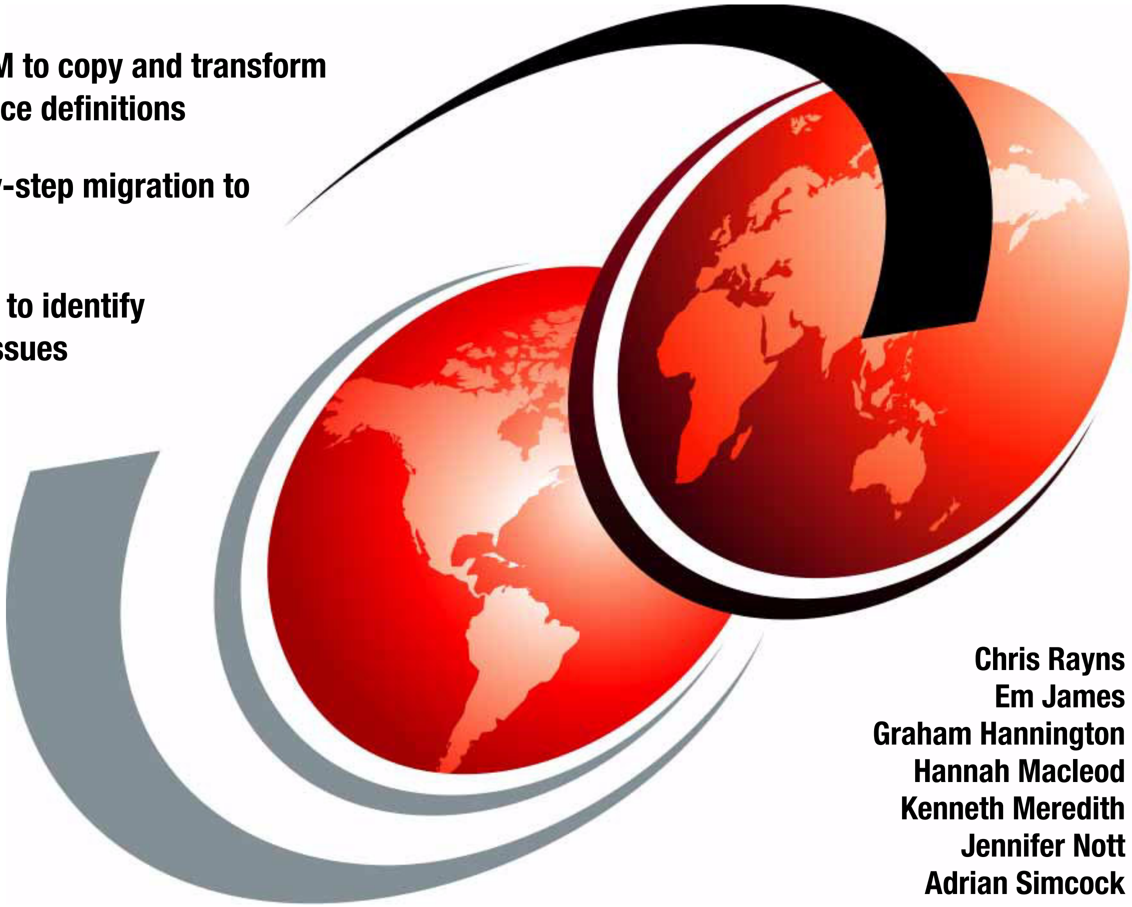
IBM

Migration Considerations for CICS Using CICS CM, CICS PA, and CICS IA

Use CICS CM to copy and transform CICS resource definitions

Use step-by-step migration to CICS TS 3.1

Use CICS IA to identify migration issues



Chris Rayns
Em James
Graham Hannington
Hannah Macleod
Kenneth Meredith
Jennifer Nott
Adrian Simcock

Redbooks

ibm.com/redbooks

International Technical Support Organization

Migration Considerations for CICS Using CICS CM, CICS PA, and CICS IA

October 2006

SG24-7294-00

Note: Before using this information and the product it supports, read the information in “Notices” on page xi.

First Edition (October 2006)

This edition applies to CICS Transaction Server Version 3 Release 1.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Redbooks (logo) ™	DB2®	RACF®
developerWorks®	IBM®	S/390®
ibm.com®	IMS™	SupportPac™
z/OS®	Language Environment®	Tivoli®
zSeries®	MQSeries®	TXSeries®
C/370™	MVS™	VisualAge®
CICS/ESA®	MVS/ESA™	VTAM®
CICS®	OS/390®	WebSphere®
CICSPlex®	QMF™	
DB2 Universal Database™	Redbooks™	

The following terms are trademarks of other companies:

EJB, Java, JVM, J2EE, ONC, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

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

Preface

This IBM® Redbook focuses on CICS® Migration to CICS TS 3.1, showing you how the CICS Tools (CICS Configuration Manager, CICS Interdependency Analyzer, and CICS Performance Analyzer) can help you with your migration.

Part 1, "Introduce CICS TS 3.1 and the CICS Tools" on page 1, gives an overview of the new functionality available in CICS TS 3.1 and an overview of the CICS Tools individually.

Part 2, "Migration" on page 111, looks at migration, discussing migration considerations and CICS TS 3.1 exploitation. It also looks at three migration scenarios:

- Migrating CICS TS 2.3 CSD to CICS TS 3.1 CSD
- Migrating CICS TS 2.3 CSD to CICSPlex® SM TS 3.1 BAS
- Migrating an Application to CICS Web Services in CICS TS 3.1

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Chris Rayns is an IT Specialist and Project Leader at the ITSO Poughkeepsie Center in New York. Specializing in security, Chris writes extensively on all areas of S/390® security. Before joining the ITSO, he worked in IBM Global Services in the United Kingdom (UK) as a CICS IT Specialist.

Em James is a Technical Specialist for CICS Tools working at IBM Hursley Labs in the UK. He has 18 years of experience working with CICS as both an Application Programmer and a Systems Programmer. He most recently worked as the Lead Developer on the CICS IA product. He holds a degree in Computer Science from Loughborough University.

Graham Hannington is a Technical Writer at Fundi Software, in Perth, Western Australia, where CICS CM and CICS PA are developed. He has 18 years of experience in information development. He wrote the CICS CM user's guide and has developed a SupportPac™ for CICS PA. His areas of expertise include technical writing and illustration; XML-related technologies such as XSLT, XML schema, and XML DOM programming; and VBScript and VBA programming.

Hannah Macleod is a CICS Tools Project Manager, responsible for CICS Configuration Manager, CICS Business Event Publisher, and CICS VSAM Copy. She works in the CICS Project Office at the IBM Hursley Laboratory in the United Kingdom. Since graduating from Oxford University with an MPhys (Physics) degree in 2002, Hannah has worked in the CICS TS Level 3 Service Team, and subsequently as the Technical Coordinator for the CICS TS 3.1 Beta Program.

Kenneth Meredith is a Senior Software Developer at Fundi Software in Perth, Western Australia. Ken has 25 years of experience in CICS system programming and application development. He has extensive experience managing the life cycle of CICS environments from a customer perspective. He is a member of the development team for CICS CM.

Jennifer Nott is an IT Availability Specialist with the IBM Support Centre in Sydney, Australia, where she provides defect and non-defect support for CICS, CICSPlex SM, MQSeries®, and TXSeries®. Jennifer has 16 years of experience with information technology, and holds an Honors degree in Applied Science from the University of NSW, majoring in Physical Geography.

Adrian Simcock is a Software Engineer working at the IBM Australian Programming Centre in Perth, Australia. He has 20 years of experience working with CICS and related products for a variety of IBM internal and external customer accounts. He is currently working as a developer for IBM Fault Analyzer. He holds a degree in Electrical and Electronic Engineering from Plymouth University.

Thanks to the following people for their contributions to this project:

Rich Conway
International Technical Support Organization, Poughkeepsie Center

Elena Wood, AIM Enterprise Software Platform Marketing Manager
IBM Hursley

Chris Baker Software Engineer
IBM Hursley

Peter Siddell, CICS Tools Specialist
IBM Hursley

Satish Tanna CICS Tools technical specialist
IBM Hursley

Steve Zemblowski CICS Advanced technical support specialist
IBM Dallas

Phil Hanson CICS Product Manager
IBM Hursley

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Learn more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Part 1

Introduce CICS TS 3.1 and the CICS Tools

In this part of the book we discuss CICS Transaction Server V3.R1, we review the new functionality delivered in this version, and we look at the benefits of migrating to CICS TS 3.1

We also take a look at some of the CICS Tools that can assist you in the migration:

- CICS Configuration Manager
- CICS Interdependency Analyzer
- CICS Performance Analyzer
- CICS Debug tool

1

Overview of CICS TS 3.1

In this chapter we provide an overview of the major enhancements and new capabilities delivered in CICS Transaction Server Version 3 Release 1 (CICS TS 3.1). We highlight the benefits of migrating to the new version and discuss the business value of the new capabilities.

1.1 CICS TS 3.1 themes

This section illustrates how the majority of the features delivered in CICS TS 3.1 can be grouped into three main themes:

- Access to CICS
  - Web Services support and Web Services sample application
  - Enhanced HTTP support
  - Improved SSL support
  - Support for mixed case passwords
  - Improved user ID checks for START
- Application transformation
  - Enhanced C/C++ support
  - Enhanced open transaction environment
  - Language Environment® MAIN support for Assembler
  - Enhanced inter-program data transfer
  - Threadsafe Web API commands
  - 64 bit addressing toleration
  - Code page conversion enhancements
  - Information Center on an Eclipse based platform
- Enterprise management
  - CICSPlex SM WUI enhancements
  - CICSPlex SM batchrep access enhancements

1.1.1 Access to CICS

CICS TS 3.1 includes a range of new and improved capabilities that enhance access to CICS. Standard interfaces and communications protocols mean that, following migration to CICS TS 3.1, you have the facilities to reuse your CICS applications within a flexible on demand operating environment. Potential benefits of this include simplified development processes, reduced development costs, and reduced time to deployment.

CICS TS 3.1 delivers major new support for Web Services, which is an evolution of the functions previously provided as the SOAP for CICS optional feature. These capabilities allow CICS-based applications to be exposed as Web Services, thus enabling existing applications to be integrated within a service-oriented architecture (SOA).

CICS TS 3.1CICS TS 3.1 supports the WS-Atomic Transaction specification enabling distributed transaction coordination for partners complying with this standard. A message-level security function that complies with the WS-Security specification has also been provided in CICS TS 3.1.

Improvements to CICS Web support include support for HTTP 1.1 and the addition of outbound HTTP support. CICS TS 3.1 also delivers enhancements to the existing support for Secure Sockets Layer (SSL), including support for the TLS 1.0 protocol.

1.1.2 Application transformation

This second group of important enhancements to CICS TS provides a range of new functions that enable further development of existing applications, and construction of new applications, using contemporary programming languages, constructs, and tools.

In CICS TS 3.1, support is introduced for Language Environment enabled Assembler application programs.

CICS TS 3.1 also provides a new mechanism for inter-program data transfer, using constructs known as channels and containers. These provide an alternative to COMMAREAs and are not subject to the same 32-KB restriction.

All the EXEC CICS Web API commands have been made threadsafe. Support for the XPLink feature of z/OS can lead to improved performance when running applications written in C/C++.

More efficient use of z/OS multiprocessor capabilities is enabled by extension of Open Transaction Environment (OTE) support to use open TCBs.

The Information Center is now provided as a plug-in to the Eclipse platform. The benefits of this include commonality with the framework now being employed by many other IBM products.

1.1.3 Enterprise management

The third main area of enhancements introduced in CICS TS 3.1 is improvements to the enterprise management capabilities of CICSPlex SM. These new capabilities enable effective management of large runtime configurations by the use of modern interfaces, so that demanding service level objectives can be met.

Numerous improvements have been made to the CICSPlex SM Web User Interface (WUI), providing new functions and enhancing its usability. This makes it the interface of choice for all system management actions. A new interface has been provided for the CICSPlex SM data repository batch update facility. With these enhancements, CICSPlex SM can be configured, set up, and run without involving the TSO or CAS components, saving time and effort for both existing and new users.

Additional capabilities have also been added to CICSPlex SM to support the new functions introduced in CICS TS 3.1.

1.2 Web Services support

The availability of the new CICS Web Services capabilities is one of the main benefits of migrating to Version 3.1. In this section we discuss these capabilities and introduce key new concepts. The SOAP for CICS function was available as an optional feature and was orderable with CICS TS Version 2. It has evolved and is now integrated into Version 3.1 as part of the CICS support for Web Services. Together with a range of extensions and new capabilities, this enables CICS business logic to be exposed as Web Services, as part of an service-oriented architecture (SOA) solution.

When exploited, the introduction of Web Services is very powerful. The impact Web Services can have on program-to-program interactions is analogous to the effect the Internet has had on interactions between programs and end users. The support CICS TS 3.1 provides for Web Services makes it possible for CICS applications to be integrated more rapidly, easily, and cheaply than ever before.

Web Services is a technology that enables you to invoke applications using Internet protocols and standards. The technology is called Web Services because it integrates services (applications) using Web technologies (the Internet and its standards). If we had to describe Web Services using just one sentence, it would be: Web Services are self-contained, modular applications that can be described, published, located, and invoked over a network.

The formal definition of a Web Service, provided by World Wide Web Consortium (W3C) Services Architecture Working Group, is as follows:

> A Web Service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web Service in a manner prescribed by its description using SOAP messages, *typically conveyed using HTTP with an XML serialization* in conjunction with other Web-related standards.

Note: It is important to note from this definition that a Web Service is not constrained to use SOAP over HTTP or HTTPS as the transport mechanism. WebSphere® MQ can also be used as the transport mechanism.

1.2.1 Service-oriented architecture

Increasingly, businesses are adopting a service-oriented architecture (SOA) approach. With a SOA, your programs can be on different systems and be provided by different vendors and yet communicate and exchange data with each other. Web Services technology is an ideal technology choice for implementing an SOA. Therefore, by deploying Web Services, your valuable CICS applications can evolve to participate in new, more flexible business models.

As shown in Figure 1-1, a service-oriented architecture has three basic components:

- A service provider

  The *service provider* creates a Web Service and possibly publishes to the service broker the information needed to access and interface with the Web Service.

- A service broker

  The *service broker* (also known as a *service registry*) makes the Web Service access and interface information available to any potential service requester.

- A service requester

  The *service requester* binds to the service provider in order to invoke one of its Web Services having optionally located entries in the broker registry using various find operations.

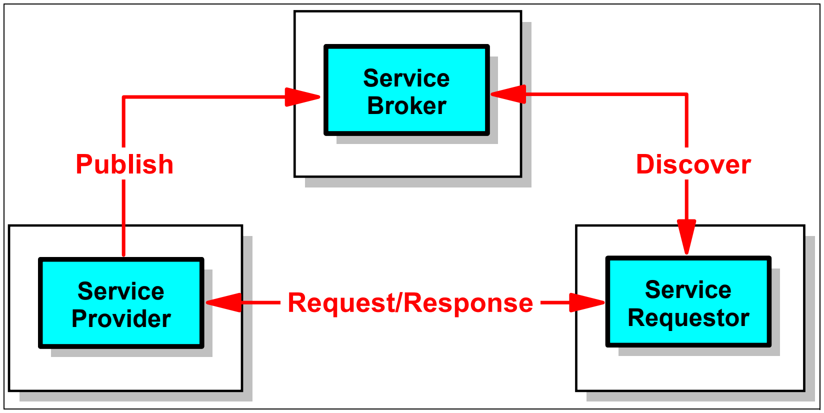


Figure 1-1 Service-oriented architecture components and operations

Each component can also act as one of the other two components. For example, if a service provider needs information that it can only acquire from some other service, it acts as a service requester while still serving the original request.

For more information about the relationship between Web Services and service-oriented architectures, refer to the IBM Redbook *Patterns: Service-Oriented Architecture and Web Services*, SG24-63033.

1.2.2 Core standards

Web Services are built upon four core standards:

- Extensible Markup Language (XML)

  XML is the foundation of Web Services. However, since much information has already been written about XML, we do not describe it in this document. You can find information about XML at:

  http://www.w3.org/XML/

- SOAP

  SOAP provides an XML, text-based, platform and language neutral message format. Originally proposed by Microsoft®, SOAP was designed to be a simple and extensible specification for the exchange of structured, XML-based information in a decentralized, distributed environment. As such, it represents the main means of communication between the three actors in an SOA: the service provider, the service requester, and the service broker.

  There are currently two versions of SOAP: Version 1.1 and Version 1.2.

  The SOAP 1.1 specification contains three parts:

  – An *envelope* that defines a framework for describing message content and processing instructions. Each SOAP message consists of an envelope that contains an arbitrary number of headers and one body that carries the payload. SOAP messages might contain faults, faults report failures, or unexpected conditions.

  – A set of *encoding rules* for expressing instances of application-defined data types.

  – A *convention* for representing remote procedure calls and responses

  A SOAP message is, in principle, independent of the transport protocol which is used, and can, therefore, potentially be used with a variety of protocols such as HTTP, JMS, SMTP, or FTP. Right now, the most common way of exchanging SOAP messages is through HTTP.

- Web Services Description Language (WSDL)

  WSDL uses XML to specify the characteristics of a Web Service: what the Web Service can do, where it resides, and how it is invoked. WSDL can be extended to allow descriptions of different bindings, regardless of what message formats or network protocols are used to communicate.

WSDL enables a service provider to specify the following characteristics of a Web Service:

– Name of the Web Service and addressing information

– Protocol and encoding style to be used when accessing the public operations of the Web Service

– Type information: operations, parameters, and data types comprising the interface of the Web Service, in addition to a name for this interface

WSDL is not bound to any particular protocol or network service. It can be extended to support many different message formats and network protocols. However, because Web Services are mainly implemented using SOAP and HTTP, the corresponding bindings are part of this standard.

- Universal Description, Discovery, and Integration (UDDI)

  The Universal Description, Discovery, and Integration standard defines a means to publish and to discover Web Services. At the time of writing, UDDI Version 3.0 has been finalized, but UDDI Version 2.0 is still more commonly used. For more information refer to:

  http://www.uddi.org/
  http://www.oasis-open.org/specs/index.php#wssv1.0

Additional standards

Figure 1-2 provides a snapshot of the rapidly changing landscape of Web Services-related standards and specifications. We do not intend it to be a strictly correct stack diagram — it just attempts to show the various standards efforts in terms of the general category to which they belong.

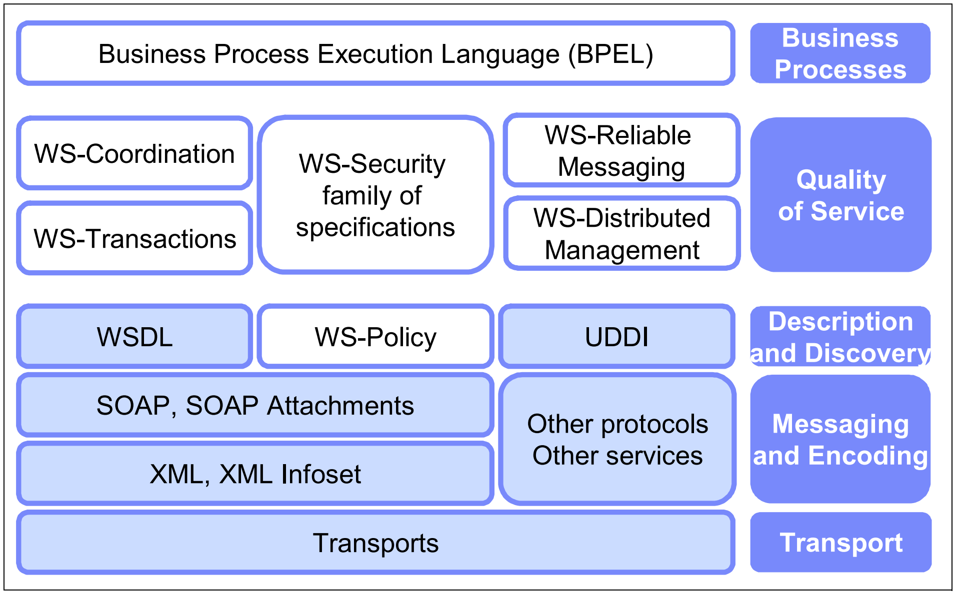


Figure 1-2 Web Services standards

Given the current momentum behind Web Services and the pace at which standards are evolving, you may also wish to refer to an online compilation of Web Services standards. An online compilation is available on the IBM developerWorks® Web site at:

`http://www.ibm.com/developerworks/views/webservices/standards.jsp`

Of particular interest to those developing Web Services in CICS are:

- WS-Transactions (the family of specifications that relate to transactional Web Services)
- WS-Security (the family of specifications that relate to securing Web Services)

1.2.3 Support for SOAP

In this section we discuss support for SOAP.

What is new in CICS TS 3.1 SOAP support

CICS TS 3.1 provides the following new functions:

- It includes a new Web Services Assistant utility.

  The Web Services Assistant utility contains two programs, DFHWS2LS and DFHLS2WS. DFHWS2LS helps you map an existing WSDL document into a high-level programming language data structure, while DFHLS2WS creates a new WSDL document from an existing language structure. The Web Services Assistant supports the following programming languages:

  - COBOL
  - PL/I
  - C
  - C++

- It supports two different approaches to deploying your CICS applications in a Web Services environment.

  - You can use the Web Services Assistant.
  - You can take complete control of the processing of your data.

- It reads a pipeline configuration file created by the CICS system programmer to determine which message handlers should be invoked in a pipeline. A message handler is a program in which you can perform your own processing of Web Service requests and responses. A pipeline is a set of message handlers that are executed in sequence.

  Note: A pipeline can be configured as a service requester pipeline or a service provider pipeline but not both. You cannot configure a CICS pipeline to function as an intermediary node in a SOAP message path.

- It supplies message handlers designed especially to help you process SOAP messages.

  Whether you use the Web Services Assistant or take complete control of the processing yourself, you can write your own message handlers to perform additional processing on your request and response messages, or you can use CICS-supplied message handlers.

  The CICS-provided SOAP message handlers can be configured to invoke one or more user-written header processing programs and to enforce the presence of particular headers in the SOAP message.

- It enables you to configure many different pipelines.

  You can configure a pipeline to support SOAP 1.1 or SOAP 1.2. Within your CICS system, you can have some pipelines that support SOAP 1.1 and others that support SOAP 1.2.

- It provides the following new resource definitions to help you configure your support for Web Services:
  - PIPELINE
  - URIMAP
  - WEBSERVICE

  If you used the SOAP for CICS feature, you may be able to use CICS resource definitions to replace the logic you provided in your pipeline programs to distinguish one application from another. For example, in a service provider, you may be able to replace code that distinguishes between applications based on a URI, with a suitable set of URIMAP resources.
- It provides the following new EXEC CICS application programming interface (API) commands:
  - SOAPFAULT ADD | CREATE | DELETE
  - INQUIRE WEBSERVICE
  - INVOKE WEBSERVICE
- It conforms to open standards including:
  - SOAP 1.1 and 1.2
  - HTTP 1.1
  - WSDL 1.1

  It ensures maximum interoperability with other Web Services implementations by conforming with the Web Services Interoperability Organization (WS-I) Basic Profile 1.0. This profile is a set of non-proprietary Web Services specifications that promote interoperability between different implementations of Web Services. Because SOAP 1.2 is not included in WS-I Basic Profile 1.0, most Web Service runtimes still support and recommend using SOAP 1.1. CICS TS 3.1 has support for both SOAP 1.1 and SOAP 1.2.
- It supports the WS-Atomic Transaction specification because the associated infrastructure includes a distributed transaction coordination capability.

1.2.4 CICS Web Services Assistant

The Web Services Assistant helps you deploy an application with the least amount of programming effort. It is a build-time facility that consists of two programs, DFHWS2LS and DFHLS2WS. If you want to expose an existing application as a Web Service, you can start with a high-level language data structure and use DFHLS2WS to generate the Web Services description (WSDL). Alternatively, if you want to communicate with an existing Web Service, you can start with its Web Service description and use DFHWS2LS to generate a high-level language structure that you can use in your program.

In both these cases, the CICS Web Services Assistant also generates a Web Services binding (WSBIND) file. This is a file that is used by CICS to enable automatic runtime conversion of the SOAP messages to containers and COMMAREAs, and vice versa. Containers are a new capability introduced in CICS TS 3.1 that can be used with channels to provide additional capacity when transferring data between programs. Further information about channels and containers is in 1.10, “Enhanced inter-program data transfer” on page 20.

You can write your own code to map between your application data and the message that flows between the service requester and provider. For example, if you want to use non-SOAP messages within the Web Service infrastructure, you can write your own code to transform between the message format and the format used by your application.

The CICS Web Services Assistant supports the following programming languages:

- COBOL
- C
- C++
- PL/I

The CICS Web Services Assistant includes Eclipse technology.

1.2.5 Web Services development approaches

CICS Web Services support can greatly assist you in the following situations:

- You have an existing application that you wish to expose as a Web Service.
- You wish to develop a new application and make it available as a Web Service.
- You want to access an existing Web Service, possibly on some other platform.

In all three of these cases you would have an existing language structure, a Web Service description language, or both. By language structure we mean the data definition statements for the programming language that defines both the inputs and outputs of the current application.

Chapter 7, “CICS TS 3.1 exploitation” on page 149, discusses these approaches further and provides a discussion on how to address the development requirements in the different situations.

1.2.6 Web Services versus CICS TCP/IP connectivity

At this point it is worth looking at the key differences between CICS Web Services and the original CICS Web Support. The main components of CICS Web Support are:

- HTTP Server Support
- CICS Transaction Gateway
- z/OS® Communications Server IP CICS Socket Interface
- z/OS Unix System Services
- Link3270 Bridge

All of these components involve a high-function tightly coupled approach. The interface is such that programs must have detailed knowledge of each others' COMMAREAs and in many cases have to be Web-aware.

This contrasts with the philosophy of Web Services where, due to the published WSDL, the application can determine the required interface and is totally unaware of the language and environment of the runtime executable.

1.3 Enhanced HTTP support

CICS Web support has been enhanced by upgrading the support for HTTP. It is now conditionally compliant with the HTTP 1.1 specification (RFC 2616). This means that CICS satisfies all the *must* level requirements but not all the *should* level requirements. CICS is only conditionally compliant because the HTTP/1.1 specification includes functions that relate to roles that are not relevant to CICS and can therefore be ignored. For example, CICS was not designed to act as a proxy, a gateway, a tunnelling server, a caching server, or a browser, so *should* conditions relating to these are not relevant. Outbound HTTP support has been added so that CICS can act as both an HTTP server (CICS Web inbound support) or as an HTTP client (CICS Web outbound support).

Persistent sessions are now the default for interactions between CICS and a remote partner, and CICS supports pipelining and chunking of messages. As well as serving HTTP requests as an HTTP server, outbound requests can be created using EXEC CICS commands for a CICS application as an HTTP client.

New Resource Definition Online (RDO) definitions for objects known as URIMAPs can manage the HTTP server facility. CICS automatically creates virtual hosts using these definitions, so multiple host names can be provided at the same IP address, which can be managed using CICS system commands. Static responses can be provided for HTTP requests, formed from a document template or Hierarchical File System (HFS) file.

This means that CICS application programs can be written using a common HTTP protocol for business-to-business communication, to control hardware or software, or to access information in non-browser HTTP applications.

Changes to the management of connected, but inactive, IP sockets allow many more clients to connect to a CICS system. Using an internal pseudo-conversational model no CICS task resources are consumed by IP sockets waiting for a message from a partner.

1.4 Improved SSL support

CICS TS 3.1 introduces a range of improvements to security. In addition to the existing support for Secure Sockets Layer (SSL) 3.0, support is provided for the Transport Layer Security (TLS) 1.0 protocol. This includes support for the Advanced Encryption Standard (AES) cipher suites, which offer 128-bit and 256-bit encryption.

Resource definitions have been enhanced to allow the user to specify the precise list of cipher suites to be used in the negotiation. This capability is also included in the new URIMAP resource definition. To support management of the new capabilities and resources, there are new System Programming Interface (SPI) commands.

Certificate revocation lists (CRLs) are now supported when negotiating with clients, allowing any connections using revoked certificates to be closed immediately. A new supplied transaction, CCRL, is provided for updating the CRL in an LDAP server. More flexibility is offered in these negotiations. For example, a minimum as well as a maximum encryption level can be specified for negotiation.

It is now possible to specify whether session IDs are shared across a sysplex, improving the current use of the cache at the address space level. Caching enables an SSL handshake to be optimized based on a previous negotiation, thereby improving the performance when setting up connections.

An increased number of simultaneous SSL connections can now be active, as a result of the introduction of support for pthreads within the Language Environment enclave from which System SSL is invoked. This achieves better throughput and improves the support for new functions such as Web Services.

All these functions also apply with the new outbound HTTP function described in 1.3, “Enhanced HTTP support” on page 14.

1.5 Support for mixed case passwords

CICS TS 3.1 has the ability to support an underlying capability for case-sensitive passwords. When this function is active, it will be indicated on the CICS-supplied signon panel.

> **Note:** This capability will also be made available on CICS TS V2.2 and CICS TS V2.3 via the service channel.

1.6 Improved user ID checks for START

The revoked status of a user ID or group connection is now tested for by the EXEC CICS START USERID() command when it is issued, in order for the issuer to be notified by the USERIDERR condition if applicable.

1.7 Enhanced C/C++ support

New support for C/C++ has been introduced, which brings the performance of these applications to a level comparable to that obtained with COBOL, PL/I, or Assembler applications. This is provided by the Extra Performance Linkage (XPLink) feature of z/OS, which provides high performance subroutine linkage mechanisms and guard pages for stack extension, resulting in highly optimized execution path lengths. These benefits are achieved by running these applications in the CICS Open Transaction Environment (OTE), instead of in the Quasi-Reentrant (QR) task control block (TCB). This also has the benefit that the applications can be run on dedicated TCBs. The applications use standard Language Environment services, with CICS storage management.

In order to run in the OTE environment these applications must be written to threadsafe standards. Maximum performance will be achieved only if the applications are limited to the use of threadsafe CICS commands.

XPLink support under CICS enables the latest compiler and optimization technologies included with C/C++ to be exploited. In particular, XPLink DLLs used outside CICS can now be used inside CICS as well. This means greater C/C++ code reusability.

1.8 Enhanced open transaction environment

CICS TS 3.1 extends the use of the open transaction environment (OTE) by providing support for COBOL, PL/I, Assembler, and non-XPLink C/C++ OPENAPI application programs. The program will run on its own OTE TCB from the start. This is enabled by the new API resource attribute on the PROGRAM definition. API(OPENAPI) requires the application to be coded to threadsafe standards. Use of any non-threadsafe CICS commands will cause a switch to the QR TCB. Then CICS will switch back to the OTE TCB before returning control to the program.

The main benefit of this support allows application workloads to be moved off the single QR TCB onto multiple OTE TCBs, thereby allowing better utilization of machine resources to achieve better throughput. Note that existing recommendations concerning use of non-CICS APIs continue to apply when executing on an OTE TCB.

The prospect of improving transaction throughput by eliminating TCB switching is enormous. However, it is important to emphasize that to achieve this goal careful consideration must be given to the suitability of the application. Aside from system performance, application integrity (that is, achieving consistent results regardless of system load) is a major consideration when implementing threadsafe applications. IBM Redbook *Threadsafe Considerations for CICS*, SG24-6351, discusses this further.

z/OS Communications Server Version 1 Release 7 has been enhanced to allow the IP CICS Sockets Task Related User Exit (TRUE) to be enabled as OPENAPI. At the time of writing we now have two TRUEs that can be enabled as OPENAPI: DB2® and IP CICS Sockets.

1.8.1 Why migrate to threadsafe

In this section we identify and outline the potential business drivers that will lead CICS customers to migrate their applications to a threadsafe environment.

There are three principle drivers, which are covered in the following sections:

- Improve performance.
- Reduce cost.
- Position for the future.

This section concludes with a warning: There is a risk associated with defining an application as threadsafe, and this risk must be understood and eliminated before migration is attempted.

Improve performance

Note: We discuss CP SHARE here. We work this out in the following way: CP SHARE is the amount of a CP an LPAR is guaranteed, before it is eligible to have the CP removed. For CICS to perform well, the CP SHARE for the LPAR where it is executing must be fairly high (90+% is great, 80% is good, 70% is workable). CP SHARE = ((# available physical CP * 100)/(# logical CP in LPAR)) * FAIR SHARE.

Customers who should benefit most from migrating to a threadsafe environment are those who experience poor response times for any of the following reasons:

- The CICS QR TCB is CPU constrained.

  Under this scenario, the CICS QR TCB is consistently reaching system CP SHARE (QR TCB is running at 100% CPU) and has to wait to be dispatched by the operating system. Every task running under the QR TCB is being delayed.

  Defining transactions as threadsafe, processing as many tasks as possible on an open TCB will remove this constraint on the QR TCB, and reduce the response times of both threadsafe and non-threadsafe transactions.

- Application tasks are waiting excessively for the QR TCB.

  Under this scenario, the QR TCB is not CPU constrained, but application tasks are contending for their share of QR.

  Again, defining transactions as threadsafe and moving as many tasks as possible to an open TCB will reduce contention for the QR TCB, and reduce the response times of both threadsafe and non-threadsafe transactions.

- The CICS region in general is CPU constrained.

  Under this scenario, the system as a whole is at or approaching 100% busy, and CICS is being constrained along with everything else.

  Depending on how an application is designed, defining it as threadsafe can significantly reduce the path length of application tasks. The transactions that will achieve the greatest CPU reduction are likely to be DB2 applications that have the following characteristics:

  – A significant number of EXEC SQL calls are invoked per task.

  – All programs invoked between the first and last EXEC SQL call in each task are defined as threadsafe.

  – All exits invoked as part of an EXEC SQL call are defined as threadsafe and only contain threadsafe EXEC CICS commands.

  – All exits invoked between the first and last EXEC SQL call in each task are defined as threadsafe.

- All EXEC CICS statements invoked between the first and last EXEC SQL call in each task are threadsafe.

Defining transactions with the preceding characteristics as threadsafe will all but eliminate TCB switches for the associated CICS tasks.

Reduce the cost of computing

Reducing the CPU utilization of an application does not always necessarily result in improved response times. An application may be a heavy user of CPU, but if the processor has spare capacity and the application is not CPU constrained, then a reduction in path length may have a negligible impact on response times.

However, for many customers, the financial cost incurred running their applications is related to the amount of CPU consumed. Under these circumstances, the CPU savings gained by migrating appropriate applications to a threadsafe environment can equate to a financial saving.

1.8.2 Stages of OTE implementation

OTE in CICS has been implemented in three stages, over several releases of CICS TS:

- Stage 1 - OTE function introduced - delivered in CICS TS V1.3
- Stage 2 - TRUEs can exploit OTE - delivered in CICS TS V2.2
- Stage 3 - Full application use of open TCBs - delivered in CICS TS 3.1

Applications that can be defined as threadsafe in CICS TS V2 will be able to exploit the enhancements provided at CICS TS 3.1 with minimum migration effort. Moreover, IBM recommends that all new application programs should be written to threadsafe standards at whatever level of CICS they are developed.

1.8.3 Understand the application

What do we mean when we say an application is threadsafe? A threadsafe program is defined as a program that does one of following:

- Uses appropriate serialization techniques, such as compare and swap or enqueue, when accessing any shared application resources. It must be capable of running concurrently on multiple TCBs, and must not rely on quasi-reentrancy to serialize access to shared resources and storage.
- Uses no shared application resources whatsoever.

For an application to meet these conditions and therefore be considered threadsafe, the application must:

- Incorporate threadsafe application logic (which means that the native language code in between the EXEC CICS commands must be threadsafe).
- Be defined to CICS as threadsafe.

Important rule: Only once it is understood whether an application is threadsafe, and all access to all shared resources are serialized, should any of its programs be defined as threadsafe. Failure to follow this rule may result in unpredictable results, and put the integrity of application data at risk.

Prior to exploiting the new OTE capabilities introduced in CICS TS 3.1, we suggest that you refer to IBM Redbook *Threadsafe Considerations for CICS*, SG24-6351, for further information about this subject.

1.9 Language Environment MAIN support for Assembler

Support has been introduced to enable coding of totally Language Environment enabled application programs in Assembler. In other words, Language Environment MAIN support is provided for Assembler programs. A new translator option LEASM is provided, which causes the Language Environment function to be used to set up the program's environment. This improves the ease of integration of these applications into the Language Environment so that Language Environment services can be run more easily. Improved Debugger support is available.

1.10 Enhanced inter-program data transfer

The restriction of a maximum of 32 KB that has previously applied to the amount of data that can be passed between programs by using a COMMAREA has been removed by the introduction of *containers* and *channels* in CICS TS 3.1. Containers are named blocks of data for passing information between programs. Any number of containers can be passed between programs. Containers are grouped together in named channels.

Channels can be used as a standard mechanism for exchanging data between programs. A channel can be passed on EXEC CICS LINK, START, XCTL, and RETURN commands. Data can be exchanged on a DPL, remote START, or pseudo-conversation between CICS TS 3.1 systems connected by either MRO or ISC.

Channels provide a more flexible and more structured method of passing data between program components. Variation in the size and number of containers can conveniently be accommodated to allow easier evolution of the interfaces between programs. The size of a container is limited only by the amount of storage available. There is no limit to the number of containers that can be added to a channel. This mechanism also removes the need for programs to know the exact size of the data returned. When containers go out of scope, they are automatically destroyed, so that the programmer is relieved of storage management concerns.

Channels can be used by applications written in any of the programming languages supported by CICS. Options on the container and related API commands are provided for data conversion, providing a much simpler mechanism than that employed with a COMMAREA. Moreover, while in COMMAREA applications data conversion is controlled by the system programmer. With the new mechanism it is controlled by the application programmer.

This mechanism can only be used for communication between programs running under CICS TS 3.1. Communication with programs running under earlier levels of CICS TS still requires the use of COMMAREAs.

Note: Channels and COMMAREAs can coexist within the same task.

This new approach introduced in CICS TS 3.1 provides an easy and more flexible mechanism for exchange of large volumes of structured parameter data between CICS programs. This new approach is provided by the two new capabilities known as channels and containers.

A container is a named reference to a CICS-managed storage area that can hold any form of application parameter data. A container may be any size and can hold data in any format required by the application. An application can reference any number of containers. CICS provides EXEC API verbs to create, delete, reference, access, and manipulate a container as well as to associate it with a channel.

A channel is a uniquely named reference to a collection of application parameter data held in containers. A channel is analogous to a COMMAREA but it does not have the constraints of a COMMAREA. CICS provides EXEC API that associates a named channel with a collection of one or more containers — an easy way of grouping parameter data structures to pass to a called application. CICS will destroy a channel when it can no longer be referenced – when it becomes *out of scope*.

1.10.1 General concepts

General concepts are:

- Containers are named blocks of data designed for passing information between programs. You can think of them as *named COMMAREAs*.
- Programs can pass any number of containers between each other, and the size of the containers is limited only by the amount of storage available.
- Containers are grouped together in sets called channels. A channel is analogous to a parameter list.
- To create named containers and assign them to a channel, a program uses the **EXEC CICS PUT CONTAINER(container-name) CHANNEL(channel-name)** command. It can then pass the channel (and its containers) to a second program using the CHANNEL(channel-name) option of the EXEC CICS LINK, XCTL, START, or RETURN commands.

Example 1-1 Passing a channel on a LINK

```
EXEC CICS PUT CONTAINER(structure-name)
          CHANNEL(channel-name)
          FROM(structure)
EXEC CICS LINK PROGRAM(PROG2)
          CHANNEL(channel-name)
```

- The second program can read containers passed to it using the **EXEC CICS GET CONTAINER(container-name)** command. This command reads the named container belonging to the channel that the program was invoked with.

Example 1-2 Receiving a container

```
EXEC CICS GET CONTAINER(structure-name)
   CHANNEL(channel-name)
   INTO(structure)
```

- If the second program is invoked by a LINK command, it can also return containers to the calling program. It can do this by creating new containers or by reusing existing containers.

Example 1-3 Returning a container

```
EXEC CICS PUT CONTAINER(structure-name)
          FROM(structure)
EXEC CICS RETURN
```

- Channels and containers are visible only to the program that creates them and the programs they are passed to. When these programs terminate, CICS automatically destroys the containers and their storage.

- Channels and containers are not recoverable. If you need to use recoverable containers, use CICS business transaction services (BTS) containers.
- Channels and COMMAREAs are mutually exclusive, in the sense that EXEC CICS LINK, EXEC CICS XCTL, and EXEC CICS RETURN can only pass a channel *or* a COMMAREA. However, program A can pass data in a COMMAREA to program B, which then creates a channel to pass the data on to program C. Since program B will receive the data from program C in a container, it will have to move the container data into a COMMAREA, which is where program A will expect to find it.

Note: COBOL dynamic program calls are not suitable for channels and containers usage. Only the EXEC interface calls (LINK, XCTL, START, RETURN) can pass channels and containers to a separate program. However, the COBOL dynamically called program can access channel and containers data.

1.10.2 Benefits of using channels and containers

The life cycle and scope of channels and containers are completely under the control of the CICS system, ensuring data integrity and storage resources management.

These are the major benefits obtained by applications exploiting the capabilities of the channels and containers methodology:

- An unconstrained, CICS-supported method of passing parameter data
- Segregation of parameter data structures, each part represented by a named container structure
- A loose functional coupling approach
- The freedom to dynamically determine the nature of the passed data and to select the appropriate processing required
- A CICS standard API for optimized exchange of data between CICS programs implemented in any CICS-supported language
- CICS-managed life cycle for channel and container resources
- Ease of parameter passing by use of unique named references
- Ease of understanding by use of unique named references to parameter payload
- Explicit codepage conversion operations

The internal CICS implementation of channels and containers is optimized for efficient memory management and data transfer. CICS will ensure that only the necessary new and modified containers are transferred between the calling applications, to optimize the performance of the calling mechanism. Providing that containers separate different parameter structures, the calling applications benefit from complete access to the data content in all containers that are in scope.

1.11 Threadsafe Web commands

All the EXEC CICS Web API commands have been made threadsafe. These are WEB READ, WEB WRITE, WEB SEND, WEB RECEIVE, WEB RETRIEVE, WEB STARTBROWSE, WEB READNEXT, WEB ENDBROWSE, WEB EXTRACT, EXTRACT WEB, EXTRACT TCPIP, and EXTRACT CERTIFICATE. This removes the requirement for CICS to return to the quasi-reentrant task control block (QR TCB) to execute these commands, so applications (both Java™ and non-Java) that use these commands should be able to obtain the performance improvements resulting from reduced TCB switching. Also threadsafe are the new Web API commands in support of outbound HTTP: WEB OPEN, WEB CLOSE, WEB CONVERSE, and WEB PARSE URL.

1.12 64-bit addressing toleration

Although CICS TS 3.1 does not support execution of 64-bit applications, support is introduced that allows 64-bit code (such as in Task Related User Exits (TRUEs)) to execute in a CICS address space. Extensions are provided to the CICS abend capture mechanisms to allow the contents of the full 64-bit general purpose registers to be reported.

1.13 Codepage conversion enhancements

To the existing CICS codepage conversion capabilities, which enable conversion between a range of EBCDIC and ASCII codepage combinations, are added the conversion of data between EBCDIC or ASCII and Unicode, in either direction. This support makes use of z/OS conversion services. The capability applies to either UTF-8 or UTF-16, and support is also provided for conversion between these forms of Unicode. Little endian to big endian transpositions for UTF-16 data are carried out if needed.

This capability is expected to be used mainly for HTML, XHTML, and XML data, as part of the CICS support for HTTP 1.1 (see 1.3, “Enhanced HTTP support” on page 14) by Web Services and by the new channel container commands introduced in this version of CICS (see 1.10, “Enhanced inter-program data transfer” on page 20). However, the enhanced codepage conversions are available for any application need so long as the application can identify the source and target codepage CCSIDs and the specific conversion is enabled in z/OS conversion services.

1.14 Information Center on Eclipse platform

In CICS TS 3.1, the Information Center is powered by Eclipse technology. It consists of an Eclipse Help System, with the information for CICS TS as a plug-in. This brings a range of benefits to the user. A major benefit is the use of a common framework, which is now the infrastructure of choice adopted by many IBM products, offering a common look and feel, together with consistency of behavior and a new search engine. This infrastructure also allows users to customize their own Information Centers using plug-ins from multiple products, or to write their own plug-ins. The CICS TS 3.1 Information Center also delivers plug-ins for other products from the CICS portfolio. The new Information Center enables direct links (eSupport) from CICS TS information to support information.

The Information Center is also now supported on a wider range of platforms, including z/OS. New functions included are:

- A *What's New* section organized by major functional area, available through the navigation and welcome page. This is similar to the long-established *Release Guide,* but is not a separate document, having integrated links into the rest of the Information Center.
- Learning Paths: a sequence of topics that help a user learn about a new area of the product. In this release, they are provided for Web Services, CICSPlex SM, and channels and containers.
- Information Roadmaps: a topic that provides a set of comprehensive links, role-based or function-based, to information from a variety of sources. In this release they are provided for Web Services, Java in CICS, and CICSPlex SM.
- A troubleshooting and support section: a self-help resource that consists of components for searching external support sites, getting fixes, and contacting IBM support. It will also contain a selection of Technotes.

The Information Centers for CICS TS for z/OS V2.2 and V2.3 will also be offered as plug-ins for the Eclipse platform, enabling them to obtain some of the benefits of using that base. The Information Centers for these products on the current technology base will continue to be available, though those on the new base will be required in order to obtain the latest updates.

1.15 CICSPlex SM WUI enhancements

The Web User Interface already provides important functions that are not available with the old TSO End User Interface, in addition to its greatly improved usability. CICS TS 3.1 introduces a further range of improvements to the Web User Interface that deliver significant user benefits. New functions added are:

- Improvements to screen design. These enhancements maximize the use of screen space in views and menus:
  - The view editor now allows detailed views to be displayed in two-column format. Users are able to create their own detail views in two columns, if they wish.
  - The Select All and Deselect All buttons have been replaced in tabular views by icons in the record heading of the table, thereby reducing white space.
  - Filters on tabular views can now be collapsed, so that more screen space is available for the display of data.
- User favorites. These allow the saving by the user of tabular and detail views to a menu. This menu can be edited and is easily accessible, allowing the chosen views to be accessed with a single click.
- User group profiles. Profiles for groups of users, containing information such as default context, scope, CMAS context, and result set warning count, can now be set by administrators. This allows them to configure the Web User Interface in ways that are tailored to the needs of particular groups of users.
- Result set warning counts. These can be set to allow a warning to be issued before a view is opened that would generate a large number of records. This allows a filter to be altered on the view in order to reduce the number of records returned, avoiding unnecessary waits.
- Filter confirmation. The view editor now allows the user, when creating or updating views, to include a filter confirmation panel before a view is opened. This means that, when navigating to a view, the user will have the opportunity to enter filters, whatever the size of the record set that will be returned.
- Dynamic selection lists. Usability is enhanced by the Web User Interface now generating lists of valid potential values for users to select attributes in input panels. Users no longer have to remember values that could be entered.
- The previous set of samples known as the starter set is now included as a fully documented set of IBM-supplied views.
- The BAS administration views (introduced in CICS TS V2.3) have been restructured to improve their usability. They have been divided into two groups: basic BAS (which emulates RDO function) and advanced BAS (which exploits the advanced features of CICSPlex SM).

