonly installed than DB2 but there are still significant (and maybe more importantly, stable numbers of) IMS installations around the world.



Sun Microsystems' Java programming language. They say: "A complete environment for application development and deployment". You can probably "hear" the difference in tone between IBM's mainframe comments and Sun's sunny internet proclamation. This is a source of some friction within old mainframe development teams. Tread gently when you approach the audit.

Sarbanes Oxley IT Auditing Standard. The Sarbanes-Oxley Act was signed into law on 30th July 2002, and introduced highly significant legislative changes to financial practice and corporate governance regulation. It introduced stringent new rules with the stated objective: "to protect investors by improving the accuracy and reliability of corporate disclosures made pursuant to the securities laws".

SAS 70 IT Auditing Standard. Statement on Auditing Standards (SAS) No. 70, *Service Organizations*, is a widely recognized auditing standard developed by the American Institute of Certified Public Accountants (AICPA). It is a standard which is audited to around the world although most commonly in North America and it applies to organisations that have IT responsibility for multiple businesses. The simplest example is a Facilities Management organisation running 2 or more different company's systems on a single z processor.

Comments made with regard to Sarbanes Oxley and/or SAS 70 within this document are notes of personal experience whilst being audited to these standards. They do not represent exhaustive instructions on performing an audit to either standard.

More Detailed Technical Information

This is at heart a technical document. I have drawn heavily on the IBM documentation and would urge anyone with a reason to read this book to try these sites too:

CICS

www-01.ibm.com/software/htp/cics

publib.boulder.ibm.com/infocenter/cicsts/v3r2/index.jsp?topic=/com.ibm.cics.ts.doc/InformationCenter

z/OS

www-03.ibm.com/systems/z/os/zos/bkserv

www-947.ibm.com/systems/support/z/progportal

www-03.ibm.com/systems/z/os/zos/bkserv/lookat

Audit Standards

www.sarbanes-oxley-forum.com

www.sas70.com

Computer Associates

support.ca.com/irj/portal/anonymous

Sarbanes-Oxley
Financial and Accounting Disclosure Information





Introduction to CICS Audit requirements

This book will provide an overview of the basic CICS environment to a technically capable but non-CICS specialist auditor. The book as a whole focuses on RACF as the security product in use but there are 2 alternates – CA ACF2 and CA Top Secret. There is a later chapter which deals with differences between the products and how those relate to auditing CICS on zSeries.

What is CICS?

Immediate access to online information, whether it be about customers, suppliers, or even the status of our latest order from Amazon is something that everyone knows and understands. We could even say that we take it for granted.

Digital technology has brought all of this information into our homes and offices, so we can now query the status of any information we have access to with just a few keystrokes. Where was this step forward taken? What made it possible that we no longer had to wait for the morning report to be delivered before we could query the state of a customers invoice? When did we first not have to wait for the postman to deliver the mail before we knew why our mail order goods had been delayed (or even to deliver them)?

Digital information and even digital goods bring a level of service delivery where information is now available within seconds, not days. This revolution in computing and the way that digital information is used affects every aspect of our work and personal lives.

This expectation of, almost, immediate availability of goods and information feels like a recent development but has actually been around for more than 4 decades. While the home user had to wait for the Internet for this level of data availability to enter their lives, the business community has had this information available, using CICS, since 1968.

IBM finally released the first commercial version of CICS on July 8 1969. This pre-dates the moon landing by 13 days, and first words Neil Armstrong uttered on the moon apply just as much to CICS! It introduced a giant leap both in technological and logistical ways of doing business and the ability to react to changes to the information held regarding customers.

CICS or Customer Information Control System Transaction Server to give it its full title is a comprehensive transaction management system that is mostly used to display and manipulate data. It introduced a flexible solution where a business could query customer information regardless of their business model in real time and display this information on a screen on demand.

This data can then be assessed and any alterations that are required due to changing circumstances, such as the payment of an invoice or any transaction required by the business, can be made. These alterations to data can be made in real time to the database keeping the information held about customers as up to date as possible.

One other important innovation introduced to the world by CICS was the 2 phase commit process. This control method invokes two separate phases to data being written to the database.

During the first phase the alteration to the data is prepared for writing to the data-

CICS Pronunciations

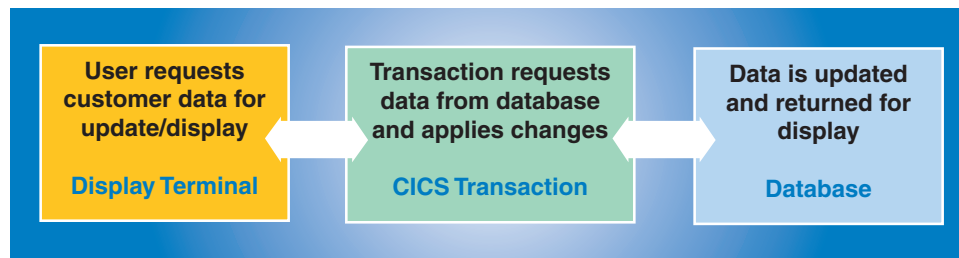
- U.S. – See-Eye-See-Ess
- UK, Australia, Belgium & Hong Kong – kicks
- Most of Spain – thicks
- Latin America – sicks
- Catalan – sicks
- Germany – tsicks
- India – kicks
- Portugal & Brazil – seeks
- Poland – kiks
- Italy – chicks



base. At this stage no actual change to the data has been written to the database, so in the event of failure the stored data does not reflect any changes.

The second phase is the commit, where the data is written to the database. This stage of the process is not entered until enough information is stored in the logs to allow any changes to be rolled back in the event of a failure. For this stage of a transaction to complete a positive response must be received from all participating resources. In the event of a positive response not being received any alterations to the data can be undone, restoring the data to its pre-transaction processing state.

A simplified, graphical representation of a basic model of CICS function is shown below:



When changes have been made they are applied to the stored data. This newly updated data can then be made accessible throughout a company's entire corporate structure if required. This allows any decisions and assessments to reflect the most up to date information available on the current state of the customer's accounts. In banking this is often used for practical applications such as the authorization of ATM requests.

CICS handles all of the communication between the user session and the data stored in the database from a central position passing data between them both. This effectively defines the CICS transaction server as middleware. Defined transactions form a controlled conduit to ensure that data is requested from and delivered to the correct locations.

In 1979 IBM revealed the CICS MRO (Multi Region Operation) feature that enabled customers to split a single CICS service into multiple regions spreading the workload between them. This enabled early adopters of the technology to avoid operating system limitations whilst supporting ever increasing user populations. The spreading of CICS workload was further enhanced by the later implementation of the parallel sysplex concept and further still by CICSplex.

A CICS service using MRO will consist of a number of different types of CICS regions - common types are AOR, DOR, FOR, TOR. For each type of region there may be multiple instances depending on the workload being supported.

How is CICS used?

CICS is used in many different business types such as Banking, Manufacturing control systems and Insurance. It is highly configure-able and can meet the requirements needed to interact with data stored in a wide variety of databases. It is used in over 90% of the top 500 companies worldwide as their primary transaction service for core business functions. This is due to its initial design requirements for high speed, high volume on-line processing of transactions.

To take just one example, within a banking environment, the CICS Transaction Server can process millions of online transactions a day, being used for many fa-



miliar functions such as ATM transactions. In fact, it is important to note that all ATM transactions will at some point invoke CICS. And with increasing numbers of web applications also connecting to CICS it is far from the “legacy software” that some people refer to.

The high availability of the System z, mainframe platform makes use of CICS Transaction Server ideal as it ensures that customers have constant access to data such as account balance, allowing cash withdrawal transactions to be performed outside of the normal banking hours.

Each CICS region may be a standalone region that performs all processing for a CICS Service or it may be one of many regions combined using MRO and parallel sysplex into a single CICS Service. Although this guide only covers an audit of a standalone region an auditor must be aware of the concept of CICS MRO and its management when making recommendations. CMAS and CICSplex may also be present and are used in the management of the entire CICS service.

While the CICS application was initially developed for use under IBM mainframe operating systems, a distributed platform version, called TXSeries runs on AIX, HP-UX, Solaris and Windows. This book deals solely with the IBM mainframe supported version.

Databases and CICS

As middleware CICS provides an intelligent, transaction processing link between the user and the stored data. This data can be stored in a number of different proprietary database structures.

CICS-defined File Control allows multiple users access to the data stored in these databases. Each accessible file, regardless of structure is defined to the CICS File Control Table. This table provides the ability to share access to data files and databases by multiple applications.

CICS also ensures that, while multiple applications can access data records for READ and UPDATE, no two users will be allowed simultaneous access to a single record at the same time.

Database Management products can also perform this function, e.g. Oracle only releases the lock on a record after COMMIT.

Access to databases is supported outside of CICS and controlled by the individual database management tools. CICS programs can make database calls requesting access to data stored.

CICS is not limited to storing data in proprietary database management systems. In fact it can access data from almost any file stored on disk.

Networks and CICS

Connection to a CICS address space was traditionally via a VTAM terminal. Developments in the methods used to gain access to z/OS systems has meant that CICS has had to continuously move forward to keep pace with advances in network technologies.

SNA evolved with the operating system and the introduction of SNA LU 6.2 allowed communication between CICS transaction servers using CICS Inter System Communications, or ISC. This allows transactions running on your CICS transaction servers to pass information between one another, meaning that related systems do not have to run on the same CICS transaction server as CICS ISC could access

Some supported DBs

IBM

- DB2
- IMS

Computer Associates

- Datacom
- IDMS

Others

- Oracle
- etc

remote data from another CICS system or even initiate remote transactions.

The Multi Region Operation function allows CICS systems to communicate with one another using cross memory facilities and also by using the coupling facility, but this can only be used between CICS Transaction servers running in the same sysplex.

The introduction of full function TCP/IP on z/OS brought with it a much more flexible framework for communications with the mainframe, allowing connection from outside the local network. In CICS release 3.2 a new TCP/IP based communications protocol called IP interconnectivity (IPIC) was introduced which allows a number of functions to be called across a TCP/IP network between CICS Transaction servers that are not in the same sysplex. This allows much of the ISC functionality to now be performed over a TCP/IP network:

- Distributed Program Links
- External Calls
- Data requests

The CICS Transaction Server is already ahead of the game, being compatible with some advancements in network technology that have not yet been implemented by the majority of sites, e.g. TCP/IP v6. This reflects IBM's requirement that the product should be built to allow for the future.

The CICS Transaction Gateway allows access to CICS using Java applications over the internet, meaning that access to your CICS transaction server is no longer limited to the traditional 3270 type terminal using an SNA network. WebSphere Application Server introduces the ability for internet based applications to be used as a front end to drive CICS transactions.

The CICS Transaction Gateway also has the functionality that allows it to be used in a load balancing capacity.

The methods of connection to the CICS transaction server have expanded massively since its initial release and will continue to do so for as long as the product is in use. As network technologies expand and grow, so will the available methods of connecting to the CICS Transaction Server.

External Security control and CICS

CICS can interface directly with all three of the major z/OS external security managers (ESM):

- CA ACF2
- RACF
- CA Top Secret

The majority of the discussion in this book is around RACF and CICS as that is the most common combination across all industry sectors. The concepts are identical across the ESMs but the implementation can be quite different.

In the early days of CICS there was no external security. All user information was stored in CICS tables and had to be maintained by CICS Systems Programmers. As soon as security became important, this became unacceptable. The CICS Systems Programmers were (and still are) very busy and so the focus was on getting things done rather than security.

The introduction of external security meant that security administration could be placed into the hands of specialist teams.

In order to enable CICS for integration with RACF without having to rewrite all of the existing applications, IBM implemented CICS external security in such a way that exactly the same information was available from within the same control blocks. This is achieved using a special version of the RACF call – FRACHECK – which loads CICS related RACF profiles into z/OS storage. This is a great example of IBM's promise to support older ways of working so that massive development cost were not encountered at every operating system (or sub-system) upgrade.

There are many points at which security can be applied with each one being enabled by a CICS SIT parameter (generally known as the *Xnnn* parameters because most of the RACF class enabling parameters start with the letter X). This allows an installation to phase the introduction of increased security.

Whilst CICS is incredibly securable, none of this happens by default. Most of the defaults switch processing off rather than on. And problems can occur if things are done in the wrong order. For example, if CICS tries to start with a RACF class which has not been fully defined, the start up will fail. This is an example of “fail safe” and is an integral part of the CICS environment.

One last point about securing SIT parameters before moving on to the actual audit information; for a very long time IBM provided a grouping of SIT Keywords into the categories (SIT Control Groups) shown below but this has recently fallen out of use.

- | | |
|--------------------------|---------------------------|
| 1 Application Issues | 16 Monitoring |
| 2 Autoinst VTAM & APPC | 17 RDO:Control Attributes |
| 3 Autoinstall Programs | 18 Security |
| 4 Basic Mapping Support | 19 Signon |
| 5 Data Interchange | 20 Storage management |
| 6 Dispatch Functions | 21 Supervisor Calls |
| 7 Dump Functions | 22 System Initialization |
| 8 Exits | 23 System recovery |
| 9 Ext. Recovery Facility | 24 System Termination |
| 10 Files (user) | 25 Temporary Storage |
| 11 Front End API | 26 Terminal/LU Mgmt. |
| 12 Intercom & MRO | 27 Trace |
| 13 Journaling | 28 Transient Data |
| 14 Loading Programs | 29 Timer |
| 15 Miscellaneous | |

I believe that one of the reasons for the decline is that the control groups no longer match with the designated function. For example, this document deals with all aspects of security but its scope is beyond that of just the group 18 SIT parameters.



What types of risk need to be considered when auditing CICS?

This chapter will explain the risks that you'll need to consider when performing an audit of the CICS environment. This includes the operating system, databases, networking, and other considerations that can have a significant impact on CICS.

z/OS elements

z/OS has a reputation for being the most securable platform commercially available. This together with the stability inherent in the operating system means that it is the platform of choice for large organisations worldwide. The complexity of System z means that no single part of the operating system can be audited on its own.

A much more holistic view of the operating system must be taken to be sure that any CICS audit also takes in all the relevant elements of the operating system and other sub-systems.

CICS needs three specific areas of functionality in order to be able to provide the service it was designed to deliver. This brings to light a number of key issues which need to be investigated outside of the CICS application to ensure that access to the business critical data addressed by the CICS transactions is protected from unauthorized access and change. In order to perform an Audit on a CICS environment you must also think “outside the box” and look at the additional areas that will require review.

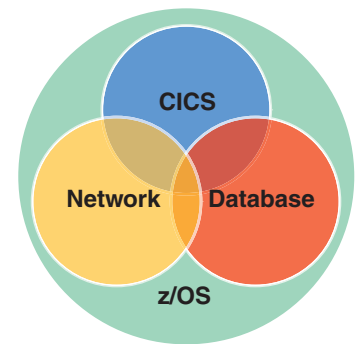
Consider a visit to the doctor where you complain of shortness of breath. First you get a check of your lungs but during your examination, you may also have your temperature taken, and your heartbeat recorded. The root cause of a problem is not always the obvious one.

This is also true with a CICS environment as it is an especially versatile product that can effectively be all things to all people dependent on requirements. With this being the case any audit of CICS will have to include an end-to-end audit of all products on the z/OS platform that interact with CICS, together with some that do not directly interact.

The z/OS base elements must be considered as part of any CICS audit, and this includes a review of external security settings regarding CICS itself, together with any resources that CICS accesses for transaction processing.

Access to the CICS files must be controlled by RACF. Programs and configuration files should also be RACF protected. Only authorised users should have access to these resources, with special consideration given to who is allowed to alter these files. There are large numbers of files used to initialize CICS and for transaction processing:

- Configurations Files
- Programs and Transactions
 - Compiled
 - Source
- Data
 - Internally stored
 - External Databases



Access to these will be handled by your security product so a review of the RACF settings for these resources will require investigation.

Integration with UNIX Systems Services (USS) also requires review. USS environments within companies are expanding exponentially, and if your CICS facility requires USS functionality then this will introduce an additional audit point to cover UNIX resources.

Data Ownership is an issue that is often flagged in an audit. With CICS the ownership of configuration datasets, as well as the data handled by the transactions must be reviewed.

A robust and regularly tested disaster recovery plan is a must for any business. This however requires the storage of data remotely, whether it be at a mirrored, dynamic site or as physical backup media. Attention must be drawn to the method used at your site to ensure that this data is stored securely, with the same degree of diligence, and meeting the same security standards. If a mirrored site is used, then all data transfer should be sent over encrypted links to ensure data security.

Access to the REXX for CICS scripting tool should be heavily restricted as it allows the development of programs capable of manipulating CICS data.

CICS users require access to the transactions and data required for them to function in their role in the business. Users who also have TSO access have the ability to browse, copy, delete and alter datasets, and proper control should be in place to ensure that access to CICS related datasets is restricted to job function.

The levels of logging and who has access to these logs is often overlooked when initially setting up a system. This can lead to information regarding transactions and userids being made available in areas where it is not required potentially exposing Customer data.

Finally the z/OS Base Control Program is directly responsible for providing operating system services that are critical to the running of all tasks. Changes to individual elements of the BCP can have a major effect on CICS.

DB2 elements

Businesses are aware that their most valuable asset is often not a tangible one, but the data that is held to ensure that they can continue to trade in an efficient and, hopefully, profitable manner. Without a secure and available data stream businesses will grind to a halt regardless of the business model, therefore the integrity of this data and its availability is a major consideration.

While this section will deal with DB2, the same levels of consideration should be given to whatever method of data storage is used at your site.

DB2 has the ability to handle security either internally or externally.

- Internal - If internal DB2 security is used access rights to all data stored in DB2 Tables accessed by CICS is administered internally by DB2 and would need reviewing to ensure that no more than the required level of authority is granted. The administration of DB2 internal security is often the responsibility of the database analyst, rather than the security team. This is often viewed as 'not ideal'.
- External - With the use of external security for DB2 a RACF resource class name is specified in DB2 related initialization processes. This allows RACF to check that the user associated with the CICS transaction has the necessary rights to be granted to access the specified records stored in the DB2 database.

Because DB2 is a completely autonomous product to CICS, data can be extracted or manipulated directly by DB2. This introduces an additional area for investigation, ensuring that data cannot be accessed by unauthorized personnel with elevated DB2 access rights.

A CICS user may need access to specific DB2 resources such as the ability to modify:

- DB2CONN
- DB2ENTRY
- DB2TRAN

Alternately they may use a transaction, accessing data that is stored in a DB2 database. This requires communication between CICS and DB2 to be configured and secured.

In order for DB2 and CICS to be able to communicate, authorization ID's have to be set up. CICS has 2 types of processes that require authorization ID's

- Connection - this is required for CICS to be able to connect to the DB2 database.
- Transaction - each transaction must be covered by a relevant definition in the security database granting it, and the associated user, access to the required data stored in the DB2 database.

Networking elements

With the ability to shape CICS to whatever environment the customer requires comes a requirement for it to integrate with any method of communication available on the z/OS platform. This requirement for integration in itself introduces a large number of additional points that require investigation as part of an Audit

Previously connection to the z/OS environment was often seen as out of scope of an audit as it was limited by the requirement of physical access to a 3270 terminal connected via a mainframe only network.

SNA architecture still exists, including LU6.2 for communication between applications. Some sites will still have terminals connected to the SNA network and the parameters for this section of the network will have to be checked. With this no longer being the predominate method of network connection to the mainframe there are number of other network architectures that should be checked,

The introduction of TCP/IP on the z/OS platform introduced a method of connecting to a mainframe session outside of the physical control offered by SNA, and in so doing brought a requirement to audit TCP/IP into the mix.

TCP/IP connections are not limited by geographical location meaning that users are now able to connect (when authorised) to a mainframe session from anywhere on Earth using a terminal emulation program. WebSphere, IBM's Web Application Server introduced an additional method to work with CICS, meaning that users did not even have to have knowledge of CICS transaction protocols as web applications can be used to drive CICS transactions.

With USS telnet is also available and the ability to access datasets from within a USS session must also be controlled. We all know how easy it is to issue an erroneous delete command, only to have to spend the next hour recovering the dataset having discovered that it, or at least part of it, is system critical.

All of these additional methods of communicating with the mainframe need thorough examination during an audit. Part of this examination should also be to ensure

that communication across any part of the network should be encrypted, not just the users ID and password, but also data handled by CICS is often business critical and may contain customer information.

Auditing CICS 101

There is no simple way to Audit a CICS installation. Any CICS audit requires a review of many aspects of the operating system. This is mainly due to the fact that CICS is a vastly configurable software package, with the ability to be linked to so many different data storage methods and accessible through any method that you can attach to the z/OS platform.

With this in mind any introduction to auditing CICS will be purely an overview. Any recommendations within this document should not be treated as the final answer for your installation until you have confirmed with more detailed documentation.

The initial part of any audit is data gathering, and a CICS audit is no exception. Unfortunately with CICS this stage involves the gathering of information on CICS itself, together with all of the associated tasks that CICS communicates with.

Initial things to look at when auditing CICS are the JCL used for the started task. This will contain details of the files that contain the settings that are used to start the CICS task.

One of the files referenced in the JCL will be the System Initialization Table (SIT). This contains the settings that are used to get the CICS task running in the first place and are therefore very important with regard to how the task is set up.

Auditing CICS - A Beginners Guide

This chapter will provide an explanation of basics, best practices, industry standards and newer functionality from an auditor's perspective.

Where to look and what to look for

CICS is a product which is designed to be used in multiple different ways according to each customer's specific requirements. Because of this flexibility it can be very difficult just to track down what is being used to configure the shape of CICS. This chapter addresses the most basic part of auditing CICS on z/OS - identifying where to find the information that needs to be audited! It focuses on RACF except for one later chapter which looks at differences for CA ACF2 and CA Top Secret.

Job Control

Job Control Language or JCL is used to explain to the operating system what requirements each individual CICS Region has - for example: files, programs, security. It does this by communicating those requirements when the job or, more usually, started task (STC) is submitted to the system. The CICS start up creates a z/OS address space which has all of the required resources available to it.

The location of JCL is not fixed. Each implementation can be different and the best way to find out where it is held is to ask the local staff. It is possible to work backwards from a STC or job which is currently running but this still may not give the whole picture. JCL for different CICS regions may have different locations. Ask the CICS Systems Programmers. They should know!

Here's an example of what you might see with a detailed explanation. It is based on the IBM supplied sample with some alteration for the sake of clarity.

```
// starting in column 1 indicates that this is a JCL statement.
//* indicates that this line is a comment.

//*****
//* PROC Symbolics *
//*****
//DFHSTART PROC START='AUTO',
// INDEX1='CICSTS32.CICS',
// INDEX2='CICSTS32.CICS',
// REGNAM='TR',
// REG='64M',
// DUMPTR='YES',
// RUNCICS='YES',
// OUTC='*',
// SIP=T
//*
//***** EXECUTE CICS *****
//CICS EXEC PGM=DFHSIP,REGION=&REG,TIME=1440,
// COND=(1,NE,CICSCNTL),
// PARM='START=&START,SY SIN'
//*
//SYSIN DD DISP=SHR,DSN=&INDEX1..SYSIN(DFHESIP&SIP)
//DFHMACD DD DSN=@dsindex@.DFHMACD,DISP=SHR
//*****
//* THE CICS STEPLIB CONCATENATION
//*****
//STEPLIB DD DSN=&INDEX2..SDFHAUTH,DISP=SHR
```

CICS Symbolics

Values assigned here are populated into the JCL at run time. In the body of the JCL you can see where the variables are **referenced** because the variable is prefixed with &

Changing these values can fundamentally change the CICS system that is brought up.

Auditors should establish that there is appropriate RACF control of all datasets specified in the whole JCL.

CICS startup and SIT input parameters

DFHSIP is the program that is run to initialise CICS.

Auditors should verify that the correct //SYSIN dataset is used. Pointing to the wrong one can change the shape of CICS security.

CICS STEPLIB Concatenation

If any of the libraries are in the z/OS linklist they don't need to be specified in STEPLIB.

Auditors should confirm minimum access to any dataset specified.

CICS DFHRPL Concatenation

CICS Relocatable Program Libraries contain all of the programs which might be executed in a particular CICS region.

Auditors should confirm the library containing the CICS startup JCL has UACC(NONE) so unauthorised users can not change this list.

IBM provides a sample with a number of entries pointing out the type of libraries that might be needed. Unless they are uncommented they will not be included.

@name@ is a place holder not a real name.

CICS System Datasets

The vast majority of these will be pre-defined z/OS datasets. They are specified like this in the JCL so that a DD name can be associated with them.

The DD name is the 1-8 characters following // in column 1 and then followed by the characters DD after a minimum of 1 space.

This DD name is then used for how CICS does most of its communication with them. But RACF protection is at the real dataset name level.

These should be defined to RACF with UACC(NONE). Very limited numbers of users should have any higher access. The CICS region userid will need to have UPDATE access at least. But basic CICS users don't need any.

CICS Dump Utility Program

During CICS shutdown, if all has gone successfully in the close, the dump utility will be run 3 times against the 2 dump datasets and the auxiliary trace dataset.

```
//          DD DSN=&INDEX2..SDFJAUTH,DISP=SHR
// *       DD DSN=@SCEERUN@2,DISP=SHR
// *       DD DSN=@SCEERUN@,DISP=SHR
// *****
// *       THE CICS LIBRARY (DFHRPL) CONCATENATION
// *****
//DFHRPL   DD DSN=&INDEX2..SDFHLOAD,DISP=SHR
// *       DD DSN=@sceecics@,DISP=SHR
// *       DD DSN=@sceerun@2,DISP=SHR
// *       DD DSN=@sceerun@,DISP=SHR
// *       DD DSN=@scsoload@,DISP=SHR
// *       DD DSN=@scsqanle@,DISP=SHR
// *       DD DSN=@scsqcics@,DISP=SHR
// *       DD DSN=@scsqauth@,DISP=SHR
// *       THE AUXILIARY TEMPORARY STORAGE DATASET
//DFHTEMP  DD DISP=SHR,DSN=&INDEX1..CNTL.CICS&REGNAM.. DFHTEMP
// *       THE INTRAPARTITION DATASET
//DFHINTRA DD DISP=SHR,
//          DSN=&INDEX1..CNTL.CICS&REGNAM..DFHINTRA
// *       THE AUXILIARY TRACE DATASETS
//DFHAUXT  DD DISP=SHR,DCB=BUFNO=5,
//          DSN=&INDEX1..CICS&REGNAM..DFHAUXT
//DFHBUXT  DD DISP=SHR,DCB=BUFNO=5,
//          DSN=&INDEX1..CICS&REGNAM..DFHBUXT
// *       THE CICS LOCAL CATALOG DATASET
//DFHLCD   DD DISP=SHR,DSN=&INDEX1..CICS&REGNAM..DFHLCD
// *       THE CICS GLOBAL CATALOG DATASET
//DFHGCD   DD DISP=SHR,DSN=&INDEX1..CICS&REGNAM..DFHGCD
// *       THE CICS LOCAL REQUEST QUEUE DATASET
//DFHLRQ   DD DISP=SHR,DSN=&INDEX1..CICS&REGNAM..DFHLRQ
// *       EXTRAPARTITION DATASETS
//DFHCXRF  DD SYSOUT=&OUTC
//LOGUSR   DD SYSOUT=&OUTC,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//MSGUSR   DD SYSOUT=&OUTC,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//CEEMSG   DD SYSOUT=&OUTC
//CEEOUT   DD SYSOUT=&OUTC
// *       THE DUMP DATASETS
//DFHDMPA  DD DISP=SHR,DSN=&INDEX1..CICS&REGNAM..DFHDMPA
//DFHDMPB  DD DISP=SHR,DSN=&INDEX1..CICS&REGNAM..DFHDMPB
//SYSABEND DD SYSOUT=&OUTC
//SYSPRINT DD SYSOUT=&OUTC
//PRINTER  DD SYSOUT=&OUTC,DCB=BLKSIZE=121
// *       THE CICS SYSTEM DEFINITION DATASET
//DFHCSD   DD DISP=SHR,DSN=&INDEX1..DFHCSD
// *       EXECUTE DUMP UTILITY PROGRAM TO PRINT THE
// *       CONTENTS OF THE DUMP DATASET A
//PRTDMPA  EXEC PGM=DFHDU640,PARM=SINGLE,
//          REGION=0M,COND=(1,NE,DTCNTL)
//STEPLIB  DD DSN=&INDEX2..SDFHLOAD,DISP=SHR
//DFHTINDX DD SYSOUT=&OUTC
//SYSPRINT DD SYSOUT=&OUTC
//DFHPRINT DD SYSOUT=&OUTC
//DFHDMPDS DD DISP=SHR,DSN=&INDEX1..CICS&REGNAM..DFHDMPA
//SYSIN    DD DUMMY
// *       EXECUTE DUMP UTILITY PROGRAM TO PRINT CONTENTS
//PRTDMPB  EXEC PGM=DFHDU640,PARM=SINGLE,REGION=0M,
//          COND=(1,NE,DTCNTL)
//STEPLIB  DD DSN=&INDEX2..SDFHLOAD,DISP=SHR
//DFHTINDX DD SYSOUT=&OUTC
//SYSPRINT DD SYSOUT=&OUTC
//DFHPRINT DD SYSOUT=&OUTC
//DFHDMPDS DD DISP=SHR,DSN=&INDEX1..CICS&REGNAM..DFHDMPB
//SYSIN    DD DUMMY
//DFHAUXT  DD DISP=SHR,DSN=&INDEX1..CICS&REGNAM..DFHAUXT
//DFHAXPRT DD SYSOUT=&OUTC
//DFHAXPRM DD DUMMY
//PRTBUXT  EXEC PGM=DFHTU640,REGION=0M,COND=(1,NE,DTCNTL)
```

```
//STEPLIB DD DSN=&INDEX2..SDFHLOAD,DISP=SHR
//DFHAUXT DD DISP=SHR,DSN=&INDEX1..CICS&REGNAM..DFHBUXT
//DFHAXPRT DD SYSOUT=&OUTC
//DFHAXPRM DD DUMMY
```

You can see that it might be easier to look at the JCL for a running CICS region which will already have all of the variables resolved. Certainly my advice would be to do so but also to make sure that all source libraries containing CICS startup JCL are adequately protected by RACF during any audit.

Another audit point is users (and other tasks like automation) who are authorised to start and stop CICS. There are multiple ways of achieving some level of lock-down; amongst the most common is to use RACF to control who can issue what commands at the z/OS and JES level.

Associated Userid

When you start a CICS region in a z/OS environment that has RACF installed, the task is associated with a userid, referred to as the CICS region userid. The authority associated with this userid determines which RACF protected resources the CICS region (rather than the users of that region) can access.

Each CICS region, for either production or test use, should be subject to normal RACF data set protection based on the region userid under which the CICS region executes. You specify the region userid under which CICS executes in one of three ways:

As a started task:

- In the RACF started procedures table, ICHRIN03
- In a STARTED general resource class profile, on the user parameter of the STDATA segment

As a job:

- On the USER parameter of the JOB statement when you start CICS as a JOB

It is recommended that the CICS userid does not match the high level qualifier (HLQ) of any datasets used in the CICS region JCL or Procedure. If the userid and HLQ match, then the region has ALTER access via RACF processing to those datasets.

This becomes relevant when developers or systems programmers submit batch jobs via CICS transactions. The RACF class PROPCNTL is used to control the propagation of the CICS region userids to such jobs. Without PROPCNTL in use, such jobs may damage CICS system datasets or access other resources.

For this reason it is recommended to control the access of the CICS region userids, including access to CICS transactions and profiles in other RACF classes.

Datasets

Dataset is the name given to a file in z/OS.

These datasets contain things like programs and data and must be protected using RACF. The JCL that is used to start each CICS region will contain the information needed to find all of the fixed datasets associated with that region. More datasets can be added dynamically.

Although, in general, CICS runs in unauthorized z/OS state, a number of programs do need to run in authorized state for part of their execution. In order to allow for this, the following suffixed CICS libraries must be identified to z/OS as APF authorised:



Note: The “trusted” or “privileged” attributes should **never** be allocated to the CICS userid.

Note: To ensure the authorizations for different CICS services are properly differentiated, each should run with a unique region userid, e.g. the userid under which you run Production CICS to process payroll and personnel applications should be the only CICS userid authorized to access production payroll and personnel data sets.

Note: This userid is also used as the prefix to resource names if SECPREFX=YES is specified.



Note: The source statements of application programs, JCL and tables are also sensitive and you should make certain that RACF protects the datasets containing them.

Library suffix

SDFHLINK
SDFHAUTH

To prevent unauthorized or accidental modification of any of the CICS system level data, you must make sure that all of the installation datasets (including SDFHLOAD and any libraries in //STEPLIB or the DFHRPL concatenation) are appropriately RACF protected. Without such protection, the integrity and security of the CICS system are at risk.

STEPLIB/STPCAT

This JCL parameter allows an installation to override specific files at CICS start up. If used, STEPLIB and STEPCAT are identified in the CICS initialization JCL with either //STEPLIB DD or //STPCAT DD starting in column 1 of the deck.