With these enhancements to the Web User Interface, together with jobstep access to batchrep (see 1.16, “CICSPlex SM batchrep access enhancements” on page 27), CICSPlex SM can be configured completely without any need to activate the CAS or TSO components. Establishing it in this configuration significantly reduces the time to exploitation of CICSPlex SM functionality for new users. For existing users, it simplifies migration to the new level of CICS TS.

A WUI equivalent for the CICSPlex SM TSO MAP command has been delivered via the CICS TS 3.1 service channel (PKxxxxx).

1.16 CICSPlex SM batchrep access enhancements

In CICS TS 3.1, a group of new facilities is introduced that provide a batch update mechanism for maintenance of definitions on the CICSPlex SM data repository. These are:

- A BATCHREP resource table, which may be accessed by the CICSPlex SM API
- Support in the Web User Interface for the BATCHREP resource table
- A z/OS utility program, which enables the definitions to be maintained from a job step

These new capabilities offer improved usability for the batchrep facility, together with introducing the ability to maintain CICSPlex SM definitions from a job step. They also provide access to the BATCHREP facility through the CPSM Web User Interface.

2

Overview of CICS PA

This chapter provides an overview of CICS Performance Analyzer for z/OS (CICS PA) and then describes how you can use it to compare CICS performance before and after migrating to CICS TS 3.1, and identify the cause of any performance issues.

2.1 CICS PA defined

CICS PA is a reporting tool that provides information about the performance of your CICS systems and applications to help you tune, manage, and plan your CICS systems effectively.

CICS PA is not an online monitoring tool. It uses data collected by your system in MVS™ System Management Facility (SMF) data sets:

- CICS Monitoring Facility (CMF) performance, exception, and transaction resource class records (SMF record type 110)
- CICS statistics and server statistics records (type 110)
- System Logger records (type 88)
- DB2 accounting records (type 101)
- WebSphere MQ accounting records (type 116)

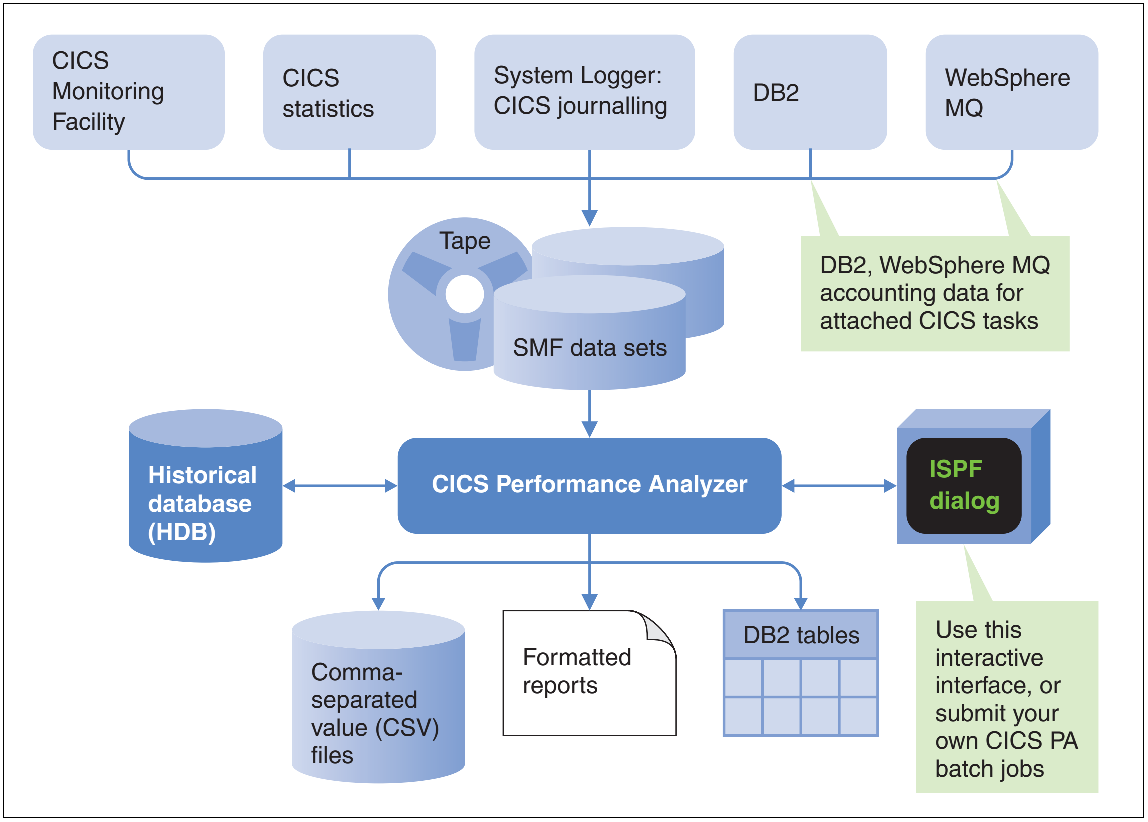


Figure 2-1 CICS PA uses SMF data to report on the performance of your CICS systems and applications

CICS PA can help:

- System programmers to track overall CICS system performance and evaluate the results of their system tuning efforts
- Application programmers to analyze the performance of their applications and the resources they use
- Database administrators to analyze the usage and performance of database systems such as IMS™ and DB2
- MQ administrators to analyze the usage and performance of their WebSphere MQ messaging systems
- Managers to ensure that transactions are meeting their required Service Levels and measure trends to help plan future requirements and strategies

CICS PA reports all aspects of CICS system activity and resource usage, including:

- Transaction response time
- CICS system resource usage
- Cross-system performance, including multi-region operation (MRO) and advanced program-to-program communication (APPC)
- CICS Business Transaction Services (BTS)
- CICS Web Support
- External subsystems, including DB2, IMS, and WebSphere MQ
- System Logger performance
- Exception events that cause performance degradation
- Transaction file and temporary storage usage

Rather than keeping large SMF data sets for reporting purposes, you can use CICS PA to load selected SMF records into a CICS PA historical database (HDB), optionally summarizing the records according to the time intervals that you require for reporting (such as hourly, daily, or weekly). You can then use CICS PA to produce reports from the HDB instead of the SMF data sets. Loading selected and summarized SMF data into an HDB allows you to accumulate the performance data you want at the level of detail you need for reporting over long periods, without requiring large amounts of storage or processing time.

In addition to producing formatted reports from SMF data sets or HDBs, CICS PA can extract data to DB2 tables or comma-separated value (CSV) text files. You can then develop your own custom reports using DB2 SQL queries, or download CSV files to your PC, where you can view and manipulate the data using PC-based spreadsheet applications such as Microsoft Excel®.

CICS PA provides both an interactive ISPF dialog interface and a batch command interface. You can use either of these interfaces to request your reports and extracts. The ISPF dialog interface uses your interactive input to prepare JCL for the batch command interface. If you prefer to work directly with a command interface rather than an interactive interface, then you can use the ISPF dialog interface to prepare JCL that you can save and use as a starting point, and then edit the JCL later without using the ISPF dialog.

2.2 Comparing performance before and after migration

To compare the performance of your CICS systems before and after migrating from CICS TS 2.3 to 3.1, you can use CICS PA to create two CSV files: one containing a summary of performance data for a time period before migration and the other containing a similar summary of performance data for a time period after migration. Then you can download these CSV files from z/OS to your Windows® PC, and use the Microsoft Excel-based Compare Before-After tool supplied with this book to compare the before and after values.

The Compare Before-After tool creates an Excel workbook that shows differences between before and after values, with color highlighting to indicate percentage increase (red) or decrease (green) (Figure 2-2).

Tran	#Tasks	Response Time Avg	Dispatch Time Avg	User CPU Time Avg	Suspend Time Avg	QR CPU Time Avg	KY8 CPU Time Avg	L8 CPU Time Avg
A101	-15	0.010186	0.011222	0.002082	-0.00291	-4.00E-05	7.00E-05	0.001558
A102	8	-0.01501	0.003454	0.001642	-0.01973	-1.00E-05	-3.90E-04	0.00105
A103	7	6.48E-03	6.13E-03	0.002096	-0.00138	0	-1.40E-04	0.001504
A104	6	-8.05E-03	8.31E-03	0.00133	-0.01808	-1.90E-04	-9.60E-04	8.76E-04
A105	0	6.51E-03	0.009418	0.001992	-0.0044	0	-1.00E-05	0.001394
A106	24	8.11E-03	0.008484	0.0029	-0.00186	-1.00E-05	6.50E-04	0.002354
A109	2	0.016988	0.014272	0.003016	0.00112	-1.00E-05	5.90E-04	0.002454
A110	0	-9.30E-03	0.004936	0.001496	-0.01728	-0.00015	-0.00037	0.00101
A115	-6	-4.48E-03	-2.05E-03	0.001734	-0.00293	-8.00E-05	-0.00027	0.00111
A116	0	0.013184	0.012018	0.001216	2.50E-04	-1.00E-05	-2.00E-05	0.000792
A117	-1	0.001882	0.00163	0.00139	-7.00E-04	1.60E-04	7.00E-05	0.00077
A118								
A124	-1	0.001362	0.000976	0.0007	0.00018	-7.00E-05	-0.00003	0.000446

Figure 2-2 Sample workbook created by the Compare Before-After tool

In this sample we can see that L8 CPU times have increased, possibly indicating issues related to DB2 or threadsafe processing. (Note that this sample was generated using data from systems in a development environment, and does not reflect typical before/after migration differences in a production environment.)

A sample workbook created by this tool is supplied with this book in the file ... - sample.xls.

When you run the Compare Before-After tool, in addition to specifying the locations of the before and after CSV files, you also specify a minimum and a maximum percentage difference for highlighting. Percentage differences less than the minimum value are not highlighted. Percentage differences up to the maximum value are highlighted in colors of increasing intensity. The deepest intensity color indicates percentage differences greater than the maximum value. Figure 2-3 shows a copy of the *Key to colors* sheet in a workbook created by this tool, illustrating how these minimum and maximum values determine the highlighting.

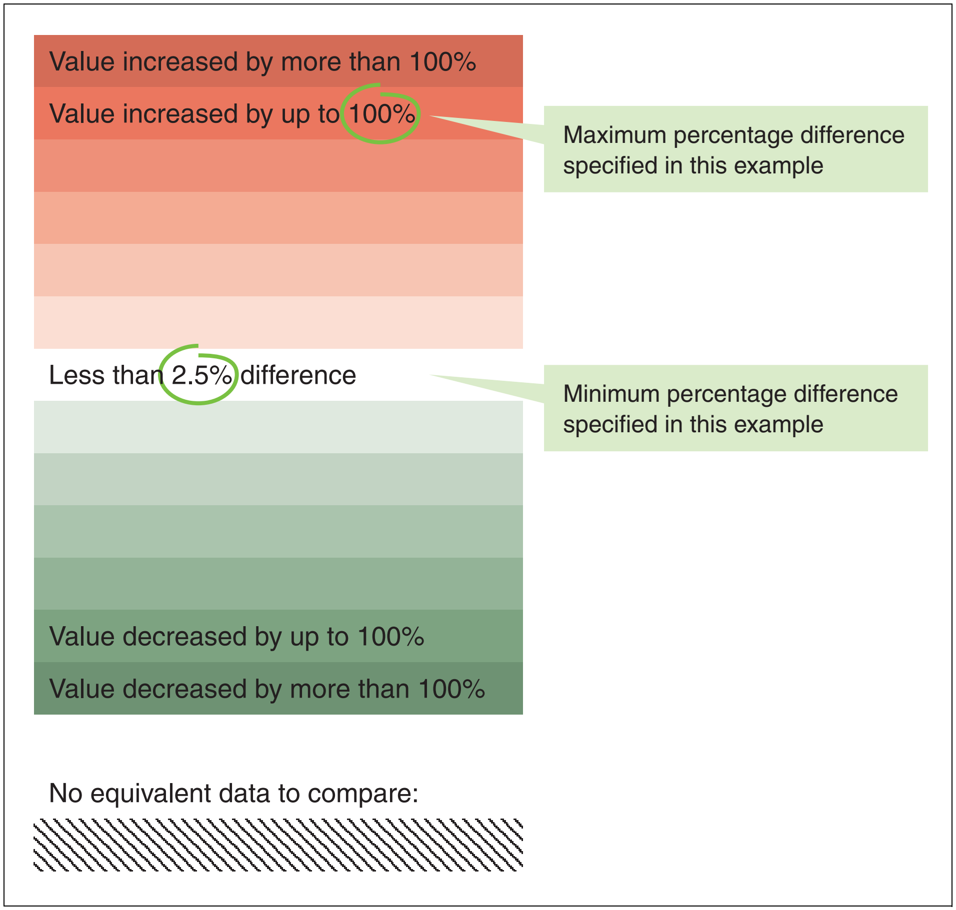


Figure 2-3 Compare Before-After tool: how minimum and maximum percentage differences affect color highlighting

Figure 2-4 illustrates this approach to comparing before and after performance data.

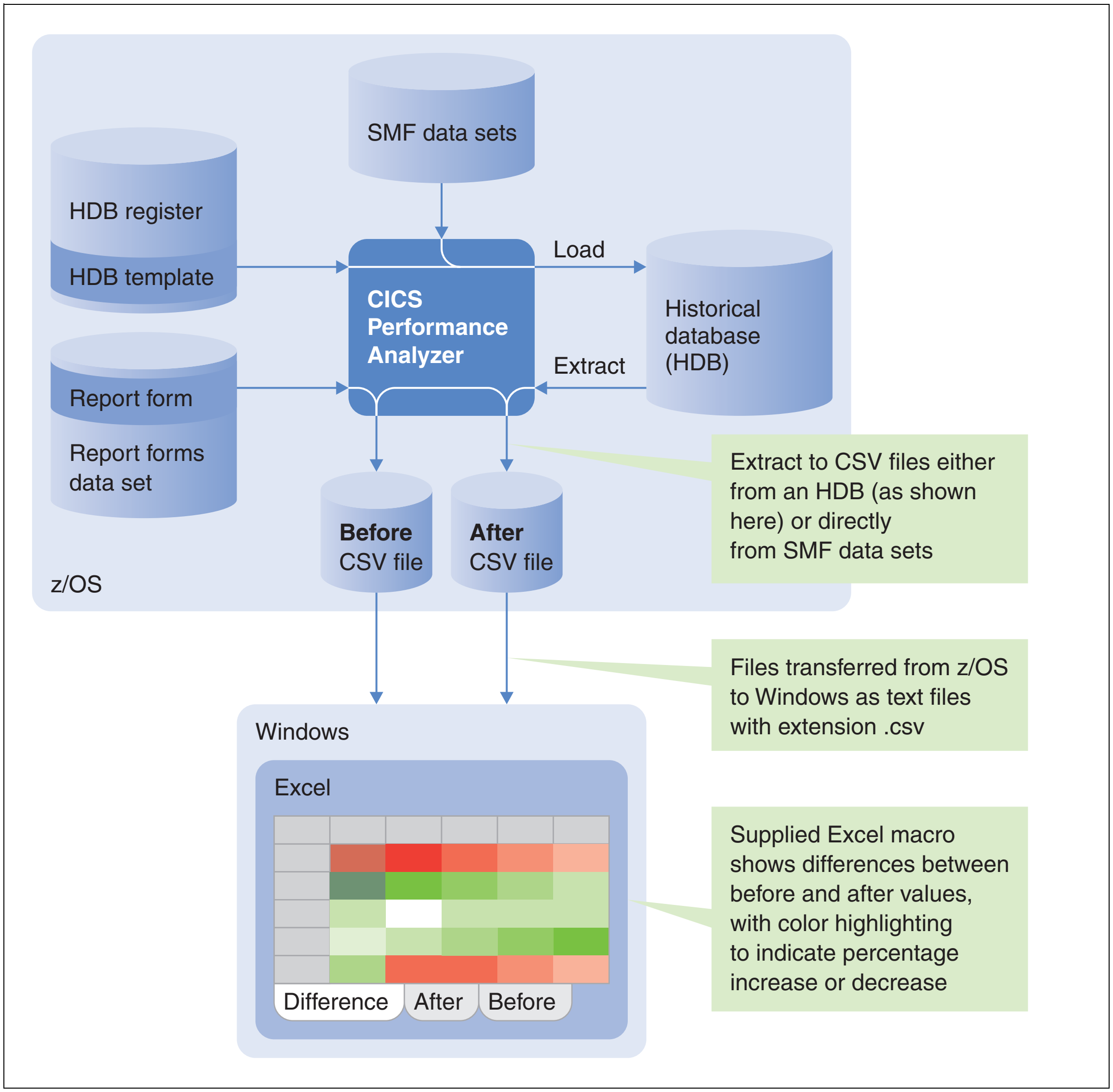


Figure 2-4 Comparing performance before and after migration using CSV files created by CICS PA

For step-by-step instructions on how to extract CSV files from an HDB, see the SupportPac *CP12: CICS Performance Analyzer for z/OS Historical Database Reporting*, available from:

http://www.ibm.com/support/docview.wss?uid=swg24011321

When extracting the CSV files, to ensure that they are compatible with the Compare Before-After tool:

- Use a summary report form that has only one key field, such as transaction ID (TRAN). For example, use the sample summary report form CPUSUM supplied with CICS PA.
- Specify a comma (,) as the extract format delimiter to ensure that the CSV file is a true *comma-separated* value file (that you can open directly in Excel, without specifying additional text import parameters).
- Include field labels so that the first line of the CSV file contains column headings.
- Do not select the (z/OS host-specific) float format for numeric fields.

The two CSV files should have similar layouts (same field used for the first column, some of the same fields for other columns), but their layouts do not need to be identical. The tool compares cells in the two CSV files that have the same row and column headings, and marks with diagonal hatching any cells that appear in one file but not the other. Column and row headings do not need to be in the same order in the two files: the comparison is based on matching column and row heading text, not on ordinal cell positions. For example, the *after* CSV file does not need to contain all of the transaction IDs that appear in the *before* CSV file.

After creating the two CSV files, download them to your PC as text files with the file extension .csv (see the samples supplied with this book, in the folder named *sample data*). Then run the Compare Before-After tool to compare them. You can run this tool using either its graphical user interface (requires Internet Explorer® 6 or later) or by editing the supplied batch (.bat) file.

> **Note:** The Compare-Before After tool was developed and tested using Microsoft Office Excel 2003 on Windows XP.
>
> To run this tool, you may need to change your Excel macro security settings to *low* (click **Tools** → **Macro Security**). For more information about macro security, see the Excel Help.

To start the graphical user interface:

1. In Windows Explorer, browse to the folder where you extracted the files supplied with this book.
2. Double-click the HTML application (.hta) file.

Figure 2-5 shows the entry form displayed by the graphical user interface.

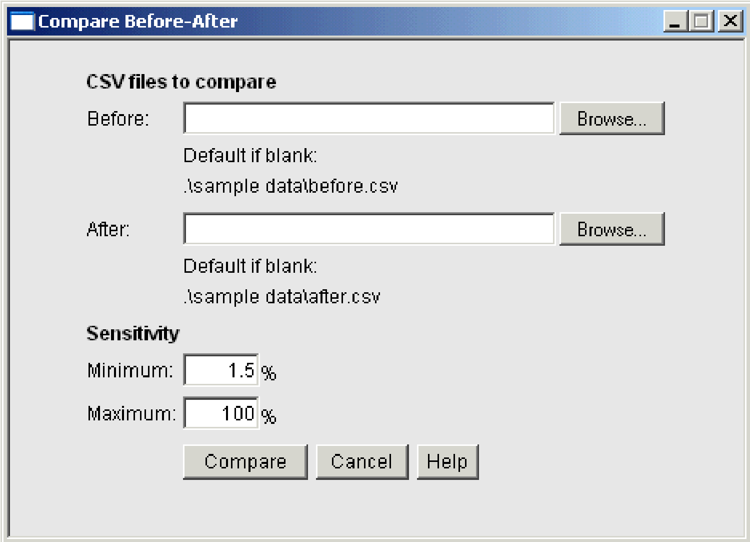


Figure 2-5 Compare Before-After tool entry form

3. Click the **Compare** button, and then wait a few seconds. The entry form window closes and an Excel window opens showing the comparison in a new workbook.

After using the Compare Before-After tool to highlight differences, you can use the formatted reports provided by CICS PA to identify the issues that caused those differences. For details, see the CICS PA user documentation.

Other uses for the Compare Before-After tool: While this tool is useful for comparing performance differences before and after migration to a new CICS release, the before and after CSV files can contain data for any two time periods. For example, you can create a CSV file summarizing CICS performance over the last year, and compare that with a CSV file summarizing CICS performance for the last week. Or you can create a *benchmark* CSV file containing values required to meet a service-level agreement, and use that as a *before* CSV file to compare with current performance.

In fact, you can use this tool to compare any two CSV files containing numerical data with row and column headings, where at least some of the headings in the two files match.

3

Overview of CICS CM

This chapter provides an overview of CICS Configuration Manager for z/OS (CICS CM) and then describes how you can use it to migrate your resource definitions to CICS TS 3.1.

Use of the term migrate: In this book, *migrate* typically refers to the process of upgrading to CICS TS 3.1. However, in CICS CM, *migrate* refers to the use of changing packages to copy resource definitions between CSD files or contexts.

3.1 CICS CM defined

CICS CM is a tool that provides a single point of control for editing, reporting, and migrating CICS resource definitions across an enterprise.

From a single interactive TSO/ISPF session, you can change CICS resource definitions stored in either CICS system definition (CSD) files or CICSPlex SM data repository contexts. The resource definitions in these CSD files or contexts can be for any version of CICS Transaction Server that is supported by CICS CM: Version 1.3, 2.2, 2.3, or 3.1. You can work with resource definitions across any combination of CSD file, context, and supported CICS TS version. For example, you can copy resource definitions from a CSD file used by a CICS 1.3 region to a context used by a CICS 3.1 CICSplex, or vice versa. CICS CM handles any underlying differences in version or storage formats.

The main components of CICS CM are the *server* (which is a CICS application that can read from and write to CSD files and contexts) and the two supplied *clients* (an interactive TSO/ISPF dialog interface and a batch command interface). The clients communicate with the server by exchanging SOAP messages over HTTP. You can also write your own client to communicate with the server and use its functions. For a more detailed description of these and other components of CICS CM, see 3.14, "Components of CICS CM" on page 69.

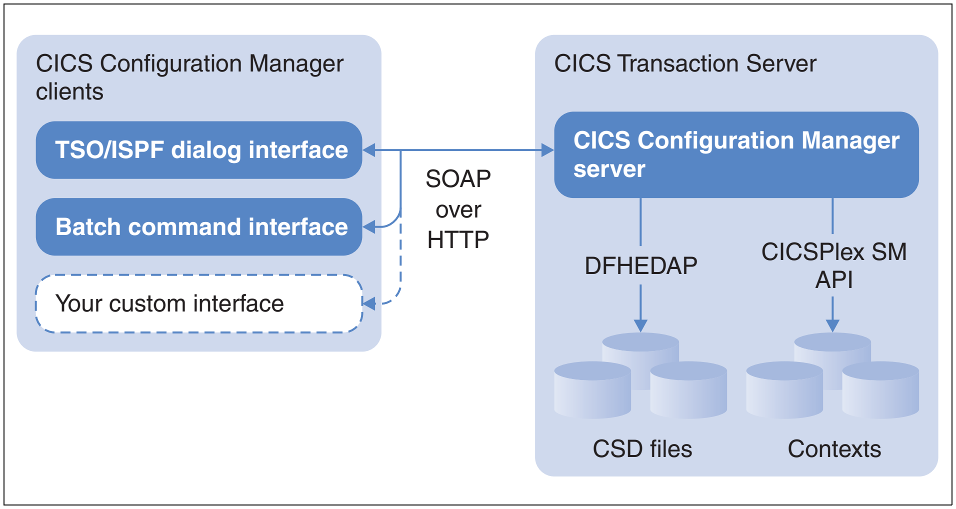


Figure 3-1 CICS CM main components

You can use CICS CM to:

- Edit resource definitions in a CSD file or a context.
- Compare resource definitions, groups, ResGroups, lists, and ResDescs across CSD files or contexts.
- Report resource definitions that match your search criteria across CSD files or contexts.
- Migrate resource definitions from one CSD file or context to another.
- Transform attribute values when migrating resource definitions, for environmental differences such as different high-level qualifiers for files, or for version differences such as inserting appropriate values for new attributes when migrating to a newer version of CICS TS.
- Optionally, require approval from authorized users before allowing a migration.
- Back out migrations.
- Log changes to resource definitions.
- Recover historical versions of resource definitions.
- Perform actions on active CICS regions:
  - Install resource definitions.
  - Discard resource definitions.
  - *Newcopy* or *phase in.* (Make new versions of programs available.)
- Export resource definitions to separately managed remote sites.

3.2 Benefits of CICS CM

CICS CM offers the following benefits:

- A single point of control for resource definitions across your enterprise.

  From a single TSO/ISPF session, you can change resource definitions in any CSD file or CICSPlex SM context that is accessible to CICS CM.

- A common interface to CSD files and CICSPlex SM contexts.

  In CICS CM, you define a CICS configuration for each CSD file or CICSPlex SM context that you want to manage. Thereafter, you refer only to the CICS configuration. CICS CM transparently handles the differences between CSD files and CICSPlex SM contexts.

- Enhanced editing of resource definitions, using a TSO/ISPF interface.

  The resource definition editor provided by CICS CM has many advantages over the resource definition online (RDO) CEDA transaction provided by CICS. You can:

  – Edit resource definitions while the CICS regions that use those definitions are active or inactive.

  – Edit RDO definitions from CSD files or BAS definitions from CICSPlex SM data repositories.

  – Get extensive field-sensitive help for all resource definition attributes.

  – View and compare previous versions of resource definitions.

  – Filter resource definitions by a combination of list, group, type, and name.

- Explore the logical hierarchy of a CICS configuration.

  Expand a CICS configuration to show its lists (or ResDescs, for CICSPlex SM), expand a list to show its groups (or ResGroups), expand a group to show its resource definitions, and expand a resource definition to show its attributes.

- Copy a set of resource definitions across multiple CICS configurations with a single command.

  In CICS CM, you can define one or more pairs of source and target (*from* and *to*) CICS configurations as a migration scheme, and you can group resource definitions into a change package. With a single command, you can migrate a change package according to a migration scheme, copying a set of resource definitions across multiple CICS configurations, regardless of whether the CICS configurations refer to CSD files or contexts.

- Transform resource definitions during migration.

  For each pair of source and target CICS configurations in a migration scheme you can select a set of transformation rules. These rules can adjust a resource definition during migration to match its target environment. For example, if a resource definition in your development environment specifies a file as SYS**<u>DEV</u>**.VSAM.FILEA, but the same file in the test environment is stored in SYS**<u>TEST</u>**.VSAM.FILEA, then you can define a rule to transform file names that match the mask *DEV.* to *TEST.*.

  You can also use transformation rules to search and replace within a single CICS configuration, by using a migration scheme with the same source and target CICS configuration. You can specify criteria that limit the resource definitions to which a transformation rule applies. You can also specify processing options for a rule, to allow or disallow further rules to be applied, or to exclude certain resource definitions from migration.

- Detect unexpected changes to resource definitions.

  Before migrating a change package, you must mark it as *ready*, indicating that you want no further edits to its resource definitions prior to migration. If you attempt to migrate or install a change package that is marked as ready, but the resource definitions in the source CICS configurations have subsequently changed or the migration scheme has changed, then CICS CM does not allow the migration. This protects you from migrating unexpected changes.

- Optionally, require approval before migrating.

  You can associate an approval profile with each change package, reflecting its sensitivity or potential impact (such as minor, major, or emergency). An approval profile specifies up to five approver roles for each migration scheme, identifying the types of user (such as developer, tester, or manager) who must approve a change package before it can be migrated. In addition to being marked as ready, each change package must be approved by all of the applicable approver roles before it can be migrated.

- Report resource definitions that match specified criteria.

  Select from a set of predefined criteria, such as *transactions using program name* (where all you have to specify is the program name) or define your own combination of search criteria.

- Compare resource definitions, lists, ResDescs, groups, or ResGroups.

  Compare resource definitions from one or more CICS configurations, filtered by name, type, and group. For each resource definition, the comparison shows a checksum of predetermined attribute values. Identical checksums indicate identical attribute values. Different checksums indicate that some attribute values are different. If you notice that two resource definitions have different checksums, then you can select the resource definitions and compare their attribute values side-by-side, with the differences highlighted.

3.3 Defining CICS configurations

Before you can begin using CICS CM to work with CICS resource definitions, you must define a CICS configuration for each of the CSD files or contexts in which those resource definitions are stored.

CICS CM uses CICS configurations to *abstract* access to resource definitions from their physical location and underlying storage method. In CICS CM, rather than referring to the names of contexts or the data set names of CSD files, refer to the names of CICS configurations. Using CICS configurations:

- Insulates you from the differences between CSD files and contexts. This allows you to work with resource definitions without needing to know how or where they are stored.
- Enables you to change the data set name of a CSD file or the location of a context without interrupting your workflow. For example, if you upgrade to a new version of CICS, and this involves changing the data set names of your CSD files, then all you need to do in CICS CM is update the CICS configurations to refer to the new data set names.
- Enables you to define security rules for accessing resource definitions based on a CICS configuration name. If you change the data set name of a CSD file, or the location of a context, you do not need to update the rules.

You can define CICS configurations in two ways:

- Interactively, one at a time, using the CICS CM ISPF dialog
- In batch, using the DATATAKEUP command of the CICS CM batch interface

To define a CICS configuration, simply specify a 1–8 character CICS configuration name, and then the data set name of a CSD file or the name of a context.

3.4 Editing resource definition attributes

To edit the attributes of a resource definition, display a list panel of resource definitions, and then select the particular resource definition whose attributes you want to edit. The CICS CM ISPF dialog offers several paths to a list panel of resource definitions. Figure 3-2 shows some of these paths.

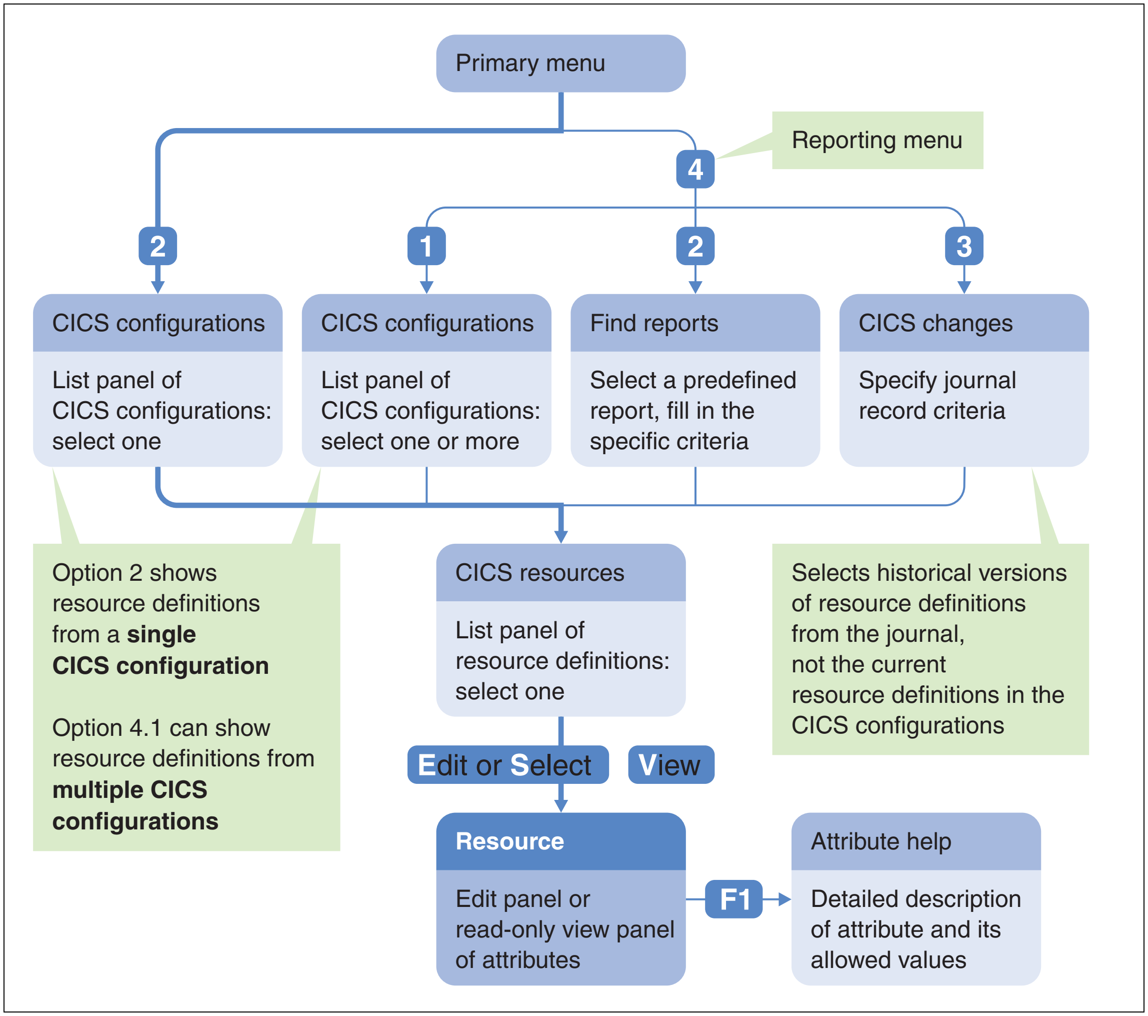


Figure 3-2 CICS CM: different paths from the primary menu to a list panel of resource definitions

To follow the path highlighted by the thick line in Figure 3-2:

1. On the CICS CM ISPF dialog primary menu, select option **2: CICS Resources**.

A list panel of CICS configurations appears.

You can optionally limit the contents of this list panel by entering values in the filter fields above the list column headings.

2. Enter S next to the CICS configuration containing the resource definitions that you want to work with.

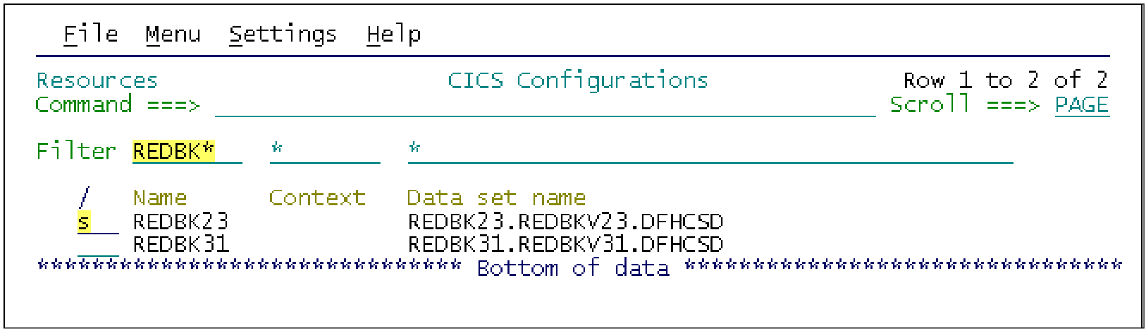


Figure 3-3 CICS CM: selecting a CICS configuration to work with

A list panel appears with filter fields and column headings, but no list items. Instead, directly under the column headings is a *bottom of data* indicator. This empty list is normal: The panel does not display any list items until you press Enter. This allows you to adjust the filter before displaying the list, avoiding delays caused by listing more resource definitions than required.

3. Type filter values for the resource name, type, and group, and then press Enter.

The filter shown in the following panel (Figure 3-4) limits the display to program resource definitions whose names begin with the characters REDBK.

```
  File  Menu  Settings  Checksum  Search  Help
 ------------------------------------------------------------------------------
 Resources                   REDBK23 CICS Resources           Row 1 to 10 of 10
 Command ===> ________________________________________________ Scroll ===> PAGE

 Filter REDBK*___  PROGRAM_____ + *________                       *_______________

        Name      Type          Group    Prompt                 --- Changed ----
 ______ REDBK1    PROGRAM       REDBOOK                         2006/06/09 08:08
 ______ REDBK1A   PROGRAM       REDBOOK                         2006/06/07 11:51
 ______ REDBK1B   PROGRAM       REDBOOK                         2006/06/07 11:51
 ______ REDBK1C   PROGRAM       REDBOOK                         2006/06/07 11:51
 ______ REDBK1D   PROGRAM       REDBOOK                         2006/06/07 11:51
 ______ REDBK1E   PROGRAM       REDBOOK                         2006/06/07 11:51
 ______ REDBK2    PROGRAM       REDBOOK                         2006/06/07 11:50
 ______ REDBK3    PROGRAM       REDBOOK                         2006/06/07 11:50
 ______ REDBK4    PROGRAM       REDBOOK                         2006/06/07 11:51
 ______ REDBK5    PROGRAM       REDBOOK                         2006/06/07 13:35
 ******************************* Bottom of data ********************************
```

Figure 3-4 CICS CM: displayed a filtered list panel of resource definitions

4. Enter `S` next to a resource definition to edit its attributes.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 CCVPCPGE                          Program
 Command ===> __________________________________________________________________

     Program . . . : REDBK1
     Group . . . . : REDBOOK
     Location  . . : REDBK23.REDBKV23.DFHCSD
     Change Date . : 2006/06/09 08:08:39
     Description . . ____________________________________________________________
                                                                  More:     +
                                              Description
     Language  . . . LE370___   +             Program language
     Reload  . . . . NO_  +                   Reload new copy on each execution
     Resident  . . . NO_  +                   In-storage residence after first use
     Usage . . . . . NORMAL___  +             Program storage release
     UseLPAcopy  . . NO_  +                   Use program from the link pack area
     Status  . . . . ENABLED_  +              Enabled for use status
     CEDF  . . . . . YES  +                   Display CEDF diagnostic screens
     DataLocation    BELOW  +                 In-memory storage address data location
     ExecKey . . . . CICS  +                  Program execution key
     Concurrency . . QUASIRENT_  +            Concurrent execution resource protection
     API . . . . . . CICSAPI  +               API interface used by the program
```

Figure 3-5 CICS CM: editing resource definition attributes

A plus sign (+) marks prompt fields. To select from a list of valid values, move the cursor to the field, and then press the Prompt (F4) key. A pop-up window appears with a list of values.

To get a detailed description of an attribute, move the cursor to the attribute field, and then press the Help (F1) key.

In addition to editing resource definitions individually, you can also select multiple resource definitions of the same resource type and alter some of their attribute values. The alter panel is similar to the edit panel, except that it displays empty fields for all the attributes of that resource type. Only the attributes for which you enter values are affected. This enables you to set the attribute values of many resource definitions with a single action, without affecting all of their attributes.

3.5 Exploring the hierarchy of a CICS configuration

You can use the CICS CM ISPF dialog to explore the hierarchy of a CICS configuration in a manner similar to using Windows Explorer to explore the folders on your PC's hard drive:

1. Expand a CICS configuration to show its lists (or ResDescs, for context-based CICS configurations).
2. Expand a list or ResDesc to show its groups or ResGroups.
3. Expand a group to show its resource definitions.

To explore the hierarchy of a single CICS configuration, select primary menu option **2 CICS Resources**. To explore the combined hierarchies of multiple CICS configurations, select option **4.1 Multiple Configs**.

The subsequent panels and available actions are similar whether you select one CICS configuration or several, except that if you select several CICS configurations, the list panels show a Config column indicating the CICS configuration to which each item belongs.

Figure 3-6 shows the primary menu options and subsequent line actions that you can use to explore a CICS configuration hierarchy.

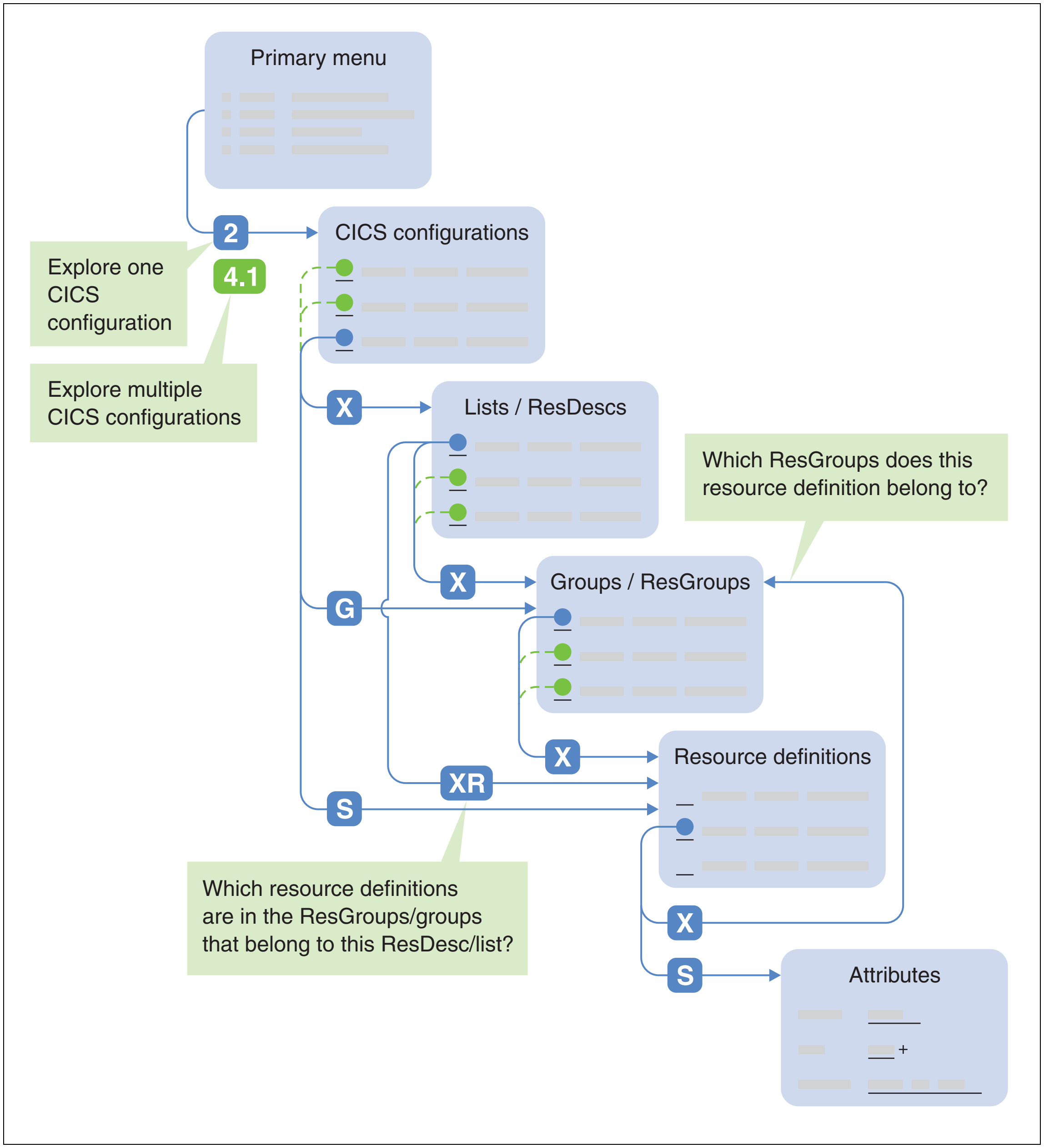


Figure 3-6 CICS CM: exploring the hierarchy of a CICS configuration

To expand an item and show the next level of hierarchy, enter `X` next to the item. To step back one level, press the Exit key (F3).

If you enter the hierarchy via option 4.1 Multiple Configs, then, even if you select only one CICS configuration, you can expand several lists/ResDescs or groups/ResGroups simultaneously to show their combined contents. For example, if you want to display a list panel of resource definitions from several groups in a CICS configuration:

1. On the CICS CM ISPF primary menu, select option **4 Reports** to display the Reports menu, and then select option **1 Multiple Configs**. (To go directly to this option, bypassing the Reports menu, enter `=4.1` on the command line of any CICS CM panel.)
2. Enter `G` next to a CICS configuration.

   A list panel of the groups/ResGroups in the CICS configuration appears.
3. Enter `X` next to several groups/ResGroups.

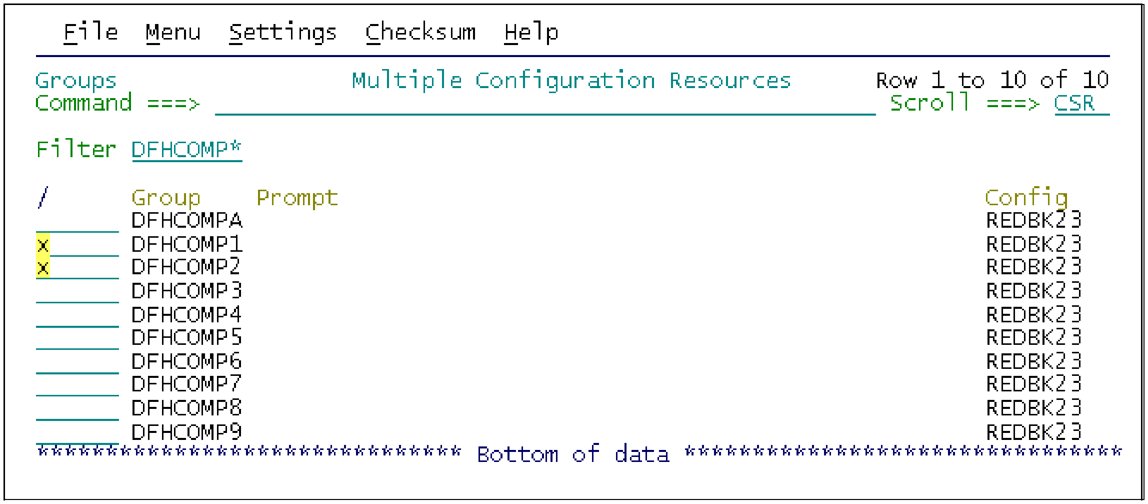


Figure 3-7 CICS CM: expanding several groups simultaneously

A list panel of the resource definitions in the selected groups/ResGroups appears (Figure 3-8).