//STEPLIB is used to override where to find the CICS initialization programs and/or any subsequently called programs.

//STPCAT changes the catalog which is used to find the required files.

Neither of these options should be used for “normal” CICS start up. If you see them used in JCL for anything other than test CICS regions then you need to ask questions to establish why.

One last point about STEPLIB/STPCAT. If an unauthorized library is concatenated with any authorised libraries, the authorised libraries will lose their APF status.

Journals and Logs

A CICS *journal* is a set of special purpose, sequential files. Journals can contain any and all data that CICS needs to facilitate subsequent reconstruction of events or data changes. This is how CICS handles recovery processing. For example, a journal might act as an audit trail, a change file of database updates and additions, or a record of transactions that are passing through the system (often called a *log*).

Journals are fundamental to the recoverability of transactions. In particular, CICS uses the system journal to log transaction commit processing and syncpoint data so that CICS can recover all necessary recoverable resources in the event of a CICS or a transaction failure.

Before considering journaling in detail, an installation needs to review the different facets of CICS logging and recovery in order to clarify the reasons for logging.

There are a number of different activities which can be logged during routine operation of CICS outside of the recoverability ones. As a minimum, an installation can, and should, log all sign-on and sign-off activity to SMF, including any invalid or unsuccessful sign-on attempts. You can only understand the logging of unsuccessful sign-on attempts by also recording successful sign-ons. For example, if a user makes one or two unsuccessful attempts followed immediately by a successful sign-on, the unsuccessful sign-ons might be interpreted as being caused by keying errors at the terminal. However, several unsuccessful attempts for a variety of userids which occur within a short space of time, and without any subsequent successful sign on activity being recorded, may well be cause for a security concern that needs investigation.



Dynamic transaction backout

If CICS abnormally terminates a transaction, all changes that the transaction makes to a recoverable resource, such as a recoverable temporary storage queue, must be backed out to the state that existed before the transaction started. This is known as dynamic transaction backout (DTB).

Recovery after a system abnormally terminates

Recovery after a system abnormally terminates makes sure that all recoverable resources and all prepared transactions are restored to their pre-failure state, before the system resumes normal operation.

For CICS this is a special case of the more general problem of recovering the state of partially finished transactions. In principle, CICS records any change that is made to a recoverable resource in the system journal as part of the two-phase commit processing so that the change can be committed from that point onward. It therefore follows that, during normal operation, CICS only writes to the system journal, thereby allowing CICS transactions to uphold their guarantees.

During recovery processing (at startup after your CICS system abnormally terminates), CICS processes the system journal to re-prepare all transactions that were in-flight at the time of the crash. CICS recovery processing reads the system journal to obtain a list of active transactions, and subsequent processing plays back the appropriate records.

CSD

The CICS System Definition or CSD file is a VSAM file that contains all of the resource definitions loaded by CICS during an initial or cold start or interactively when requested. Resource Definition Online (RDO) is the recommended method of defining resources to CICS.

There are three major transaction groups in RDO:

- CEDA allows users to modify both the active CICS system and the CSD
- CEDB allows modification of the CSD (all but INSTALL) and read-only commands
- CEDC allows only read-only commands

Resource definitions are created interactively with the CEDA transaction, or by using the batch utility DFHCSDUP. Both methods store the definitions in the CICS system definition data set (CSD).

At CICS initialization, CSD definitions are selectively installed as CICS system tables, controlled by a user supplied list of definitions. CEDA defined resource definitions can be installed while CICS is active and used immediately. The list of definitions to be loaded from the CSD is obtained from the list of groups defined on the GRPLIST system initialization parameter with each resource connected to a single group.

To control the addition of resource groups to the CICS start-up group list, you should use the CEDA or DFHCSDUP LOCK command to lock the list. This protects the group list from unauthorized additions. Also, lock all the groups that are specified in this list.

These resource definitions flesh out the shape of CICS as defined by the SIT parameters. Standard resource definitions will include entries for various resources such as terminals, files, network connections etc.

System Initialization Parameters

The System Initialization Table or SIT contains the parameters which shape CICS and can be grouped into three basic types:

1. Information used to initialize and control CICS system functions
2. Module suffixes used to load your own versions of CICS control tables
3. Special information used to control the initialization process

The primary method of providing system initialization parameters to CICS is with a system initialization table (SIT). The main parameters of the SIT, a table that you assemble into a load module, supply the system initialization program with most of the information necessary to initialize the system to suit your unique environment. You can generate more than one SIT, and at the time of system initialization select the one that is appropriate to your needs.



This is the reason that auditors must be able to verify which parameters have been used each time a CICS region is started.

The actual number varies from CICS release to CICS release but there are generally around 300 SIT parameters available. This allows for massive flexibility in the implementation of CICS but makes it very complicated to audit. The SIT parameters which directly affect the security of a CICS region are described in the next chapter - SIT Settings.

Override Parameter Settings

You can specify which SIT you want to use to start a CICS region and other system initialization parameters (with a few exceptions), in any of three ways:

JCL changes

1. In the PARM parameter of the EXEC PGM=DFHSIP statement
2. In the SYSIN data set defined in the startup job stream

z/OS Console commands

3. Through the system operator's console

You can also use these methods of input to the system initialization process to override most of the system initialization parameters assembled in the SIT. In fact this is quite common.

The base line parameters are usually assembled into a loadable SIT while specific region requirements will be applied via override. This can take the shape of changes to the start up JCL or INCLUDE statement(s) in the JCL so that changes can be applied without the need to change the JCL.

It is more unusual to override SIT parms from the z/OS console. Typically this would only happen during disaster recovery processing or new release testing.

All methods used to make updates should be secured using RACF, e.g. TSO and UPDATE to the dataset and/or the ability to issue z/OS commands.

SIT Settings



IBM specifically talks about CICS RACF set up in a number of places. However their documentation often only addresses parameters which directly affect RACF profiles. In reality, there are many more parameters which affect the security of the CICS service. This section will talk about all of the parameters which affect CICS security, i.e. the required audit points.

Comment will be made where there is a preferred setting for a SIT parameter from an audit perspective. The default values, if any, have been underlined.

The logos in the margin of this chapter show which software products are involved as well as where I have specifically seen problems during Sarbanes Oxley and SAS 70 audits at customer sites. This does not imply that these are ALL the parameters which will be considered by these audits nor that this book should be your sole source of information for auditing CICS.

CMDSEC

CICS command security controls the sub-commands of supplied transactions CEMT and CECL, respectively the CICS Master Terminal and the CICS Command Interpreter. The commands implemented by these transactions may also be issued by application program transactions using the EXEC CICS interface and are subject to RACF checking when called this way. Usually these commands are used by systems programming or development personnel to dynamically control the configuration of the CICS system or applications. As such, these commands can be powerful and are considered sensitive enough to warrant an additional level of security above basic transaction security. There are two options on this parameter:

ALWAYS

CICS overrides the CMDSEC option, and always calls its command security checking routine to issue the appropriate call to the SAF interface.

ASIS

CICS calls its command security checking routine only when CMDSEC(YES) is specified in a transaction resource definition.

CONFDATA

CONFDATA does not typically show up as a security parameter. It has been included here because a number of organisations have had problems with Sarbanes Oxley Auditors over its use.

The parameter is used to say whether an organisation wants user data to appear in any trace entries or dumps that might be taken against the CICS region. This data is used to resolve problems but specifying SHOW puts data into the stream which isn't strictly necessary to debug most issues but which could be used to hack the systems. There are two options for this parameter:

HIDETC

CICS is to 'hide' user data from CICS trace entries. It also indicates that VTAM RAIAs (Receive Any Input Area) are to be suppressed from CICS dumps.

SHOW

Data suppression is not in effect. User data is traced regardless of the CONFDATA option specified in transaction resource definitions. And this is the setting that SoX Auditors have the problem with.

CONFTXT

Is related to CONFDATA but not affected by it, as it too concerns data which could be used to hack the systems. The parameter allows an organisation to define whether it wants user data to be included in any VTAM traces that are run against the CICS region. There are two options for this parameter:

YES

CICS prevents VTAM from tracing user data.

NO

CICS does not prevent VTAM from tracing user data.

Note: XCMD is the related parameter which implements the RACF classes.

Sarbanes-Oxley
Financial and Accounting Disclosure Information



Note: XEJB is the related parameter which implements the RACF class.



Sarbanes-Oxley
Financial and Accounting Disclosure Information

DFLTUSER

This parameter allows an organisation to specify a userid to be used in a number of “default” situations such as if users are not required to sign on. If DFLTUSER is not specified it will be set to the value CICSUSER.

The DFLTUSER should always have no or very low levels of access to the CICS region.

EJBROLEPRFX

This parameter can be extremely useful if an installation is running multiple CICS Web Server instances with differing security requirements. It is used to specify a prefix that is used to qualify the security role defined in an enterprise bean's deployment descriptor as defined within RACF.

ENCRYPTION

This SIT parameter specifies the level of encryption to use for TCP/IP connections using the secure sockets layer. Possible values are:

STRONG

Specifies a 128 bit encryption key. This option used to be available when you had installed the North American encryption feature, which was available in the USA and Canada only. However, it can now be used across the world.

MEDIUM (or its older equivalent NORMAL)

Specifies a 56 bit encryption key, which is available worldwide.

WEAK

Specifies a 40 bit encryption key, which used to be the only option available in France.

ESMEXITS

This parameter allows you to specify additional processing to be carried out whenever a call is made to the external security manager, i.e. RACF (CA ACF2 or CA Top Secret). An installation can use ESMEXITS to specify whether they want CICS to pass installation data for use by the RACF installation exits.

Some third party applications which run under CICS make use of the installation data field in the RACF userid to make further security decisions. Local staff will know if there are any applications that work this way. If not, the setting for this parameter must either be left to default or be set to NOINSTLN.



There is no right or wrong for this setting from an audit perspective. It will depend on whether there is a defined requirement within an application for RACF installation data.

There are two available options for this parameter:

NOINSTLN

User installation data is not sent through to RACF. This is the default.

INSTLN

CICS-related and installation-supplied data is passed to RACF. This data is intended for use in exits written for RACF.

GMTRAN

Again, we come to a parameter which is not typically thought of as a security related one. However, the impact of the setting is to drive CICS users to a specific first transaction, i.e. the “**good morning transaction**”.

It is vital that any audit checks this value to make sure it complies with the local standard. There are three possible values:

Sarbanes-Oxley
Financial and Accounting Disclosure Information

CESN

The CICS signon transaction which also displays the text specified in the GMTEXT parameter.

CSGM

Displays the text specified in the GMTEXT parameter. This is the default.

transaction-id

The transaction must be capable of being automatically initiated (ATI).

KEYRING

The KEYRING parameter comes into play only when CICS has been configured to use SSL, which is often done to enable CICS Web Support. The SSL service on z/OS must also be activated for CICS to be able to use it.

SSL is employed to provide for secure transfer of data over an insecure network. The parameter supersedes the earlier KEYFILE SIT parameter, the use of which was removed at CICS TS 2.2, where digital certificates were stored in an HFS file rather than within RACF. All digital certificates, for use with CICS, must now be stored within an external security manager.

This is not an on/off parameter. The value supplied must be the name of a key ring held within RACF. The CICS associated user must have authority to read keys held in the ring specified.

PLTPIUSR

When a CICS region starts up it has the ability to run programs before becoming available for general work. Start up programs for IBM or ISV products like Omegamon and Abend-aid are examples of when this might be used. "Normal" CICS security isn't in operation yet as CICS isn't "fully up" so these PLTPI programs are essentially "batch" programs.

If you do not specify the PLTPIUSR parameter, CICS runs PLTPI programs under the authority of the associated CICS region userid and no surrogate (RACF SURROGAT class profiles) checking will be done. From an audit perspective this would make it very difficult to track down where/when secondary transactions are instigated.

PLT programs are run under the CICS internal transaction, CPLT. Before the CPLT transaction is attached, CICS performs a surrogate user check against the CICS region userid (the userid under which the CICS region is executing). This is to ensure that the CICS region is authorized as a surrogate for the userid specified on the PLTPIUSR parameter. This ensures that you cannot arbitrarily specify any PLT userid in any CICS region - each PLT userid must first be authorized to the appropriate CICS region.

PLTPISEC

When an installation is running any PLTPI programs, this parameter is used to define what attributes to operate with. PLTPISEC works hand in hand with the previous parameter, PLTPIUSR. If PLTPISEC is used then PLTPIUSR should also be specified otherwise the CICS region id will be used for all PLT programs.

There are four possible values:

ALL

You want CICS to perform both command and resource security checking.

CMDSEC

You want CICS to perform command security checking only.

RESSEC

You want CICS to perform resource security checking only.

NONE

You do not want any security checking on PLT initialization programs.

Note: Using the PLTPISEC=CMDSEC, RESSEC or ALL options can significantly degrade CICS startup performance.

Note: If you require DL/I security checking, you must specify the XPSB system initialization parameter as XPSB=YES or XSPB=name.

Note: Using the RESSEC=ALWAYS option can significantly degrade performance

Note: When CICS is being initialized, it requests RACF bring resource profiles into main storage for all the resource classes that specify YES in the system initialization parameters. Except for XAPPC and XDB2, YES is the default in the system initialization parameters and CICS will use the default class-names, e.g. GCICSTRN. So you need to supply RACF profiles for all those resources for which you do not specify Xname=NO explicitly.

If CICS tries to load a general resource class that does not exist or is not correctly defined, it issues a message indicating that external security initialization has failed, and terminates CICS initialization.



auditing CICS

PSBCHK

This parameter is only seen in installations that employ the use of IMS from CICS. In a case where IMS is used, the organisation can choose to specifically secure it using RACF. Under these circumstances it is vital that any security authentication request comes to IMS from the initiating userid rather than the CICS region associated id. There are two possible values:

YES

The remote terminal is checked if RESSEC(YES) is coded in the definition of the transaction in the CSD.

NO

No check is made against the remote terminal. This is the default.

RESSEC

The RESEC SIT parameter can be used to force security checking to a greater depth than that which comes with basic transaction level security.

For most simple (or single function) transactions, this extra layer of security is not strictly necessary. For example, if the transaction is designed to enable the terminal user to update a personnel file and nothing else, it is enough to authorize access to the transaction without controlling access to the file too.

But, if you have complex transactions offering users a choice of functions, or you are unsure about all the options available within a transaction, you may want to add the extra layer of security to restrict access to the data as well as to the transaction. The vast majority of installations using CICS have complex transactions in use. The appropriate RACF classes need to be activated using other SIT parameters.

In order to fully understand the impact of this parameter will require cooperation with both the CICS Systems Programmers and Applications Developers. There are two options available for this parameter:

ALWAYS

CICS overrides the RESSEC option, and always calls its resource security checking routine to issue the appropriate call to the RACF. Use this option only if you need to control or audit all accesses to CICS resources.

ASIS

CICS honors the RESSEC option defined in a transaction's RDO definition. CICS calls its resource security checking routine only when RESSEC(YES) is specified in the RDO transaction definition.

SEC

This parameter supplies the primary security setting for the CICS region. You use the SEC system initialization parameter to "switch on" RACF (CA ACF2 or CA Top Secret) for your CICS region. There are only two options:

YES

This means that RACF is "switched on", and control of CICS security is determined by the other security-related SIT options.

NO

Specifying NO means that there is no security checking of whether users are allowed to access CICS (and non-CICS) resources from this region, and that sign on cannot take place.

SECPRFX

This parameter is extremely helpful if the installation is running multiple CICS regions with differing security requirements. You use the SECPRFX system initialization parameter to specify whether you want CICS to prefix the resource

names that it passes to RACF for authorization. The prefix that CICS uses is the RACF userid under which the CICS region is running.

The prefix allows the security administrators to easily segregate the access of separate regions.

There are only 2 options on the SECPRFX parameter:

YES

CICS prefixes all resource names with the CICS region userid when talking to RACF.

NO

CICS doesn't prefix resource names in requests it passes to RACF from this region.

SECPREFIXID

This parameter is useful in conjunction with the XFCT one. It allows an installation to specify an alternative prefix that the server is to use for security checks on coupling facility data table access by CICS regions, instead of the server region user ID. The parameter is coupled with XFCT. The value specified must be between 1 and 8 characters and is used as the high level qualifier in any FCICS-FCT profiles.

SNSCOPE

At the heart of this parameter is whether a single userid can have multiple active sessions with a single CICS region, or within an MVS image or sysplex. It is effectively a compatibility option from when CICS did not limit how many times a user could sign on.

The world that we operate in now should not (normally) be allowing multiple signons from the same userid. If an individual requires multiple CICS sessions then they should be assigned multiple userids. This would make the situation controllable. Allowing multiple signons leaves a door for hackers to break into an organization's systems because a user would not know if their id was already in use when they logged on.

SNSCOPE is restricted to users signing on at local terminals, or signing on after using the CRTE transaction to connect to another system. There are four possible values for the parameter.

SYSPLEX

Each userid can be signed on once only, and to only one of the set of CICS regions within an MVS sysplex that also specify SNSCOPE=SYSPLEX. A signon is rejected if the user is already signed on to another CICS region in the same MVS sysplex. This is the most secure option.

MVSIMAGE

Each userid can be signed on once only, and to only one of the set of CICS regions in the same MVS image that also specify SNSCOPE= MVSIMAGE. A signon request is rejected if the user is already signed on to another CICS region in the same MVS image.

CICS

Each userid can be signed on once only in the same CICS region. A signon request is rejected if the userid is already signed on to the same CICS region. However, the userid can be used to signon to another CICS region in the same, or another, MVS image.

NONE

Each userid can be used to sign on for any number of sessions on any CICS region. This is the compatibility option, providing the same signon scope as in releases of CICS before CICS Transaction Server for OS/390 Release 3.



TCPIP

This parameter specifies whether the CICS TCP/IP listener service is to be activated at CICS startup. The default is NO, meaning that these services cannot be enabled.

TCPIP must be set to YES so that HTTP and IIOF services can process work.

TCPIPSERVICE resource definitions must be provided to define each active port and the type of service associated with it. The CICS TCP/IP listener is activated for the specified ports when the TCPIPSERVICE is installed, if TCPIP (YES) has also been specified.

You can change the resolver configuration of CICS either by altering system TCP/IP configuration files, or by adding or changing the DD name SYSTCPD in the CICS start up JCL. This sets the RESOLVER_CONFIG environment variable to the MVS dataset you have specified. For this reason, access to the SYSTCPD dataset should be firmly controlled using a RACF dataset profile. There are only 2 options on the SECPRFX parameter:

NO

CICS doesn't start the TCP/IP listener.

YES

CICS starts the TCP/IP listener.

Deciding what is appropriate will be dependant on what services the CICS region is expected to be used for. Again, cooperation with the CICS Systems Programmers will be useful.

USRDELAY

Another parameter which is not normally considered to be a security related one here. USRDELAY allows an installation to set a timer on user persistence (how long the user's security details will remain available to CICS). It specifies the maximum time, in the range 0 through 10080 minutes (up to 7 days), that an eligible userid and its associated attributes are to be retained in the user table if the userid is unused. The default is 30 minutes.

USRDELAY can also help to improve system performance in CICS multi-region operation, where many CICS environments communicate to provide a unified business function. USRDELAY is designed to facilitate persistent logon from one region to another (remote) region and hence avoid the need to revalidate the user ID/password for each subsequent transaction. It works by caching the user ID credentials at the remote region. This functionality is exploited by products like MQSeries for example.

You should be aware that high values of USRDELAY may affect your security administrator's ability to change the authorities and attributes of CICS users, because those changes are not reflected in CICS until the user instance is refreshed in CICS by being flushed from the timeout queue after the USRDELAY interval.

Some audits may require you to specify USRDELAY=0. This still allows **some** sharing of user instances if the usage count is never reduced to zero. Generally, however, remote users are flushed out immediately after the transaction they are executing has terminated, so that their user control blocks have to be reconstructed frequently. This results in poor performance but potentially better security control.



XAPPC

Application Peer to Peer Communication or APPC is a proprietary networking protocol which allows programs to communicate with programs on other systems with compatible communications support. It is a forerunner to TCP/IP in the z/OS environment but is still used in CICS environments.

In an APPC environment, when a user or application on one system requests access to another system, the two systems set up a session. To establish the session, the systems must link two matching APPC device descriptions.

Bind level security is controlled using RACF profiles in the APPCLU class. It is effectively a form of user ID surrogacy - when only bind level security is in place between CICS regions, the userid originating the transaction isn't checked for access to transaction resources at the remote CICS region where the transaction **actually** runs. Rather, the user ID specified on the link between the two regions is used to access resources.

This means that if the user is allowed to initiate the transaction at the originating region, we trust the transaction to be well behaved and any subsequent transactions it invokes won't require the original user to have access.

This is a common - if not entirely secure - method of establishing security in multi region CICS environments. It avoids the necessity of defining all potential user access across all CICS regions. Typically the user IDs associated with the link (or the CICS default user if none specified) will be granted access to all region transactions and access will occur under the link user ID.

The XAPPC parameter enables APPC partner-LU verification a.k.a. RACF LU6.2 bind-time (also known as APPC) security. It is not a commonly implemented RACF class (APPCLU). There are only two available values:

YES

RACF session security can be used.

If you specify BINDSECURITY=YES for a particular APPC connection, a request to RACF is issued to extract the security profile. If the profile exists, it is used to bind the session.

NO

RACF session security cannot be used.

XCMD

Just like the more familiar transaction security, command security is implemented in RACF by grouping and member classes. By default these are the VCICSCMD (grouping) and CCICSCMD (member) classes.

The XCMD parameter specifies whether you want CICS to perform command security checking, and optionally the RACF resource class name in which you have defined the command security profiles - it is related to the CMDSEC parameter described previously.

CICS command security controls the use of system programming (SP) commands such as CEMT with INQUIRE, CREATE, DISCARD, PERFORM and SET. Varying levels of RACF authorization are required to each of the commands depending on the required action. Regardless of this, the user must also have authority to run the CEMT transaction.

When an alternate class to the default is desired the XCMD parameter is set to the name of the RACF-defined member class you want to use where the first character of the class name isn't specified, only a suffix is used. The first character of a CICS command security member class is always C. Likewise the first character of the grouping class is always V. So if you want to use a custom class pair of CCICSPRD/VCICSPRD, then the XCMD SIT parameter would be specified as XCMD=CICSPRD. The three possible values of XCMD are:

name

CICS calls RACF, using the specified resource class name prefixed by C or



V, to verify that the userid associated with a transaction is authorized to use a CICS command for the specified resource. The member resource class name is Cname and the grouping class name is Vname. The resource class name specified must be 1 to 7 characters.

YES

CICS calls RACF to check whether the userid associated with a transaction is authorized to use a CICS command for the specified resource in the default classes.

NO

CICS does not perform any command security checks, allowing any user to use commands that would be subject to those checks.

As with CICS transactions it's strongly recommended to take advantage of RACF grouping classes for command security. RACF resource names such as PROD_SUPPORT, DBA and SYSPROGS that are relevant to the installation may then be employed to unify the access requirements of different CICS commands and simplify the security administration.

In general, RACF READ-level access to a CICS command implies the capability to view but not change the information managed by the command - UPDATE access is required to change CICS definitions via the commands. The tables below show a typical grouping of these command resources:

Inquiry only

These commands are candidates for grouping under one VCICSCMD resource profile and granting READ access to developers:

BEAN	MVSTCB	STREAMNAME	UOWENQ
CFDTPOOL	RRMS	SUBPOOL	
EXCI	STORAGE	UOWDSNFAIL	

DB2 related commands

These are candidates for a grouping profile and granting UPDATE access to DB2 support staff:

DB2CONN	DB2ENTRY	DB2TRAN
---------	----------	---------

Terminal Monitor and TCPIP-related commands

Can be granted to systems programming staff at UPDATE:

CONNECTION	TCPIP	TSMODEL	VTAM
IRC	TCPIPSERVICE	TSPOOL	WEB
MONITOR	TERMINAL	TSQNAME	

These commands have been superseded or replaced in the current CICS version. If you're running an older CICS release they can be granted to systems programmers at UPDATE:

IRBATCH	JOURNALNAME
---------	-------------

General Tech Support staff

May require access at READ to these, systems programmers should be granted UPDATE:

AUTINSTMODEL	DOCTEMPLATE	MODENAME	TASK
AUTOINSTALL	DSNAME	PARTNER	TCLASS
BRFACILITY	DUMPDS	PROFILE	TDQUEUE
CORBASERVER	ENQMODEL	PROGRAM	TRANDUMPCODE
DELETSHIPED	FILE	REQUESTMODEL	TRANSACTION
DISPATCHER	JOURNAL	SYSDUMPCODE	UOW
DJAR	JVMPOOL	SYSTEM	

Generally the following commands will only be used by systems programmers and UPDATE will be required:

EXITPROGRAM	STATISTICS	TRACEFLAG	TSQUEUE
REQID	TRACEDEST	TRACETYPE	

The following CEMT-related commands also require UPDATE by systems programming staff:

DUMP	LINE	UOWLINK
JOURNALMODEL	PROCESSTYPE	

These next CEMT-related commands have higher sensitivity and require UPDATE. They could be candidates for senior systems staff only, depending on the levels of security segregation your site employs:

FEPIRESOURCE	MAPSET	RESETTIME	SHUTDOWN
LSRPOOL	PARTITIONSET	SESSIONS	TYPETERM

Security-related commands

Generally used by either security admin personnel or systems programmers, UPDATE access is required:

SECURITY

The format of RACF resources for these commands is simply the command name, defined to the member class. If SECPRFX is in use, the command name is prefixed with the CICS region userid.

XDB2

The XDB2 parameter allows installations to protect use of CICS DB2ENTRY resources.

These RDO definitions link the CICS transaction to a DB2 SQL query or other process. Unlike the other Xaaa SIT parameters, XDB2 does not provide a YES option that implies a default CICS resource class name for DB2ENTRY resources. You have to specify your own DB2 resource class name.

There are two options for the parameter:

name

CICS calls RACF, using the specified general resource class name, to check whether the userid associated with the CICS DB2 transaction is authorized to access the DB2ENTRY referenced by the transaction. The resource class name specified must be 1 through 8 characters, it may be have an associated RACF grouping class, and the resource profile name checked is the name of the DB2ENTRY definition in the CICS System Definition file.

NO

CICS does not perform any DB2ENTRY resource security checks.

IBM supplies a sample DFH\$RACF exit with member resource class name of XCICSDB2 with ZCICSDB2 as the grouping class. These classes are not supplied with RACF. So whether your installation chooses to use these defaults or implement their own names, preparation work is needed to RACF.

Most installations now choose to use dynamic class definitions for this purpose although older definitions may not have been migrated to this newer RACF capability.

XDCT

The XDCT parameter allows installations to protect Transient Data Queues. Queues are sequential storage facilities, generally transitory in nature because of



the dynamic nature of transaction processing. They are typically used to process requests or to pass data from one transaction to another. For example print data produced as part of a transaction is usually not printed until well after its task has been completed. The data waits in a queue for the print program to process it when there is no more urgent work to be done.

Popular “in the old days” for mainframe channel attached printing, transient data queues provide general queue functions. It is now often used for offloading work for asynchronous updates with the trigger level set to one.

This facility is not often implemented at customer sites. There are three values which can be specified:

name

CICS calls RACF, using the specified resource class name, to check whether the userid associated with the transaction is authorized to access the specified destination. The member resource class name is Dname and the grouping class name is Ename. The resource class name specified must be 1 through 7 characters.

YES

CICS calls RACF to verify whether the userid associated with the transaction is authorized to access the specified destination.

The member resource class name is DCICSDCT and the grouping class name is ECICSDCT.

NO

CICS does not perform any transient data security checks, allowing any user to access any transient data destination.



XEJB

Enterprise JavaBeans (EJB) was developed by Sun and – in their words – is a managed, server-side component architecture for modular construction of enterprise applications. The EJB specification provides a standard way to implement the back-end 'business' code typically found in enterprise applications (as opposed to 'front-end' user-interface code). Enterprise JavaBeans were intended to handle such common concerns as persistence, transactional integrity, and security in a standard way, leaving programmers free to concentrate on the particular problem at hand.

It can be seen as a “bolt on” to the Java programming language which contains all sorts of frequently used functionality. It represents one of the first truly collaborative approaches taken by IBM to finding increasing relevance for System z in the 21st century. It also represents a huge advance in programming in the internet enabled world we find ourselves in.

Implementation of EJB in a z/OS CICS environment would indicate the use of CICS Web Server and/or other “internet facing” applications. Implementation of RACF control of EJB would most likely indicate a well established z/OS security environment with good controls on security in new application roll out. It is not a commonly implemented parameter yet.

There are only two values which can be specified:

YES

CICS support for security roles is enabled:

- When an application invokes a method of an enterprise bean, CICS calls RACF to verify that the userid associated with the transaction is defined in at least one of the security roles associated with the method.

- When an application invokes the `isCallerInRole()` method, CICS calls RACF to determine whether the userid associated with the transaction is defined in the role specified on the method call.

NO

CICS support for security roles is disabled:

- CICS does not perform enterprise bean method level checks, allowing any userid to invoke any enterprise bean method.
- The `isCallerInRole()` method always returns a value of true.

XFCT

XFCT allows CICS to call RACF to verify that the userid associated with a transaction is authorized to access File Control managed files. If active, such checking is performed every time a transaction tries to access a file managed by CICS File Control.

CICS application programs process files which are logical views of physical VSAM or BDAM datasets on z/OS. A file is identified to CICS by an 8 character file name, and you can define many files to CICS that refer to the same physical data set. For example, you can define file resource definitions called FILEA, FILEB, and FILEC, all of which refer to one physical VSAM data set, but with each file definition specifying different attributes.

CICS transactions access the data in physical data sets using the CICS file control name. Therefore, you control access to CICS managed files by defining profiles in the RACF general resource classes for CICS files, not in the RACF data set class. You define the profiles using the CICS 8 character file name to identify the resource.

Very few organisations implement XFCT control at present but environments with complex transactions should consider it. There are three values which can be specified in this parameter:

name

CICS calls RACF, using the specified resource class name, to verify that the userid associated with a transaction is authorized to access files referenced by the transaction. The member resource class name is *Fname* and the grouping class name is *Hname*. The resource class name specified must be 1 through 7 characters.

YES

CICS calls RACF, using the default CICS resource class name of FCICSFCT, to verify that the userid associated with a transaction is authorized to access files referenced by the transaction. Resources defined to the RACF grouping class of HCICSFCT may have been used in building the actual RACF profiles used for verification.

NO

CICS does not perform any file resource security checks, allowing any user to access any file.

XHFS

XHFS is one of the newer SIT parameters. It is used to specify whether CICS is to check with RACF if the transaction user is authorised to access files in the USS file system. At present, this checking applies only to the user ID of the Web client when CICS Web support is returning z/OS UNIX file data as the static content identified by a URIMAP definition. There are only two values which can be specified:

YES

CICS is to check whether the user identified as the Web client is authorized



to access the file identified by the URIMAP that matches the incoming URL. This check is in addition to the check performed by z/OS USS against the CICS region user ID. If access to the file is denied for either of these user IDs, the HTTP request is rejected with a 403 (Forbidden) response.

NO

CICS is not to check the client user's access to z/OS UNIX files. The CICS region user ID's access to these files is still checked by z/OS USS.

XJCT

The CICS log manager provides facilities to write to and read from:

- The CICS system log
- The CICS general logs, which include user journals, forward recovery logs, and autojournals



The system log is used only for recovery purposes - e.g. during dynamic transaction backout, or during emergency restart. It should not be used for any other purpose. Developers should not, therefore, write to it from a user application using the WRITE JOURNALNAME command. The only way to fully establish if this is the case would be a full examination of all of the application code called from CICS.

CICS uses journal identifier DFHLOG for its primary system log. You should not permit user transactions to write to this.

In addition to the automatic journaling and forward recovery logging that CICS performs for user transactions, applications can also write user journal records.

Users needing to write journal records must have authority to write to the JOURNALNAME. CICS calls RACF to perform a security check only for attempts to access a user journal by a CICS API command, i.e. a user process. CICS does not reference RACF for its internal logging.

The CICS API does not provide a READ command for reading journals from a CICS transaction. For this reason, with proper exercise of control over the installation of applications on your CICS systems, most installations consider it unnecessary to add RACF protection for journals that cannot be read from within CICS.

The WRITE JOURNALNUM command is supported in CICS Transaction Server for z/OS, Version 2 Release 3 for compatibility with earlier releases: the WRITE JOURNALNAME command is preferred for new applications. If resource security applies to a transaction executing WRITE JOURNALNUM, the journal number is prefixed with 'DFHJ' before the security check is applied. Thus, writing to journal number 2 requires UPDATE access to the resource DFHJ02.

If you specify YES, or a RACF resource class name, CICS calls RACF to verify that the userid associated with a transaction is authorized to access the referenced journal. There are three values which can be specified:

name

CICS calls RACF, using the specified resource class name, to verify that the userid associated with a transaction is authorized to access CICS journals. The member resource class name specified must be 1 through 7 characters.