```
  File  Menu  Settings  Checksum  Search  Help
 ------------------------------------------------------------------------------
 Group Resources        Multiple Configuration Resources     Row 66 to 81 of 128
 Command ===> ________________________________________________ Scroll ===> CSR

 Filter * _______ * ____________ +

        Name     Type          Group     Prompt                      Config
 ______ DFHEMA   PROGRAM       DFHCOMP1                              REDBK23
 ______ DFHEMB   PROGRAM       DFHCOMP1                              REDBK23
 ______ DFHEMC   PROGRAM       DFHCOMP1                              REDBK23
 ______ DFHEMD   PROGRAM       DFHCOMP1                              REDBK23
 ______ DFHEME   PROGRAM       DFHCOMP1                              REDBK23
 ______ DFHEMF   PROGRAM       DFHCOMP1                              REDBK23
 ______ DFHEMG   PROGRAM       DFHCOMP1                              REDBK23
 ______ DFHEMH   PROGRAM       DFHCOMP1                              REDBK23
 ______ DFHEMI   PROGRAM       DFHCOMP1                              REDBK23
 ______ DFHEOP   PROGRAM       DFHCOMP2                              REDBK23
 ______ DFHESE   PROGRAM       DFHCOMP2                              REDBK23
 ______ DFHESN   PROGRAM       DFHCOMP2                              REDBK23
 ______ DFHETRX  PROGRAM       DFHCOMP2                              REDBK23
 ______ DFHFCS   PROGRAM       DFHCOMP1                              REDBK23
 ______ DFHMATOC PROGRAM       DFHCOMP2                              REDBK23
 ______ DFHMTPA  PROGRAM       DFHCOMP1                              REDBK23
```

Figure 3-8 CICS CM: viewing a list panel of resource definitions that belong to selected groups

3.6 Copying definitions between CICS configurations

You can use CICS CM to copy resource definitions across any combination of CSD file, context, and supported CICS TS version.

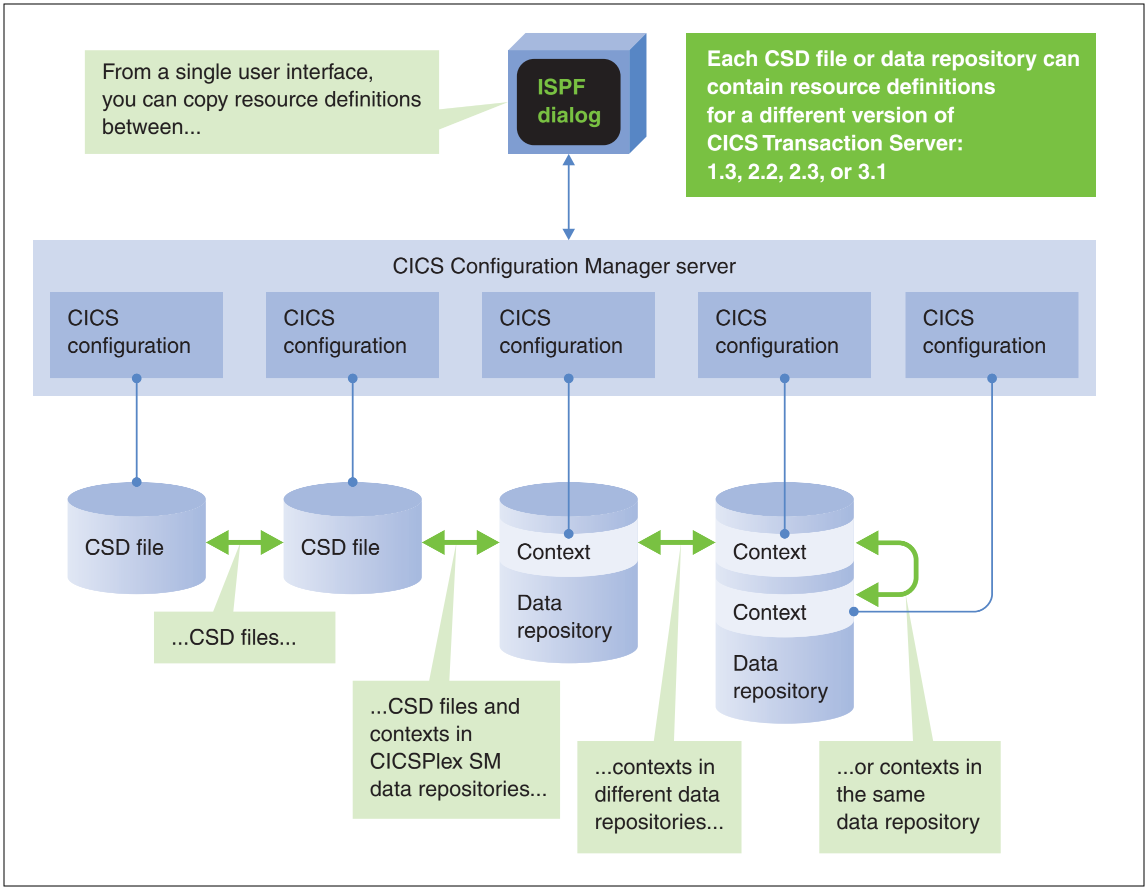


Figure 3-9 CICS CM: copy resource definitions between any combination of CSD file or context, and between any supported versions of CICS TS

In CICS CM, each CSD file or context is simply another CICS configuration. To copy resource definitions from one CICS configuration to another, select the source CICS configuration, select the resource definitions that you want to copy, and then select the target CICS configuration.

Here is the step-by-step procedure for copying resource definitions from one CICS configuration to another:

1. On the CICS CM primary menu, select option **2 CICS Resources**.

2. Enter `S` next to a CICS configuration.
3. Enter `C` next to the resource definitions that you want to copy.

> **Tip:** If the resource definitions that you want to copy can be specified by a combination of name, type, and group filters, then specify the filter values, and then enter `C *` (the letter C, a space, and then an asterisk) on the command line. This enters the line action C next to each resource definition in the list panel, saving you from typing this line action.

```
  File  Menu  Settings  Checksum  Search  Help
 ------------------------------------------------------------------------------
 Resources                    REDBK23 CICS Resources              Row 1 to 10 of 10
 Command ===> _______________________________________________    Scroll ===> CSR

 Filter REDBK*_    PROGRAM____  + *_______                        *____________

        Name      Type          Group     Prompt                  --- Changed ----
 c_____ REDBK1    PROGRAM       REDBOOK                           2006/06/09 08:08
 c_____ REDBK1A   PROGRAM       REDBOOK                           2006/06/07 11:51
 c_____ REDBK1B   PROGRAM       REDBOOK                           2006/06/07 11:51
 c_____ REDBK1C   PROGRAM       REDBOOK                           2006/06/07 11:51
 c_____ REDBK1D   PROGRAM       REDBOOK                           2006/06/07 11:51
 c_____ REDBK1E   PROGRAM       REDBOOK                           2006/06/07 11:51
 c_____ REDBK2    PROGRAM       REDBOOK                           2006/06/07 11:50
 c_____ REDBK3    PROGRAM       REDBOOK                           2006/06/07 11:50
 c_____ REDBK4    PROGRAM       REDBOOK                           2006/06/07 11:51
 c_____ REDBK5    PROGRAM       REDBOOK                           2006/06/07 13:35
 ******************************* Bottom of data ********************************
```

Figure 3-10 CICS CM: selecting resource definitions to copy

A pop-up window appears requesting the name of the target CICS configuration (and, optionally, a target group name, if you wanted to copy the resource definitions to a different group).

4. Select the target CICS configuration, and then press Enter.

Using this method to copy resource definitions is direct and straightforward. However, if you want to undo the copy, then you need to individually select the changed resource definitions in the target CICS configuration, and restore their previous versions. And if you want to copy to more than one target CICS configuration, then you need to repeat the copy procedure.

For more flexibility and control, rather than using this *ad hoc* copy method, you may wish to consider using change packages to perform the copy. Using change packages to copy resource definitions between CICS configurations is known as *migrating*.

3.7 Migrating definitions using change packages

A *change package* identifies a set of resource definitions that you want to process together. For example, suppose that you are a CICS application developer and you have just edited and created resource definitions for a CICS application in your development environment. Now you want to migrate these resource definitions to the test environment. Using a change package, you can migrate, install, and, if required, back out the resource definitions together.

A change package does not contain resource definitions; rather, it contains *selection keys* that refer to resource definitions stored in a CSD file or a CICSPlex SM context. A selection key identifies a resource definition by its name, group, type, and CICS configuration.

Migrating resource definitions between CICS configurations using change packages offers more flexibility and control than the ad hoc method of copying resource definitions:

- You can use a change package to identify a set of updates.

  When developing an enhancement to a CICS application, you can use a change package as a container for accumulating the required new or updated resource definitions, progressively adding resource definitions to the change package as you work on the application code. After coding the enhancement, you can use the change package to migrate the required resource definitions from your development environment to your test environment, and then onto production. Using a change package allows you to identify precisely which resource definitions are associated with an application enhancement or fix, and migrate only those resource definitions, rather than, for example, coding an enhancement and then copying resource definitions based on their *last changed* date, or performing a copy of all resource definitions, regardless of whether they might have changed.

- Change packages protect you from migrating unexpected edits.

  When you have finished adding or making changes to resource definitions in a change package, mark the change package as being ready, causing CICS CM to calculate checksum values from the attributes of its resource definitions. When you instruct CICS CM to migrate the change package (which may be some time later), CICS CM calculates new checksum values and compares them with the original values. If the values are different, it means that the resource definitions have been edited since you marked the change package as ready, and CICS CM disallows the migration. This protects you from migrating unexpected edits.

- Back out a change package migration with a single command.

  To undo a migration, simply instruct CICS CM to back it out. CICS CM restores any resource definitions that were updated by the migration to their pre-migration versions and deletes from the target CICS configurations any resource definitions that were created by the migration.

- Migrate changes systematically according to well-defined paths.

  To migrate the resource definitions in a change package between CICS configurations, specify a migration scheme, containing one or more pairs of source and target CICS configurations.

  Typically, organizations migrate (or promote) changes between environments according to well-defined paths (for example, from development to the test environment and then from test to production). Using change packages enables you to systematically follow these paths, reducing the risk of *ad hoc* copying of resource definitions to incorrect locations.

- Simplify migration across complex CICS system topologies.

  You can use a single migration scheme to migrate resource definitions in a change package from multiple source CICS configurations to multiple target CICS configurations. For example, suppose that you have a single development environment on which you have created or updated a set of resource definitions. You now want to migrate these resource definitions to your multiple test environments. Rather than copying the resource definitions separately to each environment, you can use a change package with a migration scheme to migrate the resource definitions to all of your test environments with a single command.

- Transform resource definition attributes during migration.

  You can define transformation rules to tailor resource definition attribute values during migration, or block certain resource definitions from migration. For example, if the high-level qualifiers of your application data sets are different in your development and test environments, then you can define rules for file resource definitions that automatically change data set names with a high-level qualifer of DEVT.* to TEST.*.

- Use the same change package to migrate from development to test, and then test to production.

  After migrating a change package, CICS CM automatically adds to that change package the keys of the migrated resource definitions in the target CICS configurations. This enables you to reuse the change package later with a migration scheme that refers to that CICS configuration as a source CICS configuration, and migrate those resource definitions onwards to other target CICS configurations.

For instance, suppose that you define a new change package and then add to it some resource definitions that you have created or edited in your development environment. You then use the change package with a *development-to-test* migration scheme to migrate those resource definitions to your test environment. After this migration the change package now refers to both the *original* resource definitions in the development environment, that you added to the change package, and the *migrated* resource definitions in the test environment, that CICS CM has automatically added to the change package. You can then use the same change package with a *test-to-production* migration scheme to migrate those resource definitions in the test environment to your production environment.

You can think of this reuse as *package once, migrate many times*.

- Optionally, require approval before migration.

  Optionally, you can require each change package to be approved by up to five people before it can be migrated. If you decide to use change package approvals, then each change package must refer to an *approval profile* that specifies up to five *approver roles*. For more information about approver profiles and approver roles, see the *CICS CM User's Guide*.

Figure 3-11 shows how CICS CM uses change packages to control the migration of resource definitions.

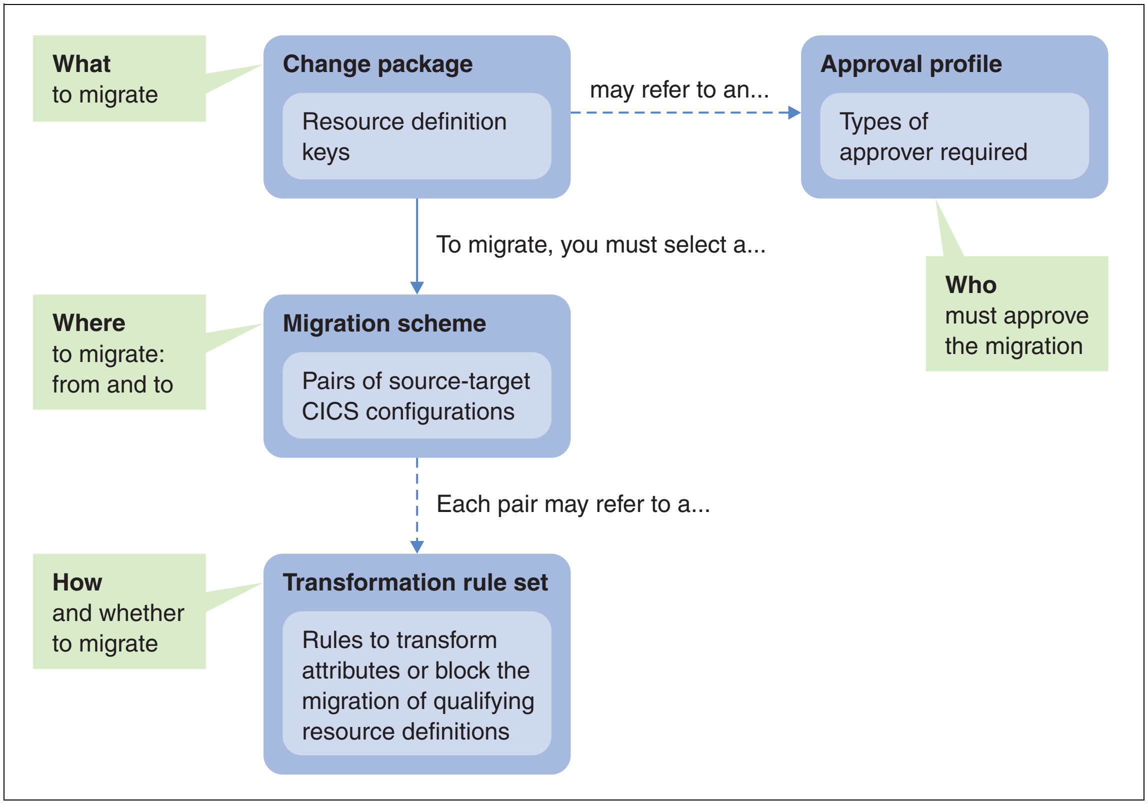


Figure 3-11 CICS CM: using change packages to control the migration of resource definitions

Figure 3-12 shows the typical workflow for a change package, including how CICS CM uses checksums to avoid migrating or approving unexpected edits.

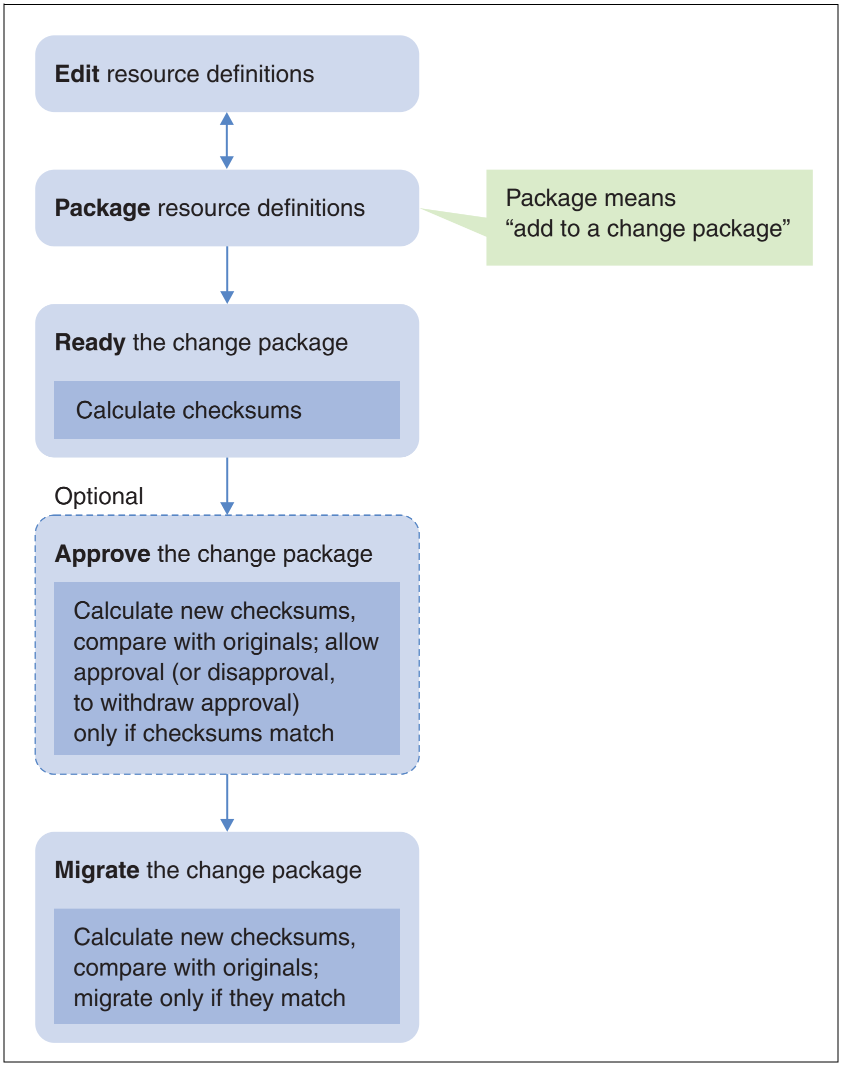


Figure 3-12 Change package workflow

In addition to using change packages to migrate resource definitions between CICS configurations, you can also use change packages to install resource definitions in active CICS regions. For example, after migrating a change package, you can then install the migrated resource definitions in the affected active CICS regions.

3.8 Comparing objects in a CICS configuration

You can compare the following types of objects in a CICS configuration:

- Lists or ResDescs
- Groups or ResGroups
- Resource definitions

You can compare objects in two ways:

- Display a list panel of the objects, and then sort by the checksum column.

  On a list panel of the objects that you want to compare, use the Checksum action bar item to display the optional checksum column, and then sort the list panel by that column so that objects with the same checksum appear consecutively. The checksum column provides an easy way to compare objects. For example, if two resource definitions of the same resource type have the same checksum, then either all of their attribute values are identical, or a subset of their attribute values are identical, depending on the type of checksum you select:

  – FULL: Include all resource definition attributes in the checksum calculation.

  – LIST: Include your choice of attributes in the checksum calculation.

  – NAME: Include only the resource definition names and types in the checksum calculation.

  – PARTIAL: Include a selection of attributes, as predefined by CICS CM.

  To display the following panel we selected CICS CM primary menu option **4.1 Multiple Configs**, entered `S` next to the CICS configurations REDBK23 and REDBK31, set the filter to display files in the group REDBOOK, and then select Checksum FULL to compare all attributes in the resource definitions. The different checksum values for the two resource definitions indicate that some of their attributes have different values.

```
  File  Menu  Settings  Checksum  Search  Help
 ------------------------------------------------------------------------------
 Resources              Multiple Configuration Resources           Row 1 to 2 of 2
 Command ===> ________________________________________________    Scroll ===> CSR

 Filter * _______  FILE _________  + REDBOOK
                                                               Checksum
        Name     Type            Group    Prompt                  Full     Config
 ______ REDBOOKF FILE            REDBOOK                       1A5A4DA2 REDBK23
 ______ REDBOOKF FILE            REDBOOK                       D8866094 REDBK31
 ******************************* Bottom of data ********************************
```

Figure 3-13 CICS CM: using the Checksum column to compare resource definitions on a list panel

You can select the two objects on a list panel and compare their details side-by-side, as described below.

- Compare the details of two objects side-by-side.

  On a list panel showing the two objects that you want to compare, type `CM` next to each object, and then press Enter. The objects appear side-by-side, with the differences highlighted. The following screen shows a side-by-side comparison of two resource definitions (Figure 3-14).

```
  File  Menu  Settings  Hilite  Help
 ------------------------------------------------------------------------------
 Compare                            File
 Command ===> _________________________________________________________________

     File  . . . . : REDBOOKF                      REDBOOKF
     ResGroup  . . : REDBOOK                       REDBOOK
 ==> Location  . . : REDBK23.REDBKV23.DFHCSD       REDBK31.REDBKV31.DFHCSD
 ==> Change Date . : 2006/06/08 15:13:10           2006/06/12 15:48:50
     Description . :
                                                                 More:     +
    VSAM Parameters
 ==> DSname  . . . : REDBK23.REDBKV23.VSAM         REDBK31.REDBKV31.VSAM
     Password  . . :
     RLSaccess . . : NO                            NO
     LSRpoolID . . : 1                             1
     ReadInteg . . : UNCOMMITTED                   UNCOMMITTED
     DSNsharing  . : ALLREQS                       ALLREQS
     Strings . . . : 100                           100
     NSRgroup  . . :

    Remote Attributes
     RemoteSystem  :
     RemoteName  . :
```

Figure 3-14 CICS CM: comparing two resource definitions side-by-side

Similarly, you can compare two lists (or two ResDescs, or a list and a ResDesc) side-by-side to view differences in their groups, and two groups (or two ResGroups, or a group and a ResGroup) to view differences in their resource definitions.

You can use list, group, and resource definition comparisons in sequence, progressively focussing on specific differences. For example, to identify differences in resource definitions between, say, two lists, you can begin with a side-by-side comparison of the two lists. You notice two groups with differences, so you enter `CM` next to those two groups to get a side-by-side listing of the resource definitions in those groups:

You notice two resource definitions with differences, and then enter `CM` next to those two resource definitions to get a side-by-side comparison of their attributes.

3.9 Searching for attribute values

You can limit a list panel of resource definitions to displaying only those resource definitions whose attribute values meet your search criteria. Search criteria consist of one or more conditions of the form:

attribute_name comparison_operator test_value

For example, the following condition limits the list panel of resource definitions to resource definitions changed after April 2005:

```
CHANGETIME GT 2005/04
```

Each resource type has separate search criteria, containing up to three sets of four conditions each. Within each set, conditions are grouped by Boolean AND operators. Sets are grouped by Boolean OR operators. You can select or deselect each set, so you can use them separately or in combination.

CICS CM stores search criteria in your ISPF user profile. Each CICS CM user has his own search criteria.

To define search criteria for a resource type, display a list panel of resource definitions, select the resource type from the type filter field, and then enter `SEARCH` on the command line. The search criteria panel for the selected resource type appears. See Figure 3-15.

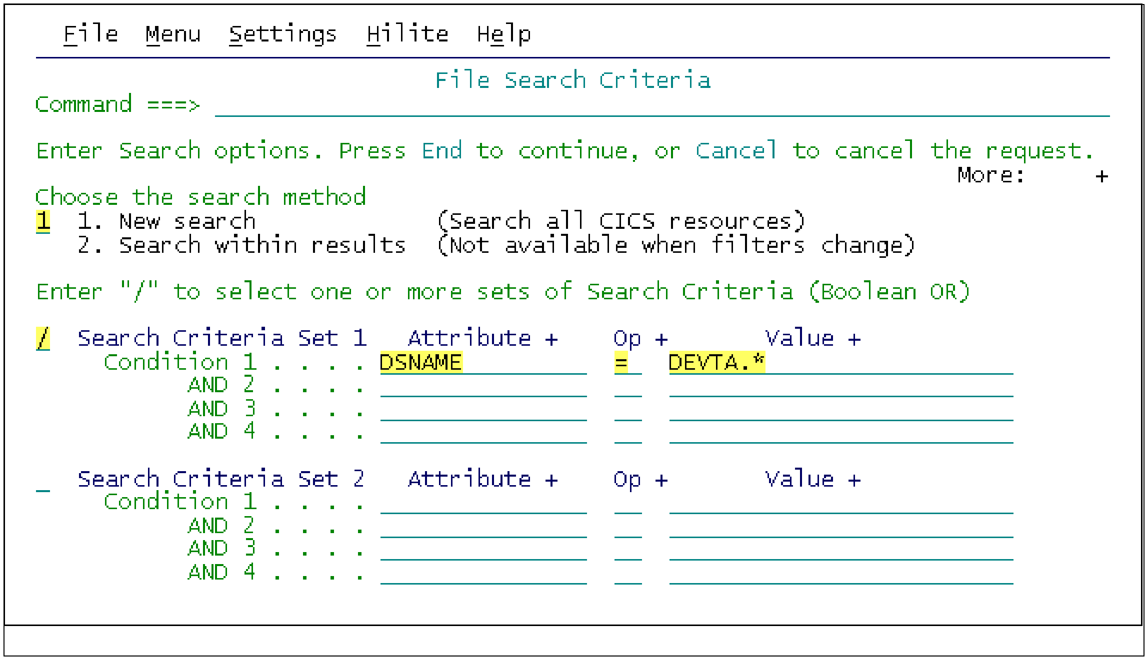


Figure 3-15 CICS CM: search criteria for file resources

As an alternative to defining your own search criteria, CICS CM menu option 4.2 Find offers predefined search criteria that you can use to display a list panel of resource definitions. See Figure 3-16.

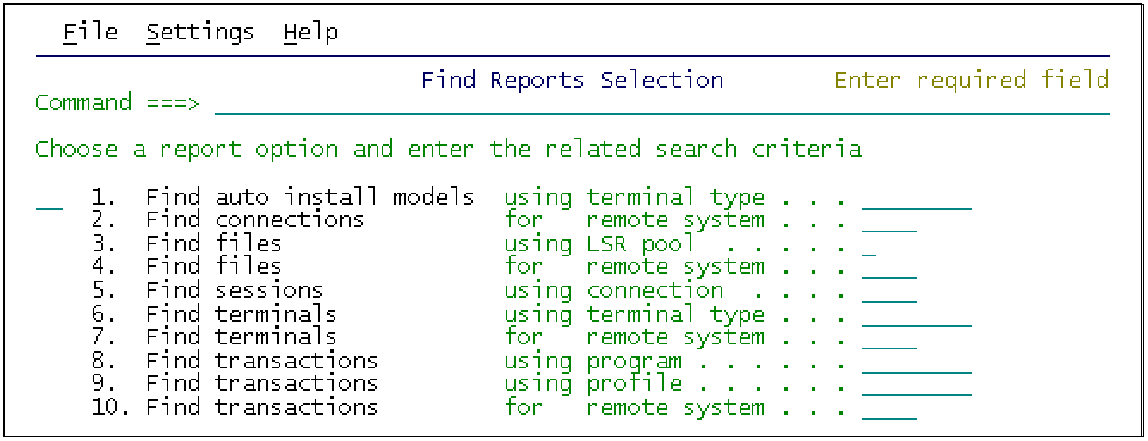


Figure 3-16 CICS CM: finding resource definitions that match predefined search criteria

3.10 Retrieving historical resource definitions

Whenever you use CICS CM to change a resource definition, whether by editing or copying an individual resource definition, or by migrating a change package of resource definitions, CICS CM records the change in the CICS CM journal.

You can browse the historical versions of resource definitions stored in the journal. You can compare a historical version of a resource definition side-by-side with the current version, or compare two historical versions. You can also restore a historical version, overwriting the current version.

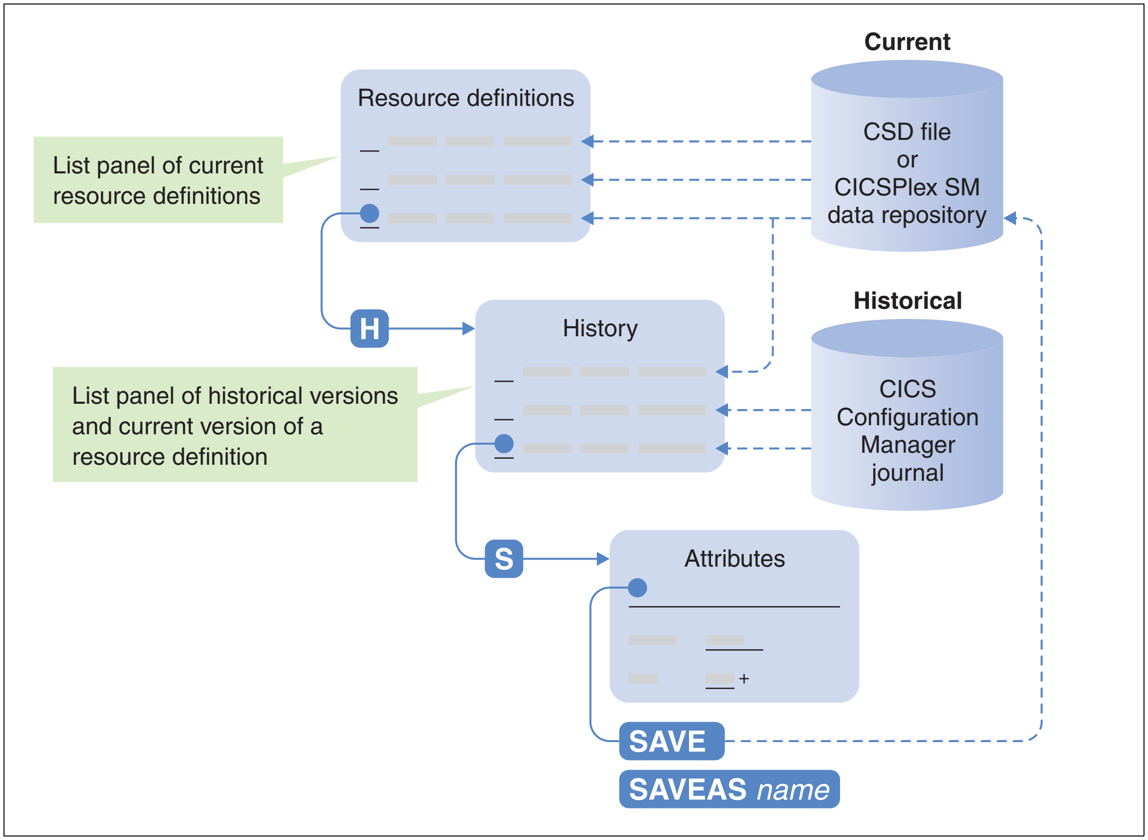


Figure 3-17 CICS CM: restoring a historical version of a resource definition

3.11 Exporting resource definitions to remote sites

Instead of referring to a CSD files or a context, a CICS configuration can refer to an export file. You can copy or migrate resource definitions to a CICS configuration that refers to an export file, using exactly the same methods you use to copy or migrate to other CICS configurations.

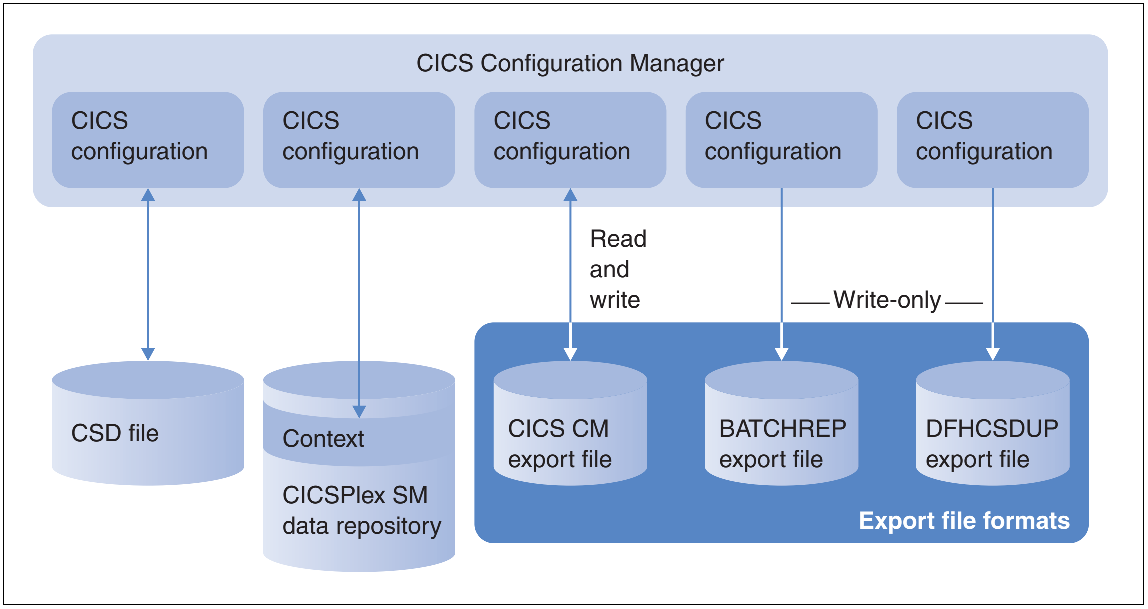


Figure 3-18 CICS Configuration Manager treats export files as CICS configurations

You can transfer an export file to a remote site, and then import the resource definitions from the file. You can create export files in three formats: CICS CM (for exporting to another CICS CM system), BATCHREP (for use with the CICSPlex SM BATCHREP utility), or DFHCSDUP (for use with the CICS TS DFHCSDUP utility).

If you migrate a change package to a CICS configuration that refers to a CICS CM export file, then the export file contains not just the migrated resource definitions, but also details of the change package. When importing the export file on another CICS CM system, you can register the change package on that system, and then use the change package to migrate the resource definitions on that system. This enables you to use change packages to seamlessly manage resource definition changes across separate CICS CM systems.

3.12 Extending and customizing the CICS CM server

You can extend and customize CICS CM server processing by attaching your own CICS programs to the following user exit points:

- Resource attribute update

  This exit point occurs before CICS CM updates the attributes of a resource definition, including creating or deleting a resource definition.

  You can write user exit programs to:

  – Allow or disallow the API command.

  – (Except for delete and rename) Allow the API command, while overriding some of the resource attribute values.

  You can use this exit point to enforce site standards upon resource definition names and attributes. For example, when a user edits and then attempts to save a resource definition, your user exit program can check the resource definition name and attributes, and then disallow the update, or allow the update as requested, or allow the update with some attribute values updated to match site standards.

- Change package pre-processing and post-processing

  These exit points occur before and after processing a change package. You can use these exit points to augment change package workflow. For example:

  – Notify users of planned or completed change package activity.

  – Log information regarding completed activity.

  – Trigger an event to an external change management system.

  – Security-check a user’s authority to perform the designated function.

  – Obtain a proceed-or-terminate decision from an external change management product.

  – Overrule change package processing checks.

  – Automatically update change package ready or approve states.

  For the change package pre-processing exit point, you can write user exit programs to disallow or allow the API command.

  You can write user exit programs in any language supported by CICS.

3.13 What business issues CICS CM addresses

Every CICS region refers to a set of resource definitions. Typically, organizations maintain each of their CICS regions in at least three separate environments: development, test, and production. Changes to resource definitions are migrated from development to test, and then from test to production. As shown in Figure 3-19, even if each environment contains only one CICS region, this means three sets of resource definitions, and two migrations to move each change into production.

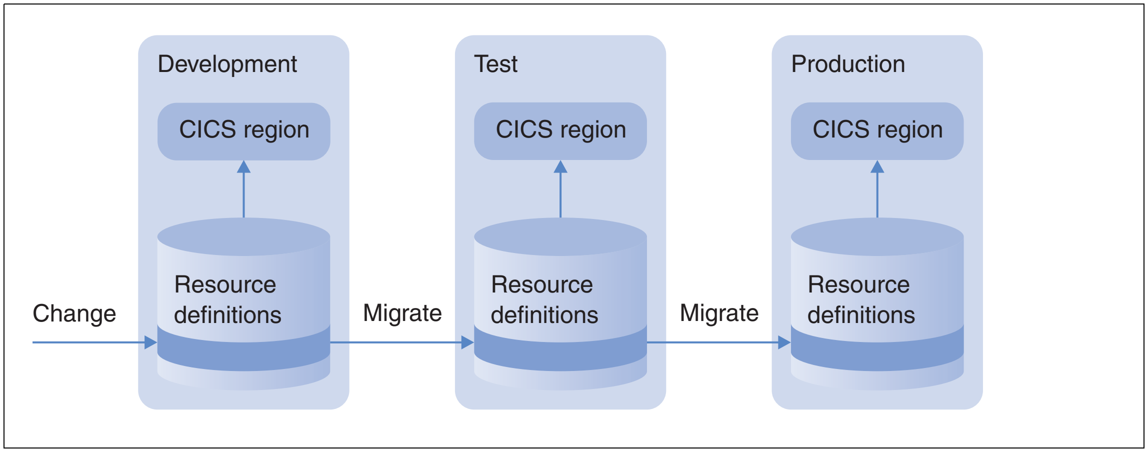


Figure 3-19 Migrating resource definitions between development, test, and production CICS environments

Real-life systems are typically much more complex, but even a simple system such as this raises issues. For example, if your developers enhance a CICS transaction, requiring a new resource definition for an additional file:

- How do you migrate the new resource definition from development to production?
- Do you simply copy the complete set of resource definitions from one environment to another?
- What if the resource definitions in each environment are intentionally different? For example, what if the high-level qualifiers for the file are different in each environment (DEV., TEST., and PROD.)? Do you edit the resource definitions for each environment separately?
- How do you ensure that you migrate only the changes that are ready for migration?
- What if some environments are controlled by CICSPlex SM, but not others? Do you need to use different tools to update these environments?

- How do you back out a change?
- How do you keep a systematic record of changes, for reporting and auditing?
- How do you avoid overwriting unexpected changes in the environment you are migrating to?
- How do you compare resource definitions?
- Who approves the migration?

These issues grow as your CICS system topology becomes more complex. Your organization might have several development, test, and production environments, with different numbers of CICS regions in each environment, and a variety of migration paths. For example, you might have resource definitions that are shared by regions in one environment, but not in another. So you might need to migrate resource definitions not just from one CSD file to another, but also from one CSD file to several:

- Your development environment might combine the typical CICS responsibilities of application owning, file owning, and terminal owning into a single region, with resource definitions stored in a single CSD file.
- Your test environment might assign these responsibilities to separate regions (AOR, FOR, TOR), sharing a single CSD file.
- Your production environment might have separate CSD files for each AOR, FOR, and TOR.

As shown in Figure 3-20, when migrating from a shared CSD file to unshared CSD files, you must ensure that you migrate the resource definitions to the appropriate locations.

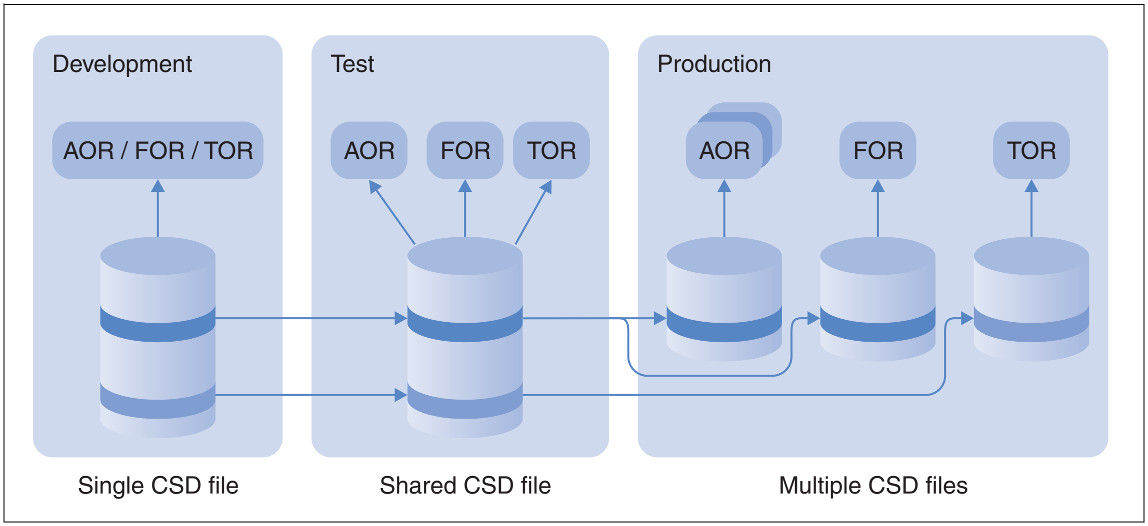


Figure 3-20 Migrating resource definitions from shared to unshared CSD files

Environments with many CICS regions often use CICSPlex SM, which stores resource definitions in data repositories rather than CSD files. So you might need to migrate resource definitions between CSD files and contexts in CICSPlex SM data repositories. See Figure 3-21.

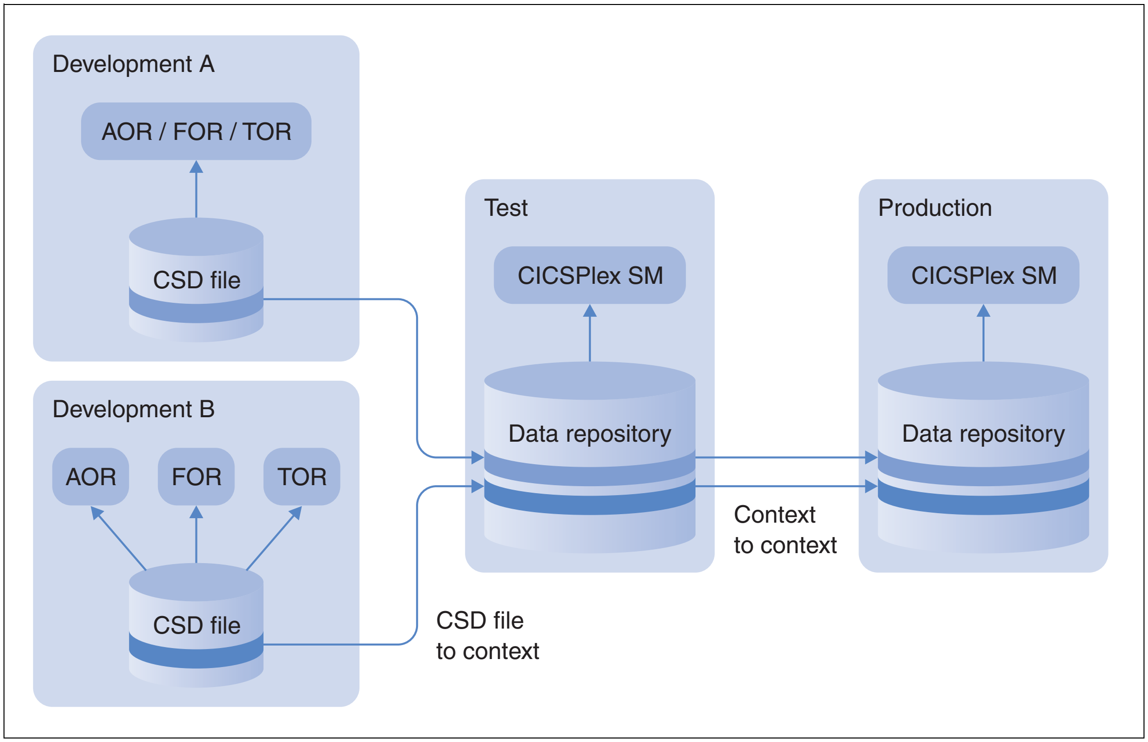


Figure 3-21 Migrating resource definitions from CSD files to CICSPlex SM data repository contexts

CICS CM addresses all of these issues.

3.14 Components of CICS CM

CICS CM consists of a client, a server, an agent, a repository, and a journal:

- Client

  A user interface that allows you to send commands to, and receive responses from, the CICS CM server. The client and server communicate by exchanging SOAP messages over HTTP. CICS CM is supplied with two clients: an ISPF dialog interface and a batch command interface.

- Server

  A CICS application that performs the actions requested by a CICS CM client, such as reading or updating resource definitions in a CSD file or a CICSPlex SM context.

- Repository

  A VSAM key-sequenced data set (KSDS) that stores current CICS CM data:

  – System options
  – CICS configurations
  – Migration schemes
  – Approval profiles
  – Transformation rules
  – Change packages and related records

- Journal

  A VSAM KSDS that records historical CICS CM data:

  – Summaries of processing events, such as updates to resource definitions

  – Before and after copies of CICS resource definitions that have been updated by CICS CM

- Agent

  A CICS CM program, running in an application CICS region, that performs actions on that region on behalf of the CICS CM server. When a CICS CM client requests install, newcopy, or discard actions for an application region, the server uses a CICS distributed program link (DPL) to invoke the agent in that region. This agent is required only if you want to perform install, newcopy, or discard actions on an active CICS region whose resource definitions are stored in a CSD file. This agent is not used for CICS regions that are managed by CICSPlex SM. For those regions, CICS CM uses the CICSPlex SM API to perform these actions.

Figure 3-22 shows the components of CICS CM and how they fit into your existing system environment. In this figure, the items inside the area marked *CICSPlex SM* only apply if your system uses CICSPlex SM.

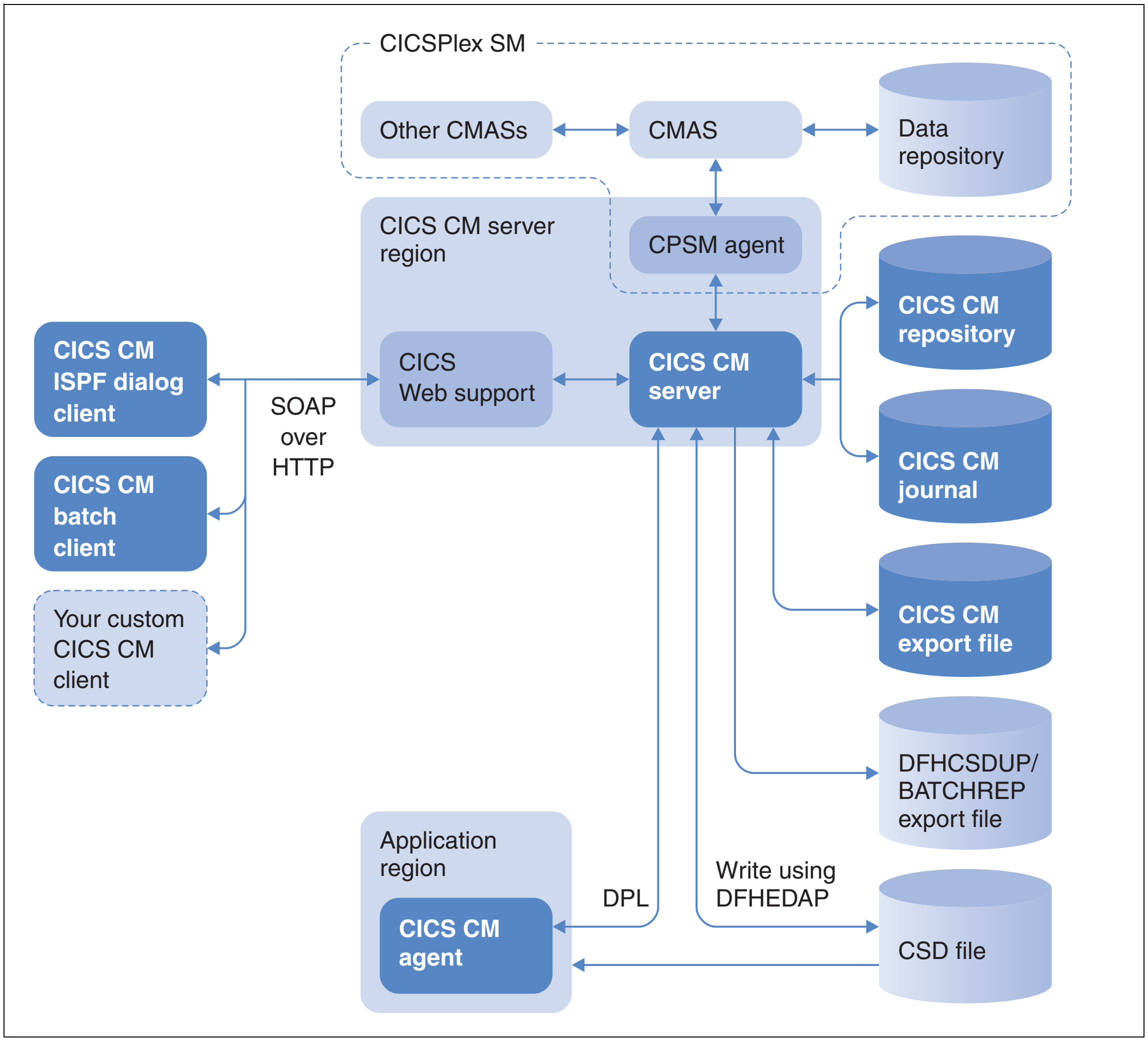


Figure 3-22 CICS CM components

The CICS CM server only uses existing, proven methods to update resource definitions. To update resource definitions stored in CSD files, the server calls the DFHEDAP program provided by CICS. This is the same program used by CICS RDO transactions such as CEDA. To update resource definitions stored in CICSPlex SM data repositories, the server calls the CICSPlex SM API.

3.15 How CICS CM can help migrate to CICS TS 3.1

The same CICS CM features that help you to manage resource definitions across multiple environments can also help you to migrate to CICS TS 3.1:

- Package the resource definitions that you want to migrate to the new CICS TS 3.1 CSD file or CICSPlex SM data repository, leaving behind any resource definitions for applications that you do not want to migrate yet.
- Transform attributes, or specify values for new attributes, for the CICS TS 3.1 CSD file or CICSPlex SM data repository.
- Compare resource definitions in your CICS TS 2.3 and 3.1 environments, to identify possible causes for differences in system behavior.

For detailed, step-by-step demonstrations of how to use CICS CM to help migrate to CICS TS 3.1, see Chapter 8, “Migrating CICS TS 2.3 CSD to CICS TS 3.1 CSD” on page 197, and Chapter 9, “Migrating CICS TS 2.3 CSD to CPSM 3.1 BAS” on page 269.

4

Overview CICS IA

This chapter provides an overview of CICS IA and covers the following topics:

- What business issues CICS IA addresses
- What CICS IA is
- What questions CICS IA answers
- What the components of CICS IA are
- CICS IA architecture

4.1 What business issues CICS IA addresses

There are many business reasons for using CICS IA, and they vary by industry. In this section we discuss some of the business imperatives facing corporations today.

4.1.1 Mergers/acquisitions

Many banks have been involved in mergers and acquisitions. The result is that they have had to consolidate workloads and move CICS applications around for isolation reasons or to spread the workload for performance reasons. Because these applications may not be well understood by the acquiring bank and documentation may be inadequate, there is a need to understand all the resources that are associated with a given application.

4.1.2 Outsourcing

Large outsourcing companies are continually facing the problem of running CICS applications with which they are unfamiliar. Often naming standards are lacking or conflicting with other applications running on the same LPAR. Documentation may be non-existent or incomplete. Again, there is a need for a tool to facilitate the understanding of the resource interdependencies and affinities involved.

4.1.3 Maintenance or enhancement of applications

During the normal application life cycle, CICS applications require maintenance and enhancement. When a programmer who is unfamiliar with the application that he is required to modify starts the modification process, much time can be spent in trying to understand the application and the inherent flow of transactions. This learning curve can be greatly reduced through the use of CICS IA. CICS IA identifies the resources that are affected directly and indirectly.

4.1.4 Workload balancing using CPSM

Many companies are looking to balance their workload across CICS application owning regions using CICSPlex SM. In order to this they need to identify any affinities that applications may have to a particular region. CICS IA can identify these affinities.

4.2 CICS IA defined

The CICS Interdependency Analyzer for z/OS (CICS IA) is a runtime and batch system for use with CICS Transaction Server for z/OS (CICS TS) and CICS Transaction Server for OS/390® (CICS TS). It has two purposes:

- To identify CICS application resources and their interdependencies

  This function enables you to understand the makeup of your application set, such as:

  - Which transactions use which programs
  - Which programs use which resources (files, maps, queues, and so on)
  - Which resources are no longer used
  - What applications does a CICS region contain

- To analyze transaction affinities

  Affinities require particular groups of transactions to be run either in the *same* CICS region or in a *particular* region.

  Affinities information is useful in a dynamic routing environment, since you need to know of any restrictions that *prevent* particular transactions from being routed to particular application-owning regions (AORs) or that *require* particular transactions to be routed to particular AORs.

CICS IA captures either affinity or interdependency information while CICS is running and stores it in VSAM files. Subsequently the VSAM files are used to load the DB2 database tables. Sample SQL queries are provided to analyze the DB2 tables or the users can use the online query interface. Detailed batch reports can be produced from the VSAM files, if desired.

Many large organizations have been using CICS since the early 1970s — their systems growing and evolving with the business. During this time, many techniques of implementing applications have been used as a result of new functions, changing corporate standards, technical requirements, and business pressures.

Frequently, this growth has not been as structured as it might have been, with the result that many applications and services share common resources, and changes in one area typically affect many others. This can reach such a level that the system can no longer develop in a controlled manner without a full understanding of these inter-relationships. CICS IA can help you achieve this understanding.

For example, if you need to change the content or structure of a file, you need to know which programs use this file, as they will need to be changed. CICS IA can tell you this, as well as the transactions that drive the programs. CICS IA records the interdependencies between resources (such as files, programs, and transactions) by monitoring programming commands that operate on resources.

The application that issues such a command has a dependency on the resource named in the command. For example, if an application program issues the command EXEC CICS WRITE FILE(myfile), it has a dependency on the file called myfile. It might have similar dependencies on transient data queues, temporary storage queues, transactions, other programs, and so on.

The commands that are monitored are typically CICS application programming interface (API) and system programming interface (SPI) commands that operate on CICS resources. However, you can also instruct CICS IA to monitor some types of non-CICS commands that operate on non-CICS resources, for example:

- MQ calls to WebSphere MQ resources
- DLI calls to IMS Database resources
- DB2 calls
- Dynamic COBOL calls to other programs

Potentially, the inclusion of any non-CICS resources gives you a fuller picture of the resources used by a transaction.

The Collector component of CICS IA collects the dependencies or affinities that apply to a single CICS region — that is, a single application-owning region (AOR) or a single, combined routing region/AOR. It can be run against production CICS regions, and is also useful in a test environment, to monitor possible dependencies introduced by new or changed application suites or packages.

From the interactive interface of CICS IA you can control collectors running on multiple regions.

Note: To ensure that you monitor as many potential dependencies or affinities as possible, use CICS IA with all parts of your workload, including rarely used transactions and abnormal situations. It is possible to store the collected information from several CICS regions into the same database. You can then review the collected dependencies/affinities using CICS IA's query interface, or produce your own SQL queries based on samples provided.

4.3 New in CICS Interdependency Analyzer for z/OS V2.1

In this section we discuss what is new in CICS Interependency Analyzer for z/OS V2.1:

- An Eclipse-based client interface and improved query management facilities
  - Make it easy for you to access the collected data and use it in the day-to-day analysis.
  - The client is based on the XML application programming interface (API), so automated processes can query the database as well.
- Timer-based collector control
  - Allows the user to start the collector for a given time of day to enable targeted data collection

    For example, you can set the tool to schedule collection in different regions throughout the data collection process.
  - Helps you to:
    - Work around high volume time periods.
    - Target collection for when an application is active.
- Enhanced single point of control capabilities
  - You can turn data collection for multiple CICS regions on and off with a single CINT command to speed selection.
  - You can select default options for all your CICS regions with a single setting or you can specify collection options to be region specific.
- A selective program and transaction Exclude list to eliminate extraneous data and reduce overhead during data capture
- Provision of CSD data set name and group-list information
- Automation of tracking of runtime impact on application change by providing program version information, enabling removal of old data by version and comparison of data by program version
- Improved installation and customisation, as well as other enhancements

4.4 What questions CICS IA answers

The following are some of the questions that can be answered with CICS IA. The purpose of listing these questions here is to give you a sense of what is possible with CICS IA.

- What regions does a particular CICS application run in?
- What are all the CICS resources used by a given application?
- What are all the CICS resources used by a given transaction?
- What transactions belong to a given application?
- What programs does a given transaction invoke?
- What transactions access a particular file and how?
- What are the resources used by a specific program?
- How is a file accessed by a particular program?
- Which affinities does a transaction have?
- What DB2, MQ, or IMS resources are used by a given application?

The relationship data collected in real time by CICS IA is ultimately loaded into a DB2 database. The database can then be queried with the CICS IA transaction CINQ or through the use of SQL in SPUFI or other DB2 query tools from IBM or other ISVs.Sample SQL queries are provided.

4.5 The components of CICS IA

The CICS IA architecture is described later in this chapter. This section is simply a high-level overview.

The design of CICS IA centers around the concept of examining the EXEC CICS commands used by applications and systems programmers. Each command and its parameters indicates the resources that will be used by the program. An analysis of these calls provides a view of resource interdependencies.

The scanner component

It is possible to write a program to examine the program load modules and report on the EXEC CICS commands and their parameters. The Scanner component of CICS IA is just such a program. It produces a report that tells for each program the commands issued, the programming language used, and the resources involved. The scanner also indicates whether the command is a possible affinity, a possible dependency, or both.

The collector component

However, there is a problem with only using the *scanner*. It does not show the execution-time path through the code and which commands are, in fact, executed. So an approach is needed that intercepts the commands as they are executed and captures the name of the program and its context (for example, which program called it, which transaction initiated it, and so on). The *collector* component is that part of CICS IA that does this capture function and stores the data in an MVS data space.

The collector function can be activated across multiple CICS regions from a single point of control and the data can be collected across these regions and written to a VSAM file shared between these regions using a file owning region (FOR) or using RLS. The collector can either collect dependency or affinity information. It cannot collect both at once. At specified intervals or on operator command, the data space is written to VSAM files.

Once the data is collected, CICS IA provides a set of utilities to enable this data to be loaded into a DB2 database. Having the data in DB2 provides many opportunities for detailed analysis using standard SQL queries or using the online CICS BMS interface that CICS IA provides. Using this analysis can help to:

- Use CICS resources more efficiently.
- Balance application workload for continuous availability.
- Improve the speed and reduce cost of application maintenance.
- Minimize the impact of routine application maintenance for the end user.
- Plan reuse of existing applications as e-business applications or build new applications more efficiently.

The reporter component

The reporter component is a set of batch programs that can produce reports from these files. A summary report can be run or, if desired, a detailed report can be run.

The query component

Subsequently, the VSAM files are loaded into a DB2 database. Once the data is available in the DB2 database, the query component can be used to view resource interdependencies. This component comprises a set of CICS transactions (COBOL/BMS).

CICS IA also provides sample SQL queries for use with SPUFI or other DB2 query tools from IBM or other ISVs.

The CICS IA client

CICS IA V2.1 provides an Eclipse-based client to query the DB2 database.

4.6 CICS IA architecture

The components of CICS IA described above are shown in Figure 4-1. This section describes these components in detail.

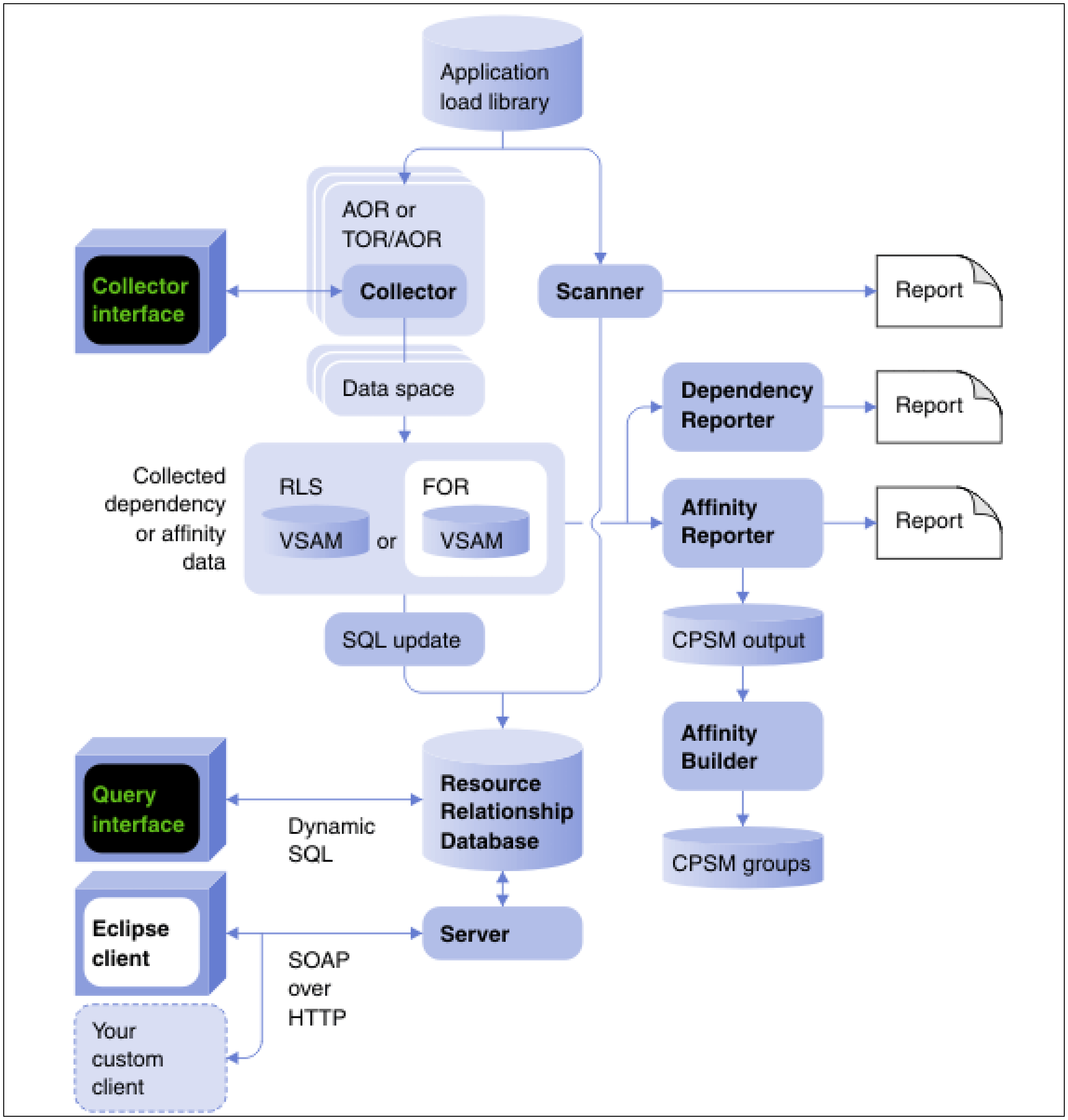


Figure 4-1 CICS IA component architecture

4.7 How CICS IA can help migrate to CICS TS 3.1

CICS IA identifies resources for each set of applications that needs to be migrated (in runtime and via the load lib scanner). This information can be used to:

- Identify non-LE and OS VS Cobol programs. If any are found, Debug Tool Utilities and Advanced Functions can be used to convert these.
- Identify applications that do not conform to threadsafe standards. In order to improve performance (if needed) on CICS TS 3.1, CICS applications need to conform to threadsafe standards.
- Identify a group of resources for each application that needs to be migrated from the current CICS TS test environment to the CICS TS 3.1 test environment. This information will be used by CICS CM to create CICSPlex SM BAS definitions to be stored in CICSPlex SM repository.
- Identify affinities and use this information to build CICSPlex SM definitions. CICS IA creates and enables you to manage affinity groups. Alternatively, this information can be used to eliminate affinities prior to CICSPlex SM enablement.

4.8 Product information

CICS IA is a one-time-charge (OTC) product and is *not* included as a part of CICS TS. The program product number for CICS IA is 5697-J23. The corresponding support and subscription number is 5697-J23.

CICS TS releases supported:

- CICS Transaction Server for z/OS Version 3
- CICS Transaction Server for z/OS Version 2
- CICS Transaction Server for OS/390 Version 1

5

Overview CICS Debug Tool and Debug Tool Utilities and Advanced Functions

Debug Tool lets application programmers trace through an application program to determine if and where any errors exist and to identify areas of potential problems.

Debug Tool Utilities and Advanced Functions provides enhancements to Debug Tool. Of specific interest for migration are the following functions:

- Debug Tool COBOL Modernization Utility (CCCA)
- Debug Tool Coverage Utility

These functions are described in more detail in this chapter, as is a summary of Debug Toll and the other features of Debug Tool Utilities and Advanced Functions.

5.1 Debug Tool

Debug Tool helps the developer test programs and examine, monitor, and control the execution of programs written in assembler, C/C++, COBOL, and PL/I on a z/OS or OS/390 system.

Applications can also include other languages. For such applications, Debug Tool provides a disassembly view that lets the user debug, at the machine code level, the corresponding portions of applications. Of course, in the disassembly view, debugging capabilities are limited.

Debug Tool can be used to debug programs in a batch mode, interactively in a full-screen mode, or in a remote debug mode. The latter mode cannot be used with some compilers, for example, PL/I for MVS and VM.

You must use the correct compiler options to be able to use Debug Tool.

5.1.1 Debug Tool interfaces

The terms *batch mode*, *remote debug mode*, and *full-screen mode* identify the types of debugging interfaces provided by Debug Tool.

Batch mode

To run Debug Tool in batch mode, a command file is prepared in advance, with a predefined series of Debug Tool commands, which will be performed on a running batch application.

No terminal input or user interactive intervention is possible in a batch debugging session.

The results are saved in a log data set and can be reviewed and analyzed when a batch debugging session is finished.

There are several ways to define the TEST runtime option with the commands file specified, including:

- In the PARM parameter of the JCL EXEC statement
- Using the PLIXOPT string in the PL/I source application
- Using CEEUOPT CSECT linked with an application

Remote debug mode

In remote debug mode, the host application starts Debug Tool, which uses a TCP/IP connection to communicate with a remote debugger on your Windows workstation. Not all compilers are compatible with this mode.

Debug Tool, in conjunction with a remote debugger, provides users with the ability to debug host programs, including batch programs, through a graphical user interface (GUI) on the workstation.

The following remote debuggers are available:

- Compiled Language Debugger component of WebSphere Studio Enterprise Developer

  This remote debugger is the recommended choice since it offers more functionality than the IBM Distributed Debugger.

- IBM Distributed Debugger

  This remote debugger is available through several products, for example:

  – OS/390 C/C++ Productivity Tools
  – VisualAge® for Windows family of products

Both remote debuggers run on Windows NT® 4.0, Windows 2000, and Windows XP.

WebSphere Studio Enterprise Developer Debugger

This debugger is available as part of the WebSphere SDK. On entry to the SDK, there is a small box on the left-hand side of the screen that looks like a divided screen. This box is titled Open a Perspective. To debug, click the small bug icon and the screen will reformat, as shown in Figure 5-1.

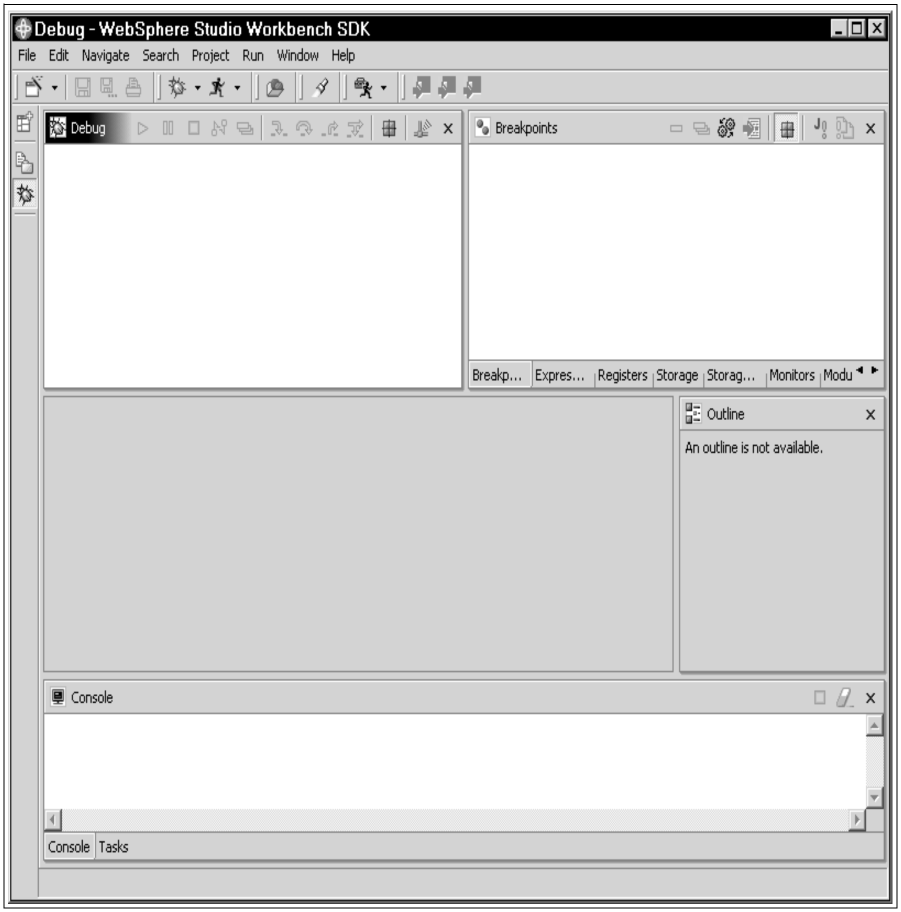


Figure 5-1 The WebSphere Studio Workbench Debug Perspective

At the top of the upper left-hand corner window, entitled Debug, there is a series of icons, as shown in Figure 5-2.

Figure 5-2 The debugger control icons

The final icon is the Listen button. If you click this button, the debugger will begin to listen on channel 8001. The next step to debugging is going back to your TSO, batch, IMS, or CICS session and starting the job you wish to run with a suitable TEST runtime option. The test runtime option will need to specify the TCP/IP address of the machine that is running the WSED debugger. In this case, the program invocation is as shown in Figure 5-3.

```
  REDBK1.SPFLOG1.LIST has been deleted.
 READY
call dev.load(knightm) 'TEST(,,,TCPIP&9.30.62.149%8001:*) / 8'
```

Figure 5-3 Invoking the program to be debugged on the mainframe

In this example we have invoked the program in TSO, but we could have invoked it in batch just as easily. The TEST runtime option contains enough information that the runtime on the mainframe can contact WSED on our PC.

The WSED screen pops up, along with a dialog box to warn about the state of the program, as shown in Figure 5-4.

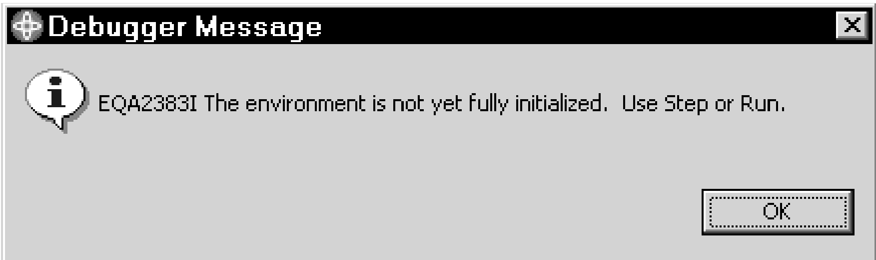


Figure 5-4 WSED initial program status message

Once we have clicked **OK** and stepped into the program, we have a fully functional point-and-click debugging environment, with many different views such as breakpoints, variables, registers, and so forth, as shown in Figure 5-5.

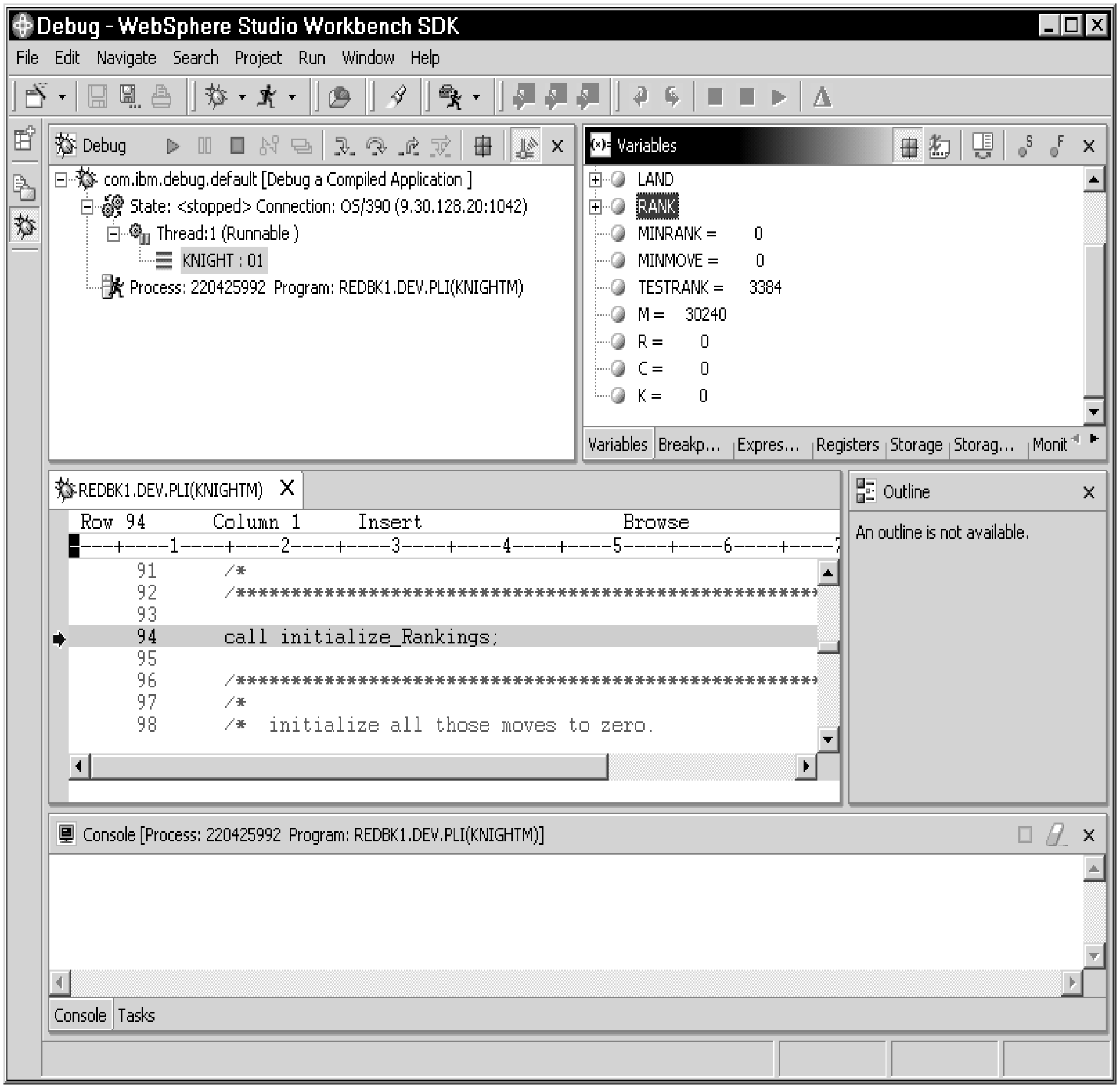


Figure 5-5 The WSED Debug screen

IBM Distributed Debugger

Any version of the IBM Distributed Debugger should work with Debug Tool and all of its supported languages.

Using this interface, your host application running under control of the Debug Tool will start a TCP/IP connection to a Windows NT, 2000, or XP workstation on which the IBM Distributed Debugger program was started and listens for requests on a specified port (default is 8000). This interface is supported by all environments including CICS, IMS, and UNIX® System Services (USS).

> **Note:** In some TCP/IP installations, a SYSTCPD DD card may be needed to point to your installation's TCPIPDATA.

Debugging with the IBM Distributed Debugger is a two-step process:

1. Start the remote debugger on your workstation. For the IBM Distributed Debugger on a workstation, this involves a command like:

   ```
   C:\IBMDebug\bin\idebug.exe -qdaemon -quiport=8000
   ```

2. Run your program using the TEST runtime option. The runtime option to be used needs to define your workstation's TCP/IP address. For example, with a COBOL program this might be:

   ```
   /TEST(ALL,*,PROMPT,VADTCPIP&9.30.40.117%8000:*)
   ```

A sample session of the IBM Distributed Debugger is shown in Figure 5-6.

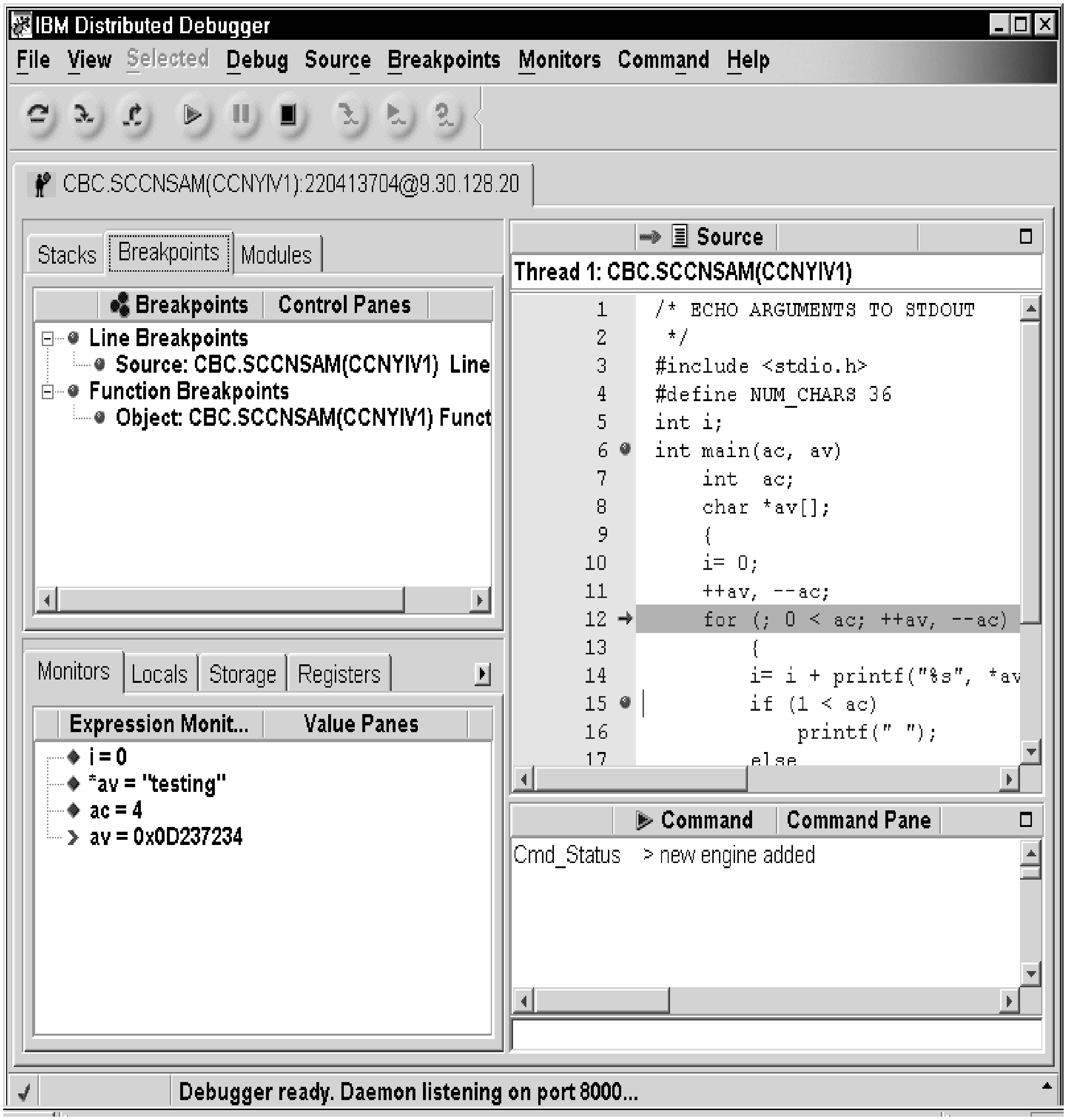


Figure 5-6 A view of the IBM Distributed Debugger screen

Full-screen mode

Debug Tool provides an interactive full-screen interface on a 3270 device, with debugging information displayed in the following three windows:

- Source window: Displays the program source or listing
- Log window: Displays a record of commands and other interactions between Debug Tool and the program
- Monitor window: Indicates changes in the program

Programs written in all languages supported by Debug Tool can be debugged in full-screen mode.

Source window

The Source window displays the source file or listing. It has four parts:

Header area Identifies the window, shows the compile unit name, and shows the current position in the source or listing.

Prefix area Occupies the left-most eight columns of the Source window. Contains statement numbers or line numbers that can be used when referring to the statements in your program. The prefix area can be used to set, display, and remove breakpoints with the prefix commands AT, CLEAR, ENABLE, DISABLE, QUERY, and SHOW.

Source display area Shows the source code (for a C/C++ program), the source listing (for a COBOL or PL/I program), a pseudo assembler listing (for an assembler program), or the disassembly view (for programs without debug information) for the currently qualified program unit. If the current executable statement is in the source display area, it is highlighted.

Suffix area A narrow, variable-width column at the right of the screen that Debug Tool uses to display frequency counts. It is only as wide as the largest count it must display. The suffix area is optional. To show the suffix area, enter `SET SUFFIX ON`. To hide the suffix area, enter `SET SUFFIX OFF`. It can also be set on or off with the Source Listing Suffix field in the Profile Settings panel.

Log window

The Log window records and displays user interactions with Debug Tool. All commands that are valid in line mode, and their responses, are automatically appended to the Log window. The following commands are not recorded in the Log window:

- PANEL
- FIND
- CURSOR
- RETRIEVE
- SCROLL
- WINDOW
- IMMEDIATE
- QUERY prefix command
- SHOW prefix command

If SET INTERCEPT ON is in effect for a file, that file's output also appears in the Log window. The user can exclude STEP and GO commands from the log by specifying SET ECHO OFF.

By default, the Log window keeps 1000 lines for display. To change this value, enter SET LOG KEEP *n*, where *n* is the number of lines you want kept for display. The maximum number of lines is determined by the amount of storage available.

Monitor window

The Monitor window is used to continuously display output from the MONITOR LIST, MONITOR QUERY, MONITOR DESCRIBE, and SET AUTOMONITOR commands.

If this window is not open, Debug Tool opens it when the MONITOR or SET AUTOMONITOR commands are entered. Its contents are refreshed whenever Debug Tool receives control and after every Debug Tool command that can affect the display.

When the MONITOR command is issued, it is assigned a reference number between 1 and 99, then added to the monitor list. The user can specify the monitor number. However, it must either replace an existing monitor number or be the next sequential number.

When the user issues the SET AUTOMONITOR ON command (if available), the following line is displayed at the bottom of the list of monitored variables:

```
********** AUTOMONITOR **********
```

Variables that are added to the Monitor window as a result of the SET AUTOMONITOR command are displayed underneath this line.

While the MONITOR command can generate an unlimited amount of output, bounded only by your storage capacity, the Monitor window can display a maximum of only 1000 scrollable lines of output.

If a window is not wide enough to show all the output it contains, either SCROLL RIGHT (to scroll the window to the right) or ZOOM (to make it fill the screen) can be used.

In most cases, the user can update the values of monitored variables by typing new values over the displayed values.

A sample of the Debug Tool screen in the full-screen mode is shown in Figure 5-7.

```
COBOL    LOCATION: PROG1 :> 48.1
Command ===>                                                   Scroll ===> PAGE
MONITOR --+----1----+----2----+----3----+----4----+----5----+----6 LINE: 1 OF 1
******************************* TOP OF MONITOR ********************************
0001  1 TEMP    'TOP   '
****************************** BOTTOM OF MONITOR ******************************



SOURCE: PROG1 ----1----+----2----+----3----+----4----+----5----+ LINE: 46 OF 53
      46              MOVE "TOP" TO STR1 MOVE "BEG" TO STR2 MOVE "UP" TO STR3 .
      47              ADD 1 TO VARBL1                                         .
      48              SUBTRACT 2 FROM VARBL2                                  .
      49              ADD 1 TO R                                              .
      50              MOVE "BOT" TO STR1 MOVE "END" TO STR2 MOVE "DOW" TO STR .
      51            END-PERFORM.                                              .
LOG 0----+----1----+----2----+----3----+----4----+----5----+---- LINE: 19 OF 22
0019    LIST ( "Exiting ", %CU ) ;
0020  STEP ;
0021  MONITOR
0022    LIST TEMP ;
PF  1:?         2:STEP       3:QUIT       4:LIST       5:FIND       6:AT/CLEAR
PF  7:UP        8:DOWN       9:GO        10:ZOOM      11:ZOOM LOG  12:RETRIEVE
```

Figure 5-7 Debug Tool full-screen session

5.1.2 Compiler options

Each programming language has a comprehensive set of compiler options. It is very important to use the correct compiler options to debug the program:

C/C++ TEST(ALL) provides maximum debugging capability. There are suboptions to refine debugging capabilities.

COBOL TEST(ALL,SYM) provides maximum debugging capability. There are suboptions to refine debugging capabilities. Some suboptions are used only with a specific version of COBOL. When using Enterprise COBOL for z/OS and OS/390 Version 3 or COBOL for OS/390 and VM Version 2 Release 2 compilers, the TEST(NONE,SYM,SEPARATE) compiler option retains most of the Debug Tool's capabilities. The suboption SEPARATE instructs the compiler to store debugging information and symbol tables in a separate file. The suboption NONE specifies that there are no compiled-in hooks, so the Dynamic Debug facility must be activated during a debug session.

PL/I TEST(ALL,SYM) provides maximum debugging capability. Programs compiled with the PL/I for MVS or OS PL/I compilers must specify the SOURCE suboption. The suboptions BLOCK, STMT, PATH, and ALL regulate the points at which compiler inserts hooks. The suboption SYM controls the insertion of symbol tables into the object file. These tables are used by Debug Tool to obtain information about program variables. The syntax for the TEST compiler option of the Enterprise PL/I compilers is slightly different. Refer to the documentation that corresponds to the version of the compiler you are using for a description of the TEST compiler option.

Assembler The ADATA option must be specified. This option generates a SYSADATA file, which the EQALANGX postprocessor needs to create a debug file (also called the EQALANGX file).

5.1.3 Link-edit options

In most cases, Debug Tool does not require specific link-edit options for application programs.

When using the DTCN transaction to manage debugging profiles in CICS, the main programs to be debugged should be link-edited with the object module EQADCCXT if they are written in PL/I or C/C++. When using the CADP transaction, which is available with CICS Transaction Server for z/OS V2.3 and later, this is not required.

5.1.4 Runtime TEST option

> **Note:** The following information is based on the description of the TEST runtime option provided in *Debug Tool V5R1 Reference and Messages*, SC18-9304-00.

About runtime TEST option

The TEST runtime option is used to specify the conditions under which Debug Tool will assume control of an application. The basic format of the instruction is as follows:

NOTEST Specifies that Debug Tool is not started at program initialization. However, it can still be started with CEETEST, PLITEST, or __ctest(). The suboptions specified with NOTEST are used when Debug Tool is stared (if it is started). Note that if the TEST option is specified somewhere that has a higher precedence than where the NOTEST option is, the values on the NOTEST option will be taken as defaults.

TEST Indicates that Debug Tool is given control according to the specified sub-options.

test_level

The test_level suboption has three possible values:

ALL This default value specifies that Debug Tool gains control, even without defined breakpoints, at the attention function, any Language Environment condition of severity 1 or above, application termination.

ERROR Without a defined AT OCCURRENCE for a particular condition, Debug Tool will only get control at the occurrence of the attention function, any Language Environment condition of severity 1 or above, application termination.

NONE This specifies that no condition will cause Debug Tool to gain control without a defined AT OCCURRENCE for a particular condition or AT TERMINATION.

commands_file

The commands_file designator is a valid DD name or file name that gives the name of the primary commands file for this program run. If this parameter is empty, requests for commands will go to the user terminal.

If an asterisk (*) is specified instead of a commands file, then no commands file is expected.

prompt_level

The prompt_level suboption is used to specify whether an initial commands list is unconditionally executed during program initialization or to specify particular Debug Tool commands:

PROMPT This default value specifies that Debug Tool is invoked at Language Environment initialization.

NOPROMPT This specifies that the Debug Tool is not invoked at Language Environment Initialization.

***** Equivalent to NOPROMPT.

; Equivalent to PROMPT.

command_list This is a character string that specifies a valid Debug Tool command. It has a maximum of 250 characters. It should be enclosed in single or double quotation marks whenever it contains embedded blanks, commas, or parenthesis. The use of a preferences file is recommended rather than putting a command list in the third sub-option.

preferences_file

The final suboption is the preferences_file. This suboption controls the interface and location of the debugger as well as the location of a preferences file that becomes the first source of Debug Tool commands after Debug Tool has started. This suboption has a complex format, as shown in Figure 5-8.

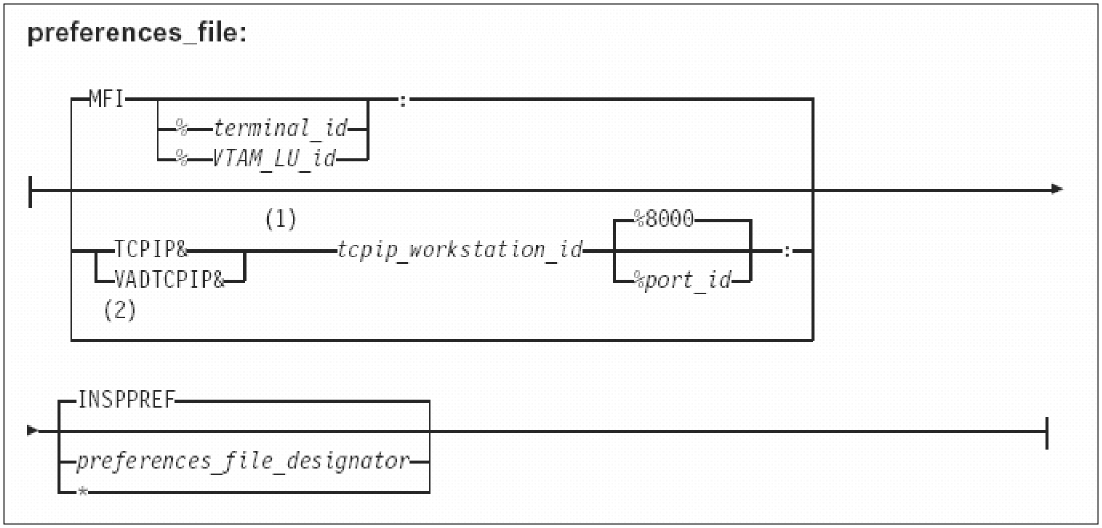


Figure 5-8 Preferences file syntax

In Figure 5-8 on page 96:

- (1) specifies remote debug mode using WebSphere.Studio Enterprise Developer.
- (2) specifies remote debug mode using the VisualAge Remote Debugger or the IBM Distributed Debugger.

The terms used are as follows:

- MFI (Main Frame interface): This specifies that Debug Tool should be started in full-screen mode through a VTAM® terminal for your debug sessions.
- terminal_id (CICS only): This specifies up to a four-character terminal ID to receive Debug Tool screen output during dual terminal debugging of a CICS application.
- VTAM_LU_id (full-screen mode using a VTAM terminal only): This is a VTAM logical unit (LU) name for a terminal. This cannot be used to debug CICS applications. It is used for full screen debugging. The application will continue in its original location and the new unused VTAM terminal will be used for the full screen MFI debugger. This can be used for batch or IMS debugging as well as traditional debugging.
- INSPREF (or blank) is the default DD name for the preference file.
- preferences_file_designator: This is a valid DD name, data set name, or file name specifying the name of the preference file. The preference file is a command file that can be used to specify the Debug Tool commands to be executed on entry to your environment.
- *: Instead of a preference file, the asterisk indicates that no preference_file is supplied.
- %port_id: This specifies the TCP/IP port on your workstation to be used by the remote debugger. The default port for WebSphere Enterprise Developer Compiled Language debugger is %8001. The default port for the IBM Distributed Debugger or the VisualAge Remote Debugger is %8000.

Sample runtime options

Some options are:

- NOTEST

  Debug Tool is not started unless there is a call to CEETEST, PLITEST, or __ctest().

- NOTEST(ALL,MYCMDS,*,*)

  Debug Tool is not started at initialization. If it is started by a call to CEETEST, PLITEST, or __ctest(), the suboptions will be used and the instructions in the file allocated to DD name MYCMDS.

- TEST

  This searches through runtime options specified in any level (CEEUOPT, pragma runopts, CEEROPT) and brings up the debugger using the options found on a prior TEST instruction or the default TEST values.

- TEST(ALL,*,*,*)

  Debug Tool is not started initially, but any condition in the code (or a call to CEETEST, PLITEST, or __ctest) will cause it to be started. No preference file or command file is expected.

- TEST(NONE,,*,*)

  Debug Tool is not started until a call to CEETEST, PLITEST, or __ctest() occurs.

- TEST(ALL,test.scenario,PROMPT,prefer)

  Debug Tool is started after environment initialization but before program initialization. The first commands executed are found in the file referred to by the DDNAME prefer.

- TEST(ALL,,,MFI%F100:*)

  For CICS Dual Terminal and Batch CICS transactions, Debug Tool will be started at CICS terminal F100 after initialization. Alternatively, F100 could be a very short VTAM LU ID that is used to define a terminal for debugging a batch or IMS transaction. No preference file is expected.

- TEST(ALL,,,MFI:foo) or TEST(ALL,,,foo)

  These are equivalent instructions when used in TSO that cause the full screen interface to be used in single screen mode (the same place that the program was called from). In both cases, the preference file is the file referenced by the foo DD.

- TEST(ALL,,,MFI%LU000001:*)

  This is for environments other than CICS. The MFI full screen debugger will be started on the VTAM LU whose ID is LU000001. The LU must be known to VTAM and not in session when the debugger is started.

- TEST(,,,VADTCPIP&9.30.62.149%8000:*)

  Bring up the IBM Distributed Debugger or the VisualAge Remote Debugger listening on channel 8000 on the terminal at TCP/IP address 9.30.62.149.

- TEST(,,,TCPIP&cello%8001:*)

  Bring up the WebSphere Studio Enterprise Developer listening on port 8001 on the machine named cello. This method does not work in all environments and requires an up-to-date name server to be available. It was found that using the actual machine address is more consistently effective.

Specifying TEST runtime option

Language Environment has several ways of specifying the runtime options for a program. The following list gives them in ascending order of precedence (that is, things lower in the list can override things higher in the list):

- CEEDOPT: Language Environment options specified at installation
- CEEROPT: Region-wide CICS or IMS default options
- CLER: Transaction under CICS
- CEEUOPT: Also #pragma runopts, or PLIXOPTS
- Command Line, or _CEE_RUNOPTS
- CEEBXITA: Used by IMS Single Point of Control (SPOC) and Debug Tool IMS utilities
- Language Environment Storage Tuning User Exit
- Options defined at installation time that have the non-overridable attribute

For more information about how to specify a Language Environment runtime option see the *Language Environment Programming Guide*.

In some cases, where runtime options cannot be passed to the Language Environment, a CEEUOPT must be generated including a specific TEST runtime option. In Example 5-1 the original sample can be found as member CEEUOPT on your SCEESAMP.

Example 5-1 Defining runtime options using CEEUOPT

```
*/****************************************************************/
*/* LICENSED MATERIALS - PROPERTY OF IBM                         */
*/*                                                              */
*/* 5694-A01                                                     */
*/*                                                              */
*/* (C) COPYRIGHT IBM CORP. 1991, 2001                           */
*/*                                                              */
*/* US GOVERNMENT USERS RESTRICTED RIGHTS - USE,                 */
*/* DUPLICATION OR DISCLOSURE RESTRICTED BY GSA ADP              */
*/* SCHEDULE CONTRACT WITH IBM CORP.                             */
*/*                                                              */
*/****************************************************************/
CEEUOPT  CSECT
CEEUOPT  AMODE ANY
CEEUOPT  RMODE ANY
         CEEXOPT TEST(ALL,*,PROMPT,MFI%LUOTCP08:INSPREF)
         END
```

This member must be customized to reflect the actual terminal ID to be used in the debugging session. The name of this member also can be customized to reflect the intention of its use.

When specifying the TEST runtime option on a JCL PARM there are dependences on what is the language of the program's main entry point:

- For C/C++ and PL/I, the PARM should start with the runtime options and have a slash (/) before the program parameters.
- For COBOL, the PARM should start with the program parameters and have a slash before the runtime options.

The user can specify RPTOPTS(ON) in the JCL PARM to have the runtime options report generated. This report lists all runtime options that were in effect when the program was executed.

For CICS, the user can use the Language Environment provided CICS transaction CLER to have runtime options displayed on the terminal, as shown in Figure 5-9.