YES

CICS calls RACF, using the default CICS resource class name of JCICSJCT, to verify that the userid associated with a transaction is authorized to access journals referenced by the transaction. Remember that resources defined to the RACF grouping class of KCICSJCT may have been used in building the

actual RACF profiles used for verification, this is the case for any SIT parm that uses RACF grouping classes.

NO

CICS does not perform any journal resource security checks, allowing any user to access any CICS journal.

XPCT

A CICS transaction initiated by a terminal user can start other transactions by means of an EXEC CICS START command. Transactions started in this way are known as **started transactions**. The START command enables a CICS application program to start another transaction associated with a terminal other than the one from which the start command is issued. The XPCT parameter allows an installation to control who has authority to issue these started transactions.

CICS requires a minimum authorization of READ for started transactions. There are three values which can be specified:

name

CICS calls RACF, using the specified resource class name, to verify that the userid associated with a transaction is authorized to use **started transactions** or related EXEC CICS commands. The member resource class name is *Aname* and the grouping class name is *Bname*. The resource class name specified must be 1 through 7 characters.

YES

CICS calls RACF to verify that the userid associated with a transaction is authorized to use started transactions or related EXEC CICS commands.

The member resource class name is ACICSPCT and the grouping class name is BCICSPCT.

NO

CICS does not perform any started task resource security checks, allowing any user to use started transactions or related EXEC CICS commands.

XPPT

You control access to the initial program specified in the transaction resource definition by authorizing the user to initiate the transaction (XTRAN). But CICS application programs can invoke other programs. XPPT would be the parameter to implement if you wanted to control all programs run by an individual user.

Also, the load status of programs can be altered by the CICS RELEASE, ENABLE, and DISABLE commands. However there is no separate security check on the RELEASE of programs loaded for task lifetime. This is done on the corresponding LOAD.

Very few installations implement program control. However, any one with an environment containing complex transactions might like to think about implementing XPPT. There are three values which can be specified:

name

CICS calls RACF, with the specified resource class name, to verify that the userid associated with a transaction is authorized to use LINK, LOAD, or XCTL commands to invoke other programs. The member resource class name is *Mname* and the grouping class name is *Nname*. The resource class name specified must be 1 through 7 characters.

YES

CICS calls RACF to verify that the userid associated with a transaction is authorized to use LINK, LOAD, or XCTL commands to invoke other programs.

Note: There can be performance implications when implementing file control security.

Note: If CICS finds that a program referenced on a LINK command is a remote program, it does not perform the security check in the region in which the link command is issued. The security check is performed only in the CICS region in which the linked-to program finally executes. For example if CICS function ships a distributed program link (DPL) command to CICSB, where the program then executes, CICSB issues the security check. If the DPL request is function shipped again to CICSC for execution, it is CICSC that issues the security check.

Note: *PSBCHK=YES must also be specified if you want full security for PSBs that are accessed in transaction routed transactions. This applies to both remote and DBCTL types of DL/I interface. If you specify PSBCHK=NO, the authority of the remote user is **not used** in transaction routed transactions.*



The member resource class name is MCICSPPT and the grouping class name is NCICSPPT.

NO

CICS does not perform any application program authority checks, allowing any user to use LINK, LOAD, or XCTL commands to invoke other programs.

XPSB

DL/I program specification blocks (PSBs) are IMS control blocks that describe databases and logical message destinations used by an application program. PSBs consist of one or more program communication blocks (PCBs), which describe an application program's interface to an IMS database. Although PSB scheduling requests are sent to IMS for processing, CICS does PSB authorization checking.

The parameter is only relevant to an installation that employs the use of IMS. There are three values which can be specified:

name

CICS calls RACF, using the specified resource class name, to verify that the userid associated with a transaction is authorized to access PSBs. The member resource class name is Pname and the grouping class name is Qname. The resource class name specified must be 1 through 7 characters.

YES

CICS calls RACF to verify that the userid associated with a transaction is authorized to access PSBs. The member resource class name is PCICSPSB and the grouping class name is QCICSPSB.

NO

CICS does not perform any PSB resource security checks, allowing any user to access any PSB.

XRES

XRES is used to secure a number of web service type resources. When active, the RACF profile names for this class consist of 3 parts:

security_prefix.resource_type.resource_name.

security_prefix is the value specified on the SECPRFX SIT parameter.

resource_type specifies the type of CICS resource, such as ATOMSERVICE, BUNDLE, DOCTEMPLATE, EVENTBINDING, JVMSERVER, or XMLTRANSFORM.

Further details on these resources types follow:

ATOMSERVICE

CICS can serve Atom feeds to Web clients. The Atom feeds consist of data that is supplied by CICS resources or application programs. When you expose a CICS resource or application program as an Atom feed or collection, users can read and update the data by making HTTP requests from external client applications, such as feed readers or Web “mashup” applications.

An Atom service document informs clients about the collections that are available from your server. It lists only Atom feeds that you want to make available as collections for editing. It does not include ordinary Atom feeds that are not available for editing.

You normally create only one Atom service document for the collections that are available through a CICS region. The Atom service document is stored in z/OS UNIX System Services.

BUNDLE

A bundle is the unit of deployment for an application. The BUNDLE resource defines where the bundle is deployed on z/OS UNIX and its status.

DOCTEMPLATE

A document template is a complete document or a portion of a document which is created offline, or by a CICS program. Document templates are often used to supply Web pages through CICS Web support, either as static pages provided by a URIMAP definition, or as part of a dynamic Web page created by an application program

EVENTBINDING

The process of specifying business events has a natural workflow, from business manager to application analyst and application programmer. You use the CICS event binding editor to create an event binding that specifies your business events. The event binding specifies the events you want CICS to produce.

JVMSERVER

Java Virtual Machine (JVM) requires a Language Environment enclave. The runtime environment is represented in CICS by a resource called JVM-SERVER. The JVMSERVER resource defines the runtime options for the JVM, including the location of the JVM profile and the maximum number of threads that the JVM supports.

XMLTRANSFORM

The XMLTRANSFORM service allows XML data to be transformed into application data for use in CICS programs.

There are three values which can be specified with the XRES SIT parameter:

name

CICS calls RACF using the specified resource class name prefixed by the letter R, to check whether the userid associated with a transaction is authorized to use the resource it is trying to access. The resource class name is Rname and the grouping class name is Wname. The resource class name specified must be 1 through 7 characters.

YES

CICS calls RACF, using the default CICS resource class name to check whether the userid associated with a transaction is authorized to use the resource it is trying to access. The resource class name is RCICSRES and the grouping class name is WCICSRES.

NO

CICS does not perform any security checks for these resources, allowing any user to access any of them.

XTRAN

This is the most commonly implemented resource protection class in CICS environments across the world. It is used to control who can execute what transactions and, alone, is considered to be a **minimum** security implementation.

The following recommendations are made by IBM. Customers with more TCICSTRN than GCICSTRN profiles defined should be asked to justify this situation:

- Define transactions in the resource group class, GCICSTRN. This minimizes the amount of effort needed to define and maintain transaction profiles and their associated access lists.
- Add RACF GROUPs to access lists not individual users and define access as READ.
- Minimise the use of generic profiles for member (transaction) names unless direct security administration advantage is obtained.

There are 3 possible values associated with this parameter.

name

CICS uses the Tname and Gname user defined resource class profiles for transaction attach security checking. The value has a maximum length of 7 characters. Using different resource classes is an alternate approach, which can also be used in conjunction with profile prefixing (SECPRFX).

YES

CICS calls RACF, using the default CICS resource class name of TCICSTRN, to verify that the userid associated with a transaction is authorized to access the transaction. Resources defined to the RACF grouping class of GCICSTRN may have been used in building the actual RACF profiles used for verification.

NO

CICS does not call RACF to check transaction attach security. As transaction level security is the absolute minimum which should be in place, this value should never be used.

XTST

Temporary storage queues (TSQ) are typically used for shared reading, writing, and updating by multiple transactions; for example, as a scratchpad for shared data.

Transactions can write, update, read, and delete data in a temporary storage queue any number of times until the queue is deleted.

Data stored in recoverable auxiliary storage is retained **after** a CICS region terminates and can be recovered in a subsequent restart. Data stored in *nonrecoverable* auxiliary storage is retained only across a normal shutdown, but not across an immediate shutdown or system failure unless a database is being used as the file manager. Data stored in main storage is not retained across any type of shutdown and so cannot be recovered.

Very few installations use RACF to protect temporary storage queues. There are three values which can be specified:

name

CICS calls RACF, using the specified resource class name, to verify that the userid associated with a transaction is authorized to access temporary storage queues. The member resource class name is *Sname* and the grouping class name is *Uname*. The name specified must be 1 through 7 characters.

YES

CICS calls RACF to verify that the userid associated with the transaction is



Note: There are no CICS parameters that allow you to control transaction attach security at the individual transaction level. When XTRAN is in use CICS issues an authorization request for every transaction. It does this how ever the transaction is started

Note: The **CEBT** transaction (the master terminal transaction used to control the alternate CICS system in an XRF environment) **is not subject to transaction security checking.**



authorized to access temporary storage queues referenced by the transaction. The member resource class name is SCICSTST and the corresponding grouping class name is UCICSTST.

NO

CICS does not perform any temporary storage security checks, allowing any user to access any temporary storage queue.

XUSER

The SIT parameter XUSER activates surrogate user security for a number of different circumstances including DB2 AUTHTYPE checking. It is only relevant for DB2 if:

- DB2 is installed and in use
- DB2 is using external security

When you install a DB2CONN resource definition that specifies the AUTHID, SIGNID, or COMAUTHID attribute, or when you install a DB2ENTRY definition that specifies AUTHID, or when you modify one of these attributes, CICS checks that the userid performing the operation is authorized as a surrogate user of AUTHID, COMAUTHID, or SIGNID. This also applies to the CICS region userid during group list install on a CICS cold or initial start.

The XUSER parameter is also used to control access to the AUTHTYPE and COMAUTHTYPE attributes, but the security control for these parameters is managed through the FACILITY general resource class.

There are only two valid values for XUSER:

YES

CICS is to perform surrogate user checking in all those situations that permit such checks to be made (for example, on EXEC CICS START commands without an associated terminal). Surrogate user security checking is also performed by CICS against userids installing or modifying DB2 resource definitions that specify AUTHID or COMAUTHID.

NO

CICS is not to perform any surrogate user checking.

External Security

As discussed previously, CICS can interface directly with all three of the major z/OS external security managers - CA ACF2, RACF and CA Top Secret. The majority of the discussion in this book is around RACF and CICS as that is the most common combination across all industry sectors. The concepts are identical across the ESMs but the implementation can be quite different. A brief discussion of the differences can be found later in this chapter.

The world that CICS was born into was a very different one to where we are at today. Datacentres were isolated places with few having any external communications connections.

At my first job working on an IBM mainframe we didn't even have terminals on our desks. Changes to CICS SIT parms were written in pencil onto coding sheets and given to "The Data Entry Girls" to produce a punched card deck. We had about fifty hard wired (BSC for the geeks among you) terminals used to sign on to CICS at various locations throughout the factory and knew each user personally.

Security was not built in to CICS originally. There was no need for it!



Note: The XUSER parameter is also used by CICS to control access to the AUTHTYPE and COMAUTHTYPE parameters on DB2 resource definitions, although not through surrogate user checks.

Note: If XUSER=YES is specified and the SURROGAT class is not activated in RACF, CICS fails to initialize.



Today, the roll out of a new CICS web application can lead to thousands of users being given access to systems with only the most rudimentary of checks to make sure they are an actual human being!

Embarrassingly public failures of corporations across the globe have brought home the need for a proactive approach to security. Local legislation is in place in most countries that deals specifically with securing corporate data. We have to take a new approach.

Retrograde application of external security protocols to a widely used, established software package like CICS was never going to be a smooth transition but it has allowed for the ongoing inclusion of new CICS based applications into the RACF fold much more quickly.

As a result the implementation of external security controls in CICS can feel a little “clunky” at first. It is possible to implement elegant and flexible security solutions in the System z environment - once the basic principals have been grasped. Additionally, RACF integration allowed security to be handled by security specialists rather than over worked CICS Systems Programmers.

Userids

At its most basic level, a userid is how a security package authenticates that someone (or something) is who they say they are. CICS has had the ability to authenticate users and authorize them to use resources for a lot longer than it has had RACF integration. Userids, passwords and other relevant information used to be hard coded into a table that CICS used for its own authentication process.

However, many implementations of CICS did not force users to sign on. The default userid (SIT parameter DFLTUSER) concept in CICS exists for just this reason. It allowed an installation to decide what default actions could be taken at a CICS terminal even if the person in front of it didn't have a userid of their own.

At my first job the factory workers didn't have userids. Only the supervisors needed to sign on to CICS in order to gain access to transactions which allowed them to update information. The CICS default userid could issue all of the read only transactions needed to see where parts were stored.

Another thing that our factory supervisors used to take advantage of was the fact that CICS allowed you to sign on with your userid as many times as you wanted. This meant that a supervisor could sign on at all of the control terminals in the factory allowing the regular factory workers to do some of his work while he did other things with his time.

When they started writing the userid and password (along with the command they used if CICS wasn't responding quickly enough - CEMT P SHUT – used to stop CICS) on a blackboard on the factory wall we implemented a “Userid for all” policy along with RACF protection for CICS commands! This was the same, evolving environment that led to the widespread use of UACC(READ) in many RACF installations. But I digress.

When you run CICS with RACF security checking active then you control a user's access to CICS resources through levels of authorization you define in RACF managed resource profiles. You define these authorizations for specific users by adding individual RACF userids (or preferably, RACF group IDs) to the resource access lists or, for unsigned on users, by adding the default CICS userid to selected resource access lists.

There is slightly more to a userid in CICS. The RACF USERID contains a specialized

CICS Segment Data

- *OPIDENT*
- *OPPRTY*
- *NOFORCE*
- *TIMEOUT*
- *OPCLASS*
- *LANGUAGE*
- *TLS_KEY*
- *RLS_KEY*

segment which describes all of the entries that would have previously been found in the CICS internal security tables. These values can be referenced from the RACF userids CICS segment and CICS can then make further security decisions depending on the value returned by RACF.

If you find yourself auditing a CICS system which has been around for some decades, you are more likely to find examples of CICS segment data usage. Most modern applications do not rely on these fields for additional security requirements.

RACF classes

There are three basic types of records which are held in a RACF database:

- Userids
- Groups
- Resources

Userids and Groups represent individuals (or tasks) and groups of individuals with common access requirements. The final type is used to represent anything that isn't a person. Within this last group there is a further declination between dataset profiles and general resource profiles. In other words, anything in RACF that isn't a userid, group of userids or dataset is a general resource. CICS implements many such general resource classes.

Default RACF Class	Customized RACF Class	Usage
APPCLU	n/a	LU6.2 bind-time (also known as APPC) security
CCICSCMD	Cnnnnnnn	SP command security – member class
VCICSCMD	Vnnnnnnn	SP command security – grouping class
n/a	nnnnnnnn	DB2 – no default class name
DCICSDCT	Dnnnnnnn	Transient data queue – member class
ECICSDCT	Ennnnnnn	Transient data queue – grouping class
EJBROLE	n/a	Enterprise Java Beans
FCICSFCT	Fnnnnnnn	CICS file control – member class
HCICSFCT	Hnnnnnnn	CICS file control – grouping class
UID and GID	n/a	HFS
JCICSJCT	Jnnnnnnn	CICS journal control – member class
KCICSJCT	Knnnnnnn	CICS journal control – grouping class
ACICSPCT	Annnnnnn	Started transaction control – member class
BCICSPCT	Bnnnnnnn	Started transaction control – grouping class
MCICSPPT	Mnnnnnnn	CICS program control – member class
NCICSPPT	Nnnnnnnn	CICS program control – grouping class
PCICSPSB	Pnnnnnnn	IMS - DL/I control blocks – member class
QCICSPSB	Qnnnnnnn	IMS – DL/I control blocks – grouping class
RCICSRES	n/a	CICS web services – member class
WCICSRES	n/a	CICS web services – grouping class
TCICSTRN	Tnnnnnnn	Transaction control – member class
GCICSTRN	Gnnnnnnn	Transaction control – grouping class
SCICSTST	Snnnnnnn	CICS temporary storage control – member class
UCICSTST	Unnnnnnn	CICS temporary storage control – grouping class
SURROGAT	n/a	DB2 AUTHTYPE checking

Many of the Xnn SIT parameters allow for the use of customized RACF classes but also have default class names. The list of available classes can expand when new functionality is introduced to CICS. So any auditor should make sure



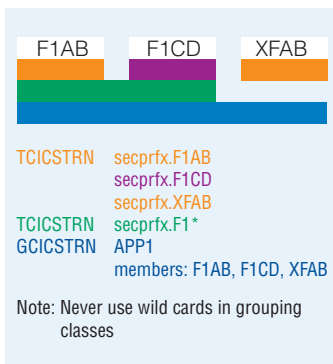
that they are in possession of the latest information for a site before starting a CICS Audit.