```
CLER                                                               PAGE 5 OF 6

           Language Environment Region Level Runtime Options.

Current Settings

LAST WHERE SET          OPTIONS
--------------------    ---------------------------------------------------------
Installation default    TERMTHDACT(TRACE,CESE,96)
Installation default  NOTEST(ALL,"*","PROMPT","INSPPREF")
Installation default    THREADHEAP(4096,4080,ANYWHERE,KEEP)
Installation default    THREADSTACK(OFF,4096,4080,ANYWHERE,KEEP,4096,4080)
Installation default    TRACE(OFF,4096,DUMP,LE=0)
Installation default    TRAP(ON,SPIE)
Installation default    UPSI(00000000)
Installation default  NOUSRHDLR(,)
Installation default    VCTRSAVE(OFF)
Installation default    VERSION()
Installation default    XPLINK(OFF)
Installation default    XUFLOW(AUTO)



PF:  1=Help   3=Quit   7=Back   8=Forward   10=RPT->CES
```

Figure 5-9 Transaction CLER used to display runtime options

5.1.5 Special files

There are four special files used by Debug Tool in full-screen mode:

- Save file (INSPSAFE)

  This file, if allocated by the user, is used by Debug Tool to save the sizes of panels, colors, PF keys setting, and so forth between debugging sessions. CICS does not support this file.

- Preference file (INSPPREF)

  This file contains Debug Tool commands used to customize the debugging session. The information about the user's preference file is passed to Debug Tool by specifying it in the TEST runtime option.

- Commands file (INSPCMD)

  This file contains Debug Tool commands that control the debugging session. It can be used to set breakpoints or set up monitoring for variables. The information about this file should also be specified in the TEST runtime option.

- Log file (INSPLOG)

  This file is used by Debug Tool to record the progress of the debugging session. The results of the executed commands are saved as comments, which allows you to use the log file as a commands file in later debugging sessions. Since this file is written to by the Debug Tool, we recommend allocating it as a sequential file, which will eliminate any contentions.

There is one additional special type of files used by Debug Tool. A separate debug file SYSDEBUG is produced by the compiler when compilation is performed with the SEPARATE suboption of the TEST compiler option. Currently this option is available only for COBOL for OS/390 and VM and Enterprise COBOL compilers.

5.1.6 Global preferences file enhancement

Debug Tool provides a mechanism where an installation-wide default preferences file can be specified and processed. The purpose is to have:

- A mechanism that is easy to set up at Debug Tool installation and customization time, but that is transparent to Debug Tool users.
- Consistent tailoring of the debug session, such as PF key assignments and window configuration. It provides a set of installation-wide preferences to all users.

For users who want personal customization, the existing preferences file parameter in the TEST runtime option provides a way to add additional preferences or override the global settings.

Restrictions

The mechanism works in the:

- Debug Tool supported host subsystems (TSO, Batch, CICS, and IMS)
- MFI debug mode, but not in the remote debug mode (like the current INSPPREF)

Global preferences file location

The global preferences file is a sequential file or a PDS member residing on the host. The name of the file is coded in an Assembler CSECT called EQAOPTS. During Debug Tool installation and customization, you have to code the name of the global preferences file in the EQAOPTS CSECT, assemble it, and build the load module, EQAOPTS. The load module is then placed in a private data set concatenated in the 'load module search path' before hlq.SEQAMOD.

During initialization time, Debug Tool loads in the EQAOPTS module and retrieves the global preferences file name.

Using EQAOPTS options file

EQAOPTS uses the EXAXOPT macro to define the global preferences. Two option are available, as shown in Example 5-2.

Example 5-2 EQAOPTS

```
EQAOPTS  CSECT      ,
EQAOPTS  AMODE     31
EQAOPTS  RMODE     ANY
                   EQAXOPT  GPFDSN,'DEVELP.TEST.GLBLPREF'
                   EQAXOPT  SVCSCREEN,CONFLICT=NOOVERRIDE
                   EQAXOPT  END
                   END,
```

The options are:

- Global preferences file data set name

  GPFDSN provides the data set name.

  To have a consistent tailoring of the debug session such as PF key assignment, window configuration, or other installation-wide default preferences, the global preferences file makes sure that every debug session is initialized with the preferences in the global preferences file.

- SVC screening filter

  SVCSCREEN(parm1,parm2)

parm1	Enablement. Possible values are ON and OFF.
parm2	Conflict resolution. Possible values are CONFLICT=OVERRIDE and CONFLICT=NOOVERRIDE.

  The filter allows you to enable SVC screening and to override the SVC screening already put in place by other vendor product. More information is available in the *Customization Guide*.

Global preferences file content

The Debug Tool commands allowed in the current preference file (INSPPREF) are eligible in the global preferences file.

Function

Debug Tool processes the global preferences file at initialization time, like the existing user preferences file and the commands file. The order of processing is as follows:

1. Global preferences file
2. User preferences file
3. Commands file

If a command is specified multiple times in a file, or in multiple files, the last instance is used.

If a user adds or modifies a preference by issuing the command directly in the command line, it is valid only in the current session and is not persistent across sessions.

5.1.7 Finishing a Debug Tool session

There are several ways to finish working with the Debug Tool. It is important to choose an appropriate one because it affects what actions will be performed in relation to the databases used in the program.

QUIT

Soft termination of the program occurs at the current statement, with a prompt message.

QQUIT

Soft termination of the program occurs at the current statement, with no prompt message.

QUIT ABEND

The program will be abended (ABENDU4038) at the current statement, with a prompt message.

> **Note:** When using QUIT ABEND, any non-committed database updates will be rolled back. This is the recommended setting to be used as the default for PF3.

The user can associate QUIT ABEND with the PF3 key by issuing the following command:

```
SET PF3 ‘ABEND’ = QUIT ABEND;
```

> **Note:** This PF key setting will be saved to the INSPSAFE file, if one was allocated. CICS does not support INSPSAFE. Therefore this command should be placed in the INSPPREF file when running under CICS. This approach can also be used for other environments.

QUIT DEBUG

The debugging session will be terminated, but the program will continue to run to completion.

5.1.8 Built-in functions

There are several built-in functions defined in Debug Tool. Two of them are presented here.

- %HEX returns the hexadecimal value of the operand.
- %GENERATION (PL/I) returns a specific generation of a controlled variable in the program.

5.1.9 Dynamic Debug facility

The Dynamic Debug facility enables the user to debug COBOL programs compiled with the NONE suboption of the TEST compiler option, assembler, and disassembled programs.

The user must activate the Dynamic Debug facility (by using the command SET DYNDEBUG ON) to debug programs that run without the Language Environment runtime.

The Dynamic Debug facility can be used to improve the performance of programs with compiled-in hooks (compiled with certain compilers) while debugging them.

Programs written in C/C++ and PL/I must be compiled with the TEST option.

If the Dynamic Debug facility has been installed, the initial setting is ON. If it was not installed, the initial setting is OFF and the facility cannot be activated by the user.

5.2 Debug Tool Utilities and Advanced Functions

Debug Tool Utilities and Advanced Functions adds tools to help the user with the following tasks:

- Preparing high-level language programs for debugging by helping convert, compile, and link.
- Preparing assembler programs for debugging by helping assemble, create debug information, and link.
- Conducting analysis on test cases to determine how thoroughly test cases test programs (Debug Tool Coverage Utility).
- Starting and running a program in foreground or batch by storing and using setup information. Setup information can be the runtime parameters, libraries, and names of input and output data sets.
- For IMS Version 8, browsing and editing the Language Environment runtime parameters table.
- Creating a batch job for a private IMS message region with customized load libraries and region attributes.
- Converting old COBOL source code and copybooks to new versions of COBOL by using COBOL and CICS Command Level Conversion Aid (CCCA).

5.2.1 Debug Tool conversion utility CCCA

CICS TS 3.1 has removed support for OS/VS COBOL. Hence, all such applications must be converted to be compliant with a supported LE COBOL compiler. CCCA can assist in this task by identifying COBOL language elements and CICS commands in the input source program that are:

- Not supported by the target language
- Supported in a different manner

Having identified any such elements CCCA can:

- Convert them to the equivalent in the target language.
- Remove them.
- Flag them.

How CCCA works

CCCA is an interactive system comprising ISPF panels that enable you to access a batch (MVS) or foreground (VM) conversion application. You use CCCA online ISPF panels to:

- Define the type of conversion you want.
- Submit a batch job (MVS) or run CCCA in foreground (VM) to convert your programs.

Figure 5-10 shows the three phases of a conversion job.

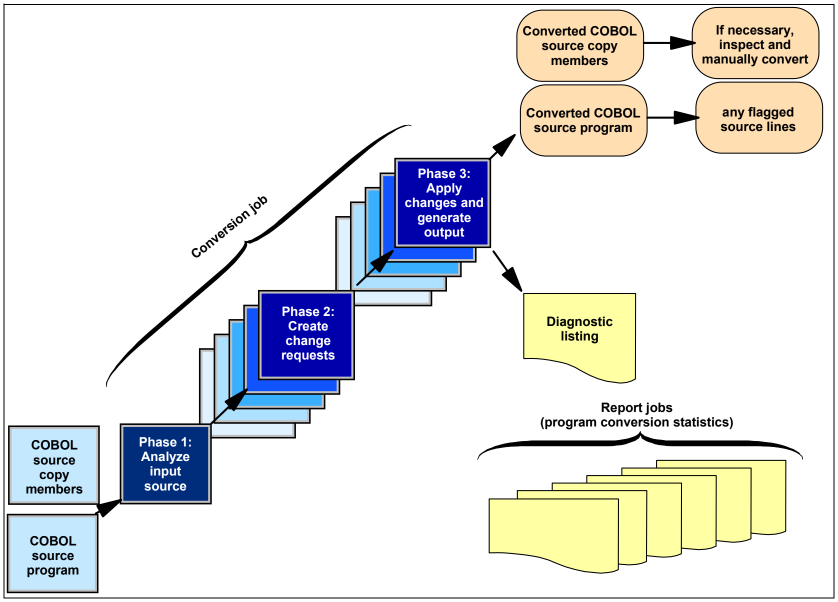


Figure 5-10 The three conversion phases

Phase 1: Analyze input source

At the start of a conversion job, phase 1:

- Extracts copy members from the appropriate copy libraries and merges them with the source program
- Translates the original source program and copy books into a set of character strings known as tokenized source
- For each language element in the tokenized source, identifies whether conversion is required, and if so, which Language Conversion Program (LCP) to use

Phase 2: Create change requests

For each item that needs converting, phase 2:

- Loads an LCP
- Runs the LCP
- Generates change requests

Phase 3: Apply changes and generate output

Finally, phase 3:

- Applies the change requests from phase 2, creating new source programs and, if required, new copy members
- Generates the Diagnostic listing

5.2.2 Debug Tool Coverage Utility

Debug Tool Coverage Utility (Coverage Utility) is a tool that measures test coverage in application programs that have the following characteristics:

- Written in the COBOL, PL/I, C/C++, and assembler languages
- Compiled by certain IBM COBOL, PL/I, and C/C++ compilers or assembled by the High Level Assembler or Assembler H

Coverage Utility enables you to run application programs in a test environment and retrieve information to determine which code statements have been executed. This process is called measuring test case coverage.

Coverage Utility has the following advantages:

- Low overhead

  For a test case coverage run, Coverage Utility typically adds very little to the execution time of your program. Coverage Utility inserts SVC instructions into your application object modules as breakpoints, and then is given control by MVS when these SVCs are executed. Most breakpoints are removed after their first execution. The increase in test program execution time is minimal because of this technique.

- Panel-driven user interface

  You can use an ISPF panel-driven interface to create JCL for executing Coverage Utility programs.

Monitoring coverage overview

Running Coverage Utility consists of the following steps:

1. Setup. Prepare to monitor programs.
   a. Compile the source code that you want to analyze, using the required compiler options.
   b. Generate Coverage Utility JCL by using the Coverage Utility ISPF dialog:
      i. Edit the Coverage Utility control file.
      ii. Create the setup JCL.
      iii. Create the start monitor JCL.

iv. Create the report or summary JCL.

c. Edit the link-edit JCL to include the modified object modules that are created when the setup JCL is run. Alternatively, you can instrument load modules after your build process.

d. Edit the GO JCL (or program invocation) to point to the instrumented load module that was provided in step 1c.

2. Execution. Run a monitor session.

   a. Run the setup JCL (created in step 1bii).
   b. Run the link-edit JCL (created in step 1c).
   c. Run the JCL to start a monitor session (created in step 1biii).
   d. Run your application using the load modules from step 2b.
   e. Stop the monitor session (with the EQACUOSP command).

3. Report. Obtain Coverage Utility reports.

   a. Run the report or summary JCL (created in step 1biv).

   b. Optional: Run the export utility to save the output in XML format for use by other programs.

Figure 5-11 is a flow diagram of the entire process.

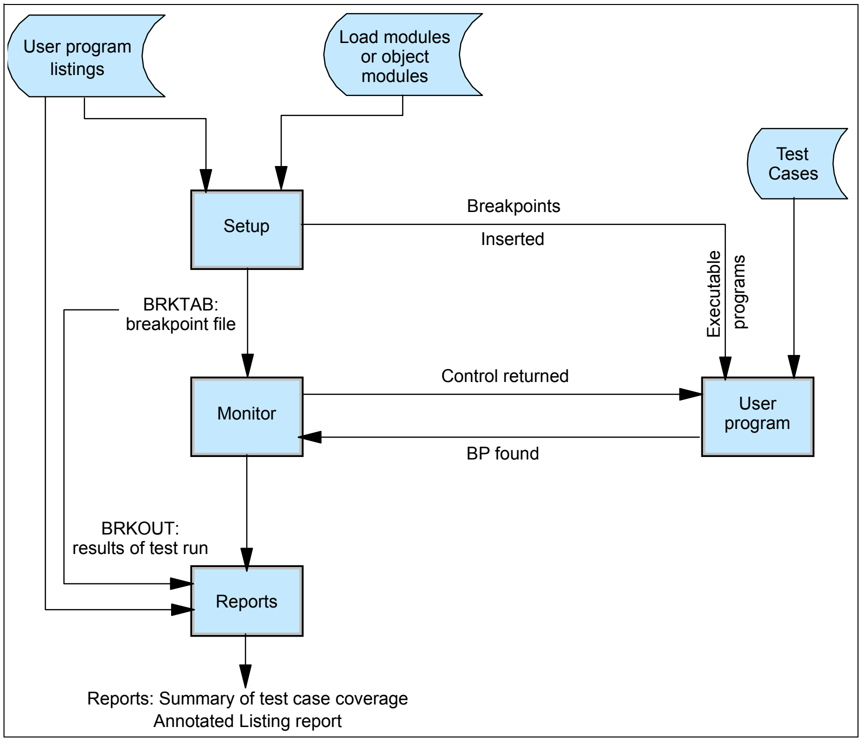


Figure 5-11 Monitoring coverage process

Part 2

Migration

In this part of the book we first discuss migration considerations when migrating from CICS TS V2 to CICS TS 3.1. Then we take you step-by-step through a few migration scenarios, showing the migration and how the CICS tools have helped us achieve this.

6

Migration considerations

In this chapter we discuss the new, changed, and removed functionality when migrating from CICS Transaction Server V2.R3 (CICS TS 2.3) to CICS Transaction Server V3.R1 (CICS TS 3.1). We also highlight other considerations if you are migrating from lower releases such as CICS Transaction Server V2 R2 (CICS TS 2.2). For migration from CICS Transaction Server V1R3 (CICS TS 1.3) see Appendix B, “Migrating from CICS TS 1.3 considerations” on page 429. The areas we focus on are:

- Software prerequisites
- General external changes
- RDO
- Application and systems programming interfaces
- Global user exits
- Monitoring and statistics
- Functional changes
- Language Environment
- Obsolete function removal

6.1 CICS Transaction Server V3.1 elements

CICS Transaction Server V3.1 Elements

- **CICS 0640**
- **CICSPlex SM 310**
- **CICS Information Center**
- **REXX Development System and Runtime Facility for CICS/ESA**
- **CICS Application Migration Aid Version**
- **CICS Integrator Adapter for z/OS**
- **WebSphere Studio Enterprise Developer V5.1 promotion**
  - 1 unrestricted entitlement, no service entitlement
  - Integrated development environment for CICS and WebSphere
    - COBOL, PL/I, Java for CICS and J2EE applications

Figure 6-1 CICS TS 3.1 elements

When you buy CICS Transaction Server V3 R1 you get the following:

- CICS Functional level CICS TS V3 R1 (Internal level CICS 0640)
- ONC™ RPC support, CICS Web interface, CICS DB2 attachment facility, CICS/DDM
- CICSPlex SM at functional level CICS TS V3 R1

  Updated to support new levels of function in CICS. CICSPlex SM becomes an exclusive element in CICS TS Release 3. IBM CICSPlex System Manager for MVS/ESA™ Version 1 Release 3 continues to be available for customers that are not yet ready to migrate to CICS TS (for example, customer with CICS/ESA® Version 4 Release 1 or earlier).
- Application Migration Aid at functional level CICS TS V1 R1

  First available in 1990, this element is still available stand-alone as IBM Customer Information Control System (CICS) program offering, CICS Application Migration Aid, program number 5695-061.

- REXX for CICS at functional level CICS TS V1 R2 (REXX for CICS/ESA V1R1)

  Separately available as REXX for CICS, program number 5655-B54.
- IBM CICS Integrator Adapter for z/OS

  Server runtime environment to those Adapter services that are modelled, generated, and deployed using the Service Flow Modeller plug-in of the WebSphere Developer for zSeries® product.
- One unrestricted entitlement to WebSphere Studio Enterprise Developer V5

6.2 Software prerequisites

The prerequisites are:

- z/OS V1.4 or later
  - CICS will not initialize unless the minimum prerequisite level of the operating system is installed.
  - Some components of CICS are installed in PDSE and HFS files:
    - The OMVS address space, UNIX Systems Services, must be active in full-function mode during the install process.
    - The jobs to create the HFS files and directories require superuser authority.
  - LE library SCEERUN must be available to CICS during CICS initialization.
  - z/OS Conversion Services must be enabled.
- IBM SDK for z/OS, Java 2 Technology Edition, Version 1.4

  Must be at the 1.4.2 level. PTF UQ90449.

The CICS installation process does not alter if you have data conversion requirements.

However, to get the benefits of z/OS conversion services, if your system requires support for the conversion of UTF-8 or UTF-16 data to EBCDIC, you must enable the z/OS conversion services and install a conversion image that specifies the conversions that you want CICS to perform.

Refer to the instructions in the *z/OS Support for Unicode: Using Conversion Services manual*, SA22-7649, to see the steps needed to set up and configure conversions supported though the operating system services.

CICS TS 3.1 requires the IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.2. The 1.4.2 level is available by applying PTF UQ90449.

6.2.1 Optional software minimum levels

For WS-Security support, the IBM XML Toolkit for z/OS V1.7 is required. This is a no-charge product, program number 5655-J51.

The following levels of products are supported for use with CICS TS for z/OS Version 3.1:

- IMS Database Manager V7 (5655-B01), IMS Database Manager V8 (5655-C56), IMS Database Manager V9 (5655-J38).
- DB2 Universal Database™ Server for OS/390 V6.1 (5645-DB2). For SQLJ/JDBC support, with PTF for APAR PQ84783 DB2 V6 does not support DB2 Group Attach.
- DB2 Universal Database Server for OS/390 V7.1 (5675-DB2). For SQLJ/JDBC support, with PTFs for APARs PQ84783 and 86525. For DB2 Group Attach, with APARs PQ44614, PQ45691, and PQ45692.
- DB2 Universal Database for z/OS V8.1 (5625-DB2). For SQLJ/JDBC support, with PTFs for APARs PQ84783 and 86525.
- WebSphere MQ for z/OS V5.3 (5655-F10).
- Tivoli® Decision Support for OS/390 (5698-ID9) V1.6, with necessary service applied.
- Tivoli Business Systems Manager V3.1.
- CICS Universal Client Version 5.0 or later.
- CICS Transaction Gateway Version 5.0 or later.

6.3 Installation process

This release of CICS Transaction Server is installed using the SMP/E RECEIVE, APPLY, and ACCEPT commands. The SMP/E dialogs may be used to accomplish the SMP/E installation steps.

The process is described in the CICS TS 3.1 Program Directory. It is in line with IBM Corporate Standards, and may be familiar to those who have installed other z/OS products.

The traditional method, DFHISTAR, of installing CICS Transaction Server is still available.

6.4 New SIT parameters for CICS TS 3.1

The new SIT parameters for CICS TS 3.1 are listed in Figure 6-2.

Systems Initialization Table: New parameters

- **CLINTCP**
  - Default client code page when DFHCNV CLINTCP=SYSDEF
- **CRLPROFILE (PK04622)**
  - Name of a profile in the RACF LDAPBIND class
- **LOCALCCSID**
  - Default CCSID for local region
- **MAXSSLTCBS**
  - Maximum number of S8 TCBs for use with SSL
- **MAXXPTCBS**
  - Maximum number of X8 and X9 TCBs for use by XPLINK programs
- **SRVERCP**
  - Default server code page when DFHCNV SRVERCP=SYSDEF
- **SSLCACHE**
  - Specifies scope of SSL caching

Figure 6-2 SIT new parms

The default values for these parameters are designed to have minimal impact when you are migrating from an earlier release of CICS.

- CLINTCP={437|codepage} specifies the default client code page to be used by the DFHCNV data conversion table but only if the CLINTCP parameter in the DFHCNV macro is set to SYSDEF.
- CRLPROFILE is the 246-character name of a profile in the RACF® LDAPBIND class that contains bind information about an LDAP server that will be used by CICS SSL support to obtain certificate revocation list information.
- LOCALCCSID={037|CCSID} specifies the default CCSID for the local region. The CCSID is a value of up to eight characters. If the CCSID value is not specified, the default LOCALCCSID is set to 037.

- MAXSSLTCBS={8|number} specifies the maximum number of S8 TCBs that can run in the SSL pool. The default is 8, but you can specify up to 1024 TCBs.
- MAXXPTCBS={5|number} specifies the maximum number, in the range 1 through 999, of open X8 and X9 TCBs that can exist concurrently in the CICS region.
- SRVERCP={037|codepage} specifies the default server code page to be used by the DFHCNV data conversion table, but only if the SRVERCP parameter in the DFHCNV macro is set to SYSDEF.
- SSLCACHE={CICS| SYSPLEX} specifies whether SSL is to use the local or sysplex caching of session IDs.

6.5 Systems initialization table: changed parameters

ENCRYPTION specifies the cipher suites that CICS uses for secure TCP/IP connections. When a secure connection is established between a pair of processes, the most secure cipher suite supported by both is used.

- Use ENCRYPTION=STRONG when you can tolerate the overhead of using high encryption if the other system requires it.
- Use ENCRYPTION=WEAK when you want to use encryption up to 40 bits in length.
- Use ENCRYPTION=MEDIUM when you want to use encryption up to 56 bits in length.

For compatibility with previous releases, ENCRYPTION=NORMAL is accepted as an equivalent to ENCRYPTION=MEDIUM. CICS can use only the cipher suites, which are supported by the underlying z/OS or OS/390 operating system.

FORCEQR specifies whether you want CICS to force all CICSAPI user application programs that are specified as threadsafe to run under the CICS QR TCB, as though they were specified as quasi-reentrant programs. This parameter applies to all application programs that are restricted to the current CICS programming interfaces (that is, those that specify API(CICSAPI)), and does not apply to any of the following:

- Java programs that are run in a JVM™
- C/C++ programs using XPLINK
- OPENAPI programs

None of these can run on the QR TCB.

6.6 Systems initialization table: obsolete parameters

The obsolete parameters are:

- MAXHPTCBS

  Runtime support for Java program objects and hot-pooling (HPJ) has been removed. The system initialization parameter MAXHPTCBS is not required, and is removed. The open TCB mode H8, which was used for hot-pooling Java program objects and was controlled by MAXHPTCBS, no longer exists.

- SSLTCBS

  This parameter is now obsolete and is only kept for compatibility. If it is specified, it is rejected with a message and MAXSSLTCBS is assumed.

- TCAM

  This parameter is now obsolete and is only kept for compatibility. If it is specified, it is rejected with a message and TCAM=NO is assumed.

6.7 CICS-supplied transactions

This section looks at the new and changed CICS-supplied transactions in CICS Transaction Server 3.1.

6.7.1 Changes to CWXN (Web attach transaction)

There are several changes to the processing carried out by the CICS-supplied transaction CWXN, the Web attach transaction. The most significant of these are:

- If a matching URIMAP definition is found for an HTTP request, CWXN now invokes the analyzer program only if instructed to do so by the URIMAP definition.
- Where the HTTP version of the request is HTTP/1.1, CWXN carries out some of the responsibilities of an HTTP server by performing basic acceptance checks on the request. In response to these checks, CWXN might take action to return a response to the request without involving a user-written application program.
- CWXN pre-processes chunked and pipelined messages received from a Web client so that user-written applications do not have to perform this processing.
- Chunked messages are single messages split up and sent as a series of smaller messages (chunks). CWXN receives and assembles the chunks of the message to create a single HTTP request. CWXN checks that the

message is complete before passing it to the user application. The user application can then process the request like any other HTTP request.

- Pipelined messages are multiple messages sent in sequence, where the sender does not wait for a response after each message sent. A server must respond to these messages in the order in which they are received. To ensure this, CWXN holds pipelined requests and releases them one at a time to the user application. The user application must send a response to the first request before receiving the next request from CWXN.
- Persistent connections are now the default behavior. The connection is only closed if the Web client requests closure, or if the timeout period is reached, or if the Web client is an HTTP/1.0 client that does not send a Keep-Alive header.

6.7.2 New CICS-supplied transactions

CCRL, the certificate revocation lists transaction, is used to create and update the certificate revocation lists (CRLs) that are stored in an LDAP server. You only need to use CCRL if you are implementing SSL in your CICS regions and want each connection checked for a revoked certificate during the SSL handshake:

- CPIH, internal alias transaction for inbound Web Services over http
- CPIR, Internal alias transaction for inbound Web Services using WMQ

In CICS Transaction Server for z/OS Version 3 Release 1 processing for HTTP requests and processing for non-HTTP requests are kept separate. This ensures that CICS can perform basic acceptance checks on HTTP requests and responses, and that non-HTTP requests are not subjected to these checks. Processing for non-HTTP requests must now be carried out under the user-defined (USER) protocol, which is specified on the TCPIPSERVICE definition for the port that receives the requests.

The new CICS-supplied transaction CWXU, the CICS Web user-defined protocol attach transaction, is the default when the protocol is defined as USER. CWXU executes the CICS program DFHWBXN.

6.7.3 New CEMT command options

- INQUIRE
  - HOST
  - PIPELINE
  - URIMAP
  - WEBSERVICE
- SET
  - HOST

  - PIPELINE
  - URIMAP
  - WEBSERVICE
- DISCARD
  - PIPELINE
  - URIMAP
  - WEBSERVICE
- PERFORM
  - PIPELINE SCAN

CEMT supports the standard inquire, set, and discard commands for the new pipeline, Urimap, and Web Service resources.

A perform pipeline command initiates a scan of the Web Service binding directory that is specified in the WSBIND attribute of a pipeline definition.

6.7.4 Changed CEMT command options

The changed CEMT command options are:

- INQUIRE

  Dispatcher, Doctemplate, Program, System, Tcpip, Tcpipservice, Workrequest

- SET

  Dispatcher, Doctemplate, Program, System, Tcpipservice, Workrequest

- PERFORM

  Statistics

- INQUIRE SYSTEM
  - CICSTSLEVEL returns 030100.
  - RELEASE returns 0640.

In terms of obsolete options, inquire/set commands for dispatcher and program have any parameters relating to Java hot-pooling and HP TCBs removed.

The dispatcher command now has new parameters relating to XP TCBS for XPLINK.

For programs, a new APIST keyword shows whether the program is defined as OPENAPI or CICSAPI. The existing RUNTIME keyword has a new value of XPLINK.

For DOCTEMPLATES, the HFSFILE keyword returns the full-qualified name of the HFS file where the template resides. TCP/IP commands support the new CRLSERVER and SSLCACHE keywords.

A new MAXDATALEN parameter for TCPIPSERVICE specifies the maximum length of data that may be received by CICS as an HTTP server as a result of upgrading our support to HTTP 1.1

Statistics now support the new pipeline and Web Service resources.

- CETR

  Activate trace for pipeline manager domain (PI).

- New CICS RACF category 1 transactions

  The CICS region user ID must be authorized for these transactions:

CPIH	CICS SOAP HTTP inbound router transaction
CPIL	CICS SOAP MQ inbound listener transaction
CPIQ	CICS SOAP MQ inbound router transaction
CPIR	CICS pipeline resolution transaction
CRTP	CICS persistent session sign-on
CWXU	CICS Web attach - user-defined protocol

CETR allows tracing to be activated for the new pipeline manager domain.

There are some new CICS internal system transactions added to the list of category one transactions. These are the transactions that need to be defined to RACF, and to which the CICS region user ID must be authorized, to enable CICS to initialize successfully when you are running CICS with security enabled (SEC=YES).

6.8 Resource definition

This section looks at the new, changed, and removed resource definitions in CICS Transaction Server 3.1.

6.8.1 CICS System Definition (CSD)

- Define new CSD.
- REPRO existing CSD to new data set.
- Run DFHCSDUP UPGRADE.
  - Use the DFHCSDUP SCAN command to check for user changes.
  - Review the CEE group.
- Share the CSD.

CICS TS 3.1 CSD can be shared with prior releases. There is no requirement for a DFHCOMPx group to share with CICS TS 2.3.

Run the DFHCSDUP utility program, specifying the UPGRADE command, to upgrade the CICS-supplied definitions in your CSD to the latest CICS TS level. You can create a new CSD using the DFHCSDUP INITIALIZE command.

Upgrading other IBM-supplied resource definitions

If you have resource definitions in your CSD that support other IBM products, you may need to upgrade these also. For example, if your Language Environment resource definitions are not at the z/OS Version 1 Release 4 level, we recommend that you delete and replace the CSD group containing these.

You can find the Language Environment resource definitions in the SCEESAMP library in member CEECCSD.

6.8.2 Obsolete IBM-supplied resource groups

In this section we discuss obsolete IBM-supplied resource groups.

DFH$JAVA

IBM-supplied sample application program group DFH$JAVA is removed. This group contained the resource definitions needed for the sample applications for Java support using VisualAge for Java, Enterprise Edition for OS/390. The same sample applications are defined for use with a JVM by the DFH$JVM group.

DFHAUGRP

IBM-supplied group DFHAUGRP is removed. This group contained the resource definitions for the CICS transaction affinities utility.

DFH$AFFY

IBM-supplied sample group DFH$AFFY is removed. This group contained sample resource definitions for the CICS transaction affinities utility that you could modify to suit your requirements.

Obsolete definition groups have been removed from the CICS-supplied default start-up group list.

DFHLIST

If you use customized startup group lists, you must remove any obsolete definition groups from them.

6.8.3 Changes to resource definition

In this section we discuss changes to resource definitions.

CORBASERVER CIPHERS Keyword added

This specifies a string of up to 56 hexadecimal digits that is interpreted as a list of up to 28 2-digit cipher suite codes. The attribute value is automatically populated with the list of acceptable codes, depending on what level of encryption has been specified by the ENCRYPTION system initialization parameter. For ENCRYPTION=WEAK, the default value is 03060102. For ENCRYPTION=MEDIUM, the default value is 0903060102. For ENCRYPTION=STRONG, the default value is 0504352F0A0903060102.

DOCTEMPLATE HFSFILE attribute

This allows the template to reside on an HFS file.

PROGRAM API attribute

This specifies what application programming interfaces the program will use.

CICSAPI means that the program uses CICS application programming interfaces only. CICS determines whether the program runs on the quasi-reentrant (QR) TCB or on another TCB. This depends upon the value of the CONCURRENCY attribute in the PROGRAM resource definition. If the program is defined as threadsafe it may run on whichever TCB, in use by CICS at the time, is determined suitable.

OPENAPI means that the program is not restricted to the CICS application program interfaces. Programs defined with API(OPENAPI) run almost independently of the QR TCB.

Such programs run on an L8 or L9 open TCB, depending upon their EXECKEY value. Because OPENAPI programs can potentially use non-CICS APIs, the key of the TCB is important and must match the execution key. This is unlike API(CICSAPI) threadsafe programs that can execute in CICS key or user key irrespective of the TCB key. CICS services are implemented irrespective of the key of the TCB they are running under, unlike MVS services, which care about the TCB key. If, while executing a CICS command, CICS requires a switch to the QR TCB, it returns to the open TCB before handing control back to the application program. OPENAPI requires the program to be coded to threadsafe standards and defined with CONCURRENCY(THREADSAFE).

TCPIPSERVICE CIPHERS keyword added

This specifies a string of up to 56 hexadecimal digits that is interpreted as a list of up to 28 2-digit cipher suite codes. The attribute value is automatically populated with the list of acceptable codes, depending on what level of encryption has been specified by the ENCRYPTION system initialization parameter. For ENCRYPTION=WEAK, the default value is 03060102. For ENCRYPTION=MEDIUM, the default value is 0903060102. For ENCRYPTION=STRONG, the default value is 0504352F0A0903060102.

TCPIPSERVICE MAXDATALEN keyword added

This defines the maximum length of data that can be received by CICS as an HTTP server on the HTTP protocol or the USER protocol. The default value is 32 K. The minimum is 3 K and the maximum is 524288 K. To increase security for CICS Web support, specify this option on every TCPIPSERVICE definition for the HTTP protocol. It helps to guard against denial of service attacks involving the transmission of large amounts of data.

TCPIPSERVICE USER value added to protocol keyword

Processing for all non-HTTP requests must now be carried out under the USER protocol. No parsing is carried out for messages received on the USER protocol, and requests that have been divided up for transmission across the network are not automatically assembled. This is the same behavior as when handling non-HTTP messages in earlier CICS releases.

TCPIPSERVICE change of recommendation for SOCKETCLOSE(0)

In previous releases the recommendation was that if you are using the TCPIPSERVICE for CICS Web Support and are processing only standard HTTP requests, SOCKETCLOSE(0) should be specified to avoid unnecessary CWXN transactions remaining in the system.

However, in CICS TS 3.1, the socket can remain open without involving a CWXN transaction taking up a max task slot. Also, with the upgrade to HTTP 1.1, the recommendation is that if you are using a TCPIPSERVICE for CICS Web Support with the HTTP protocol, SOCKETCLOSE(0) should not be specified. A zero setting for SOCKETCLOSE means that CICS closes the connection immediately after receiving data from the Web client, unless further data is waiting. This means that persistent connections cannot be maintained

6.8.4 New definitions

In this section we discuss new definitions.

PIPELINE definition

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. The information about the processing nodes is supplied indirectly—the PIPELINE specifies the name of an HFS file that contains an XML description of the nodes and their configuration. An inbound Web Service request (that is, a request by which a client invokes a Web Service in CICS) is associated with a PIPELINE resource by the URIMAP resource.

UIRMAP definition

URIMAP definitions are resource definitions that match the URIs of HTTP or Web Service requests and provide information about how to process the requests. URIMAP definitions are used to provide three different Web-related facilities in CICS:

- Requests from a Web client to CICS as an HTTP server
- Requests to a server from CICS as an HTTP client
- Web Service requests

WEBSERVICE definition

A WEBSERVICE resource defines aspects of the runtime environment for a CICS application program deployed in a Web Services setting, where the mapping between application data structure and SOAP messages has been generated using the CICS Web Services assistant. Although CICS provides the usual resource definition mechanisms for WEBSERVICE resources, they are typically installed dynamically, using the output produced by the assistant. The aspects of the runtime environment that are defined by the WEBSERVICE resource are:

- A pipeline
- A Web Service binding file
- A Web Service description

See *Implementing CICS*, SG24-7206.

6.9 Application Programming Interface

This section looks at the new and changed API in CICS Transaction Server 3.1.

6.9.1 EXEC CICS

In CICS TS 1.3 and earlier, CICS recognizes the sign-on immediately, and establishes the specified user's security and operating attributes for the terminal. The transaction (and any associated task-related user exits, function shipping, or distributed transaction processing) may have invoked other resource managers (for example, IMS, DB2, or VSAM). It is unpredictable whether these other RMs recognize the sign-on before the transaction terminates, and thus you can only be sure that the new user attributes apply for all resource managers invoked by subsequent transactions at the terminal. Hence, since CICS TS V2, the behavior of EXEC CICS ISGNON and SIGNOFF changed in that SIGNON and SIGNOFF commands do not affect the current transaction issuing the command.

- SIGNON/SIGNOFF
  - Since CICS TS V2 operation is terminal-related only
  - Executing transaction security and user ID set at task attach time

    XSNEX Global User Exit (migration aid retained for compatibility)
- VERIFY PASSWORD

  CICS now enforces the revoked status of a user ID or a user's group connection.

If you have applications that cannot tolerate the change in the SIGNON and SIGNOFF process, CICS provides a global user exit point (XSNEX) and sample global user exit program that will enable CICS to handle EXEC CICS SIGNON and SIGNOFF, as in CICS TS 1.3 and earlier releases. Note that XSNEX is a migration aid only, and you should consider removing all application dependency on the old behavior. CICS TS 3.1 continues to ship this migration aid.

When the command EXEC CICS VERIFY PASSWORD is issued, CICS now enforces the revoked status of a user ID or a user's group connection. For example, if a user has tried to log on too many times, the ID is revoked and the user cannot access the system or resources

6.9.2 High performance Java (HPJ) programs

Non-IIOP applications must be converted to JVM programs.

Run-time support for Java program objects and for hot-pooling (HPJ) is withdrawn in CICS TS 3.1. Any Java programs that you had processed using the VisualAge for Java Enterprise Edition for OS/390 bytecode binder (hpj) to run as Java program objects in CICS must be migrated to run in a Java Virtual Machine (JVM).

6.9.3 C/C++ programs

CICS provides support for C and C++ programs compiled with the XPLINK option by using the multiple TCB feature in the CICS Open Transaction Environment (OTE) technology. X8 and X9 mode TCBs are defined to support XPLink tasks in CICS key and USER key, respectively. Each instance of an XPLink program uses one X8 or X9 TCB.

- Activated via the XPLINK compiler option.
- New CICS-supplied procedures for translate, compile, and linkedit.
- Programs run on X8 or X9 TCBs using MVS LE services.
- Programs must be threadsafe to use XPLINK and be defined as threadsafe.

To use XPLink, your C or C++ application code must be reentrant and threadsafe. The same code instance can be executing on more than one MVS TCB and, without threadsafe mechanisms to protect shared resources, the execution behavior of application code is unpredictable. This cannot be too strongly emphasized.

CICS provides procedures DFHYITFL for C programs, and DFHYITGL for C++ Programs wanting to use XPLINK.

6.10 Systems Programming Interface

A new SPI command, EXTRACT STATISTICS, handles statistics for URIMAP, PIPELINE, and WEBSERVICE resources. Use the EXTRACT STATISTICS command to retrieve the current statistics for a single resource or global statistics for a class of resources. The EXTRACT STATISTICS command performs a function equivalent to COLLECT STATISTICS for the URIMAP, PIPELINE, and WEBSERVICE resources. To collect statistics for other resources use the existing COLLECT STATISTICS command. The syntax of the EXTRACT STATISTICS differs from that of COLLECT STATISTICS.

All CICS SPI commands are restricted in the number of distinct options they can support. As new resources have been added to CICS over time, the limit has been reached for the COLLECT STATISTICS command, and it is not possible to accommodate the new URIMAP, PIPELINE, and WEBSERVICE resources on the existing command.

The EXTRACT STATISTICS command uses the RESTYPE option, with a CVDA, to specify a CICS resource. As a result, there is no limit on the number of resources that the command can potentially support, although in this release, only the three new resources are supported.

6.11 Global user exits

We *highly* recommend that *all* global user exits be analyzed to ensure that they are threadsafe and that their program definitions have been changed to specify CONCURRENCY(THREADSAFE).

All user programs defined by a program resource definition have a concurrency attribute, which can be either QUASIRENT or THREADSAFE. By default, global user programs are defined as quasi-reentrant, which means that they are given control on the CICS QR TCB. If the task under which the global user exit is invoked is executing on an open TCB, and the exit program is defined as quasi-reentrant, CICS switches back to the QR TCB for the execution of the exit program.

To avoid unnecessary TCB switching, we strongly recommend that you make sure that your global user programs conform to threadsafe programming standards. When you are satisfied that your exit programs are threadsafe, ensure that they are defined as CONCURRENCY(THREADSAFE). This is particularly important for exits that are invoked by tasks that are using the CICS DB2 interface and running under an L8 TCB.

Exit parameter UEPGIND passed to global user exits includes reference to the mode of the TCB the exit is running on. With the new types of open TCB introduced, exits can now run on these new types of TCB if they are threadsafe and defined to CICS as such.

For more information about *Threadsafe Considerations for CICS*, SG24-6351.

6.11.1 New global user exits

New global user exits are:

- XWBOPEN

  This is called during WEB OPEN, before the session is established. It can be used to bar access to a whole host.

- XWBSNDO

  This is called during WEB SEND or WEB CONVERSE. It enables systems administrators to specify a security policy for HTTP requests by CICS.

There are two new global user exits for CICS as an HTTP client: XWBOPEN in the WEB OPEN command and XWBSNDO in the WEB SEND command. (Note that XWBSNDO only applies when the WEB SEND command is used for CICS as an HTTP client, and not for CICS as an HTTP server.)

XWBOPEN enables systems administrators to specify proxy servers that should be used for HTTP requests by CICS as an HTTP client, and to apply a security policy to the host name specified for those requests. XWBOPEN is called during processing of an EXEC CICS WEB OPEN command, which is used by an application program to open a connection with a server.

XWBSNDO enables systems administrators to specify a security policy for HTTP requests by CICS as an HTTP client. XWBSNDO is called during processing of an EXEC CICS WEB SEND or EXEC CICS WEB CONVERSE command. The host name and path information are passed to the exit, and a security policy can be applied to either or both of these components.

6.11.2 Changed global user exits

Global user exit programs cannot access containers created by application programs. They can, however, create their own channels and pass them to programs that they call.

- Parameter list changes
  - Existence bits with channel name passed to exits:
    - XICEREQ, XICEREQC
    - XPCREQ, XPCEREQC
    - XPCTA, XPCFTCH, XPCHAIR, XPCABND
  - Exits may not access contents of channels.
- XPlink programs

  XPCTA does not allow a resume address. The new flag is PCUE_NO_RESUME in PCUE_CONTROL_BITS.
- XPCFTCH does not allow a modified entry address.
  - The new flag is PCUE_NO_MODIFY in PCUE_CONTROL_BITS.
  - The alternative is CEEBXITA.

When the exit XPCTA is invoked from a C or C++ program that was compiled with the XPLINK option, a flag is set indicating that a resume address, if specified by the exit, will be ignored. This is because XPLINK runs with MVS LE, which has it own recovery procedures, which percolates to CICS. By the time CICS recovery gets control, the program environment has gone. When the exit XPCFTCH is invoked from a C or C++ program that was compiled with the XPLINK option, a flag is set indicating that any modified entry point address, if specified by the exit, will be ignored. It is not supported because XPLINK uses MVS LE with CEEPIPI pre initialized interface and PIPI will reject the signature of any assembler program.

6.11.3 Removed global user exits

Here we discuss remove global user exits.

XTCTIN terminal control program

This exit was invoked on TCAM input events. It is no longer called because CICS TS 3.1 does not support the TCAM/ACB interface, and it only supports the TCAM/DCB interface indirectly.

XTCTOUT terminal control program

This exit was invoked on TCAM output events. It is no longer called because CICS TS 3.1 does not support the TCAM/ACB interface, and it only supports the TCAM/DCB interface indirectly.

6.12 User replaceable modules

The user replaceable modules are:

- Removed URMs
  - DFHAPH8O (HPJ Hotpooling)
  - DFHJHPAT (HPJ)
- New URMs

  DFHAPXPO (XPLINK)
- Changed URMs

  User-replaceable programs cannot access containers created by application code.
- DFHCNV

  Added SYSDEF operand to TYPE=INITIAL

The user-replaceable programs DFHAPH8O and DFHJHPAT are removed.

DFHAPH8O was provided to allow you to alter the default Language Environment runtime options for the Language Environment enclave where a Java program object was to be run.

DFHJHPAT was optional and could be used for your own purposes, such as tracing. It was called before a Java program object was invoked.

The new user replaceable module DFHAPXPO allows you to alter the default Language Environment runtime options for the Language Environment enclave where an XPLINK program is to run.

The new operand SYSDEF has been added to the TYPE=INITIAL and TYPE=ENTRY macro parameters CLINTCP and SRVERCP. These macros define the user-replaceable data conversion table DFHCNV. The DFHCNV TYPE=INITIAL macro defines the beginning of the conversion table. It gives a list of valid code pages. The DFHCNV TYPE=ENTRY macro specifies a name and type to uniquely identify a data resource. There must be one for each resource for which conversion is required.

6.13 Monitoring and statistics

Some performance data fields are added to performance class data records. The result of all these additions is that the record length of performance class data records has increased significantly, with the maximum record length now up to 1836 bytes per record.

- Performance class data

  Record size increases to 1836 bytes. Reduce using the INCLUDE and EXCLUDE options on the MCT.

- Changes to statistics record

  New and changed DSECTs:

  – DFHCHNLContainer usage
  – DFHPROGProgram statistics
  – DFHSOCKTCP/IP statistics
  – DFHTASKTask statistics
  – DFHWEBBWeb support statistics

SMF data sets can quickly fill with unwanted data.You can reduce the amount of data written to SMF by using a monitoring control table (MCT) to selectively include or exclude specified fields.

There are changes to CICS statistics records. These are usually because of new domains, or they are a result of enhancements to CICS. As a result, a number of statistics DSECTs have new or changed fields. The changed DSECTs are:

- DFHDSGDS dispatcher global statistics
- DFHPIPDS pipeline resource statistics
- DFHPIWDS Web Service resource statistics
- DFHWBSDS urimap global statistics
- DFHMNTDS transaction performance monitoring resource statistics
- DFHWBRDS urimap resource statistics
- DFHSORDS TCP/IP resource statistics
- XPLINK CPU time will be included in CICS 110 records:
  – X8 CPU, dispatch and delay times

  - X9 CPU, Dispatch and delay times
- OPENAPI
  - New L9 CPU, dispatch and delay times as well as existing L8 times
- SP and S8 CPU time
  - SP time will be included in miscellaneous.
  - S8 CPU, dispatch and dispatch delay times.

The CICS 110 record includes new fields to record the CPU time consumed on X8 and X9 TCBs used by XPLINK programs. These contribute to the overall CPU total for the transaction.

Similarly for openapi programs, L9 TCBs contribute to the overall time, as well as the existing L8 TCBs. For SSL a new SP TCB and S8 TCBS CPU time is captured.

6.14 CICS SOAP feature

If you use the SOAP for CICS feature, you can continue to do so. The feature continues to be fully supported in CICS TS 3.1 independently of Web Services in CICS.

The SOAP for CICS feature can interoperate with the support for Web Services in CICS TS 3.1. The feature can be the service requester or the service provider. This is not orderable with CICS TS 3.1. The existing Version 2 feature may be used with 3.1. The intent is to aid migration. This is not intended as a substitute for Web Services.

6.15 CICSPlex Systems Manager

CICSPlex migrations similar to previous releases, CAS, CMAS, and MAS agent code must all be at 3.1 level,.

The WUI server and its connected CMAS must be at the 3.1 level.

Migrate the contents of the WUI Server repository:

1. At the prior CICS level, export view set and menu definitions.
2. Create a new WUI server repository for 3.1.
3. Start the 3.1 WUI server.
4. Import the new starter set definitions.
5. Review the new view formats with your changes:
   a. Import the previous release view set and menu definitions.
   b. Specify SKIP in the Duplicate Names field of the COVC panel.

Important: Maintenance point CMAS must be upgraded first.

You must migrate your CICSPlex SM CMAS to CICS TS 3.1 at the same time at which you migrate the CICS system on which it runs. This is because since CICS Transaction Server for z/OS Version 2 Release 3 a CICSPlex SM CMAS will run only in a CICS system at the same release level.

Both the Web User Interface server and the CMAS that it connects to must be at the highest level of CICSPlex SM within the CICSplex. This means that both must be at the same level as the maintenance point CMAS.

Before you migrate a Web User Interface server, you must migrate the CMAS that it connects to. You must migrate the Web User Interface server before you migrate any other MASs. If the CMAS that the Web User Interface server connects to is not the maintenance point CMAS, you must migrate the maintenance point CMAS at the same time.

As the CICS system that acts as your Web User Interface server is a local MAS, all the considerations that apply to a local MAS also apply to a Web User Interface server.

See *Using the Web User Interface in CICSPlex SM*, SG24-6793.

6.16 Language Environment

Runtime support for OS/VS COBOL programs is withdrawn. OS/VS COBOL programs, which had runtime support in CICS Transaction Server for z/OS Version 2, cannot run under CICS TS 3.1.

OS/VS COBOL programs must be upgraded to Language Environment conforming COBOL, and recompiled against a level of COBOL compiler supported by CICS. Enterprise COBOL for z/OS and OS/390 Version 3 is the recommended compiler.

You can now produce assembler MAIN programs that are Language Environment conforming. Until now, the only way to use Language Environment conforming assembler programs within CICS was to use a call from a COBOL, PLI, or C Language Environment conforming program and linkedit the assembler program with the high-level language (HLL) program. This made the assembler program a Language Environment subroutine. It had to have MAIN=NO on CEEENTRY. The user had to specify NOPROLOG and NOEPILOG and then code the CEEENTRY and CEETERM calls separately. A CICS PROGRAM resource could not be defined as both ASM and LE370.

CICS now supports the coding of Language Environment conforming assembler MAIN programs. A new translator option LEASM causes the Language Environment function to be used to set up the program's environment. Such programs must be linkedited with stub DFHELII rather than DFHEAI.

This support also enables the use of the Debugger for Assembler programs.

- CICS TS 2.3
  - CICS interfaces for the VS COBOL II, OS PL/I, and C/370™ runtimes removed
  - Need runtime libraries distributed with LE to execute current load modules
- CICS TS 3.1
  - CICS interfaces for the OS/VS COBOL are removed.
  - CICS will terminate any OS/VS COBOL program with an ALIK abend.
  - Support for LE conforming Assembler main programs is added.

To enable Language Environment support to be installed correctly by CICS:

1. Specify enough storage for the ERDSA to run CICS and the Language Environment together. They need a minimum of 3500 KB. To this minimum, add an amount of storage sufficient for your own requirements.
2. Ensure that the CICS-Language Environment interface module, CEECCICS, and the Language Environment modules CEEPIPI and CEECTCB are installed in an APF-authorized library defined in the STEPLIB concatenation in the CICS startup JCL. You can do this by including the Language Environment SCEERUN library in an APF-authorized library in the STEPLIB concatenation of your CICS startup job (for example, in the CICSTS31.CICS.SDFHAUTH library) or in an APF-authorized library in the MVS LNKLSTnn concatenation.
3. Ensure that the program resource definitions for the Language Environment language interface modules have been added to the CICS CSD. These definitions are in the CEE group. The CEE group is added automatically to the CSD and to the grouplist DFHLIST during CICS installation, as part of the DFHCOMDS job. The definitions are also supplied as DEFINE statements in the CEECCSD member of the SCEESAMP library. You can add the CEE group to any CICS startup group list named in the GRPLIST system initialization parameter.
4. Define the Language Environment transient data destinations, CESE, and CESO (DD names CEEMSG and CEEOUT). The CICS-supplied resource definition group, in the CSD, DFHDCTG, contains entries for CESE and CESO. For information about the attributes needed for Language

Environment transient data destinations, see *IBM Language Environment for MVS & VM Programming Guide*, SC26-4818.

5. Define the Language Environment runtime libraries on the CICS STEPLIB and DFHRPL DD statements as follows: Add the SCEERUN library, which contains CEECCICS and CEECTCB, and the SCEERUN2 library, which contains support that is required for the IBM Java Virtual Machine (JVM) and also support for other programming languages, to STEPLIB or to a library in the MVS LNKLSTnn concatenation. Both the libraries, SCEERUN and SCEERUN2, must be APF-authorized. Add the SCEECICS, SCEERUN2, and SCEERUN libraries to DFHRPL, with SCEECICS and SCEERUN2 concatenated before SCEERUN.

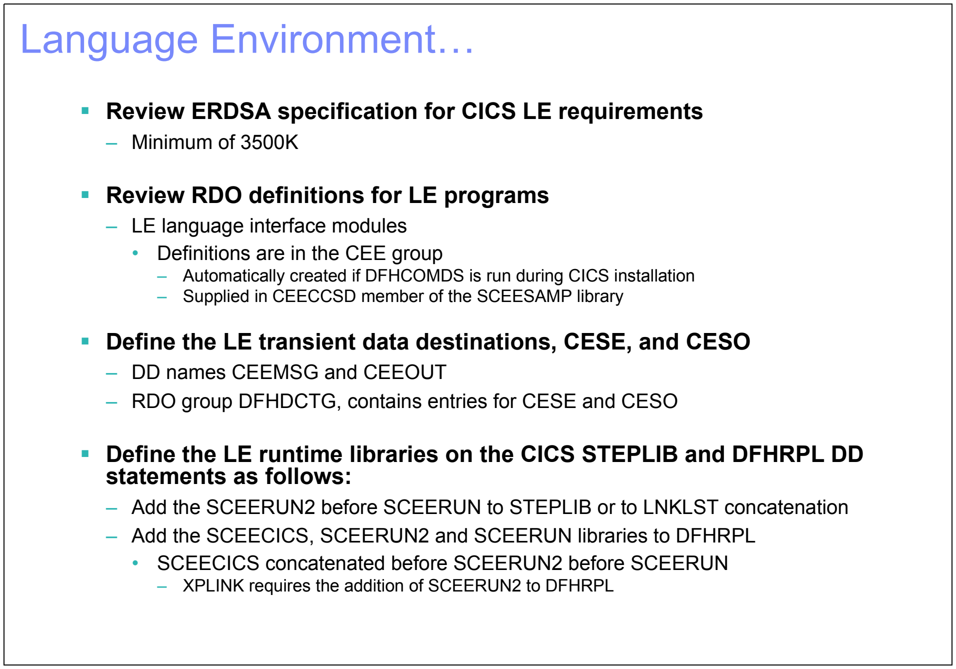


Figure 6-3 LE support

6.17 Open transaction environment

Applications wanting to use XPLINK or OPENAPI support must be coded to threadsafe standards. Applications have to worry about concurrent access to their resources such as shared storage. Unless an application requires to overwrite itself (in which case it has to provide serialization of such code; it is a type of shared storage) then ensure applications are read-only. The CICS read-only DSA can be used to ensure this.

CICS provides a load module scanner utility with a sample table called DFHEIDTH that looks for applications that issue EXEC CICS ADDRESS CWA, EXEC CICS GETMAIN SHARED, or EXEC CICS EXTRACT EXIT. All these commands give access to shared storage and hence have the potential for the application logic not being threadsafe if the storage is not subsequently updated in a threadsafe way. Applications can use ENQUEUE and DEQUEUE to serialize updates to shared storage. In assembler applications compare and swap instructions can be used.

- OPENAPI and C/C++ XPLINK applications have to be THREADSAFE.
  - CICS will ensure threadsafe access to its managed resources: VSAM files, TS, TD, DLI databases, and DB2 tables.
  - Applications have to ensure threadsafe access to their resources: shared storage (for example, CWA, GETMAIN SHARED).
- Ensure that applications are read-only.
  - Put them in the CICS read-only DSA (linkedit with RENT).
  - Set the SIT option RENTPGM=PROTECT.
- Serialize access to shared resources.
  - CWA or shared storage. Use the load module scanner to look for use of global storage.
  - Use services such as EXEC CICS ENQUEUE and DEQUEUE.

For more information about OTE see *Threadsafe Considerations for CICS*, SG24-6351.

6.18 Function removal

Support for the CICS Connector for CICS TS, introduced in CICS TS for z/OS Version 2.1, is withdrawn.

A CICS connector is a software component that allows a Java client application to invoke a CICS application. CICS TS for z/OS Version 2.3 introduced a new CICS connector, the CCI Connector for CICS TS, that performs a similar role to the CICS Connector for CICS TS—that is, it enables a Java program or enterprise bean running on CICS Transaction Server for z/OS to link to a CICS server program. However, while the old CICS Connector for CICS TS implemented the IBM-proprietary Common Connector Framework (CCF) interface, the new CCI Connector for CICS TS implements the industry-standard Common Client Interface (CCI) defined by the J2EE™ Connector Architecture Specification Version 1.0.

The ECI Base Classes (ECIREQUEST, which were introduced for compatibility with the CICS Transaction Gateway) are not included in CICS TS 3.1. The recommended replacement is the COMMON CLIENT INTERFACE CONNECTOR FOR CICS TS (CCI Connector for CICS TS), introduced in CICS TS V2.3, when it was announced that ECIREQUEST would be removed.

CICS TS 3.1 does not include the detector and reporter components previously provided as part of the CICS Transaction Affinities utility. These components are now incorporated in IBM CICS Interdependency Analyzer for z/OS V1.3, announced in August 2004, which has the capability of analyzing both interdependencies and affinities. The load library scanner component of the CICS Transaction Affinities utility remains in CICS TS 3.1 and can produce reports on application programs that have potential affinities.

Support for defining terminals using the 1-byte console ID is withdrawn. The CONSOLE attribute on the TERMINAL resource definition is obsolete, but is supported to provide compatibility with earlier releases of CICS. You can define terminals using the CONSNAME(name) attribute on the TERMINAL resource definition.

If you have a network of terminals connected by the ACB interface of TCAM to a back-level CICS TOR, you will not be able (as you were in previous CICS releases) to route transactions from them to a CICS TS for z/OS Version 3.1 AOR. You must migrate your connections to use TCAM/DCB or (preferably) ACF/VTAM, or route to a previous version of CICS. (All terminals that support TCAM/ACB also support ACF/VTAM.)

If you have a network of terminals connected by the DCB interface of TCAM to, for example, a CICS TS 2.3 TOR, you will not be able to migrate the TOR to CICS TS for z/OS Version 3.1. To do so, you must migrate your connections to use ACF/VTAM.

If you have a network of terminals connected by the DCB interface of TCAM to a back-level CICS TOR, you will (as in previous CICS releases) be able to route transactions from them to a CICS TS for z/OS Version 3.1 AOR. However, we recommend that you migrate your connections to use ACF/VTAM.

If you have a network of BTAM terminals connected to a back-level CICS terminal-owning region (TOR), you will not be able (as you were in previous CICS releases) to route transactions from them to a CICS TS for z/OS Version 3.1 application-owning region (AOR). You must either upgrade your terminals or route to a previous version of CICS.

6.19 Extra considerations when migrating from CICS TS 2.2

There are a few additional considerations for the user migrating from CICS TS 2.2 to CICS TS 3.1. The additional changes fall into the areas of systems initialization table changes, new and changed CICS-supplied transactions, resource definition changes, new user exits, and changes to the CICS Java interface:

- Systems initialization table
- CICS-supplied transactions
- RDO
- User exits
- Java

6.19.1 Systems initialization table: new parameters

Figure 6-4 reviews the systems initialization table: new parameters.

Systems Initialization Table: New parameters

- **DEBUGTOOL**
  - Specifies if debugging profiles will be used to select programs
- **INFOCENTER**
  - Universal Resource Locator for the CICS Information Center
  - Used to provide help information for the web based application debugging profile manager
- **JVMCCPROFILE**
  - JVM profile for the master JVM that initializes the shared class cache
- **JVMCCSIZE**
  - Size of the shared class cache
- **JVMCCSTART**
  - Specifies how the shared class cache is to be started
    - Auto: at CICS initialization
    - Yes: at first JVM request
    - No: by CEMT PERFORM CLASSCACHE START
- **JVMPROFILEDIR**
  - Specifies the HFS directory that contains the CICS JVM profiles
- **JVMLEVEL0TRACE, JVMLEVEL1TRACE, JVMLEVEL2TRACE**
  - Specifies the default level for JVM level 0. 1 and 2 tracing, corresponds to CICS SJ trace levels 29-31
- **JVMUSERTRACE**
  - Specifies the default level for JVM user tracing, corresponds to CICS SJ trace level 32

Figure 6-4 SIT new parms

The default values for these parameters are designed to have minimal impact when you are migrating from an earlier release of CICS.

- DEBUGTOOL {NO|YES} specifies whether debugging profiles will be used to select programs that will run under the control of a debugging tool.
- INFOCENTER {infocenter_url} specifies the Universal Resource Locator (URL) of the root of the CICS Information Center directory structure.
- JVMCCPROFILE {DFHJVMCC|profile} specifies the JVM profile to be used for the master JVM that initializes the shared class cache.
- JVMCCSIZE {24M|number} specifies the size of the shared class cache on an initial or cold start of CICS.
- JVMCCSTART {AUTO|YES|NO} determines whether the shared class cache is started during CICS initialization and sets the status of autostart for the shared class cache.
- JVMLEVEL0TRACE, JVMLEVEL1TRACE, JVMLEVEL2TRACE, JVMUSERTRACE {option} specify the default options for the JVM trace levels.
- JVMPROFILEDIR {/usr/lpp/cicsts/cicsts23/JVMProfiles|directory} specifies the name of an HFS directory that contains the JVM profiles for CICS.

6.19.2 Systems initialization table: changed parameters

The parameters are:

- EDSALIM

  The default size is now 30 M.

- MAXJVMTCBS

  This specifies the maximum number of J8 and J9 TCBs.

  – The minimum value is now 1.
  – The master JVM (JM) does not count towards MAXJVMTCBS.

  MAXJVMTCBS {5|number} specifies the maximum number of open TCBs that CICS can create in the pool of J8-mode and J9-mode TCBs for use by Java programs that run in a JVM (the JVM pool). Within this limit, there are no constraints on how many of the TCBs in the JVM pool are J9 TCBs, and how many are J8 TCBs.

STNTR, STNTRxx, SPCTR, and SPCTRxx {level numbers}: The SJ component (JVM domain) now has trace levels 29–32, which are reserved to indicate the JVM trace levels 0, 1, and 2, plus a user-definable JVM trace level. We recommend that you use only the SPCTRSJ system initialization parameter to activate JVM tracing, so that it is only activated for special transactions. Selecting tracing levels 29, 30, 31, 32, or ALL for standard tracing for the JVM domain (SJ) component (using the STNTR or STNTRSJ system initialization parameters) is not recommended, because JVM trace can produce a large amount of output.

- New domain specifications
- The JVM domain (SJ) now has trace levels 29–30 corresponding to JVM trace levels 0, 1, 2, plus a user definable JVM trace level.
- The recommendation is to only use SPCTRSJ to activate JVM tracing due to the amount of output trace data.

6.19.3 CICS-supplied transactions

Figure 6-5 lists CICS-supplied transactions.

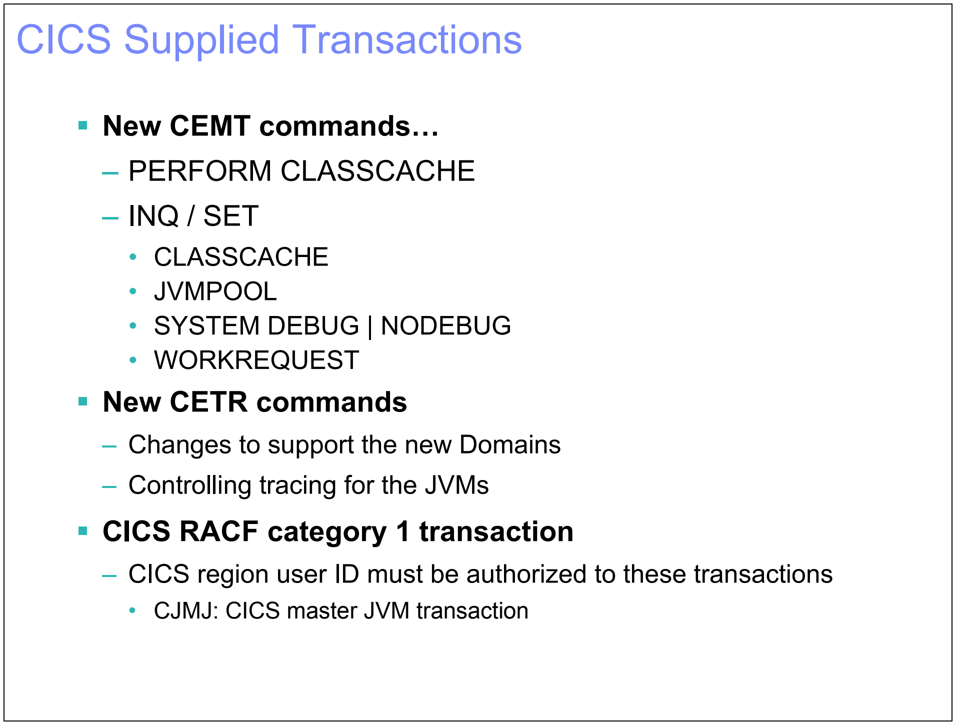


Figure 6-5 CICS-supplied transactions

There are changes to a few of the CICS-supplied transactions.

- INQUIRE CLASSCACHE

  The INQUIRE CLASSCACHE command is added to give you information about the active shared class cache in the CICS region and report the presence of any old shared class caches that are awaiting deletion.

- INQUIRE JVM

  The INQUIRE JVM command is added to enable you to identify JVMs in a CICS region and get information about their status.

- INQUIRE WORKREQUEST

  The INQUIRE WORKREQUEST command is added to enable you to track EJB™ tasks. You can:

  – Determine which transactions are associated with a single request.

  – Correlate all transactions associated with a single request (for example, for accounting purposes).

- PERFORM CLASSCACHE

  The PERFORM CLASSCACHE command is added to enable you to start and reload the shared class cache, or to phase out, purge, or forcepurge the shared class cache and the worker JVMs associated with it. While you are performing one of these operations, you can also change the size of the shared class cache, the JVM profile that is used for the master JVM, or the autostart status of the shared class cache.

- SET CLASSCACHE

  The SET CLASSCACHE command is added to enable you to set the status of autostart for the shared class cache.

- SET WORKREQUEST

  The SET WORKREQUEST command is added to enable you to track EJB tasks. You can purge selected work requests.

CETR has new option screens to display and update trace settings for JVMs. The default JVM trace options that are provided in CICS use the JVM trace point level specifications. The default settings for JVM Level 0 trace, JVM Level 1 trace, and JVM Level 2 trace specify LEVEL0, LEVEL1, and LEVEL2, respectively, so they map to the Level 0, Level 1, and Level 2 trace point levels for JVMs. A Level 0 trace point is very important, and this classification is reserved for extraordinary events and errors. Note that unlike CICS exception trace, which cannot be switched off, the JVM Level 0 trace is normally switched off unless JVM tracing is required. The Level 1 trace points and Level 2 trace points provide deeper levels of tracing.

There is a new CICS internal system transaction added to the list of category one transactions. These are the transactions that need to be defined to RACF, and to which the CICS region user ID must be authorized, to enable CICS to initialize successfully when you are running CICS with security enabled (SEC=YES). The new transaction is CJMJ—the CICS master JVM transaction.

6.19.4 Resource definition

CICS Resources

- **Sharing the CSD**
  - CICS TS 3.1 CSD can be shared with 2.2
    - Use DFHCOMPA group
- **PROGRAM**
  - EXECKEY now applies to Java programs
  - JVM debug option moved to JVM profile
  - JVMPROFILE refers to files in the JVMPROFILEDIR
    - Case matters!
- **New IBM groups**
  - DFHDP
    - Application debugging profile manager 3270 interface
  - DFHWB
    - Application debugging profile manager web interface

Figure 6-6 RDO

Most releases of CICS make changes to the IBM-supplied groups of resource definitions that are included in the DFHLIST group list. In all such cases, the old versions of the CICS resource definitions are retained in compatibility groups, which are needed to support earlier releases if you share the CSD between different levels of CICS.

If, after upgrading a CSD, you plan to share the CSD with earlier releases of CICS, include the appropriate DFHCOMPx compatibility groups in your startup group list to provide the required support for earlier releases.

There are changes to the PROGRAM definition:

- PROGRAM EXECKEY({USER|CICS})

  The EXECKEY attribute now applies to programs that run in a JVM. You can use the same JVM profile to invoke a JVM in either of the keys.

- JVMPROFILE({DFHJVMPR|name})

  The JVM profiles that you specify using the JVMPROFILE attribute are now files in the HFS directory that is specified by the system initialization parameter JVMPROFILEDIR, and you must specify the name using the same combination of upper and lower case characters that is present in the HFS file name.

There are new groups of resource definitions added to your CSD when you run the UPGRADE command:

- DFHDP

  IBM-supplied group DFHDP contains the resource definitions for the new Application debugging profile manager 3270 interface (the CADP transaction) and for the inactivate debugging profiles utility.

- DFHDPWB

  IBM-supplied group DFHDPWB contains the resource definitions for the new application debugging profile manager Web interface.

6.19.5 Global user exits

New global user exits are:

- XICERES

  This enables the user to determine the availability of resources in a remote region for dynamically routed starts.

  XICERES is invoked by the interval control program, before CICS processes a non-terminal-related EXEC CICS START request that has been dynamically routed to this region.

- XPCERES

  This enables the user to determine the availability of resources in a remote region for dynamic distributed program links.

  XPCERES is invoked by the EXEC interface program, on the target region, before CICS processes either of the following kinds of dynamically routed link request:

  - A distributed program link (DPL) call
  - A Link3270 bridge request

6.19.6 Language Environment

Runtime support for OS/VS COBOL programs is withdrawn.

OS/VS COBOL programs, which had runtime support in CICS Transaction Server for z/OS Version 2, cannot run under CICS TS for z/OS Version 3. OS/VS COBOL programs must be upgraded to Language Environment conforming COBOL, and recompiled against a level of COBOL compiler supported by CICS.

Enterprise COBOL for z/OS and OS/390 Version 3 is the recommended compiler. Appendix B of the *CICS Application Programming Guide* provides assistance with converting OS/VS COBOL programs to Language Environment conforming COBOL.

A new abend code ALIK indicates an attempt to use an OS/VS COBOL program. In this situation, CICS abnormally terminates the task and disables the program, and CICS processing continues.

In CICS Transaction Server for z/OS Version 3 Release 1, interfaces to the VS COBOL II, OS PL/I, and C/370 runtimes are removed. Applications compiled and linked with these non Language Environment conforming products usually execute successfully under the Language Environment in compatibility mode.

Review the ERDSA specification for CICS LE requirements. The minimum is 3500 K.

Review the RDO definitions for LE programs. The LE language interface modules' definitions are in the CEE group, supplied in the CEECCSD member of the SCEESAMP library.

Define the LE transient data destinations, CESE and CESO:

- DD names CEEMSG and CEEOUT.
- RDO group DFHDCTG contains entries for CESE and CESO.

Define the LE runtime libraries on the CICS STEPLIB and DFHRPL DD statements as follows:

- Add the SCEERUN and SCEERUN2 to STEPLIB or to LNKLST concatenation.
- Add the SCEECICS and SCEERUN libraries to DFHRPL.
  - SCEECICS concatenated before SCEERUN
  - SCEERUN2 library does not need to be added to DFHRPL

Java

CICS Transaction Server for z/OS Version 3 Release 1 supports the JVM created by the IBM Software Developer Kit for z/OS, Java 2 Technology Edition Version 1.4.2 or later, which features the persistent reusable JVM technology.

CICS Transaction Server for z/OS Version 2 Release 2 supported the JVM created by the IBM Developer Kit for OS/390 Java 2 Technology Edition Version 1.3.1s, which also featured the persistent reusable JVM technology. Java programs that ran under CICS Transaction Server for z/OS Version 2 Release 2 can also run under CICS Transaction Server for z/OS Version 3 Release 1.

The library SDFJAUTH is now required for Java support. SDFJAUTH is the partitioned data set extended (PDSE) version of SDFHAUTH, and it contains some of the components of the SJ domain. A separate library is needed because these components are now built using XPLink (Extra Performance Linkage). As for the SDFHAUTH library, the SDFJAUTH library must be APF-authorized by adding it to the list of APF-authorized libraries in an appropriate member in SYS1.PARMLIB, and a STEPLIB DD statement must be provided for it in your startup job stream.

JVM profiles, which contain the JVM initialization options, are now kept as HFS files, rather than as members of a partitioned data set (PDS). The DFHJVM DD card in the CICS startup JCL, which referred to the PDS for the JVM profiles, is no longer required and should be removed. You can use several different JVM profiles in the same CICS region, and each is stored as a separate HFS file. The name of each JVM profile (that is, the name of the HFS file) must still be eight characters or less, so that it can be used in program definition. Use the JVMPROFILE attribute of a PROGRAM resource definition to name the JVM profile that is used to construct the JVM that runs the program.

CICS-defined programs now have their own JVM profile, DFHJVMCD, to make them independent of any changes you make to the default JVM profile DFHJVMPR. DFHJVMCD is used by the default request processor program DFJIIRP, which is used by the CICS-supplied CIRP request processor transaction, and by DFJIIRQ, the CICS-key equivalent of DFJIIRP. DFHJVMCD has an associated JVM properties file, dfjjvmcd.props. You need to make changes to DFHJVMCD and dfjjvmcd.props to ensure that the settings in them are suitable for your installation (including the configuration for your JNDI nameserver).

For language migration issues see:

http://java.sun.com/j2se/1.4/compatibility.html
http://java.sun.com/products/jdk/1.3/compatibility.html#incompatibilities1.3

The EXECKEY parameter on the PROGRAM resource definition is no longer ignored for Java programs. In CICS Transaction Server for OS/390 Version 1 Release 3 and CICS Transaction Server for z/OS Version 2 Release 2, CICS made all Java programs execute in CICS key, but they now execute as specified by the EXECKEY parameter. The default for this parameter is EXECKEY(USER), which means that the program runs in a JVM that executes in user key. (A new type of open TCB, the J9 TCB, is used for these JVMs.) As running applications in user key extends CICS storage protection, it could be beneficial to let most of your Java programs run in a JVM in user key.

Before setting up the shared class cache, you must check the options for semaphores that you have set in the BPXPRMxx members of SYS1.PARMLIB.

The master JVM that initializes the shared class cache uses a single semaphore ID, and requests a set of 32 semaphores, so you must:

- Ensure that the MNIDS value is enough for the maximum number of semaphore IDs that are in use at one time, including the shared class cache. Depending on the frequency with which you expect to reload the shared class cache, you might want to allow two or possibly three semaphore IDs for the shared class cache. One semaphore ID would be used by the master JVM that controls the active shared class cache, and the remainder would be used by a master JVM that controls a shared class cache that is being phased out, or by a new master JVM that controls a shared class cache that is being loaded. It is unlikely that you would need more than two semaphore IDs for the shared class cache, except in a CICS region that is being heavily used for development and testing.
- Ensure that the MNSEMS value is enough for the maximum number of semaphores that the master JVM requests in a semaphore set. The value must be 32 or greater. If you need to change the MNIDS value, you can do this by using the IPCSEMNSEMS parameter that is in the BPXPRMxx members of SYS1.PARMLIB.

7

CICS TS 3.1 exploitation

This chapter discusses the use of the new functionality in CICS TS 3.1 and how you can exploit enhancements once you have migrated to this new release. Considerations are made in functional areas of open transaction environment (OTE) and threadsafe applications, Web Services, and channels and containers.

7.1 OTE considerations

In the CICS open transaction environment in CICS TS 3.1, threadsafe application programs, task-related user exits, global user exit programs, and user-replaceable modules can run concurrently on multiple open TCBs and no longer rely on quasi-reentrancy for serialized access to resources. Previously, in CICS TS V2, only some task-related user exits enabled with the OPENAPI attribute were supported.

In CICS TS 3.1, programs are now defined with API(OPENAPI) and will run almost independently of the QR TCB—running on an L8 or L9 open TCB depending on its EXECKEY value. In addition, any program that can be defined as COncurrency(Threadsafe) can now also be defined as API(OPENAPI) and exploit the performance benefits of running on an open TCB. For this reason we recommend that *all* programs be written to threadsafe standards.

Prior to CICS TS 3.1 the OPENAPI option was only available to task-related user exits (TRUEs).

The implication of this now in CICS TS 3.1 is that since multiple tasks can potentially access shared resources simultaneously when executing under an open TCB, applications that access these resources, such as the CWA, must be responsible to ensure the integrity of those resources by implementing an appropriate serialization technique. This technique involves an agreed-upon set of standards to ensure serialized access using threadsafe commands.

The responsibility for preserving data integrity in a CICS Transaction Server V3.1 environment lies with CICS, the product, and the application owner. This is summarized as follows:

- CICS will ensure threadsafe access to its managed resources.
  - A large subset of the CICS API and SPI in CICS TS 3.1 is coded to threadsafe standards.
  - For VSAM files, temporary storage, and TDQs, CICS ensures correct serialized access.
- Applications need to ensure threadsafe access to their resources

  Shared storage (for example, CWA, GETMAIN SHARED)

A further discussion of threadsafe considerations for applications and CICS is in 7.1.2, “Threadsafe considerations” on page 152.

7.1.1 OPENAPI programs and additional TCB switching

OPENAPI programs are programs to run on an open TCB from the start of the program. This kind of program is an OPENAPI program and is defined in the program definition as API(OPENAPI).

An OPENAPI program is a program that has been written to threadsafe standards and does not rely on a call to a TRUE to move the program to an open TCB (as in CICS TS V2). An OPENAPI program is a program that *must* be run on an open TCB.

Programs defined with API(OPENAPI) run almost independently of the QR TCB. Such programs run on an L8 or L9 open TCB, depending upon their EXECKEY value. Because OPENAPI programs can potentially use non-CICS APIs, the key of the TCB is important and must match the execution key. This is unlike API(CICSAPI) threadsafe programs that can execute in CICS key or user key irrespective of the TCB key. CICS services are implemented irrespective of the key of the TCB they are running under, unlike MVS services, which care about the TCB key.

Important: Use of non-CICS APIs within CICS is entirely at the risk of the user. No testing of non-CICS APIs within CICS has been undertaken by IBM, and use of such APIs is not supported by IBM Service.

The use of OPENAPI programs can increase TCB switching within CICS. If an OPENAPI program is defined to run with an execution key of user, it is given control under an L9 TCB rather than an L8. Should the program issue SQL calls to invoke DB2, the task is switched to an L8 TCB for the duration of the EXEC SQL request. This is since OPENAPI TRUEs such as DFHD2EX1 have to run in CICS key under an L8 TCB. On completion of the SQL request, CICS returns control to the application program on its L9 TCB.

Likewise, an OPENAPI program that invokes non-threadsafe EXEC CICS commands will be switched from its L8 or L9 TCB to the QR TCB for the duration of the CICS request, then switched back to the open TCB when returning control to the application program. This is because when a program is defined as being OPENAPI it means it *must* run its application logic under an open L8 or L9 TCB. This is different from a CICSAPI threadsafe program, which does not have affinity to any one TCB and executes under whatever TCB CICS deems appropriate to use.

To avoid such additional TCB switching, user key CICS DB2 applications are best left defined as CICSAPI threadsafe programs. Other good candidates for threadsafe programs defined with API(CICSAPI) are those that invoke non-threadsafe CICS API requests.

Good candidates for threadsafe programs defined with API(OPENAPI) are those with an execution key of CICS that invoke EXEC SQL requests, those that only invoke threadsafe CICS API requests, or those CPU-intensive applications.

Important: The EXECKEY program attribute will determine the mode of open TCB that is assigned for an OPENAPI program to run under. User key programs will run under an L9 TCB, CICS key programs under an L8 TCB. There is an exception to this behavior, however. If a CICS system does not have storage protection active (that is, STGPROT=NO is specified), all OPENAPI programs will run under L8 TCBs, regardless of their EXECKEY value. This is because STGPROT=NO makes CICS operate without any storage protection, and so run in a single storage key.

7.1.2 Threadsafe considerations

One of the most important considerations in migrating to CICS 3.1, which is also widely documented in other IBM Redbooks, is the issue of threadsafe and the misconception that simply by defining programs as being threadsafe, performance with regards to response time and CPU usage will be greatly reduced. This is not necessarily the case and this remains one of the major hurdles in post migration testing because of a lack of understanding as to what a *threadsafe program* really means.

This section outlines the need for careful consideration before defining an application as threadsafe. The issue with respect to threadsafeness is twofold. There is either one of the following:

- Compromise to data integrity due to a lack of understanding of program logic and use of non-threadsafe commands
- Performance degradation with respect to transaction response and CPU time due to the nature of TCB switching that occurs

Understanding the application

Without a fundamental understanding of the application program, there is an exposure to compromise the data integrity. There is nothing to prevent you from defining an application as being threadsafe, but what do we mean when we say an application is threadsafe? A threadsafe program is defined as a program that does one of the following:

- Uses appropriate serialization techniques, such as compare and swap or enqueue, when accessing any shared application resources. It must be capable of running concurrently on multiple TCBs, and must not rely on quasi-reentrancy to serialize access to shared resources and storage.
- Uses no shared application resources at all.

For an application to meet these conditions and therefore be considered threadsafe, the application must:

- Incorporate threadsafe application logic (which means that the native language code in between the EXEC CICS commands must be threadsafe).
- Be defined to CICS as threadsafe.

The term *threadsafe application* then, is a collection of application programs that employ an agreed-upon form of serialized access to shared application resources. A single program within the application operating without the agreed-upon serialization technique can destroy the predictability and therefore integrity of an entire system of otherwise threadsafe programs. Therefore, an application system cannot be considered *threadsafe* until all programs that share a common resource implement that application's threadsafe standards.

It is important to note that an application that makes no use of any of the shared resources can be said to be threadsafe even if it uses non threadsafe CICS commands.

Defining a program as threadsafe

The steps to take to make an application threadsafe are summarized as follows:

1. Review the application code including all LINKed programs:
   a. Native code review.
   b. Check shared storage for:
      - CWA
      - GETMAIN SHARED
      - EXTRACT EXIT
      - LOAD PROGRAM HOLD
   c. Use only CICS threadsafe commands.
2. Review program definitions:
   a. API(CICSAPI) or API(OPENAPI)
   b. CONCURRENCY(THREADSAFE)

We now discuss some of these further.

Checking native code

This involves a review of the native code between the CICS API commands to ensure that it is threadsafe and to eliminate a data integrity risk in an OTE and threadsafe environment. While CICS can identify whether CICS API commands are threadsafe, it is the responsibility of the owner of the application to ensure that the native code between the API conforms to threadsafe standards. A technique to follow to identify any problems in this area is as follows:

- Compile any high-level languages with the RENT option. LE programs are guaranteed to be reentrant if compiled with RENT.
- Ensure that Assembler programs do not modify themselves. This can be achieved by compiling with RENT and then restarting CICS with RENTPGM=PROTECT. An abend ASRA will result if they are not threadsafe, as the program will be loaded into RDSA/ERDSA.

Identifying programs that use shared resources

For application-maintained shared resources, it is the responsibility of the application program to ensure that the resource is accessed in a threadsafe manner. Typical examples of shared storage are the CICS CWA, global user exit global work areas, and storage acquired explicitly by the application program with the shared option. You can check whether your application programs use these types of shared storage by looking for occurrences of the following EXEC CICS commands:

- ADDRESS CWA
- EXTRACT EXIT
- GETMAIN SHARED

Application programs using these commands *may* not be threadsafe because they allow access to global storage areas that could be updated concurrently by several tasks running on different open TCBs. To ensure that it is threadsafe, an application program must include the necessary synchronization logic to guard against concurrent update. To help you find occurrences of these commands, CICS provides DFHEIDTH, a sample command table you can use with the load module scanner utility, DFHEISUP.

> **Important:** It is very important that you understand that DFHEIDTH is not testing the scanned programs for non-threadsafe CICS commands but is merely identifying whether the application is using CICS commands that give rise to the *possibility* that the application logic is non threadsafe.

> **Important:** During your investigation process of identifying programs that use shared resources, you should include any program that modifies itself. Such a program is effectively sharing storage and should be considered at risk.

CONCURRENCY attribute

The CONCURRENCY parameter of the PROGRAM resource definition indicates that the program is threadsafe. It is important to understand that the keyword CONCURRENCY(THREADSAFE) is telling CICS that the application logic is threadsafe, not whether CICS commands are threadsafe. CICS will ensure threadsafety of its own logic and switch the execution to the QR TCB if it is necessary. In this way, CICS ensures that the resource is accessed in a threadsafe way.

A threadsafe application can use non-threadsafe CICS commands. It will suffer the overhead of TCB switching but resource integrity is maintained.

If an application containing non-threadsafe logic is incorrectly defined to CICS as CONCURRENCY(THREADSAFE), the results are unpredictable.

If you defined a program with CONCURRENCY(THREADSAFE), all routines that are statically or dynamically called from this program, for example, COBOL routines, must also be coded to threadsafe standards.

7.1.3 The CSACDTA field

Historically, the CSACDTA field provided the address of the task control area (TCA) for the currently dispatched task running within CICS. Before OTE was introduced, all tasks ran under the control of the QR TCB, and this provided a guarantee that a running task would retrieve the address of its own TCA if it accessed the CSACDTA field.

With the introduction of OTE, it is *no longer safe* to assume that the TCA address held within CSACDTA is the TCA of the task that is accessing the CSA. CSACDTA contains the address of the task currently dispatched on the *QR* TCB. The program that is referencing CSACDTA may be running under an open TCB. In this case the wrong TCA address will be used by the program, leading to unpredictable results.

Since CICS/ESA Version 4.1, direct access to CICS control blocks is not supported. The CICS system programming interface (SPI) should be used for programs wishing to access state information about a task.

In CICS Transaction Server Version 3.1, CSACDTA is renamed CSAQRTCA to further discourage its use.

> **Important:** It is the intention of IBM to withdraw the ability to reference a TCA using this field in future releases after CICS Transaction Server Version 3.1 by loading CSAQRTCA with the address of an area of fetch-protected storage. This will result in an abend ASRD with message DFHSR0618 if it is referenced.

7.1.4 Threadsafe migration scenarios

This section highlights the need to examine your use of CICS global user exits and function shipping in your CICS systems, and highlights why, if no action is taken, this can result in a performance degradation after migration. It also discusses the use of the OPENAPI program definition option.

The following scenarios demonstrate the differences between programs being defined as being threadsafe and non threadsafe and the impact they have on performance in a migrated system.

Migrating to CICS TS V3 without threadsafe consideration

One of the reasons you may be migrating to CICS Transaction Server Version 3 is to take advantage of the benefits that OTE and threadsafe implementation can offer. You will have to consider the possible need to make changes in regard to threadsafe to ensure that your applications run with at least the same performance they experienced in the earlier release.

> **Note:** Although the following discussion is related to DB2, many of the points are equally applicable to other OPENAPI-enabled TRUEs. In addition, the example program is defined as an API(CICSAPI) program.

The following scenario shows the program flow and TCB switches of a program making one DB2 call in a region with exits XRMIIN and XRMIOUT and a Dynamic Plan exit all enabled as QUASIRENT.

The transaction in Figure 7-1 on page 157 was migrated to CICS TS 3.1 with no consideration of threadsafe. What we mean by this is that the program associated with transaction TRANA is defined as QUASIRENT and all exits are enabled as QUASIRENT. This diagram shows that there are many TCB switches from the L8 TCB, which runs the DB2 true, to the QR TCB and back. The non-threadsafe exits must run on the QR TCB to ensure that serialization occurs. In this example we see eight TCB switches occurring. Each TCB switch incurs additional CPU time, which can adversely affect performance. If the exits were enabled as THREADSAFE and their associated programs were written to threadsafe standards the program would be allowed to continue running on the

L8 TCB and the additional switches would not be necessary. This is shown in the following scenario.

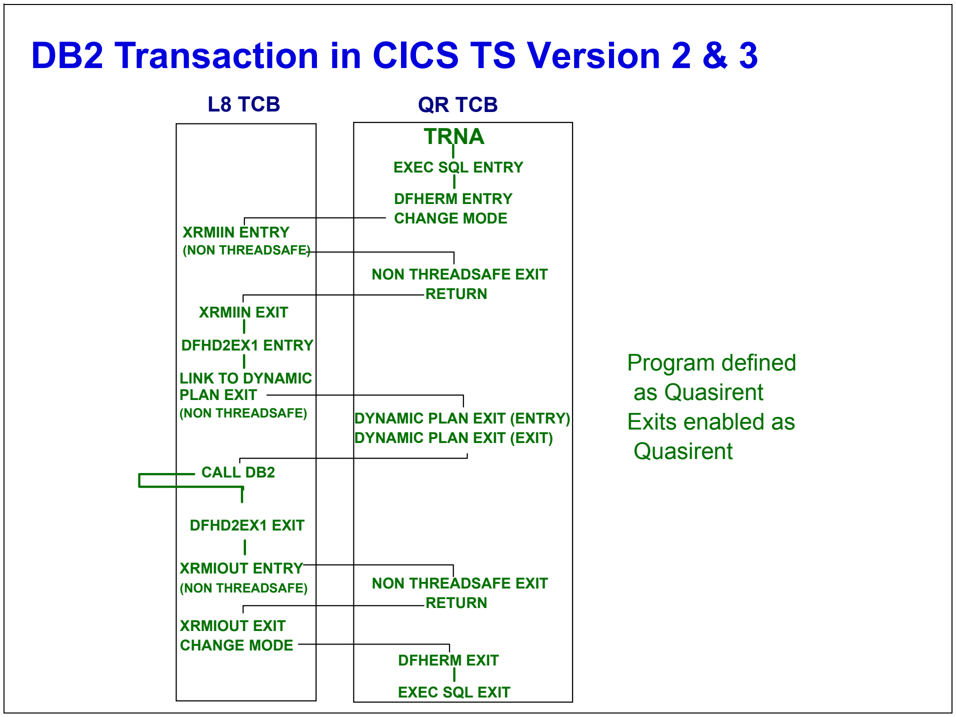


Figure 7-1 TCB switches before exits are enabled as threadsafe on CICS TS Version 3

Migrating to CICS TS V3 with threadsafe consideration

Now we consider two more scenarios:

1. The effect of the Dynamic Plan Exit and the XRMIIN and XRMIOUT exits enabled as threadsafe with their associated programs written to threadsafe standards.
2. The true benefit threadsafe has to offer, with the application and programs associated with all exits within the path being defined and written to threadsafe standards.

Scenario 1 - The CICS exits defined as threadsafe

Figure 7-2 on page 158 shows the same transaction, TRANA, running in a CICS Transaction Server Version 3 environment and making one DB2 call. The transaction was migrated to CICS TS V3 with threadsafe consideration in mind. The program associated with transaction TRANA is still defined as quasirent but the XRMIIN, XRMIOUT, and the dynamic plan exits were enabled as

THREADSAFE and their associated programs were written to threadsafe standards. This diagram shows that the number of TCB switches has been reduced to two switches. A TCB switch back to the QR TCB must still take place upon completion of the DB2 call due to TRANA's program not being threadsafe. Therefore, there are two TCB switches per DB2 call.

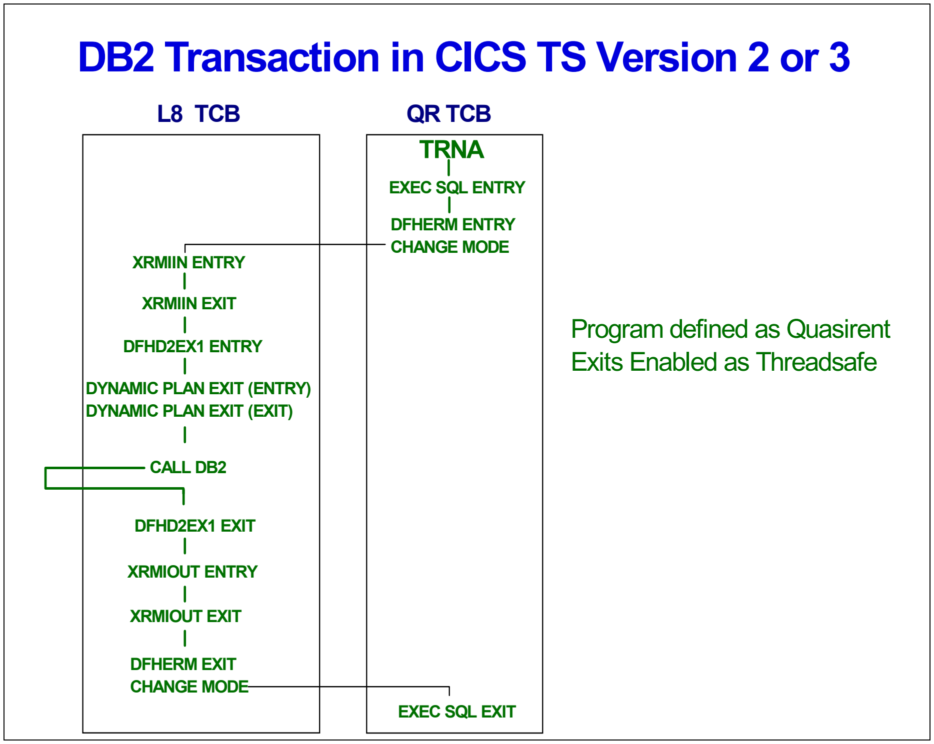


Figure 7-2 TCB switches after exits are made threadsafe on CICS TS Version 3

Scenario 2 - All programs and exits are threadsafe

Figure 7-3 on page 159 shows the same transaction, TRANA, running in a CICS Transaction Server Version 3 environment and making one DB2 call. The transaction was migrated to CICS Transaction Server Version 2 or Version 3 with threadsafe consideration in mind. The program associated with transaction TRANA and the programs associated with XRMIIN, XRMIOUT, and the dynamic plan exits are all defined as THREADSAFE and written to threadsafe standards.

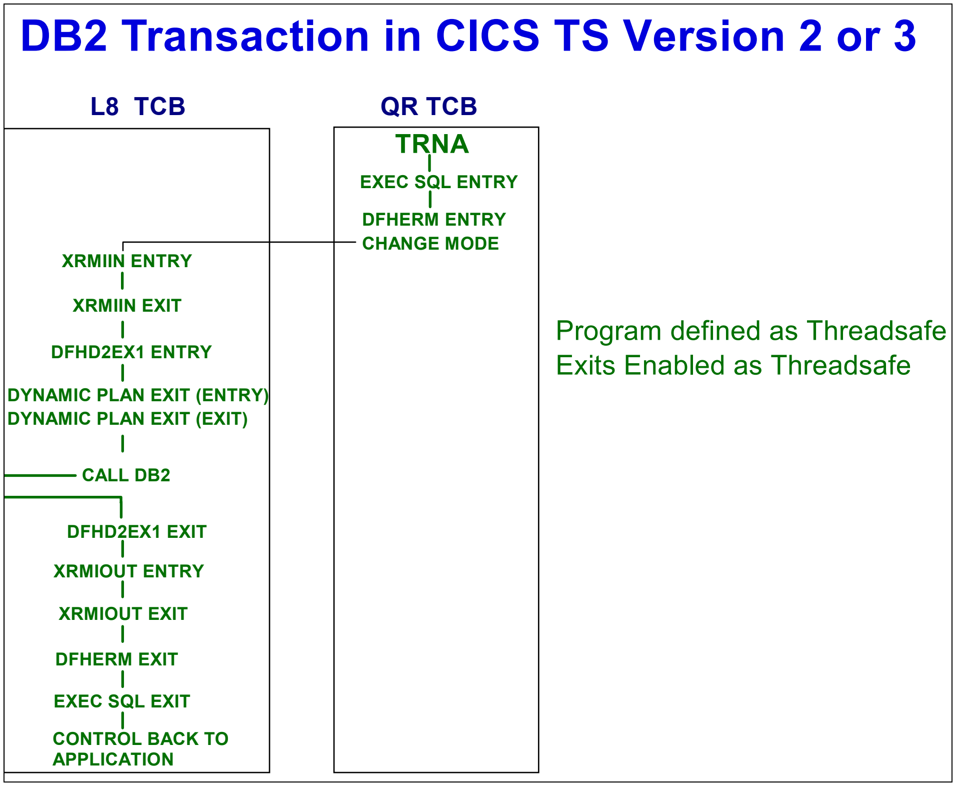


Figure 7-3 TCB switches with programs and exits running as threadsafe on CICS TS Version 3

Figure 7-3 shows a TCB switch from the QR TCB to the L8 TCB for the first DB2 call. Upon completion of the DB2 call the program remains on the L8 TCB. The number of DB2 calls that could be made without another TCB switch is only limited by the design of the application. There would only be a TCB switch back to the QR TCB at task termination time.

All of these scenarios highlight the scope as far as performance benefit that threadsafe can offer with regard to savings in both CPU and response time. These examples are to show that without threadsafe consideration, the migration of critical business applications may prove costly in terms of performance and response.

7.1.5 Function shipped commands

This section discusses the impact function shipping has on TCB mode switching, which can greatly assist in understanding why a particular program may appear to be doing a lot of TCB switching, although it may be defined as being threadsafe.

We demonstrate that the conversion of remote temporary storage queues to shared temporary storage queues within a coupling facility is a recommended solution within a threadsafe environment

The temporary storage commands are threadsafe. This is true when the commands are performed against locally defined resources or against shared temporary storage queues residing within a coupling facility. However, if these commands are performed against remote resources, they must be function shipped to the remote region to execute. This involves extra TCB switching due to Multi-Region Operation (MRO) and Intersystem Communication (ISC) CICS components not being threadsafe. The same is true for to an EXEC CICS LINK command to a remote program (that is, a DPL call).

The following examples show these commands being performed in both local and remote scenarios.

In all of these scenarios the application is defined as a threadsafe CICSAPI program that makes a single DB2 call on an open L8 TCB.

Local resource scenarios with no TCB switching

In each example, a threadsafe CICSAPI application program makes a DB2 call on an open L8 TCB.

1. The program does an EXEC CICS LINK to a program that is defined as a local program. The LINK command is threadsafe, so there is no mode switch to the QR TCB and the request is processed entirely on the L8 TCB.
2. The program issues a WRITEQ-TS request to a temporary storage queue, which is defined as a local queue. The WRITEQ-TS command is threadsafe so there is no mode switch to the QR TCB and the request is processed on the L8 TCB.
3. The program issues a WRITEQ-TS request to a shared temporary storage queue, which resides within a coupling facility. In this scenario there is no need to function ship the WRITEQ-TS request. The application continues to run on the L8 TCB with no additional TCB switches to the QR TCB. Note that the initial call to the shared temporary storage server is always issued from the QR TCB, regardless of which TCB the program is currently on.

Remote resource scenarios with TCB switching

Once again, this is a threadsafe CICSAPI application program making a DB2 call on an open L8 TCB.

1. The program performs a DPL request to a remote program. Although the link command itself is threadsafe, there is a mode switch to the QR TCB in order to ship the request to the remote region. When we return the application continues to run on the QR TCB and does not switch back to the L8 TCB. The application will not be switched to the L8 TCB until another DB2 request is made.
2. The program issues a WRITEQ-TS request to a remote temporary storage queue. Although the WRITEQ-TS command itself is threadsafe, there is a mode switch to the QR TCB in order to function ship the request to the remote region. When the WRITEQ-TS returns, the application continues to run on the QR TCB and does not switch back to the L8 TCB. The application will not be switched to the L8 TCB until another DB2 request is made.

7.1.6 COBOL considerations

First it must be noted that all the runtime CICS interfaces for OS/VS COBOL are removed in CICS Transaction Server V3.1. OS/VS COBOL programs that had runtime support in CICS Transaction Server V2.3 will no longer be able to run and CICS will terminate the program with an abend ALIK.

If your application makes use of COBOL calls to link to sub programs then you must be aware that the concurrency value used will be the value set for the program at the higher level. So if PROGA is defined as CONCURRENCY(QUASIRENT) and PROGB is defined as CONCURRENCY(THREADSAFE), the concurrency attribute that will be honored will be QUASIRENT when we call PROGB from PROGA. This can be demonstrated by looking at the following two trace examples.

This behavior can be seen when using both dynamic COBOL calls as well as static COBOL calls.

With CICS Transaction Server Version 3, the API attribute of the calling program is also inherited by the called program. If, for example, PROGA had been defined with API(OPENAPI) and EXECKEY(USER), it would have been invoked under an L9 TCB, and would have called PROGB under the L9 TCB too.

7.2 Channels and containers

The new functionality of using channels and containers in CICS TS 3.1 has already been discussed in 1.10, “Enhanced inter-program data transfer” on page 20. Essentially, it provides a solution to the 32-KB limit imposed on the traditional CICS COMMAREA in order to accommodate modern applications. There is now a need for considering how you currently handle data exchange and whether implementing this new function will benefit your application design needs.

Consider some of the COMMAREA issues you may face when handling large data objects:

- Applications must use a circumvention technique, such as using external VSAM files, or splitting the data into separate parts. This method increases risk, as well as programming time and effort.
- Passing XML documents by value throughout the request process path becomes inhibited because the size constraint applies to:
  - Calls between CICS programs both within the local system and between CICS systems
  - Parameter data passed between CICS tasks
  - External client programming interfaces such as CICS interface (EXCI and the CICS client external call interface (ECI))
- Data structures used to define a COMMAREA payload can become overloaded. Redefining structures on the same area of memory increases the risk of program errors. Similarly, confusion about the validity of fields can result in application programming errors.
- An overloaded COMMAREA structure increases transmission time between CICS regions because the structure size must account for the maximum size of the data that could be returned from the called program, and this parameter size depends on the request logic invoked.

  CICS TS must always allocate memory to accommodate the return of the maximum COMMAREA structure size.
- A code-page conversion of the COMMAREA structure is complex because binary and character data cannot be easily separated.

7.2.1 Advantages over COMMAREAs

The containers and channels approach has several advantages over COMMAREAs:

- Containers can be any size and, as a result, can extend beyond the maximum 32-KB size of a COMMAREA. There is no limit to the number of containers that can be added to a channel, and the size of the individual containers is limited only by the amount of storage available.
- A channel consists of multiple containers, enabling it to be used to pass data in a more structured way. In contrast, a COMMAREA is a single block of data.
- Unlike COMMAREAs, channels do not require the programs that use them to know the exact size of data returned, making programming easier.

7.2.2 Channels

A channel is a uniquely named reference to a collection of application parameter data held in containers. It is analogous to a COMMAREA but is not subject to the constraints of a COMMAREA.

You can choose a channel name that is a meaningful representation of the data structures that the channel is associated with. For example, in a human resource application, a channel name might be <employee-info>.

This collection of application parameter data serves as a standard mechanism to exchange data between CICS programs.

CICS TS provides an EXEC API that associates a named channel with a collection of one or more containers — offering an easy way to group parameter data structures to pass to a called application.

CICS TS removes a channel when it can no longer be referenced (when it becomes out of scope).

The current channel

A program's current channel is the channel (if any) with which it was invoked. The current channel is set by the calling program or transaction by transferring the control to the called program via a LINK, XCTL, START, and pseudo-conversational RETURN with the channel parameter.

Although the program can create other channels. the current channel, for a particular invocation of a particular program, never changes. It is analogous to a parameter list.

If a channel is not explicitly specified, the current channel is used as the default value for the CHANNEL (channel-name) parameter on the EXEC CICS command. This is shown in Figure 7-4.

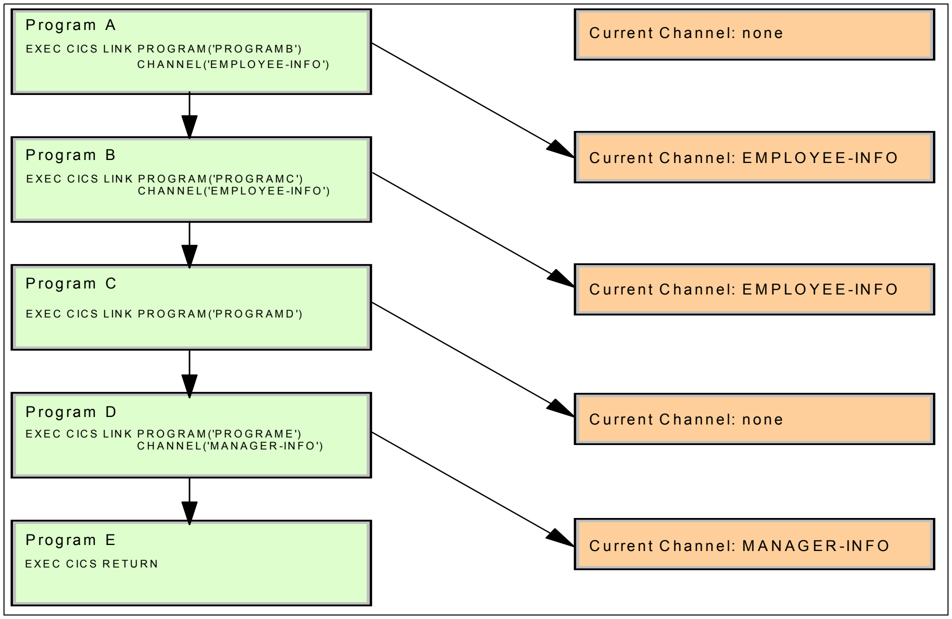


Figure 7-4 The current channel

Typically, programs that exchange a channel are written to handle that channel, that is, both client and server programs know the name of the channel, and the names and number of the containers in that channel. However, if, for example, a server program or component is written to handle more than one channel, on invocation it must discover which of the possible channels it has been passed.

A program can discover its current channel (that is, the channel with which it was invoked) by issuing an EXEC CICS ASSIGN CHANNEL command. (If there is no current channel, the command returns blanks.)

The program can also retrieve the names of the containers in its current channel by browsing, but typically this is not necessary. A program written to handle several channels is often coded to be aware of the names and number of the containers in each possible channel.

To get the names of the containers in the current channel, use the browse commands, as shown in Example 7-1.

Example 7-1 Browsing containers in a channel