RACF Grouping Classes

Throughout this document you have seen reference to RACF grouping classes. It is important to understand how authority is granted when grouping classes are used in order to properly determine which RACF userids have access to issue CICS transactions.

I'll use the CICS transaction classes in this example but the point holds true for any other class pair which has both a member class **and** a grouping class.

Let's start with a system in which I have two CICS transactions called F1AB and XFAB. These transactions are both part of the same line of business and all users requiring access to one also require access to the other. In other words, these transactions require identical RACF access lists in the profile(s) that protect them.



I could simply create two RACF profiles in the **TCICSTRN** member class with names such as **F1AB** and **XFAB**, each with its own access list. But if a new transaction is added - say transaction **F1CD** - I must create a third **TCICSTRN** profile **F1CD**, and duplicate the access list associated with the others to this new one. This would eventually become a maintenance nightmare, with multiple transactions defined using the same access list. If I need to change the access list to add a new group of business users, then I need to make this update in multiple places. Also, how will I know which transaction definitions need to be updated? There's no easy method in RACF to find profiles with identical access lists. Eventually business operations will be impacted as a result of this overly complex situation.

This is where RACF grouping classes ride to the rescue. With a single grouping class (**GCICSTRN**) profile of **APP1** you can define one access list and associate this with many transaction definitions. The transactions associated with this new grouping class profile do not have to match a generic pattern, rather they are just a list of transactions, any list. This effectively clones the access list of the single profile **APP1** through all the transactions added to it. These cloned transactions are referred to as 'members' of the grouping class profile.

Grouping classes can be particularly useful where separation of responsibilities is required - as in Sarbanes Oxley. The grouping class allows a single profile to cover the majority of access requirements for a particular line of business (or role).

In the box above we also see an example **TCICSTRN** class profile **F1***, this however would not pick up the XFAB transaction. While generic profiles may be defined in the member class, this is not preferred as CICS developers then decide security, as they may create transactions to match pre-existing profiles.

When making access checks RACF uses any resources defined in either the member class directly, or 'faked' in the member class by being added to a grouping class profile. It is possible to define a transaction more than once, and with differing access lists. How RACF handles this situation is complex. Hence it's strongly recommended to ensure transactions are defined in either a member class, or a grouping class, never both.

Differences When Using Other External Security Managers

The three major security products offer the same basic functionality, the protection of resources from unauthorized use in a z/OS environment, but in different methods.



Note: If you define new or alter existing CICS profiles you must refresh the member, not the grouping class.

First the similarities of the 3 products:

- All 3 products are called by the SAF when a request is made for access to a resource
- SAF passes the request for access to a resource on the installed security product
- All use very efficient, proven databases

RACF was first to market and was introduced by IBM in 1976. By default it allows unrestricted access to mainframe resources. Rules have to be defined to PROTECT resources on the mainframe. There are four types of security profiles:

- User - the userid is a maximum of 8 characters.
- Group - where each individual user belongs to AT LEAST one group.
- Dataset
- General Resource

Datasets and general resources are protected using RACF profiles and once protected, users or groups must be granted access using the RACF PERMIT command.

CA ACF2

CA ACF2 is part of CA's Mainframe Security suite of products. The range contains many of CA's security related software products. ACF2 was developed by SKK, Inc. Barry Schrager, Eberhard Klemens, and Scott Krueger combined to develop ACF2 at London Life Insurance in London, Ontario in 1978.

The 2 was included in the ACF2 name by Cambridge Systems (who had the North American marketing rights for the product) to differentiate it from the prototype, which was developed by Schrager and Klemens at the University of Illinois - the prototype name was ACF. Cambridge also had a product named ASM2 at the time and the new name for ACF2 was similar to ASM2. SKK and ACF2 were sold to UCCEL in 1986, which in turn was purchased by Computer Associates in 1987 who develop and maintain the product to this day.

These facts are quite important to the people who work with the leading alternative to RACF today. Knowing them may help achieve a better level of cooperation when conducting the audit.

CICS support is provided by a separate product, CA ACF2 for CICS.

Individual users have a LID, or LogonID and this is used to identify them in a similar way to the RACF userid. The LID also contains information which would be found in RACF segments.

The CA ACF2 UID string is used to determine whether a user should be granted access to a specific resource. This UID string is between 1 and 24 characters long and contains information such as **location**, **department**, **job function** and the **individual users LogonID**. This is installation defined.

Datasets and resources then need to have Access and Resource Rules defined to them to permit access. These access and resource rules can allow wildcard statements within the UID string that is used, for example using the UID string described in the grey box on the right.

CA ACF2, as it is initially installed, using defaults, will deny access to all resources. This default DENY function ensures that no user, or group has access to resources unless they are specifically granted access to it.

CA ACF2 protects resources using general resource specification rules which



LOC	DIV	DEPT	JOBF	LID
LO	A	PUR	CL	APL001
London		Purchasing		
		Accounts	Clerk	

are defined in standard, ISPF accessible, libraries. The different resources are identified using the TYPE keyword. If the TYPE keyword is not specified on a rule it represents a dataset rule.

Rules are searched when access is requested and as soon as a positive match is found the process stops. This makes it critical that rule statements are in the right order - most specific at the top with the loosest match capability at the bottom of the rule. This is the order that the CA ACF2 Rule Compiler will place them in.

CICS GRS types available (custom types can be added):

Note: The CA ACF2 Types are only three characters long while the RACF classes are eight characters long. This is because CA ACF2 had CICS support years before RACF and it was felt, by the developers, that three characters was sufficient. There is a Class Mapping Table provided by CA ACF2 which will map the RACF classes into the CA ACF2 types.

ACF2 TYPE	Default RACF Class	Usage
CKC	TCICSTRN / GCICSTRN	Transactions
CTD	DCICSDCT / ECICSDCT	Destination Control Table
CFC	FCICSFCT / HCICSFCT	File Control
CPC	MCICSPPT / NCICSPPT	Processing Program Table
PSB	PCICSPSB / QCICSPSB	Program Specification Blocks
CTS	SCICSTST / UCICSTST	Temporary Storage Table
XCD	CCICSCMD/VCICSCMD	CICS SP Commands
CMR	n/a	MRO in/outbound

If a dataset access rule for the dataset PROTECTED.DATASET.NAME was required to permit read access to accounts purchase ledger clerks in ALL offices it could be defined like this:

```
$KEY(PROTECTED) - the $KEY parameter identifies the dataset HLQ
DATASET.NAME
UID(**APUR*****) READ(A)
```

To allow write access only from the London office an additional line could be added to the access rule:

```
UID(LOAPUR*****) READ(A) WRITE(A)
```

Without this second line in the rule the London office would not be allowed to alter their own office's data. Write access to data can also be prevented using the WRITE(P) parameter, although this would not restrict READ access in the above example. This highlights the requirement that any resource access using CA ACF2 MUST be defined prior to a user or department requiring that particular level of access.

Access to CICS transactions is handled in much the same way where a UID must be granted read access to the transaction, as well as being assigned the CICS privilege. Displaying the UID string for the LogonID in the example above using the LIST LogonID command should produce output which looks something like the diagram below.

APL001	LOAPURCLAPL001 Purchase Clerk COMPANY() DEPT(A) DIV(PUR) IDNUM() JOBF(CL) LEVEL() LOC(LO) LOCATION() OLDLID() POSITION() PROJECT() SITE()
PRIVILEGES	CICS

The final entry is the PRIVILEGES one that indicates that the user has access to CICS. There may be multiple PRIVILEGES for single LIDs.

A list of all users with access to CICS can be requested from your CA ACF2 administrator, together with reports detailing levels of access to CICS associated resources using the various reporting utilities such as:

- ACFRPTSL - Selected LogonID report, e.g. with CICS privilege
- ACFRPTXR - Cross Reference report, e.g. Dataset Access level

If there are only a small number of users associated with the privilege, the “LIST IF(CICS)” command could be used as an alternative. This is not appropriate in most CICS environments due to sheer quantities.

For CICS transactions, ALLOW access should be granted where required to allow these CICS users access to individual transactions. To use a similar example to the dataset one above:

```
$KEY(secprfx) TYPE(CKC)
ABCD
UID(APL001) ALLOW
```

Any changes to the security settings and/or rules must be compiled and stored after edit before they can become active to CA ACF2.

CA Top Secret

CA Top Secret is part of CA's Mainframe Security suite of products and is often referred to as TSS (Historically the name given to its started task on z/OS). Like CA ACF2 it was first brought to market by another vendor, in this case CGA Software Products Group in 1981. In 1983 the DoDCSC assigned CA TOP SECRET/MVS version 3 a class C2 rating.

In 1985, during an extended period of market expansion through acquisitions of competing ISVs, CA purchased the company and CA Top Secret was born. CA has continued to maintain a separate team to continue development and maintenance of Top Secret and r14 of the product is the most current release as of June 9th 2009.

Although CA Top Secret has the smallest market share of the z/OS external security managers it is still very relevant to smaller mainframe enterprises, particularly those running the VSE operating system, and is actually seeing something of a resurgence in popularity. One of these reasons is its unique position amongst the external security managers of having a product that supports all of the generally available z Series operating systems: z/OS, z/VM, z/VSE, z/OS UNIX and Linux for System z. In 2000 IBM even began offering CA Top Secret for VSE/ESA 3.0 as their preferred ESM for VSE/ESA 2.4.

Conceptually CA Top Secret uses a hierarchical configuration that is designed to mirror the corporate tree structure used by client companies where by every employee has someone they report to. To do this each resource or user is allocated a unique ACessor ID or ACID. At the top most level of the configuration resides the Master Security Control ACID or MSCA of which there can be only one.

Below the MSCA all the other ACIDs in the database are separated into the following types:

- **Functional ACIDs** - which are used to perform specific tasks and report to organizational ACIDs and contains the User, Profile (similar to RACF groups) and Groups (only for USS resources, mirrors RACF group processing). Functional ACIDs are associated with a single department ACID.
- **Organisational ACIDs** - which form the upper layers of the security hierarchy and report to other organizational ACIDs, but never to functional ACIDs. It

contains the Department, and the optional Division and Zone ACIDs. Resources can be attached to organizational ACIDs, although it is not recommended that resources are attached to Zone ACIDs. The optional Division and Zone ACIDs can be used to introduce greater granularity or separation of responsibilities. There can be multiple Divisions comprising of one or more Departments. There can be multiple Zones with comprising of two or more Divisions. When Zones are used Departments must be attached to a Division and the Division is attached to the Zone.

- **Control ACIDs** - are used to define the security administrators for each of the structural layers in the hierarchy. A Control ACID can be a standard user of system facilities – such as job submission, dataset access and issue subsystem commands. Each Control ASID is associated to a layer within the security administration hierarchy and is assigned a scope of authority to control its use. There are 7 layers of authority within the hierarchy:
 - Master Security Control ACID (MSCA) - referred to as the Master Central Security Administrator
 - Central Security Control ACID (SCA) - referred to as the Central Security Administrator
 - Limit Central Security Control (LSCA)
 - Zonal Control ACID (ZCA)
 - Divisional Control ACID (VCA)
 - Departmental Control ACID (DCA)
 - User ACID

CA Top Secret offers the same default deny everything functionality as CA ACF2 whereby any requests received for access to undefined resources are rejected, although it is still possible for undefined datasets to exist given the correct combination of circumstances.

Key security related items to watch for are:

- Use of 'special' resources or keywords such as ALL and *****. For example using the CA Top Secret command PE(ALL) DSNAME('SYS1.BROADCAST') would give all users and ACIDs, including those unknown by Top Secret, read access to the SYS1.BROADCAST z/OS dataset.
- Use of security definition attributes, these permit ACIDs to bypass the entire security hierarchy. These attributes include:
 - NODSNCHK** Allows access or use of any data set.
 - NOLCFCHK** Allows use of any command, program, or transaction.
 - NORESCHK** Allows use of any terminal, program, CICS, DB2, or user resource.
- Use of CA Top Secret with CICS requires an additional product, CA Common Services for z/OS CAIENF, to be installed and active. CAIENF acts as common communications layer between various CA products and with some external products. Any audit process must include the CAIENF environment since the CA Top Secret documentation states *"Without CAIENF, CA Top Secret CICS does not function."*
- CA Top Secret is normally configured to use a database contained in a single dataset, however sites with large databases may be using a database per z/OS image and using command propagation to synchronize them. In addition to this some sites may use VSAM datasets to hold digital certificates, keyrings and Kerberos KERBSEGM and KERBLINK records.

CA recommends this route if you make heavy use of digital certificates or assign Kerberos KERBNAME to your web users.

To permit access to a dataset using CA Top Secret an ACID definition must be defined for the dataset:

```
TSS ADDTO(acid) DSNAME(datasetname)
where the ACID in this statement is the owner
```

Additional resource must then be defined to have the required level of access for their role:

```
TSS PERMIT(acid) DSNAME(datasetname) ACCESS (level)
```

In securing access to CICS transactions, users must be granted access within CA Top Secret. This is granted using the PERMIT command, e.g.

```
TSS PER(acid) OTRAN(xxxxx)
where xxxxx can be an individual transaction or a list of transactions, i.e.
(a,b,c,d)
```

This is true of all resources, where access MUST be defined in the CA Top Secret database in order for that resource to be available to users or other ACIDs.

CA Top Secret actually offers two ways to protect CICS resources. The simplest of which is by using OTRAN (resource) security. The second and somewhat more complicated but infinitely more flexible way is by using the Limited Command Facility (LCF).

LCF allows the organisation to secure the resources behind the CICS transaction – e.g. FCT. Typically a Facilities Matrix will be defined for each CICS service. Two default entries are supplied in the Facilities Matrix for CICS resources – CICSPROD and CICSTEST. These are not adequate for the majority of installations. Displaying the details for CICSPROD shows the following:

```
FACILITY DISPLAY FOR CICSPROD
INITPGM=DFH ID=C TYPE=004
ATTRIBUTES=IN-USE,ACTIVE,SHRPRF,ASUBM,NOABEND,MULTIUSER,NOXDEF
ATTRIBUTES=LUMSG,STMSG,SIGN(M),INSTDATA,RNDPW,AUTHINIT
ATTRIBUTES=NOPROMPT,NOAUDIT,RES,WARNPW,NOTSOC,LCFTRANS
ATTRIBUTES=MSGLC,NOTRACE,NOEODINIT,IJU,NODORMPW,NONPWR
ATTRIBUTES=LUUPD
MODE=WARN DOWN=GLOBAL LOGGING=ACCESS,INIT,SMF,MSG,SEC9
UIDACID=8 LOCKTIME=000 DEFACID=*NONE* KEY=8
FACMATRX=NO EXTSEC=YES EJBRPRFX=NO
XJCT=YES XFCT=YES XCMD=YES XDCT=YES XTRAN=YES XDB2=NO XEJB=NO
XTST=YES XPSB=YES XPCT=YES XPPT=YES XAPPC=NO XUSER=NO
PCTEXTSEC=OVERRIDE PCTCMDSEC=OVERRIDE PCTRESSEC=OVERRIDE
DSNCHECK=NO LTLOGOFF=NO RLP=NO SLP=NO PCLOCK=NO
MAXUSER=03000 PRFT=003 MAXSIGN=010,RETRY
CICSCACHE=TASKLIFE,NOAUDIT,0512
```

You can see the **SAF class** related data in the lines above. So, in the CICSPROD Facilities Matrix entry:

- External security is switched on
- CICS resource protection active for:
 - Journal
 - File
 - Command
 - Transient Data Queue
 - Transaction

auditing CICS

- Temporary Storage
- IMS PCB
- Program
- The whole entry is in **WARN** mode meaning access won't be denied

In addition to the two CICS default facilities, a total of 222 predefined facilities are provided by CA that you can use to define a new facility of your own. Your security administrator can easily define a facility to the Facilities Matrix by:

- Changing the name of one of the predefined USER facilities.
- Modifying the security attributes of the facility to tailor security processing for that facility.

The association of a CICS region to a facility occurs by adding a MASTFAC parameter to the region ACID as shown in the example below:

```
TSS ADDTO(CICSP1) MASTFAC(CICSPROD)
```

LCF and the Facilities Matrix are quite complex topics. And the decision on when to use them over OTRAN can be a very hard one to make. Be glad you are an auditor.

All entries in CA Top Secret are therefore definitions of resources and any that are used for CICS should be checked to ensure that the MINIMUM level of access required is defined. Your CA Top Secret Administrators and Systems programmers will be able to provide you with the details of these definitions, which can be accessed using the TSS WHOHAS command against transaction, ACID and dataset definitions.

CICS System Definition - CSD

One of the foundations of a CICS environment is the CSD file. Data in this file needs to be closely monitored. As with all key system datasets you need to ensure that all access to it is restricted to those processes that actually need it, and that the contents are adequately backed up and recoverable. This should include the restriction of read access as the contents of the CSD should be considered as being of a sensitive nature.

From the perspective of a batch task you must ensure that access to the DFHCSDUP CSD utility program is adequately restricted. Although any changes made using this utility will not come into effect until the CICS region is cold started, it creates the potential for 'sleeping' changes, either by accident or by design, that impact service or security immediately or later.

There are a number of CSD specific entries in the SIT, the majority of them relate to the physical attributes rather than being of interest to a security audit. The following entries are of interest to a CICS security audit:

CSDACC

This parameter controls the level, and type of access to the CSD for this CICS region. It should be noted that it only alters the settings when you cold start CICS. If you use the START=AUTO parameter, CICS will perform a warm or emergency restart and the new settings will not come into effect. The file resource definitions for the CSD are recovered from the CICS global catalog however you can redefine the level of access to the CSD dynamically with a CEMT SET FILE, or an EXEC CICS SET FILE, command. The default value of READWRITE grants the full range of CEDA, CEDB, and CEDC functions. Setting this to READONLY limits the CEDA and CEDB transactions to only those functions that do not require write access.

SEC

This parameter must be set to YES if CSD protection is to be achieved. *

SECPRFX

An optional parameter only relevant if set.*

XTRAN

If this parameter is set to NO, CSD protection cannot be achieved.*

* See entry in *SIT settings* section for more information.

As I have mentioned already access to the CICS CSD interface is controlled by the CEDA, CEDB and CEDC transactions. RACF protection of these transactions represents the initial layer of CSD protection. They call the exec interface programs to process the user's commands, in much the same way as CECL or a user's command-level program would. The resource security and the command security used by the command-level programming interface are thus applicable to the master terminal user.

The following commands are available when using the CEDA transaction:

ADD	DISPLAY	UNLOCK
ALTER	EXPAND	USERDEFINE
APPEND	INSTALL	VIEW
CHECK	LOCK	Route (Copy, Move and
COPY	MOVE	Rename entries)
DEFINE	REMOVE	
DELETE	RENAME	

When using the CEDB transaction, the INSTALL command is not available. This means that the CSD can be updated but not the running CICS system. The CEDC transaction only allows the read only commands DISPLAY, EXPAND, and VIEW.

CICS TS 3.2 supports the following RDO resource definitions:

- **CONNECTION** - Defines a remote system CICS connects to using ISC or MRO links.
- **CORBASERVER** - Defines an execution environment for enterprise beans and stateless CORBA objects.
- **DB2CONN** - Defines the attributes of the connection between CICS and DB2, and of the pool threads and command threads used with the connection.
- **DB2ENTRY** - Defines the attributes of entry threads used by the CICS DB2 attachment facility.
- **DB2TRAN** - Defines a transaction, or a group of transactions, associated with a DB2ENTRY, that are additional to the transactions specified in the DB2ENTRY itself.
- **DJAR** - Defines an instance of a deployed JAR file, which contains enterprise beans.
- **DOCTEMPLATE** - Defines the attributes of a document template.
- **ENQMODEL** - Defines a named resource for which the EXEC CICS ENQ and EXEC CICS DEQ commands have a sysplex-wide scope.
- **FILE** - Defines the physical and operational characteristics of a file.
- **IPCONN** - An IPCONN (also known as an IPIC connection) is a CICS resource that represents an outbound Transport Control Protocol/Internet Protocol (TCP/IP) communication link to a remote system.
- **JOURNALMODEL** - Defines the connection between a CICS journal name



(or identifier) and the associated physical log streams managed by the MVS system logger, or between the journal name and the SMF log.

- **LIBRARY** - Define the physical and operational characteristics of a LIBRARY.
- **LSRPOOL** - Defines the size and characteristics of the local shared resources (LSR) pool.
- **MAPSET** - Defines a BMS map sets.
- **PARTITIONSET** - Defines a partition set (a table that describes to CICS how to partition a display screen).
- **PARTNER** - A PARTNER resource enables CICS application programs to communicate, using APPC protocols, with a partner application program running on a remote logical unit.
- **PIPELINE** - A PIPELINE resource definition is used when a CICS application is in the role of a Web service provider or requester. It provides information about the message handler programs that act on a service request and on the response. Typically, a single PIPELINE definition defines an infrastructure that can be used by many applications.
- **PROCESSTYPE** - A PROCESSTYPE resource defines a BTS process-type. It names the CICS file which relates to the physical VSAM data set (repository) on which details of all processes of this type (and their activity instances) are to be stored.
- **PROFILE** - A PROFILE resource specifies options that control the interactions between transactions and terminals or logical units. The PROFILE is a means of standardizing the use of such options as screen size and printer compatibility, and the use of such functions as message journaling and the node error program.
- **PROGRAM** - Describes the control information for a program that is stored in the program library and used to process a transaction, or part of a transaction.
- **REQUESTMODEL** - Defines how an Internet Inter-ORB Protocol (IIOP) inbound request is mapped to the CICS transaction that is to be initiated.
- **SESSION** - Defines the logical link between two CICS systems that communicate using ISC or MRO links.
- **TCPIPSERVICE** - Defines which TCP/IP services are to use CICS internal sockets support.
- **TDQUEUE** - Defines the attributes of a transient data queue.
- **TERMINAL** - Defines the characteristics of a terminal device which communicates with CICS. Terminal devices include visual display units, printers, operating system consoles, and more specialized devices such as facsimile (FAX) machines.
- **TRANCLASS** - Defines the characteristics of a transaction class.
- **TRANSACTION** - Defines transaction attributes that relate to functions provided by CICS.
- **TSMODEL** - Defines a TS queue name prefix, and associates attributes with that name.
- **TYPETERM** - Defines a set of attributes that are common to a group of terminals.
- **URIMAP** - URIMAP definitions are resource definitions that match the URIs of HTTP or Web service requests, and provide information on how to process the requests.

- **WEBSERVICE** - Defines aspects of the run time environment for a CICS application program deployed in a Web services setting. Although CICS provides the usual resource definition mechanisms for WEBSERVICE resources, they are typically installed dynamically, using the output produced by the CICS Web services assistant.

The resources defined to the CSD should also be reviewed regularly. Errant table entries tend to accumulate over time. In most large companies where CICS has been used for 30 years or more, region tables often resemble old attics full of surprise boxes. They might include entries used in testing but never deleted, or entries made by software during installation that are now dead, mismatched or orphaned. In the current climate RDO entries for companies which have been merged into core business (such as is seen when a large Bank takes over a smaller one) are not always relevant.

History

Early versions of CICS used control tables to store all its resource definitions which required compiling and a service outage to implement changes. As the popularity of CICS grew and more functionality was added to the product single CICS regions began hitting various architectural limits in the underlying operating systems resulting in 1979 with the release of CICS MRO which allowed multiple CICS regions to act as a single CICS service. This generated a rapid expansion in the workload required to maintain CICS and in the amount of service lost due to the need to recycle CICS to pickup new definitions.

In 1982 IBM revealed RDO, Remote Definition Online, which used the new mandatory CSD file to store resource definitions previously held in the PCT, PPT and TCT macro tables. Support for other types of resource has been added to later releases of CICS. These definitions could be changed or created using an online, i.e. through a VTAM terminal session, interface and then enabled without needing to shutdown the CICS region. In addition to this the definitions could be placed into a basic grouping structure that helped further reduce the overhead of maintaining multiple CICS regions.

Future (CPSM)

With the introduction of CICSplex SM, IBM introduced a new location for resource definitions the CICSplex SM Repository. Initially this repository was restricted to CMAS link and workload related configuration information.

With the recent releases of CICSplex SM/CPSM, IBM are moving towards web based administration over the existing CICSplex SM ISPF and CICS RDO terminal transaction interfaces. This new process allows CICS sites to manage and maintain these definitions from a single interface.

CPSM is not currently in widespread use. Sites that have employed it may use it to store information, normally held in the CSD, in the CPSM repository instead. Although the content of the resource definition remains the same the task of securing CICS definitions and CPSM must now include the new web GUI access method. It should be noted that under CICSplex SM a resource is no longer restricted to a single group, as it is with CSD RDO entries, but can be a member of many groups.

Where CICSplex SM is in use there are a number of additional RACF definitions that should be present to secure the CICSplex SM application and how it interacts with the rest of CICS:

- CICSplex SM transactions for MAS and CMAS
A list of these transactions is contained in the CSD group EYU\$CDEF.



- CICSplex SM transactions for the Web user interface
A list of these transactions is contained in the CSD group EYU310GW.
- Relevant EYUWUI profiles in the RACF FACILITY class to protect the web user interfaces.
- Activate the CPSMOBJ RACF class and define the relevant profiles.
- Create RACF class CPSMOBJ profiles to protect the CICSplex SM resources
Use the formats `function.type.context` or `function.type.context.scope` depending on the function being protected.

The Web user interface uses the same authorization path as the CICSplex SM API.

IBM Supplied CICS transactions



Transaction security requirements vary across installations. All in house developed application transactions should have access policies defined in site specific documentation. However there is a class of CICS transactions which should always have the same access requirements. These are the IBM supplied transactions which perform many CICS functions (for example signing on or closing down CICS) and it's the one fixed point when auditing CICS - although you have probably already guessed there are a few exceptions even here.

All IBM supplied (i.e. not installation written) CICS transactions are associated with three security categories. The three categories contain all the required CICS transactions, which are generated in their designated groups when you initialize your CICS system definition data set (CSD). This is a CICS function and not affected by the installation. The set of transactions which should be members of each category is the same across all CICS installations.

There should be transaction grouping class profile(s) defined in RACF for each category.

Often installations have the categories broken down further than just the three discussed in this chapter. If `SECPRFX=YES` is set in the SIT there might also be 'duplicate' transaction definitions, prefixed by the CICS regions `userid(s)`, in the member class.

The profiles are defined in either the default `GICSTRN` class or an installation defined grouping class as specified in the SIT parameter `XTRAN`. The names of the RACF profiles are not fixed. You should ask the CICS Systems Programmers or RACF Administrators for the naming standards of these grouping class profiles.

Descriptions of the three categories are listed in the following sections along with the IBM supplied transactions falling into each category. It is important to ensure that all the transactions listed are defined and protected to the required standards appropriate to each category. Remember that new transactions can be added with each new version of CICS. You should consult the IBM documentation for your version of CICS to determine the appropriate list of transactions for each category. The list below is current at time of publication (CICS TS 3.2).

Category 1

These transactions are never associated with a terminal. That is, they are for CICS internal use only, and should not be invoked from a user terminal. CICS checks that the region `userid` has the authority to attach these transactions. In other words, the CICS region `userid` is the only one which should appear on the access list in RACF and the profile should have a `UACC` of `NONE`.

- | | | | |
|--------|--------|--------|--------|
| • CATA | • CIOD | • CKAM | • CSKP |
| • CATD | • CIOF | • CKTI | • CSNC |
| • CDBD | • CIOR | • CMTS | • CSNE |
| • CDBF | • CIRR | • COVR | • CSOL |
| • CDBO | • CISC | • CPIR | • CSPQ |
| • CDBQ | • CISD | • CPIS | • CSQC |
| • CDTS | • CISE | • CPLT | • CSSY |
| • CESC | • CISR | • CRMD | • CSTE |
| • CEX2 | • CISS | • CRMF | • CSTP |
| • CFCL | • CIST | • CRSQ | • CSZI |
| • CFOR | • CISX | • CRSY | • CTSD |
| • CFQR | • CITS | • CRTP | • CWBG |
| • CFQS | • CJGC | • CSFR | • CWXN |
| • CFTL | • CJMJ | • CSFU | • CWXU |
| • CFTS | • CJPI | • CSHA | • CXCU |
| • CGRP | • CJTR | • CSHQ | • CXRE |

Category 2

Transactions in category 2 are either initiated by the terminal user, or are associated with a terminal. Many of these transactions are for CICS administration, and are very powerful. Access to initiate these transactions should be restricted to a strictly limited group of users.

The following list details the requirements for protection of category 2 transactions using RACF:

- UACC(NONE) and AUDIT(FAILURES(READ),SUCCESS(READ)) in the transaction profile (AUDIT(FAILURES) is the default, and need not be explicitly specified)
- Strictly limited access list as appropriate

It's unlikely that more than a few users would require access to the entire set of category 2 transactions. Thus it is common to see these transactions divided into several subcategories.

- SYSADM, containing transactions CBRC, CDBT, CEDA, CEMT, and CETR
- DEVELOPER, containing transactions CEBR, CECI, CECS, CEDB, and CEDF
- INQUIRE, containing transactions CDBI and CEDC
- OPERATOR, containing transactions CEOT, CEST, CMSG, and CWTO
- INTERCOM, containing transactions CEHP, CEHS, CPMI, CRTE, CSMI, CSM1, CSM2, CSM3, CSM5, and CVMI

- WEBUSER, containing only transaction CWBA

If function shipping is being used, the mirror transactions must be available to remote users in a function shipping environment. This means that:

- The terminal user running the application must be authorized to use the mirror transaction
- The terminal user must also be authorized to use the data that the mirror transaction accesses
- ALLUSER, containing transactions CMAC and CSGM which are the CICS "messages and codes" and "good morning" transactions with UACC(READ). Also include your "goodnight" transaction in this group, if you defined one with the GNTRAN system initialization parameter
- NOUSER, for all remaining transactions that do not have a demonstrated requirement to be used by staff or system userids, or do not have an operator interface as noted below.

Note: The following are only examples of the way that transactions might be grouped. Unfortunately for the Auditor, this is another non-fixed point of configuration. An installation can choose to group CICS transactions in the ways that best suit their needs.



The transactions that have operator interfaces are marked by an asterisk (*). The remainder, therefore, have no operator interface which means that no userids or groups should be on the access list in RACF.

- | | | | |
|---------|---------|---------|---------|
| • CADP* | • CEHP | • CKRS | • CRTE* |
| • CBAM* | • CEHS | • CKRT | • CRTX |
| • CCRL* | • CEMN | • CKSD | • CSFE* |
| • CDBC* | • CEMT* | • CKSQ | • CSGM |
| • CDBI* | • CEOT* | • CKQC | • CSHR |
| • CDBM* | • CESD | • CMAC | • CSM1 |
| • CDBT | • CEST* | • CMSG* | • CSM2 |
| • CDFS | • CETR* | • CPIH | • CSM3 |
| • CEBR* | • CIDP* | • CPIL | • CSM5 |
| • CECI* | • CIND* | • CPIQ | • CSMI |
| • CECS* | • CIRP | • CPMI | • CTIN |
| • CEDA* | • CKBM | • CREA* | • CVMI |
| • CEDB* | • CKBP | • CREC* | • CWBA |
| • CEDC* | • CKCN | • CRPA | • CWTO* |
| • CEDF* | • CKDL | • CRPC | • CWWU |
| • CEDX* | • CKDP | • CRPM | • DSNC |

Category 3

The last category of IBM supplied CICS transactions is category 3. They are either initiated by the terminal user or associated with a terminal. All CICS terminal users, whether they are signed on or not, require access to transactions in this category. For this reason, category 3 transactions are exempt from any security check, and CICS permits any terminal user to initiate these transactions.

These transactions should be defined to RACF, but this definition does not affect actual security processing. Their RACF definitions may be used with the CICS program call EXEC QUERY SECURITY, but this is not common. Effectively, there should be no entries on the RACF profile access list with any kind of access other than READ. In a technical sense, users' access to these transactions is ALTER, meaning that CICS gives away all control over them. However ALTER and READ level access are not distinguished within general CICS access requests. Often, all users are granted READ to these transactions, purely to document the effective access in place.

- | | | | |
|--------|--------|--------|--------|
| • CATR | • CLR1 | • CPSS | • CSPK |
| • CCIN | • CLR2 | • CQPI | • CSPP |
| • CEGN | • CLS1 | • CQPO | • CSPS |
| • CEJR | • CLS2 | • CQRY | • CSRK |
| • CESF | • CLS3 | • CRSR | • CSRS |
| • CESN | • CLS4 | • CSAC | • CSSF |
| • CIEP | • CMPX | • CSCY | • CXRT |
| • CLQ2 | • CPCT | • CSPG | |

IBM additionally documents these three transactions as immune to RACF checking, behaving the same as Category 3 transactions:

- CDBN** DBCTL interface connection
- CEKL** Emergency Master terminal
- CSXM** Transaction environment management

Again, this list does change occasionally most often on a release boundary but new function can be added by SMP/e fix in exceptional circumstances.