```
EXEC CICS STARTBROWSE CONTAINER BROWSETOKEN(data-area)
EXEC CICS GETNEXT CONTAINER(data-area) BROWSETOKEN(token)
EXEC CICS ENDBROWSE CONTAINER BROWSETOKEN(token)
```

Having retrieved the name of its current channel and, if necessary, the names of the containers in the channel, a server program can adjust its processing to suit the kind of data that it has been passed.

Note: For a program creating a channel, the ASSIGN CHANNEL command will return blanks unless it was invoked via START, LINK, or XCTL specifying channel name.

The scope of a channel

The scope of a channel is the code (that is, the program or programs) from which it can be accessed.

Figure 7-5 shows the scope of channel EMPLOYEE-INFO, which consists of programs A (the program that created it), program B (for which it is the current channel), and program C (for which it is also the current channel). Additionally, we show the scope of channel MANAGER-INFO, which consists of program D (which created it) and program E (for which it is the current channel).

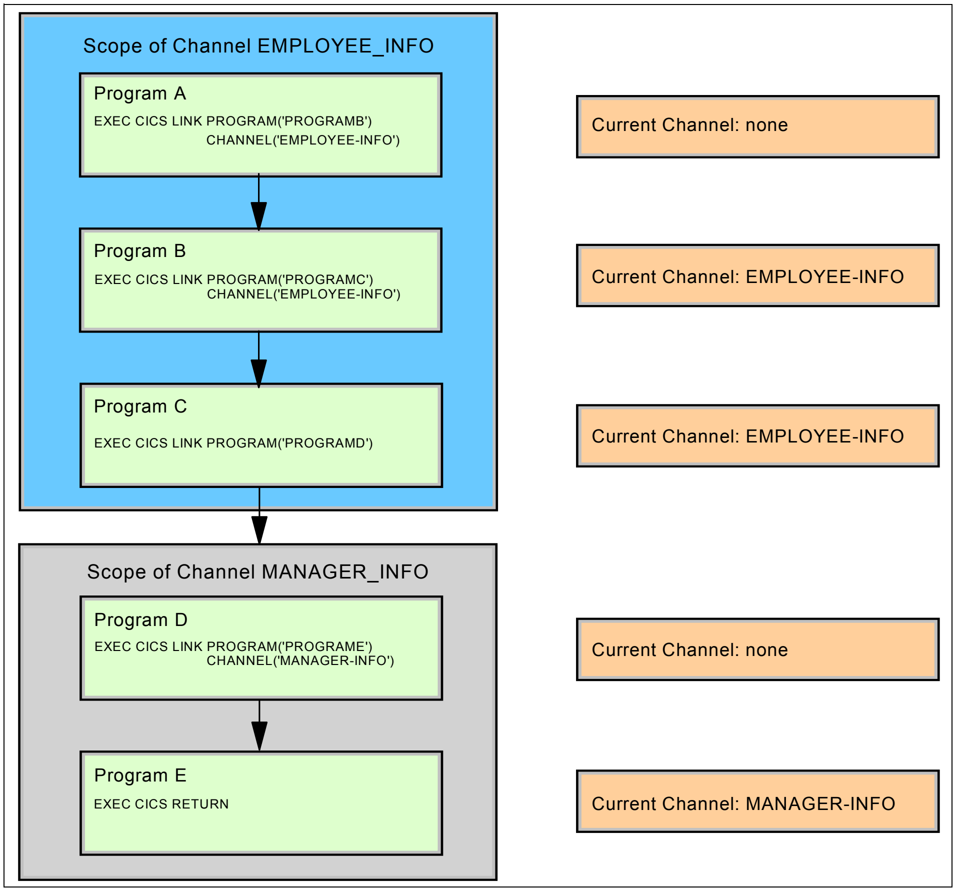


Figure 7-5 Example showing the scope of a channel

Lifetime of a channel

A channel is created when it is named on an EXEC CICS command. The usual command to create a channel is the EXEC CICS PUT CONTAINER command, in which specifying the CHANNEL parameter will create the channel and also associate the container with it.

A channel will be deleted when it goes out of scope to the programs in the linkage stack, meaning that no programs will be able to access it. This will cause the channel to be deleted by CICS.

Figure 7-6 shows the APIs used to create and manage a channel.

```
► EXEC CICS PUT CONTAINER CHANNEL
Creates a channel and places data into a container within the channel
► EXEC CICS GET CONTAINER CHANNEL
Retrieves the container data passed to the called program
► EXEC CICS MOVE CONTAINER CHANNEL AS TOCHANNEL
Moves a container from one channel to another channel
► EXEC CICS DELETE CONTAINER CHANNEL
Deletes a container
► EXEC CICS ASSIGN CHANNEL
Returns the name of the program's current channel, if one exists
► EXEC CICS LINK PROGRAM CHANNEL
Links to the program, on a local or remote system, passing the channel and
container data
► EXEC CICS XCTL PROGRAM CHANNEL
Transfers control to the program passing the channel and container data
► EXEC CICS START TRANSID CHANNEL
Starts a task, on a local or remote system, copying the named channel and
container data and passing it to the started task
► EXEC CICS RETURN TRANSID CHANNEL
Returns control to CICS, passing the channel and container data to the next
transaction
```

Figure 7-6 API to create and manage a channel

7.2.3 Containers

A container is a uniquely named block of data that can be passed to a subsequent program or transaction. It refers to a particular parameter data structure that exists within a collection of virtually any form of application parameter data.

You can choose a container name that is a meaningful representation of the data structure. For example, in a human resource application, the container name might be <employee-name>.

CICS TS provides EXEC API verbs to create, delete, reference, access, and manipulate a container as well as to associate it with a channel. See Figure 7-7 for more details.

```
► EXEC CICS PUT CONTAINER CHANNEL
Creates a channel and places data into a container within the channel
► EXEC CICS GET CONTAINER CHANNEL
Retrieves the container data passed to the called program
► EXEC CICS MOVE CONTAINER CHANNEL AS TOCHANNEL
Moves a container from one channel to another channel
► EXEC CICS DELETE CONTAINER CHANNEL
Deletes a container from a channel
► EXEC CICS STARTBROWSE CONTAINER
Start a browse of the containers associated with a channel
► EXEC CICS GETNEXT CONTAINER
Return the name of the next container associated to the channel
► EXEC CICS ENDBROWSE CONTAINER
Ends the browse of the containers associated with the channel
```

Figure 7-7 Container-related API

A container can be any length, and a container size is constrained only by the available user storage in the CICS address space. It can include data in any format required by an application. An application can create any number of containers and can use separate containers for different data types, such as binary and character data. This capability helps ensure that each container structure is based on a unique area of memory.

It also minimizes the potential for errors that commonly arise when parameter data for multiple applications is overloaded in a single memory area, by isolating different data structures, and making the association between data structure and purpose clear.

CICS read-only containers

CICS can create channels and containers for its own use, and pass them to user programs. In some cases CICS marks these containers read-only, so that the user program cannot modify data that CICS needs on return from the user program.

User programs cannot create read-only containers.

You cannot overwrite, move, or delete a read-only container. Thus, if you specify a read-only container on a PUT CONTAINER, MOVE CONTAINER, or DELETE CONTAINER command you will receive an INVREQ condition.

7.2.4 Channels and BTS

The PUT, GET, MOVE, and DELETE CONTAINER commands used to build and interact with a channel are similar to those used in CICS business transaction services (BTS) applications. BTS implemented containers as a way of passing information between activities and processes. There is no limit to the size of a container in BTS. In fact, there have been white papers written to describe how a programmer might use BTS containers as a *big COMMAREA*.

The containers used in the channel context are similar to those used in BTS, and the commands used to access the container data are similar (for example, GET, PUT, MOVE, DELETE).

It is possible to have the same server program invoked in both a channel and a BTS context. To accomplish this the server program must avoid the use of options that specifically identify the context.

The server program must *call* CICS to determine the context of a command. When a container command is executed CICS will first check to see if there is a current channel. If there is, then the context of the command will be channel. If there is no current channel, CICS will then check to see if this is part of a BTS activity. If this is part of a BTS activity, then the context will be BTS. If the program has no channel context and no BTS context then an INVREQ will be raised, so a program that issues container commands can be used, without change, as part of a channel application or as part of a BTS activity.

The BTS approach requires the adoption of a new programming approach for CICS applications with consequential application re-engineering. This may be an ambitious undertaking for a mature critical CICS business application suite. The channels and containers approach is more simple and does not require as much effort to change applications.

Note: Channels and containers are not recoverable. If you need to use recoverable containers, use CICS business transaction services (BTS) containers.

7.2.5 Channels and JCICS

CICS provides EXEC API support for channels and containers in all supported CICS programming languages. In the CICS Java environment JCICS classes are provided to enable channels and containers to be used as the mechanism for exchanging data between CICS J2EE style applications and traditional CICS procedural applications.

See Figure 7-8 for details about the JCICS classes that CICS Java programs can use to pass and receive channels.

```
► com.ibm.cics.server.Channel
A Channel class used to create new containers in a channel
► com.ibm.cics.server.Container
A Container class used to place data in a container
► com.ibm.cics.server.ContainerIterator
A ContainerInterator class used to browse the current channel
```

Figure 7-8 JCICS classes managing channels

CICS also provides the following exception classes for handling errors shown in Figure 7-9.

```
► com.ibm.cics.server.CCSIDErrorException
Class that represents the CICS CCSIDERR condition
► com.ibm.cics.server.ChannelErrorException
Class that represents the CICS CHANNELERR condition
► com.ibm.cics.server.ContainerErrorException
Class that represents the CICS CONTAINERERR condition
```

Figure 7-9 JCICS classes for channels error handling

> **Note:** You can use channel and container related JCICS commands when writing CICS enterprise beans. However, CICS does not support the transmission of channels over IIOP request streams. This means that you cannot, for example, pass a channel to an enterprise bean on a remote region.

7.2.6 Data conversion

The data conversion model used by channel applications is much simpler than that used by COMMAREA applications, as data conversion in COMMAREA applications is controlled by the system programmer, while in channel applications it is controlled by the application programmer, using simple API commands.

Here are some cases in which data conversion is necessary:

- When character data is passed between platforms that use different encoding standards (for example, EBCDIC and ASCII)
- When you want to change the encoding of some character data from one Coded Character Set Identifier (CCSID) to another

Applications that use channels to exchange data use a simple data conversion model. Frequently, no conversion is required and, when it is, a single programming instruction can be used to tell CICS to handle it automatically.

Using COMMAREAs

For applications that use the COMMAREAs to exchange data, the conversion is done under the control of the system programmer, using the DFHCNV conversion table, the DFHCCNV conversion program, and (optionally) the DFHUCNV user-replaceable conversion program.

Using channels

The data conversion model used by channel applications is much simpler than that used by the COMMAREA applications. The data in channel and containers is converted under the control of the application programmer, using API commands.

- The application programmer is responsible only for the conversion of user data—that is, the data in containers created by the application programs. System data is converted automatically by CICS, where necessary.
- The application programmer is concerned only with the conversion of character data. The conversion of binary data (between big-endian and little-endian) is not supported.
- Applications can use the container API as a simple means of converting character data from one code page to another. Example 7-2 converts data from codepage1 to codepage2.

Example 7-2 API to convert codepage

```
EXEC CICS PUT CONTAINER(temp) DATATYPE(CHAR)
              FROMCCSID(codepage1) FROM(input-data)
EXEC CICS GET CONTAINER(temp) INTOCCSID(codepage2)
              SET(data-ptr) FLENGTH(data-len)
```

7.2.7 Migrating COMMAREA to channels and containers

To migrate programs exchanging data via a COMMAREA on a LINK command, the format of the command must be changed and proper commands must be added to use channels and containers. Figure 7-10 shows an example of this.

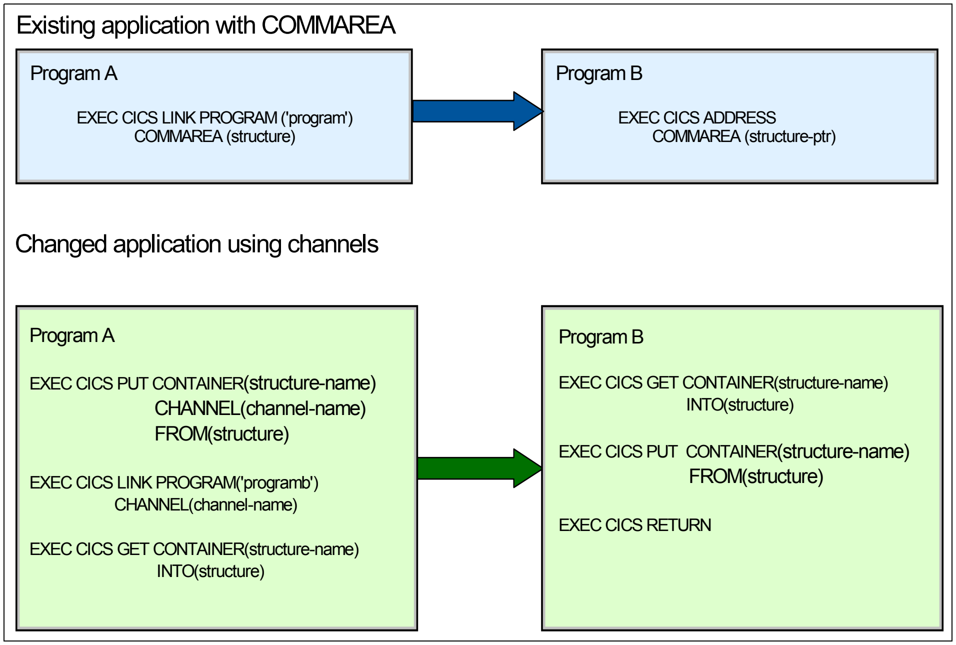


Figure 7-10 Changes from commarea to channels using LINK

The same applies to programs using the START command with the COMMAREA. Figure 7-11 shows an example of this.

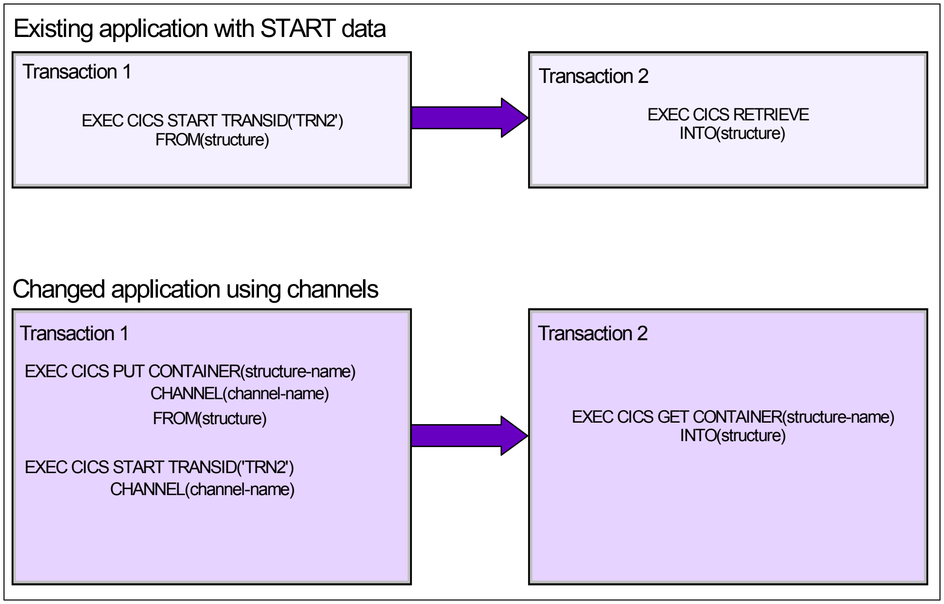


Figure 7-11 Changes from commarea to channels using START

Migration consideration

You may wish to consider the following items when migrating from a COMMAREA to channels and containers:

- CICS application programs that use traditional COMMAREAS to exchange data will continue to work as before.
- EXEC CICS LINK and EXEC CICS START commands, which can pass either COMMAREAs or channels, can be dynamically routed.
- If you employ a user-written dynamic or distributed routing program for workload management, rather than CICSPlex SM, you must modify your program to handle the new values that it may be passed in the DYRLEVEL, DYRTYPE, and DYRVER fields of the DFHDYPDS communications area.
- It is possible to replace a COMMAREA by a channel with a single container. While this may seem the simplest way to move from COMMAREAs to channels and containers, it is not good practice to do this.
- Also, be aware that a channel may use more storage than a COMMAREA designed to pass the same data. Because you are taking the time to change your application programs to exploit this new function, you should implement the best practices for channels and containers.

- Channels have several advantages over COMMAREAs and it pays to design your channels to make the most of these improvements.
- In previous releases, because the size of COMMAREAs is limited to 32 K and channels were not available, some applications used temporary storage queues (TSQs) to pass more than 32 K of data from one program to another. Typically, this involved multiple writes to and reads from a TSQ. If you migrate one of these applications to use channels, be aware that:
  - If the TSQ used by your existing application is in main storage, the storage requirements of the new, migrated application are likely to be similar to those of the existing application.
  - If the TSQ used by your existing application is in auxiliary storage, the storage requirements of the migrated application are likely to be greater than those of the existing application. This is because container data is held in storage rather than being written to disk.

Additional information can be found in the IBM Redbook *CICS Transaction Server V3R1 Channels and Containers Revealed*, SG24-7227.

7.3 Web Services

This section focuses on some of the architectural concepts that need to be considered for a Web Services project. We define and discuss service-oriented architecture (SOA) and the relationship between SOAs and Web Services.

We then take a closer look at Web Services, a technology that enables you to invoke applications using Internet protocols and standards. The technology is called Web Services because it integrates services (applications) using Web technologies (the Internet and its standards).

7.3.1 Service-oriented architecture

This section provides a short introduction to the key concepts of SOA. First we review why businesses are adopting an SOA approach.

There is a strong trend for companies to integrate existing systems to implement IT support for business processes that cover the entire business cycle. Today, interactions already exist using a variety of schemes that range from very rigid point-to-point electronic data interchange (EDI) interactions to open Web auctions. Many companies have already made some of their IT systems available to all of their divisions and departments, or even their customers or partners on the Web. However, techniques for collaboration vary from one case to another and are thus proprietary solutions. Systems often collaborate without any vision or architecture.

Thus, there is an increasing demand for technologies that support the connecting or sharing of resources and data in a very flexible and standardized manner. When technologies and implementations vary across companies and even within divisions or departments, unified business processes cannot be smoothly supported by technology. Integration has been developed only between units that are already aware of each other and that use the same static applications.

Furthermore, there is a need to structure large applications into *building blocks* that can be well-defined components within different business processes. A shift toward a *service-oriented* approach not only standardizes interactions, but also allows for more flexibility in the process. The complete value chain within a company is divided into small modular functional units, or services. A service-oriented architecture thus has to focus on how services are described and organized to support their dynamic, automated discovery and use.

Companies and their sub-units should be able to easily provide services. Other business units can use these services in order to implement their business processes. Ideally, this integration can be performed during the runtime of the system, not just at the design time.

Service-oriented architecture can help address these demands because, with SOA, the architecture makes no statements about the infrastructure or protocols it uses. Therefore, you can implement a service-oriented architecture using technologies other than Web technologies.

A service-oriented architecture enables a loose coupling between the participants. Such a loose coupling provides greater flexibility:

- Old and new functional blocks are encapsulated into components that work as services.
- Functional components and their interfaces are separated. Therefore, new interfaces can be plugged in more easily.
- Within complex applications, the control of business processes can be isolated. A business rule’s engine can be incorporated to control the workflow of a defined business process. Depending on the state of the workflow, the engine calls the respective services.

A service-oriented architecture has been used under various guises for many years. It can and has been implemented using a number of different distributed computing technologies, such as Common Object Request Broker Architecture (CORBA) and messaging middleware. However, the effectiveness of service-oriented architectures in the past has always been limited by the ability of the underlying technology to interoperate across the enterprise.

There are a number of reasons why Web Services technology is an ideal technology choice for implementing a service-oriented architecture:

- Web Services are based on standards, and standards promote interoperability. Interoperability is a key business advantage within the enterprise and is crucial in B2B scenarios.
- Web Services are widely supported across the industry. For the very first time, all major vendors are recognizing and providing support for Web Services. The Web Services Interoperability Organization (WS-I) is now working on a common implementation of Web Services.
- Web Services are platform neutral and language neutral. There is no bias for or against a particular hardware or software platform. Web Services can be implemented in any programming language or toolset. Consequently, there will be continued industry support for the development of standards and interoperability between vendor implementations.
- This technology provides a migration path to gradually enable existing business functions as Web Services.
- This technology supports synchronous and asynchronous, RPC-based, and complex message-oriented exchange patterns.

However, it is important to note that while Web Services technology is a prime choice for implementing a service-oriented architecture, many Web Services implementations are not service-oriented architectures. For example, the use of Web Services to connect two heterogeneous systems directly together is not an SOA. These uses of Web Services solve real problems and provide significant value on their own. They may or may not form the starting point of an SOA.

In general, an SOA has to be implemented at an enterprise or organizational level in order to harvest many of the benefits.

7.3.2 Web Services properties

All Web Services share the following properties:

- Web Services are self-contained.

  On the client side, no additional software is required. A programming language with XML and HTTP client support is enough to get you started. On the server side, an HTTP server and a SOAP server are required.

- Web Services are self-describing.

  A Web Service Description Language (WSDL) file provides all of the information you need to implement a Web Service as a provider or to invoke a Web Service as a requester.

- Web Services can be published, located, and invoked across the Web.

  The service requester uses established lightweight Internet standards such as HTTP to invoke the service provider. It leverages the existing infrastructure.

- Web Services are modular.

  Simple Web Services can be aggregated to form more complex ones, either using workflow techniques or by calling lower-layer Web Services from a Web Service implementation. Web Services can be chained together to perform higher-level business functions. This shortens development time and enables best-of-breed implementations.

- Web Services are language independent and interoperable.

  The client and server can be implemented in different environments. Any language can be used to implement Web Service clients and servers.

- Web Services are inherently open and standards-based.

  XML and HTTP are the major technical foundation for Web Services. A large part of the Web Service technology has been built using open-source projects. Therefore, vendor independence and interoperability are realistic goals.

- Web Services are loosely coupled.

  A service requester has to know the interface to a Web Service but not the details of how it has been implemented.

- Web Services provide programmatic access.

  The approach provides no graphical user interface. It operates at the code level.

- Web Services provide the ability to wrap existing applications.

  Existing applications can be integrated easily into the service-oriented architecture by implementing a Web Service as an interface to the application.

7.3.3 Soap

The way SOAP applications communicate when exchanging messages is often referred to as the message exchange pattern (MEP). The communication can be either one-way messaging, where the SOAP message only goes in one direction, or two-way messaging, where the receiver is expected to send back a reply.

Due to the characteristics of SOAP, it does not matter what technology is used to implement the client, as long as the client can issue XML messages. Similarly, the service can be implemented in any language, as long as it can process XML messages.

> **Note:** The authors of the SOAP 1.1 specification declared that the acronym SOAP stands for Simple Object Access Protocol. The authors of the SOAP 1.2 specification decided not to give any meaning to the acronym SOAP.

7.3.4 Support for SOAP

As discussed, application programs running in CICS TS 3.1 can participate in a heterogeneous Web Services environment as service requesters, service providers, or both, using either an HTTP transport or a WebSphere MQ transport. In this section we look at the history of the SOAP for CICS optional feature and how SOAP for CICS compares with the CICS TS 3.1 Web Services support.

History of SOAP for CICS

Early in 2003, IBM delivered its first support for SOAP in the CICS product when it announced that CICS SupportPac CA1M was available for use with CICS TS 1.3 and CICS TS 2.2. SupportPac CA1M was intended to be only a preview of how CICS might support SOAP. When IBM delivered CICS TS 2.3 late in 2003, it also delivered the optional SOAP for CICS feature for use in both CICS TS 2.2 and 2.3. Both SupportPacCA1M and the SOAP for CICS feature implement a pipeline approach to processing SOAP messages.

Pipelines

The SOAP for CICS feature consists of pipelines that support service providers and pipelines that support service requesters. A *pipeline* is a sequence of programs arranged so that the output from one program is used as input to the next.

A service provider pipeline is a pipeline of user-provided and system-provided programs that receives an inbound SOAP message, processes the contents, and sends a response.

A service requester pipeline is a pipeline of user-provided and system-provided programs that sends an outbound SOAP message, receives the response, and processes the contents of the response.

Limitations

The SOAP for CICS feature has the following limitations:

- You can have just one pipeline for all your service provider applications, and one for all your service requesters.
- You may define only one message handler per CICS region.

  > **Note:** The message handler is a user-written program that is typically used to extract information from the message or modify its contents.

- XML parsing and generation of the SOAP body must be either user-written or generated by a tool such as WebSphere Studio Enterprise Edition, or the newer WebSphere Developer for zSeries (WebSphere Developer).
- It supports only Version 1.1 of the SOAP protocol.
- It does not support either the WS-Security or the WS-Atomic Transaction specification.

Comparing CICS TS 3.1 with the SOAP for CICS feature

Table 7-1 summarizes some of the differences between the support for Web Services found in CICS TS 3.1 and the support for SOAP found in the SOAP for CICS feature.

Table 7-1 Comparison of CICS TS 3.1 Web Services with SOAP for CICS feature

Description	CICS TS 3.1	SOAP for CICS feature
Pipeline data passing mechanism	Channels and containers	BTS containers

Description	CICS TS 3.1	SOAP for CICS feature
Number of pipelines	Multiple per CICS region	One service requester pipeline per CICS region One service provider pipeline per CICS region
Number of message handlers	Multiple per each Web Service definition	One per CICS region
Pipeline container names	► DFHWS-APPHANDLER ► DFHWS-BODY ► DFHWS-DATA ► DFHWS-OPERATION ► DFHWS-PIPELINE ► DFHWS-SOAPACTION ► DFHWS-SOAPLEVEL ► DFHWS-TRANID ► DFHWS-URI ► DFHWS-USERID ► DFHWS-WEBSERVICE ► DFHWS-XMLNS ► DFHERROR ► DFHFUNCTION ► DFHHEADER ► DFHNORESPONSE ► DFHREQUEST ► DFHRESPONSE ► DFH-HANDLERPLIST ► DFH-SERVICEPLIST	► APP-HANDLER ► APP-NAMESPACES ► FAULT ► INPUT ► NAMESPACES ► OUTPUT ► PIPELINE-ERROR ► REQUEST-BODY ► RESPONSE-BODY ► SOAP-ACTION ► TARGET-TRANID ► TARGET-URI ► TARGET-USERID ► USER-CONTAINERS
SOAP protocol level	SOAP 1.1 and 1.2	SOAP 1.1
CICS resource definitions	► PIPELINE ► URIMAP ► WEBSERVICE	None

Description	CICS TS 3.1	SOAP for CICS feature
CICS API and SPI	► CREATE PIPELINE ► CREATE URIMAP ► CREATE WEBSERVICE ► INQUIRE WEBSERVICE ► INVOKE WEBSERVICE ► PERFORM PIPELINE SCAN ► SOAPFAULT ADD ► SOAPFAULT CREATE ► SOAPFAULT DELETE	None
XML parsing	CICS WSBIND file generated by either CICS Web Service assistant or WebSphere Developer	WebSphere Developer WebSphere Studio Enterprise Developer Write your own using Enterprise COBOL

Note: The SOAP for CICS feature can no longer be ordered with CICS TS 3.1. However, to assist with migration, customers who already have this feature will be able to use it with CICS TS 3.1. While this will continue to be supported, we recommend that, once you have migrated to CICS TS 3.1, you also move to Web Services to take advantage of its additional capabilities.

Using the Web Services functionality provided in CICS TS 3.1, CICS applications can participate in a heterogeneous Web Services environment as service requesters, service providers, or both. CICS's ability to act as a Web Services service provider means that it is relatively simple to transform an existing CICS application into a Web Service. Similarly, CICS's ability to act as a service requester means that a CICS application can use a Web Service provided by any external provider. To complement the CICS TS 3.1 Web Services support, a message-level security function is provided. This was delivered via the Service channel (PKxxxxx). This is a WS-Security compatible implementation for securing SOAP messages.

7.3.5 Web Services Assistant

The CICS Web Services Assistant is a set of batch utilities that can help you to transform existing CICS applications into Web Services and to enable CICS applications to use Web Services provided by external providers. It contains two utility programs:

- DFHLS2WS

  This generates a Web Service binding file from a language structure. This utility also generates a Web Service description.

- DFHWS2LS

  This generates a Web Service binding file from a Web Service description. This utility also generates a language structure that you can use in your application programs.

The assistant supports rapid deployment of CICS applications for use in service providers and service requesters, with minimum programming effort. When you use the Web Services Assistant for CICS, you do not have to write your own code for parsing inbound messages and for constructing outbound messages. CICS maps data between the body of a SOAP message and the application program's data structure.

CICS will, for the most part, generate and install the resource definitions automatically. You do have to define PIPELINE resources, but you can, in many cases, use one of the pipeline configuration files that CICS provides. These are:

- basicsoap11provider.xml

  This file defines the pipeline configuration for a service provider that uses the SOAP 1.1 message handler supplied by CICS.

- basicsoap11requester.xml

  This file defines the pipeline configuration for a service requester that uses the SOAP 1.1 message handler supplied by CICS.

The assistant can create a WSDL document from a simple language structure or a language structure from an existing WSDL document, and supports COBOL, C/C++, and PL/I. If you decide not to use the CICS Web Services Assistant, you will have to:

- Provide your own code for parsing inbound messages and constructing outbound messages (unless you use WebSphere Developer).
- Provide your own pipeline configuration file.
- Define and install your own URIMAP and PIPELINE resources.

For full details on how to implement CICS TS 3.1 Web Services we suggest that you refer to the IBM Redbook *Implementing CICS*, SG24-7206. There is also another IBM Redbook that covers Web Services in detail, *Application Development for CICS Web Services*, SG24-7126.

7.3.6 Web Services development approaches

Consider the following scenarios and read on to see how each of these situations is addressed using Web Services support in CICS:

- You have an existing application that you wish to expose as a Web Service.
- You wish to develop a new application and make it available as a Web Service.
- You want to access an existing Web Service, possibly on some other platform.

Bottom-up approach

In the first case, where we want to expose an existing application as a Web Service, we would most likely consider a bottom-up approach. We would start with the language structures for our current application and go through a process where we worked upward, developing the WSDL and other infrastructure elements until we had a fully fledged, published Web Service. This approach takes advantage of the CICS Web Services Assistant by using DFHLS2WS to generate WSDL based on the interface used by the existing application. This allows you to deploy an existing application as a Web Services application with very little effort. However, it is likely that there will still be potential to simplify the interface that is presented to the requester.

Meet-in-the-middle approach

The meet-in-the-middle approach can be used to ensure that the interface created is suitable for the requester. There may be an existing Web Service definition that we would like to use or we may just want to simplify the interface created for the requester using the bottom-up approach. We modify the WSDL and then create a wrapper program that will convert the existing application interface to and from the interface to the requester. The application is indirectly exposed as a Web Service with minimal, if any, CICS development work. Therefore, development costs are still low and we have the added advantage of being able to provide a more suitable interface to the requester.

Top-down approach

In the second case, where we wish to develop a new Web Service, we have some flexibility in our approach. However, it is likely that we would use a top-down approach and take advantage of the help and ease of implementation delivered by the Web Services Assistant.

In the third case, where we wish to access an existing Web Service, we would also consider a top-down approach. We would start with the WSDL, as published by the Web Service, and work downwards, first generating the language is most convenient for our environment. The CICS Web Services Assistant program DFHWS2LS would be used in this instance to generate the high-level language structure for use in your program. Thus, you can create a Web Services application based on the prescribed interface. The advantage of this approach is that the interface presented by the requester will be accurate. However, a small amount of development effort is required to incorporate the language structure generated by DFHWS2LS within you CICS application.

Table 7-2 summarizes when the three approaches can be used.

Table 7-2 The different approaches

Approach	**Application**	**WSDL**	**Type**
Bottom-up	Existing	New	Service provider
Top-down	New	Existing	Service provider
Top-down	New	New	Service provider
Top-down	New	Existing	Service requester
Meet-in-the-middle	Existing	Existing	Service provider

Alternatively, you can take complete control of processing your data by writing your own code to map between your application data and the message that flows between the service requester and provider. For example, if you want to use non-SOAP messages within the Web Services infrastructure, you can write your own code to transform between the message format and the format used by your application.

7.3.7 Web Services Atomic Transaction

CICS TS 3.1 Web Services supports the Web Services Atomic Transaction (WS-AT) specification. To understand what this really means, in this section we look at an example of a classic transaction and map this to the analogous WS-Atomic Transaction, highlighting the differences. This illustrates the advantages that support for WS-Atomic Transaction brings. In fact, the differences are almost cosmetic from the outside as they concern how the entities communicate with each other, not the substance of what they communicate. However, as we will see below, it is these differences in how the entities communicate that have a big impact on flexibility and interoperability.

A classic transaction

We begin by describing an example of a classic transaction. We borrow freely from the paper *Tour Web Services Atomic Transaction operations: Beginner's guide to classic transactions, data recovery, and mapping to WS-Atomic Transactions,* which Thomas Freund and Daniel House published on September 2, 2004, on the IBM developerWorks Web site at:

http://www-128.ibm.com/developerworks/webservices/library/ws-introwsat

Not losing money is quite important. Just ask Waldo. Waldo's situation typifies the need for a transaction. Waldo uses a Web browser or an Automatic Teller Machine (ATM) to move money from one account to another account. These accounts may be in different branches of the same financial institution, or they may be in different institutions.

It is never acceptable to Waldo for his money to disappear. Should Waldo ever doubt the safety of his money, he would probably switch financial institutions.

Waldo's money is represented by data in two databases that cooperate to ensure that the data they contain is always in a known and consistent state. That is, these two databases allow actions or tasks between them to be within a common activity or work scope, as shown in Figure 7-12. In other words, a single transaction can manipulate data in both databases and *something* will guarantee that only one of two possible outcomes occurs: *all* the changes are successfully made or *none* of them are made.

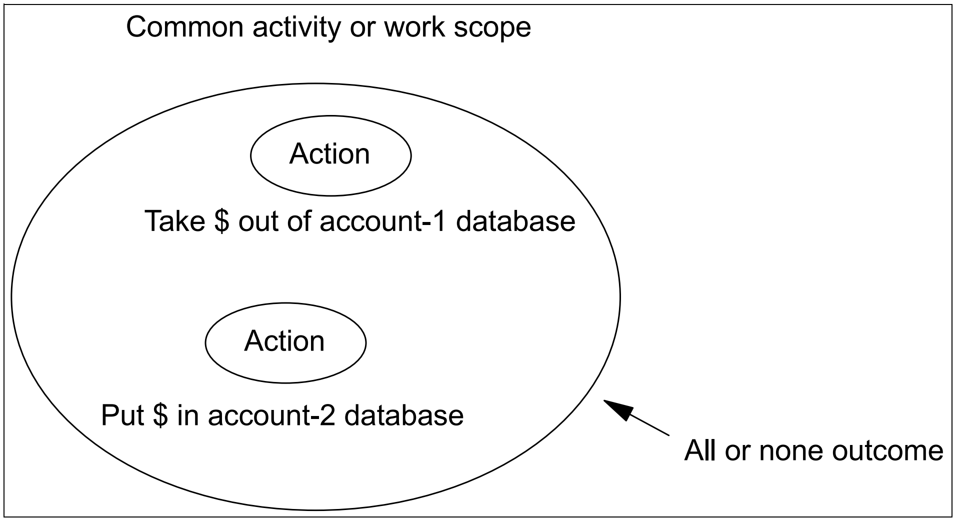


Figure 7-12 Common activity encompasses various recoverable actions

The *something* that guarantees the common outcome of all the actions is a protocol supported by both databases, and supporting middleware. The protocol the databases use to keep data (such as Waldo's balances) coordinated is called *two-phase commit*, or simply 2PC. Our example uses a common variation of 2PC called *presumed abort*, where the default behavior in the absence of a successful outcome is to roll back or undo all the actions in the activity.

From a programming perspective, there are different ways to specify that multiple actions should be within the scope of a single transaction. One particularly clear way to specify transactional behavior is shown in Example 7-3. The code is the small piece of logic running somewhere behind the ATM Waldo is using—perhaps in the data center of one of the financial institutions involved.

Example 7-3 Pseudo-code for Waldo's transaction