Access to the non-categorized transaction EXCI should also be examined, it allows batch submission of CICS transactions and security credentials may be specified by the developer or user. The CEDF tracing facility should also be examined as it can be used to view confidential data.

Securing CSD Transactions

When SEC=YES and XTRAN=YES (or an installation defined class) in the SIT parameters, all CICS transactions are subject to RACF protection with the exception of those in category 3. By default CICS will allow access to any transaction undefined to RACF unless a backstop, or catchall, profile is defined or the default return code (4) of the RACF class used has been altered (to 8).

The Category 2 transactions CEDA, CEDB and CEDC control all access, via CICS, to the contents of the CSD. While CEDC allows READ access only this should still be controlled to prevent identification of security weaknesses. Access to the Master Terminal transaction CEMT must also be strictly controlled.

The combination of both transaction protection and CICS command protection (XCMD) is currently considered 'best practice' for most common CICS usage. The additional classes, eg: File or Resource control (XFCT, RESSEC) may be implemented for an additional layer of protection where higher levels of security are required.



Glossary of Terms

This will be more than just a CICS glossary from an IBM manual. It is a collection of basic CICS and auditing terms that you should be familiar with.

4LA / FLA - Four Letter Acronyms, CICS is full of them but there is a reason. Although most of z/OS suffers from a limit of 8 characters or combinations thereof, parts of the CICS architecture are limited to 4 characters. Hence all commands are 4 characters etc.

Address space - A virtual space created and maintained by z/OS in which separate tasks run. For example both a batch job and a CICS STC would run in separate address spaces. Each address space has a unique id called the ASID (Address Space ID, not to be confused with ACID which is often pronounced the same way).

ACID - ACcessor ID, A security related term used to describe the userid of a process. When dealing with a CA Top Secret environment it is roughly equivalent to a RACF userid. Not to be confused with ASID.

AOR - Application Owning Region, logical grouping construct used by sites using a CICS MRO configuration. That is commonly used to indicate a CICS region that runs the application program workload.

API - Application Program Interface, a documented interface point that allows applications or applications and users to interact in a controlled manner.

APPC - Application Peer to Peer Communication - VTAM network protocol

APPN - Application Peer to Peer Network - VTAM network protocol

Batch job - Aka Job. A set of JCL instructions, including the execution of at least one program, that performs a unit of work as a separate task to that which created it. Batch jobs are submitted to JES which uses the supplied JCL to determine where, when and what runs. A single user may submit many batch jobs which may or may not run concurrently depending on the requirements. Batch jobs are used for short running tasks and are executed by JES in a special sub-set of address spaces called Initiators which are under the control of JES.

BCP - The Base Control Program provides all of the base functionality for the z/OS platform. It provides the foundations for all of the functions that run under z/OS, dealing with elements such as Workload Manager and the System Management Facility. Without BCP the operating system would not function.

CA ACF2 - An external security manager product from Computer Associates.

CA Top Secret - An external security manager product from Computer Associates.

CICSplex - Sysplex for CICS, the ability to connect and control multiple CICS regions running on multiple LPARs within a z/OS parallel sysplex.

CICSplex SM - aka CPSM, A software backed management layer used to control large, complex CICS sites.

CLIST - Basic scripting language found on z/OS.

CMAS - CICSplex SM Address Space - This is the started task that controls the CICSplex management layer.

DASD - Direct Access Storage Device. Traditionally a DASD unit related to a single physical drive unit. With the increase in drive capacity a single physical drive unit may contain multiple DASD units. Each DASD unit contains a single z/OS DASD Volume. These volumes are used to contain the z/OS file systems that contain all of the z/OS datasets. A DASD unit may be allocated to a single LPAR or shared by several.

Dataset - The term used to describe files within the z/OS file structure. The term File may sometimes be used in its place. Each dataset is defined with a specific organisation or format which is used to determine which access method is used to access data held within the dataset. CICS application data is mainly stored in either VSAM datasets or in a relational database such as DB2.

Dataspace - A dataspace is a range of up to 2 gigabytes of contiguous virtual storage addresses that a program can directly manipulate through ESA/390 instructions. Unlike an address space, a dataspace contains only data. It does not contain common areas or system data or programs. Programs cannot execute in a dataspace, although load modules can reside in a dataspace. To reference the data in a dataspace, a program must be in access register (AR) mode. Up to 15 dataspace can support an address space at one time.

Using dataspace minimizes the need to store active data on external devices and increases integrity by separating code from data.

DOR - Data Owning Region, aka FOR logical grouping construct used by sites using a CICS MRO configuration. That is commonly used to indicate a CICS region that runs the file ownership workload.

ESM - External Security Manager, the product used to control access to z/OS resources. See RACF, CA-TOP SECRET, CA-ACF2.

Exit - aka Exit Point, a documented interface point within z/OS or other IBM and ISV system software and the associated code, normally written in Hi-Level Assembler. To be treated

with caution as they can be used to alter the way in which any activity occurs.

FOR - File owning Region, see DOR.

Hiperspace - High performance space (hiperspace) is a data buffer that is backed either in expanded storage only or in expanded storage and auxiliary storage. It can be used for temporary or permanent data. Its maximum size is 2 gigabytes. However, unlike dataspace, which are limited to 15 active at any time, the number of active hiperspaces for an address space is limited only by address space limits defined in the IEFUSI SMF exit. For response-critical applications requiring large data structures, hiperspaces can provide almost unlimited definable storage, provided that the expanded storage is available.

IMAGE - aka z/OS image, can be used to describe any z/OS LPAR but technically it indicates a z/OS LPAR that is a member of a parallel Sysplex. Specifically where each LPAR in the SYSPLEX is an image of the others.

ISC - intersystem communication, A type of connection used by CICS to connect to remote systems.

JCL - Job Control Language. A set of instructions which when combined into a JCL stream describe an environment in which programs may be executed. STCs use a subset of the JCL instruction set whilst batch jobs have access to the entire instruction set.

Back in the days when MVS was first introduced, the only real method of entering data into the system was punched cards. JCL has not radically changed since those days. It is still an 80 character line length representing the maximum that would fit on a standard IBM punched card. For this reason it is often referred to as a JCL Deck by the Grey Haired Gurus aka sysprogs.

JES2 / JES3 - Software based resource management layer found on z/OS systems. A z/OS image will use JES2 or JES3 never both. Amongst its many responsibilities is output management.

LPAR - Logic Partition, A mainframe concept used to allocate resources such as CPU (both in number and percentage terms), physical memory or DASD units to a specific operating system instance. Each LPAR is ring fenced at the CPU and memory level to provide data integrity.

LID - Logon ID, A term from the CA ACF2 security manage product. Roughly equates to the RACF userid concept.

LSR POOL - The LSR pool is a reserve of data buffers, strings, and Hiperspace™ buffers that VSAM uses when processing access requests for certain files.

MAS - Managed Address Space, the CICSplex term for a CICS region under CICSplex control.

Mashup - A process combining information from multiple sources into a single stream. Although

originally a term from the Music Industry it can now be found being used to describe mainframe / Web 2.0 applications.

MRO - Multi Region Operation, introduced by IBM in 1979 to overcome the limitations of running CICS as a single instance. It allows multiple CICS regions to function as a single service whilst maintaining basic data integrity. MRO regions are connected using MRO links.



MVS - A term used by the older generation of IT staff instead of z/OS.

Parallel Sysplex - A hardware and software management layer that allows multiple LPARs running on multiple mainframes to be combined into a single service. A correctly configured Sysplex offers 24x7x52 availability with hardware and software fault tolerance plus dynamic workload balancing and dynamic increases in processing capacity.

POSIX - Portable Operating System Interface for Unix, a set of standards that define various aspects of the UNIX operating system. USS is fully POSIX compliant.

RACF - Resource Access Control Facility, the IBM external security manager product. Market leader amongst the External Security Managers.

REXX - A very powerful scripting language available on z/OS (and other platforms including the Open Source community). Modified versions of REXX can also be found in certain IBM and ISV products that provide a base REXX environment with additional product specific functionality, e.g. CA-OPS/MVS OPS/REXX.

SAF - System Authorization Facility, A term used to describe the process used to validate an access request to a resource controlled by the ESM.

Segment - Can be used when describing specific parts of an individual RACF userid record. For example a z/OS TSO user would have a RACF userid record with a TSO segment defined. Other segments include CICS, NetView, Lotus Notes and many others.

SMF - System Management Facility, The underlying system activity audit trail for z/OS. SMF can be configured to cut information records detailing much of the activity that occurs on z/OS. For example RACF can be configured to generate an SMF record when access to a resource is denied. These records are written out to datasets that are normally backed up and retained for audit processing.

SNA - a network protocol used by VTAM. This includes the LU 2 and 6.2 protocols.

STC - Started Task. STCs are defined using JCL and controlled by z/OS and JES in a similar manner to batch jobs. The key differences are that a batch job is designed to perform a single discreet work process and then complete where as an STC is designed to process requests on an ongoing basis. Most STCs also

interact with other tasks or users to perform designated tasks.

Surrogate - The ability to pretend to be another userid when requesting access to a resource. Particularly used in conjunction with CICS calls to DB2 in the context of this book.

Also RACF class SURROGAT used to enable things to be done legitimately on behalf of another user.

Sysplex - See Parallel Sysplex

Systems Programmer - aka Sysprog, An endangered species of technical specialists responsible for maintaining the z/OS software stack. Whilst best approached with caution and humility they can be good sources of information on vulnerabilities within a system. Often partial to beer / donuts and may be split into specific sub groups with responsibilities for specific areas such as CICS, z/OS or DB2.



TCP/IP - The Internet Protocol stack supported by z/OS which allows z/OS based applications to communicate with IP based hosts/clients. A fully functional TCP/IP stack appears to have been the driving factor in the renewed vigor of System z.

Transaction - A commonly used word with multiple and sometimes conflicting meanings. This term is often used to describe a single CICS process such as a CICS transaction that can be processed as a single entity. Confusingly it can also be used to describe a process that includes a number of CICS transactions. A transaction may be considered to be restricted to the execution of a single program or may consist of the execution of multiple programs.

UACC - aka UACC(NONE). Universal Access level. This represents the access level to be granted to a resource where the requestor has not been granted specific access nor have they been granted access by membership of a group. A UACC level above NONE allows the specified level of access unless the requestor has been specifically excluded potentially even if the requestor is not a valid userid.

UID - A UNIX security concept numeric label used to grant / reject access to UNIX based resources. Not to be confused with RACF USERID or ACF2 UID string.

USERID - A z/OS security concept label of 1 to 8 characters assigned to an entity through which all access to resources is granted or rejected. Not to be confused with UID.

USS - UNIX System Services, a fully POSIX compliant implementation of the UNIX operating system that runs as a service under the control of z/OS. A comparatively new part of the z/OS offering that became a required service when IBM rewrote their TCP/IP application to run using USS. As a result of the way USS was introduced there is a tendency for the management, in all aspects including security,

to be less well structured and documented than is expected in other areas such as CICS.

VSAM - Virtual Storage Access Method, one of a number of methods of storing and accessing data on z/OS systems. VSAM is the most popular method of storing CICS application data. Data held in databases such as DB2 is actually stored within VSAM files with DB2 providing the database table structure and access layer. Each VSAM file consists of one or more (depending on the type of VSAM file) z/OS datasets.

VTAM - Virtual Telecommunications Access Method, the basic network layer used by z/OS to transport data between various areas such as between users and the applications running on z/OS. It uses the SNA protocol and includes CICS LU 6.2 connections. VTAM network traffic is not generally encrypted. Overall use of VTAM is declining in favour of TCP/IP.

z/OS aka MVS - The most advanced operating system known to the human race, unfortunately only legal on mainframes. There is no such thing as the 'blue screen of death' under a fully configured and properly managed z/OS system - ever.



Index

- ACF2**, 5, 8, 15, 22, 24, 37, 41, 42, 43, 44
- ACID**, 43, 44, 45, 46
- Amazon**, 5
- AOR**, 6
- API**, 9, 32, 50
- APPC**, 9, 24, 26, 27, 39, 45, 48
- APPN**, 55
- ATM**, 6, 7
- Authorized**, 11, 13, 16, 17, 18, 31
- BCP**, 12
- CICS**, 1, 2, 3, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53
- CICSplex**, 6, 7, 49, 50
- CLIST**, 55
- CMAS**, 7, 49
- communication**, 6, 7, 13, 14, 16, 34, 47
- database**, 2, 5, 6, 7, 11, 12, 13, 18, 34, 36, 39, 41, 43, 44, 45
- Dataset**, 16, 17, 18, 20, 26, 31, 39, 41, 42, 43, 44, 45, 46
- Dataspace**, 55
- DB2**, 2, 7, 12, 13, 24, 28, 29, 37, 39, 44, 47
- DOR**, 6
- ESM**, 8, 22, 37, 43
- Exit**, 22, 29
- FOR**, 6
- Hiperspace**, 56
- IMAGE**, 25, 44, 56
- IMS**, 2, 7, 24, 34, 39, 46
- ISC**, 7, 8, 47, 48
- Java**, 2, 8, 30, 35, 39
- JCL**, 14, 15, 16, 17, 18, 20, 26, 55, 56
- JES2**, 56
- JES3**, 56
- LID**, 41, 42
- Logging**, 12, 18, 32, 45
- LPAR**, 55, 56
- MAS**, 49, 55
- Mashup**, 34, 56
- MRO**, 6, 7, 9, 42, 47, 48, 49, 55, 56
- MVS**, 25, 26, 43, 48
- POSIX**, 56, 57
- RACF**, 5, 8, 9, 11, 12, 15, 16, 17, 18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 47, 49, 50, 51, 52, 53
- Region**, 6, 7, 8, 15, 16, 17, 18, 20, 21, 22, 23, 24, 25, 26, 27, 29, 32, 33, 34, 36, 37, 46, 49, 50
- REXX**, 12, 56
- SAF**, 21, 41, 45
- Sarbanes Oxley**, 3, 21, 22, 23, 24, 40
- SAS 70**, 3, 21, 22, 24, 25
- Segment**, 17, 38, 39
- SIT**, 9, 14, 15, 19, 20, 21, 22, 23, 24, 27, 29, 31, 33, 34, 35, 37, 38, 39, 46, 47, 50, 53
- SMF**, 18, 45, 48
- SNA**, 7, 8, 13, 56, 57
- Surrogate**, 23, 37, 57
- Sysplex**, 6, 7, 8, 25, 47, 55, 56, 57
- Systems Programmer**, 17, 28, 29, 46, 57
- Sysprog**, 28, 56, 57
- TCP/IP**, 8, 13, 22, 26, 47, 48, 57
- telnet**, 13
- Top Secret**, 5, 8, 15, 22, 24, 37, 43, 44, 45, 46, 55
- TOP SECRET**, 43, 55
- transaction**, 5, 6, 7, 8, 11, 12, 13, 17, 18, 19, 21, 22, 23, 24, 25, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53
- Unix Systems Services**, 12, 34
- USS**, 12, 13, 31, 32, 43, 56, 57
- VSAM**, 19, 31, 44, 48, 55, 56, 57
- VTAM**, 7, 9, 21, 28, 49, 55, 56, 57
- WebSphere**, 8, 13
- z/OS**, 2, 3, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 20, 23, 26, 30, 31, 32, 34, 35, 37, 40, 43, 44, 55, 56, 57

Future Publications

In the coming months, NewEra Software will publish additional White Papers on CICS and other topics of interest to z/OS professionals. The following are slated for publication in the fourth quarter of 2009:

Alphabet Soup

CICS Security is quite a complex subject with many layers and facets. Understanding of the basic functions is essential in order to appreciate the very sophisticated way that the various SAF classes are exploited by CICS. Fifteen CICS SIT parameters are detailed.

The Importance of SITting Comfortably

The SIT parameters represent the heart of CICS. Centralised control of configuration parameters is vital in a complex CICS environment. The author relates an experience in a customer setting where security had been “MacGyver’d” together and how he helped them implement a more secure and functional environment.

Managing Complex CICS Environments

Very few installations run single CICS regions under z/OS. This white paper will cover some of the important questions that you need answer when running a complex CICSplex environment.

These White Papers will be available on the NewEra Software website – www.newera.com. Please check the website regularly for new additions to the NewEra repository of pertinent z/OS information.