```
TransferCash(fromAcct, toAcct, amount)
    BeginTransaction
    fromAcct = fromAcct - amount
    toAcct   = toAcct + amount
    CommitTransaction
Return
```

For our simple purposes, a recoverable action is anything that modifies protected data. For example, taking money out of one of Waldo's accounts (fromAcct = fromAcct - amount) is a recoverable action that can be reversed up to the end of the transaction. A classic transaction, then, is just a grouping of recoverable actions, the guaranteed outcome of which is that either all the actions are taken, or none of them is taken (see Figure 7-13 on page 188).

In Waldo's case, his transaction is composed of two actions: taking money out of one account and putting money into another account. It is okay for both of these actions to occur, and it is even okay if neither of these actions occurs. It is never okay for one action to occur without the other also occurring, which would result in corrupt data and either Waldo's net worth or the bank's assets disappearing or appearing from nowhere. Hence, both actions need to be within a single transaction with a single outcome: either both actions occur (a commit outcome) or neither action occurs (a rollback outcome).

Assuming that no errors happen, the code in Example 7-3 shows that a commit outcome is desired. The code could just as easily have specified rollback instead of commit (for when Waldo presses the Cancel key on the ATM), which means reverse all actions in the transactional work scope (between beginning and end). The transaction monitor, which is the underlying middleware helping the code in Example 7-3 support transaction processing, would automatically specify rollback if the program suffered an unhandled exception. Such an automatic rollback on the part of the transaction monitor is a protection mechanism to make sure that data is not corrupted (for example, even if the ATM application fails unexpectedly, the middleware will clean up and guarantee the outcome).

Now let us see how one common variant of 2PC (presumed abort) can be used to effect Waldo's transaction and move money from one account to another in a recoverable way. A key part of this illustration is to see that no matter what kind of failure occurs, data integrity is preserved and Waldo remains a loyal customer.

Figure 7-13 shows Waldo's transaction on a timeline with all of the interacting components needed to execute the logic shown in Example 7-3 on page 187.

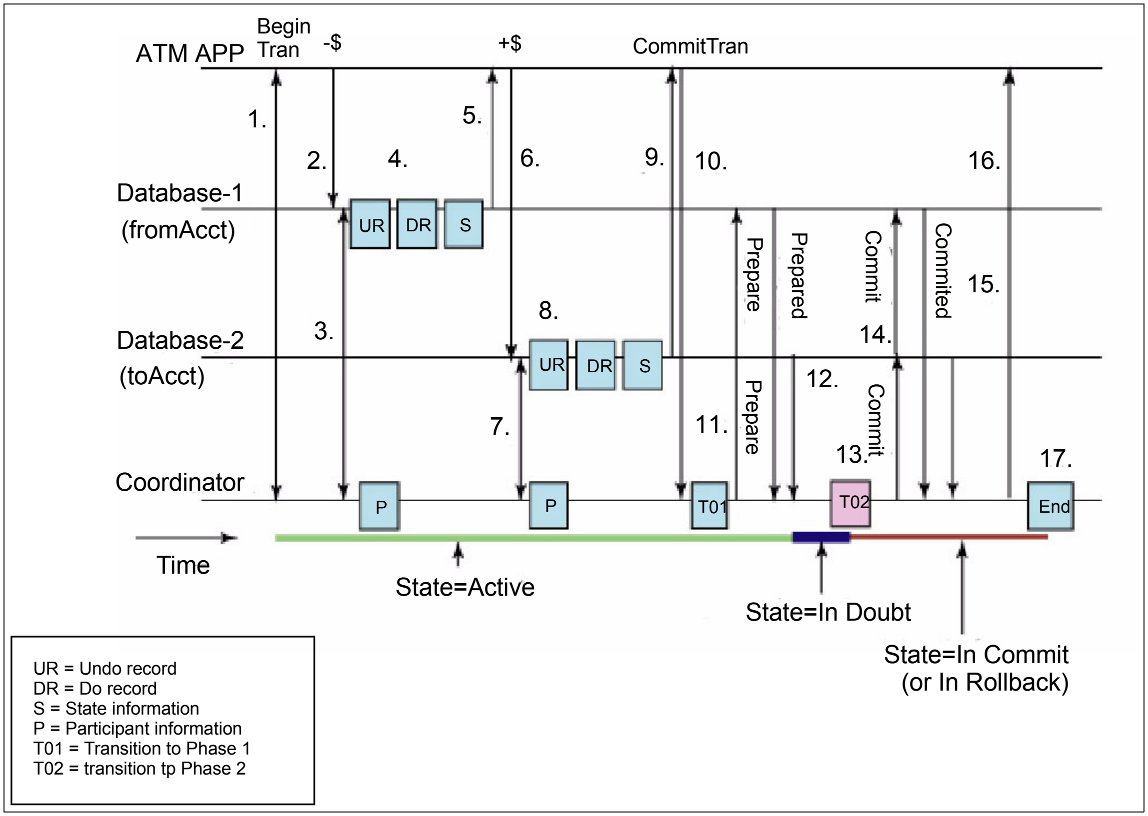


Figure 7-13 Behind the scenes of Waldo's ATM transaction

The top line represents the ATM application itself.

The next two lines represent the account databases that the application manipulates. The databases will be transactional participants.

The next line is a transactional coordinator, or middleware that will orchestrate the 2PC protocol.

The line at the very bottom indicates the state of Waldo's transaction at different points in time. The state of the transaction dictates recovery processing in the event of a failure.

The lines for Database-1, Database-2, and Coordinator represent both time (flowing left to right) and also key records recorded onto a recovery log. These records include images of the data before it is modified (Undo records), images of the data after it has been modified (Do records), and state information. The recovery log is used to ensure data integrity during recovery processing.

Now let us walk through Waldo's transaction. Below when we talk about *the ATM application*, this means either the application itself or some middleware supporting the application. For example, when we say that the application begins a transactional scope, it could be that middleware begins the transactional scope on behalf of the application.

Here we explain the numbered steps shown in Figure 7-13 on page 188:

1. The ATM application indicates the beginning of a transactional scope. The coordinator creates a context for this transaction. The context includes a unique identifier and some other information about the transaction. Importantly, this transaction context flows back to the application. The context flows with other interactions between the application and resource managers. It is the context that helps glue together a whole set of actions into one transactional activity.
2. The application takes money out of Database-1. The context (from step 1) is inserted into this flow.
3. Database-1 sees the request for action, but also sees the transactional context. Database-1 uses this context to contact the transactional coordinator and register interest in this transaction or activity (so that the coordinator will help Database-1 through 2PC processing later to guarantee a commit or rollback outcome of all actions). The coordinator remembers that Database-1 is a participant in the transaction.
4. Database-1 looks at the request to modify recoverable data. It writes records to a recovery log, plus transaction state information. One record describes the database change to be made if the decision later is to commit (the Do record). The other record describes the database change to be made if the decision is to roll back (the Undo record).
   - In this case, the Undo record says *make Waldo's balance = x* and the Do record says *make Waldo's balance = x - $*. (*x* is the amount of the balance before this transaction ever started and *$* is the amount to transfer). Notice that we are only looking at the recovery log, not database files.
   - The Do records are not strictly required if Database-1 makes database file updates when the application requests it to, instead of waiting. However, waiting to write the data can have advantages for performance and concurrency. In addition, the Do records may be used for audit or other advanced reasons. Since they are so useful, our example databases use them.
5. Return to the application.

6. Similarly to step 2, the application makes a request to manipulate the other database, Database-2. The application wants to add in the amount taken out of Database-1.
7. Database-2 registers interest in the transaction with the coordinator the same way Database-1 did. The coordinator remembers that Database-2 is a participant in the transaction.
8. Database-2 writes Undo and Do records and state information to its recovery log, again just as Database-1 did.
9. Return to the application.
10. The application chooses to commit the transaction. The coordinator now takes over. When *commit* is received, the coordinator writes a log record indicating that phase 1 of 2PC has begun.
11. In phase 1, the coordinator goes down the list of all participants (Database-1 and Database-2 in this example) who expressed interest in this transaction, asking each one to *prepare*. Prepare means get ready to receive the order to either commit or rollback.
12. Database-1 and Database-2 both respond with *prepared*, meaning that they are ready to be told the final outcome (commit or rollback all the changes made) and support it.
    - They must have committed something (at least on their logs) by this point, because responding *prepared* means they guarantee being able to commit or roll back when told. Actions up to this point were just tentative.
    - If either database had some kind of failure preparing, it would respond `Aborted` instead of `Prepared`, and the coordinator would broadcast `Rollback` to all participants.
13. The coordinator forces a log record indicating a transition to phase 2 (T02).
    - Once this record is hardened on a log, we know and have recorded that:
        - All participants are prepared to go either way (commit or rollback).
        - The ultimate outcome of the transaction is known (commit in our example).
        - The outcome is guaranteed by recovery processing.
    - If this record fails to make it to the log for any reason, the ultimate outcome will be to roll back (we are using presumed abort in this example). The recovery processing will enforce the outcome.
14. The coordinator informs each participant that the decision is to commit the changes. The participants can then do whatever they need to do, such as perhaps writing the results to the real database data.

15. The participants return to the coordinator with `Committed`. Once the coordinator knows that all the participants acknowledged the *commit* order with `Committed`, it can forget about this transaction because the transaction was acknowledged by all to be done.

16. Return to the application.

17. At some point, since it knows the participants have succeeded in the 2PC flow by acknowledging the common outcome, the coordinator writes an *end* indicator on its log.

Mapping from classical transactions to WS-ATs

In Figure 7-13 on page 188 we did not mention how Database-1 contacted the coordinator, nor did we specify how the application called the databases. In fact, we did not specify the mechanisms for anything to contact anything else. In the past, these were mostly non-universal mechanisms that sometimes only worked between certain combinations of entities (applications, resource managers, and coordinators or transaction monitors).

The combination of Web Services, WS-Coordination (WS-C), and WS-Atomic Transaction (WS-AT) maps all of the flows shown in Figure 7-13 on page 188 and specifies precise communications mechanisms for achieving the same results. However, instead of only working between certain combinations, the Web Services based flows can work with just about anything.

In Figure 7-14 the classic flows are converted to Web Services as follows. Significantly changed steps are described below. As before, when we say *application*, this means the application or helper middleware. Likewise, when we say *database*, it might mean the actual database or some helper middleware.

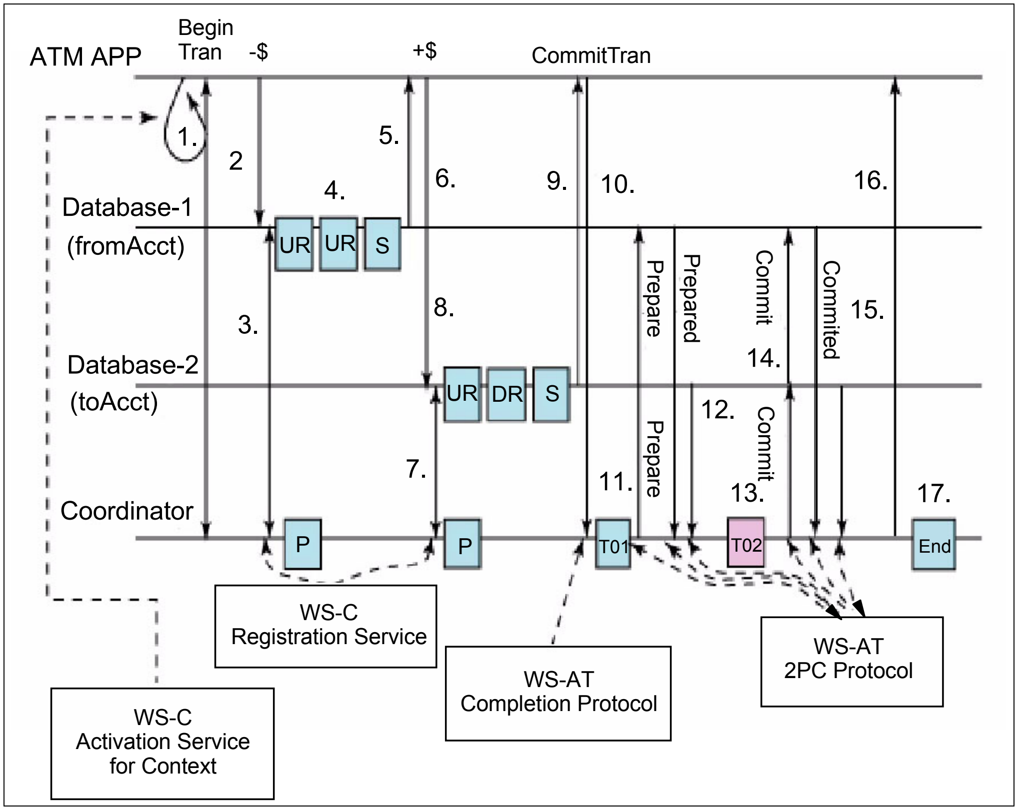


Figure 7-14 Waldo's transaction revisited

The steps are:

1. The application uses the activation service defined in WS-C to obtain a transactional context.
2. The application invokes a Web Service exposed by Database-1 (alternatively, exposed by an application server that then talks to Database-1) to subtract money from Waldo's balance. The context flows along with the Web Service invocation, although the application is not aware of that.
3. Database-1 uses information in the context to invoke the registration service defined in WS-C to register interest in this transaction.
4. No change.

5. No change.
6. The application invokes a Web Service exposed by Database-2 to add money to Waldo's balance. Just like in step 2, the context flows along with the Web Service invocation.
7. Just like step 3, Database-2 uses information in the context to invoke the registration service and register interest in this transaction.
8. No change.
9. No change.
10. The application uses the completion protocol defined in WS-AT to indicate that it wishes to commit the transaction.
11. In steps 11 through 15 the databases and the coordinator participate in 2PC flows, as defined in the WS-AT 2PC Protocol.

From Figure 7-14 on page 192 it is clear that atomic transactions using Web Services (WS-C and WS-AT) are substantially the same as without Web Services (Figure 7-13 on page 188, for example). The primary differences are almost cosmetic from the outside and involve *how entities communicate with each other, not the substance of what they communicate*. However, these differences in how the entities communicate have a big impact on flexibility and interoperability.

You can achieve universal interoperability with Web Services, because instead of changing resource manager X to interoperate with transaction monitor Y, you can change both X and Y to use Web Services and then interoperate with many other resource managers and transaction monitors. So instead of two-at-a-time interoperability, or interoperability only within a specific kind of domain, n-way universal interoperability is possible.

Recovery processing using Web Services between the interested parties is the same as before Web Services. Resource managers are the only ones who know their resources and how to commit them or roll them back.

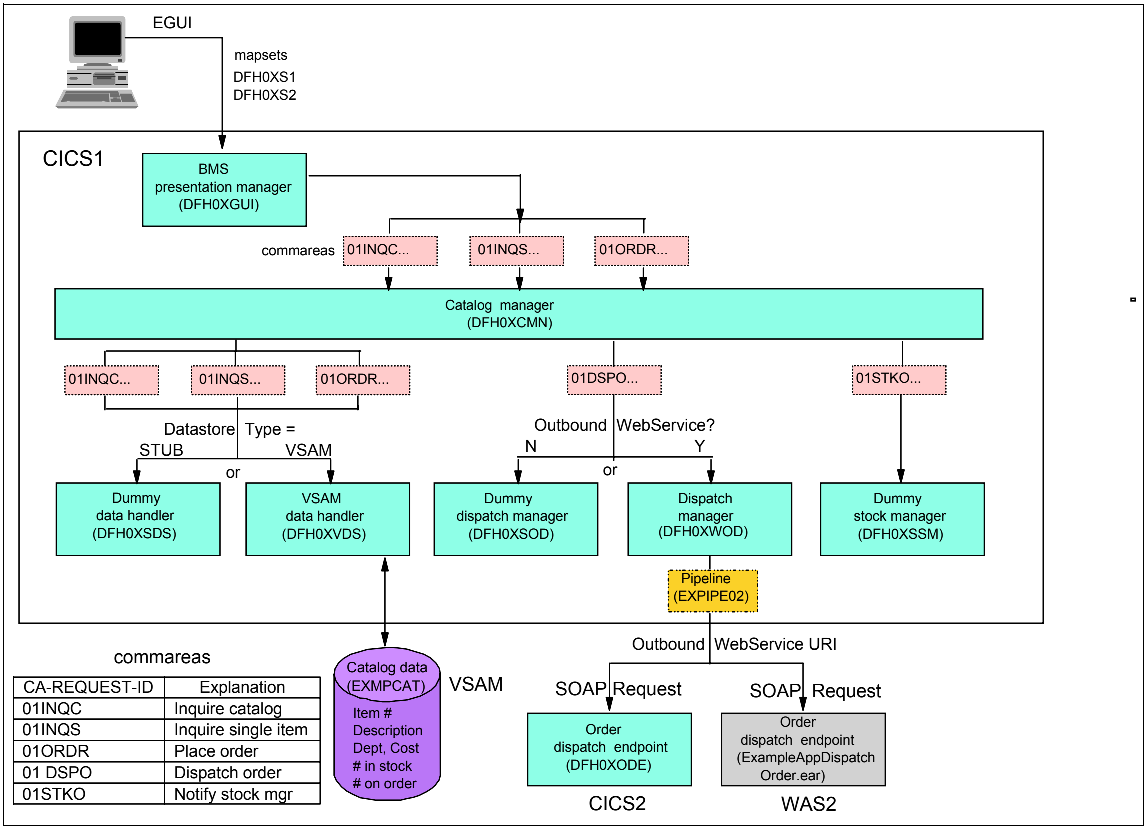
As an example, suppose that Database-1 fails between steps 5 and 6 in Figure 7-14 on page 192. Database-1 comes back up and, just like before Web Services, it reads its log, notices that it has an incomplete transaction, and realizes that it needs to contact the coordinator. Information about how to contact the coordinator is in the state saved on its recovery log. With Web Services it will be an endpoint reference. Database-1 contacts the coordinator at that endpoint reference with a message defined in WS-AT called `Replay`. Replay causes the coordinator to resend the last protocol message to Database-1, which lets Database-1 deduce the transaction state and then apply the appropriate recovery rule. In our example the coordinator tells Database-1 that it has no knowledge of this transaction. Database-1 therefore applies its Undo record, making the data consistent again.

> **Important:** WS-AT is a two-phase commit transaction protocol that is suitable for short duration transactions only. WS-AT is well suited for distributed transactions within a single enterprise, but is it is generally not recommended that WS-AT transactions be distributed across enterprise domains. Inter-enterprise transactions typically require a looser semantic than two-phase commit.

7.3.8 CICS Web Services catalog sample application

A new sample application is provided that illustrates how to code and implement a Web Services provider and requester application, together with a range of other functions including COMMAREAs and channels, as an example of suggested best practices using the new functions of CICS TS 3.1. This sample application is a catalog management purchase order style COBOL application that accesses an order catalog stored in a VSAM file. The CICS catalog example application is a working COBOL application that is designed to illustrate best practice when connecting CICS applications to external clients and servers.

It demonstrates how you can use SOAP and the Web Services to make existing, CICS-controlled information available to SOA service requestors. It is a simple application that provides the functions to list details of an item in the catalog and then select a quantity of that item to order. The catalog is then updated to reflect the new stock levels. If selected in the application configuration, an outbound Web Service call is then made to an external dispatch manager Web Service. Figure 7-15 shows a CICS implementation of this.



CA-REQUEST-ID	Explanation
01INQC	Inquire catalog
01INQS	Inquire single item
01ORDR	Place order
01 DSPO	Dispatch order
01STKO	Notify stock mgr

Figure 7-15 Catalog sample overview

8

Migrating CICS TS 2.3 CSD to CICS TS 3.1 CSD

This chapter describes the steps required to migrate an application region from a CICS TS 2.3 CSD file to a CICS TS 3.1 CSD file, and is shown in Figure 8-1 on page 198. In particular we look at ways the CICS Tools Interdependency Analyzer and CICS Configuration Manager can be used to assist migration.

8.1 The environment

In this migration scenario we use the environment shown in Figure 8-1. We demonstrate how to migrate resource definitions from a CSD in a CICS TS 2.3 region to a CSD in a CICS TS 3.1 region using CICS IA and CICS CM tools.

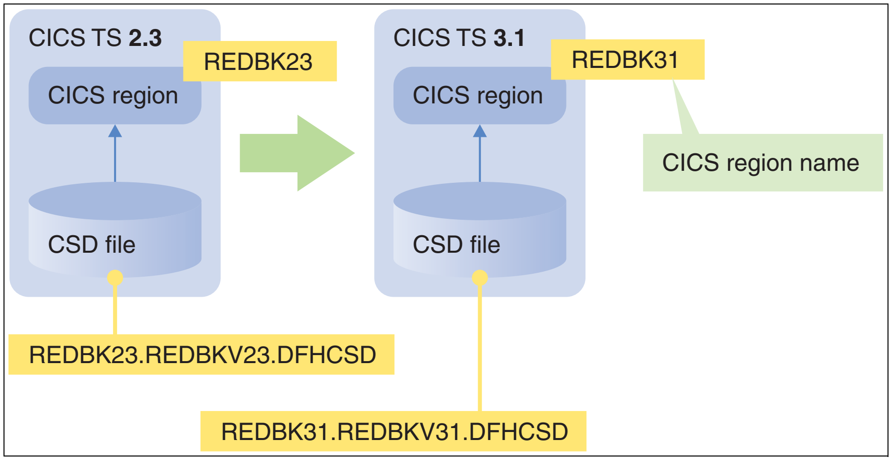


Figure 8-1 Migration of TS 2.3 CSD to TS 3.1 CSD

This chapter discusses the following:

- Configuring CICS TS 3.1
- Identifying application resources using CICS IA
- Migrating CSD resources using CICS CM
- Verifying migration changes using CICS CM and CICS IA
- Installing CSD resources using CICS CM

This chapter does not discuss the use of CICS PA for performance analysis on the CICS TS regions before or after migration. This is discussed in Chapter 2, “Overview of CICS PA” on page 29.

8.2 Configuring CICS TS 3.1

This section describes the steps involved in configuring CICS TS 3.1. For a complete description refer to the *CICS TS 3.1 Migration guide*, SC34-6458, and the *CICS TS 3.1 Installation Guide*, SC34-6425. The following steps are discussed in more detail:

- Check minimal software prerequisites.
- Create data sets.
- Review SIT parameters.
- Change JCL.
- Post-installation requirements.

8.2.1 Check minimal software prerequisite

The following are the minimal software requirements for CICS TS 3.1:

- z/OS V1.4 or later
  - CICS will not initialize unless the minimum prerequisite level of the operating system is installed.
  - Some components of CICS are installed in PDSE and HFS files.
  - The OMVS address space, UNIX Systems Services, must be active in full function mode during the install process.
  - The jobs to create the HFS files and directories will require superuser authority.
  - LE library SCEERUN must be available to CICS during CICS initialization.
  - z/OS Conversion Services must be enabled.
- IBM SDK for z/OS, Java 2 Technology Edition Version 1.4

  This must be at the 1.4.2 level. PTF UQ90449.

8.2.2 Create data sets

The following CICS-supplied sample jobs, found in hlq.SDFHINST, can be used to create the required CICS TS 3.1 data sets:

DFHCOMDS Delete and recreate data sets common to all CICS regions.

DFHDEFDS Delete and recreate data sets for each individual CICS region.

DFHCMACI Delete and recreate the CICS messages data set DFHCMACD. This data set is used by the CICS messages facility (CICS-supplied transaction CMAC).

The *CICS TS 3.1 Installation Guide* refers to the use of DFHISTAR to create different copies of the above sample jobs. For the purposes of this book we took copies of the above samples and manually changed them to conform to the appropriate local naming standards. Each job is described below.

DFHCOMDS

In this step we create and initialize a new TS 3.1 CSD file. See Figure 8-2.

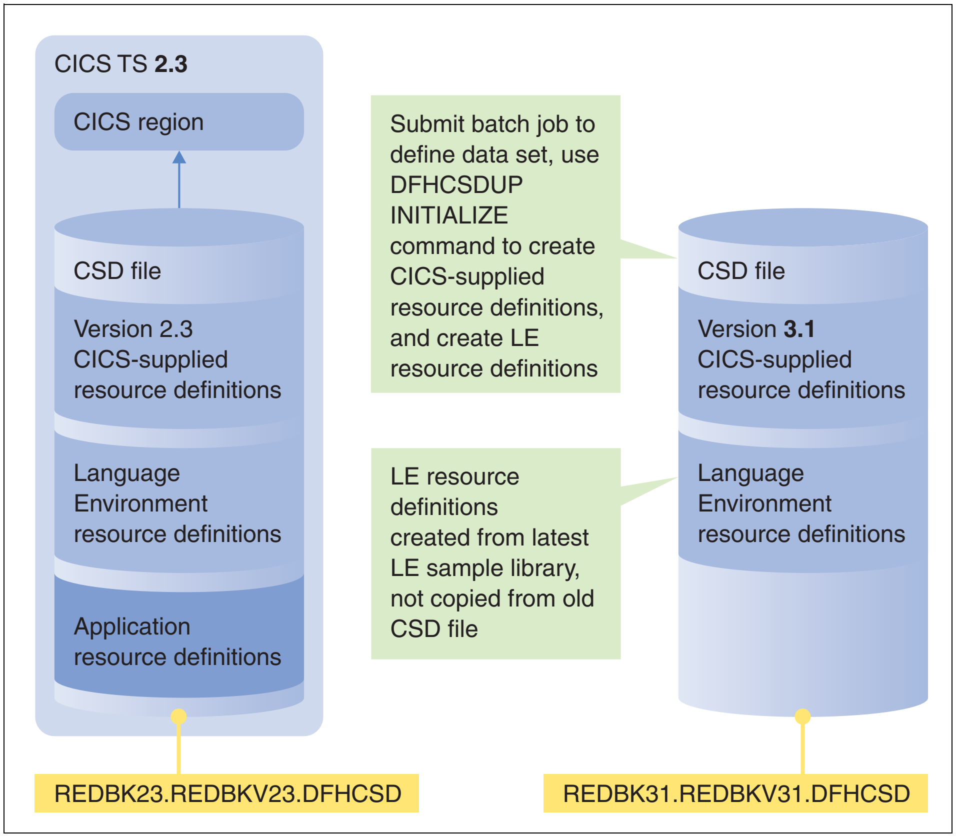


Figure 8-2 CSD initialization

We then migrate application resource group REDBOOK from the TS 2.3 CSD to the TS 3.1 CSD. See Figure 8-3.

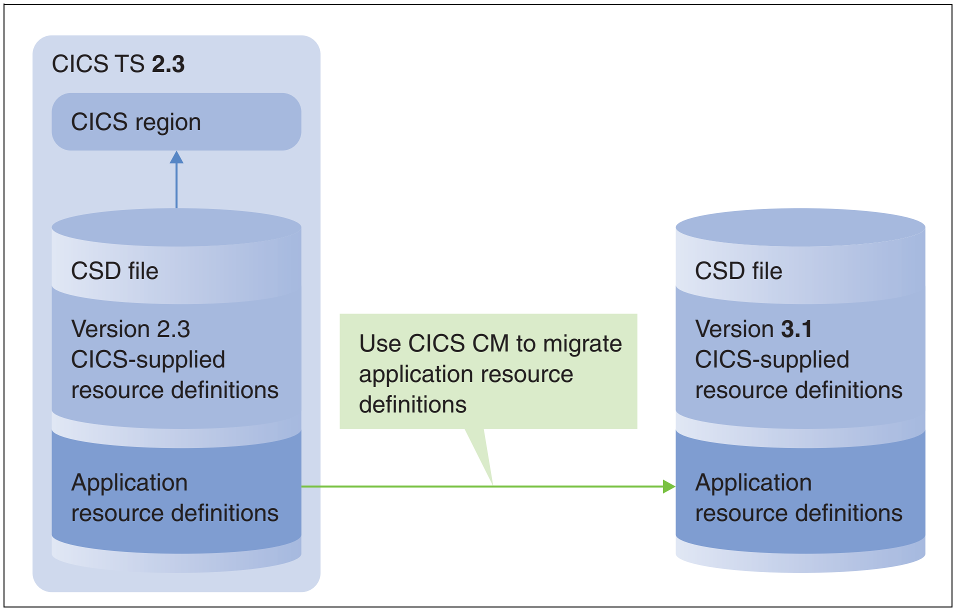


Figure 8-3 CICS CM migration

The following steps were performed:

1. We copied hlq.SDFHINST(DFHCOMDS) to AYS.REDBOOK.JCL.
2. We edited the copied member.
3. We added the appropriate jobcard.
4. We deleted all references to JES3-ONLY.
5. We issued the following change commands:
   - c '&DSINDEX' REDBK31.REDBKV31 all
   - c '&THLQ.CICS.&TQUAL' 'CICS.V640.CICS' all
   - c '&SCEESAMP' 'CEE.SCEESAMP' all
   - c ‘&COPYPRG’ IEBCOPY all
6. We deleted the UNIT and VOL information for the OUT DD in the DEFSYSIN step, as this data set is SMS managed.
7. We submitted the modified job.

The INITCSD step will give a return code of 4 and all other steps should complete with a return code of 0. After successful execution the data sets shown in Example 8-1 will have been created.

Example 8-1 DFHCOMDS data sets

```
REDBK31.REDBKV31.CICSONC.RESOURCE
REDBK31.REDBKV31.CICSONC.RESOURCE.DATA
REDBK31.REDBKV31.CICSONC.RESOURCE.INDEX
REDBK31.REDBKV31.DFHCSD
REDBK31.REDBKV31.DFHCSD.DATA
REDBK31.REDBKV31.DFHCSD.INDEX
REDBK31.REDBKV31.SYSIN
```

DFHDEFDS

The following steps were performed.

1. Copied hlq.SDFHINST(DFHDEFDS) to AYS.REDBOOK.JCL
2. Edited copied member
3. Added appropriate jobcard
4. Deleted all references to JES3-ONLY
5. Issued the following change commands:
   - c '&DSINDEX' REDBK31.REDBKV31 all
   - c '&THLQ.CICS.&TQUAL' 'CICS.V640.CICS' all
   - c 'CICS&REGNAME.' '' all
   - c 'cntl..' '' all
   - c '&dsvol' 'DEV004' all
   - c '&DSUNIT' 3390 all
6. Submitted the modified job

The job should complete with a 0 return code and after execution the additional data sets shown in Example 8-2 will have been created.

Example 8-2 DFHDEFDS data sets

```
REDBK31.REDBKV31.BANKACCT
REDBK31.REDBKV31.BANKACCT.DATA
REDBK31.REDBKV31.BANKACCT.INDEX
REDBK31.REDBKV31.DFHADEM
REDBK31.REDBKV31.DFHADEM.DATA
REDBK31.REDBKV31.DFHADEM.INDEX
REDBK31.REDBKV31.DFHAUXT
REDBK31.REDBKV31.DFHBRNSF
REDBK31.REDBKV31.DFHBRNSF.DATA
REDBK31.REDBKV31.DFHBRNSF.INDEX
REDBK31.REDBKV31.DFHBUXT
```

```
REDBK31.REDBKV31.DFHDMPA
REDBK31.REDBKV31.DFHDMPB
REDBK31.REDBKV31.DFHDPFMB
REDBK31.REDBKV31.DFHDPFMB.DATA
REDBK31.REDBKV31.DFHDPFMB.INDX
REDBK31.REDBKV31.DFHDPFMP
REDBK31.REDBKV31.DFHDPFMX
REDBK31.REDBKV31.DFHDPFMX.DATA
REDBK31.REDBKV31.DFHDPFMX.INDEX
REDBK31.REDBKV31.DFHEJDIR
REDBK31.REDBKV31.DFHEJDIR.DATA
REDBK31.REDBKV31.DFHEJDIR.INDEX
REDBK31.REDBKV31.DFHEJOS
REDBK31.REDBKV31.DFHEJOS.DATA
REDBK31.REDBKV31.DFHEJOS.INDEX
REDBK31.REDBKV31.DFHGCD
REDBK31.REDBKV31.DFHGCD.DATA
REDBK31.REDBKV31.DFHGCD.INDEX
REDBK31.REDBKV31.DFHHTML
REDBK31.REDBKV31.DFHINTRA
REDBK31.REDBKV31.DFHINTRA.DATA
REDBK31.REDBKV31.DFHLCD
REDBK31.REDBKV31.DFHLCD.DATA
REDBK31.REDBKV31.DFHLCD.INDEX
REDBK31.REDBKV31.DFHLRQ
REDBK31.REDBKV31.DFHLRQ.DATA
REDBK31.REDBKV31.DFHLRQ.INDEX
REDBK31.REDBKV31.DFHTEMP
REDBK31.REDBKV31.DFHTEMP.DATA
REDBK31.REDBKV31.FILEA
REDBK31.REDBKV31.FILEA.DATA
REDBK31.REDBKV31.FILEA.INDEX
```

DFHCMACI

This step is optional and is only required if you intend to use the CICS-supplied transaction CMAC. We did execute this and the following steps were performed:

1. Copied hlq.SDFHINST(DFHCMACI) to AYS.REDBOOK.JCL
2. Edited copied member
3. Added appropriate jobcard
4. Deleted all references to JES3-ONLY
5. Issued the following change commands:
   – c '&DSINDEX' REDBK31.REDBKV31 all
   – c '&THLQ.CICS.&TQUAL' 'CICS.V640.CICS' all

– c '&cmacvol' 'DEV004' all

6. Submitted the modified job

The job should complete with a 0 return code and after execution the data sets shown in Example 8-3 will have been created.

Example 8-3 DFHCMACI data sets

```
REDBK31.REDBKV31.DFHCMACD
REDBK31.REDBKV31.DFHCMACD.DATA
REDBK31.REDBKV31.DFHCMACD.INDEX
```

8.2.3 Review SIT parameters

Before starting the new CICS TS 3.1 region you will need to review the SIT parameters, especially if they have been copied from an existing lower CICS release level, as we did. The review of the SIT parameters is discussed in 6.4, “New SIT parameters for CICS TS 3.1” on page 117.

8.2.4 Change JCL

Rather than changing the existing CICS TS 2.3 JCL, we copied it to a suitably named CICS TS 3.1 JCL member and made the necessary changes to it. In essence the changes required are to update the JCL to reflect the data set names created in “DFHDEFDS” on page 203. The changes we made were:

- Changed ‘REDBK23.REDBKV23’ to ‘REDBK31.REDBKV31’
- Changed ‘CICS.V630.CICS’ to ‘CICS.V640.CICS’
- Changed ‘CICS.V630.CPSM’ to‘CICS.V640.CPSM’

Our CICS TS 3.1 JCL referenced the newly created SYSIN data set (see below), and so we created the REDBKV31 member by copying the corresponding CICS TS 2.3 member. Having copied the member, we reviewed the SIT parameters contained in the SYSIN member, as discussed in 8.2.3, “Review SIT parameters” on page 205. For us this highlighted the need to update parameter SSLTCBS to MAXSSLTCBS.

```
//SYSIN    DD DISP=SHR,DSN=REDBK31.REDBKV31.SYSIN(REDBKV31)
```

We also changed the application load library from ‘REDBK23.APPL.LOADLIB’ to ‘REDBK31.APPL.LOADLIB’.

8.2.5 Post-installation requirements

The *CICS TS 3.1 Installation Guide* gives details on post-installation requirements that should be reviewed. Below is a summary of the steps to follow to enable your z/OS environment to support CICS:

1. APF-authorize SDFHAUTH. Define the CICSTS31.CICS.SDFHAUTH library as an APF-authorized library.
2. Authorize CICS regions user IDs. Authorize to RACF each CICS region user ID to permit access to the required MVS resources.
3. Add SDFHLINK to LNKLST. Include the CICS linklist library, CICSTS31.CICS.SDFHLINK, in the MVS LNKLST concatenation.
4. Define CICS as a subsystem. Define CICS as an MVS subsystem if you intend to use multiregion operation (MRO), the CICS console message-handling facility, or MVS workload management.
5. Install the CICS Type 3 SVC. Define the DFHCSVC module to MVS. Schedule an IPL to install the CICS SVC routine, DFHCSVC, and other CICS-required modules in the MVS link pack area (LPA).
6. Review the requirement for HPO. Ensure that the DFHHPSVC module is included in the MVS nucleus if you are going to use the VTAM high-performance option (HPO), and ensure that the HPO SVC is defined as a Type 6 SVC in the appropriate MVS IEASVCxx PARMLIB member.
7. Define VTAM APPLs for CICS. Define to VTAM each CICS region that requires VTAM support (for example, all your terminal-owning regions) and also ensure that any VTAM terminal definitions are properly specified for connection to CICS.

8.3 Identifying application resources using CICS IA

This section describes the steps required to identify application resources that need to be reviewed before migration to CICS TS 3.1. It discusses the following:

- Using the CICS IA Scanners
- Using the CICS IA Collector
- Identifying COBOL/VS programs
- Identifying non threadsafe programs
- Identifying applications to be migrated

To install and customize CICS IA V2R1 refer to Appendix A, "CICS IA installation and customization" on page 405.

8.3.1 Using the CICS IA Scanners

In CICS IA V2R1 you have two load module scanners:

- The original load module scanner that reports on possible affinities and dependencies in a program. It also reports the program language. It produces a batch report and populates two DB2 tables:
  - CIU4_SCAN_SUMMARY
  - CIU4_SCAN_DETAIL
- A new CSECT scanner that reports on linkage and compiler attributes of all CSECTs within a program. It produces a batch report and populates two DB2 tables:
  - CIU4_CSECT_INFO
  - CIU4_PROGRAM_INFO

Running the load module scanner

To run the load module scanner we must first edit and run the customized job CIUJCLTS to produce a summary report.

The job appears in Example 8-4. The values that require editing in this job are:

scan The load library to be scanned. We scan REDBK23.APPL.LOADLIB.

ciudet The output data set to be used as input to the detailed job CIUJCLTD. We use REDBK23.APPL.DETMODS.

Example 8-4 CIUJCLTS - IA summary scanner JCL

```
//CIUJCLTS JOB USER=EYJ,NOTIFY=EYJ,
//         CLASS=A,MSGCLASS=Y,REGION=0M
//*******************************************************************
//*                                                                 *
//* JCL NAME = CIUJCLTS                                             *
//*                                                                 *
//* DESCRIPTIVE NAME = IBM CICS INTERDEPENDENCIES UTILITY           *
//*                    RUN SCANNER IN SUMMARY MODE WITH DB2 OUTPUT  *
//*                                                                 *
//* CHANGES TO BE MADE                                              *
//*                                                                 *
//*   1) CHANGE THE JOB CARD TO SUIT YOUR SYSTEM CONVENTIONS        *
//*   2) CHANGE THE FOLLOWING PARAMETERS:-                          *
//*   DB2P                                                          *
//*   THE DB2 ID                                                    *
//*   CIU                                                           *
//*   DATASET HLQ FOR CIU PRODUCT                                   *
//*   DSN710                                                        *
//*   DATASET HLQ FOR DB2 SDSNLOAD and RUNLIB.LOAD                  *
//*   _scan_                                                        *
//*   CICS LOAD DATASET TO BE SCANNED                               *
//*   _ciudet_                                                      *
//*   Output dataset created by SCANNER SUMMARY JOB                 *
//*   3) EDIT THE MEMBER CIUDB2BT IN                                *
//*      REDBK23.MIG23T31.SCIUCLIS                                  *
//*      AND CHANGE THE FOLLOWING:-                                 *
//*   CIU                                                           *
//*   DATASET HLQ FOR CIU PRODUCT                                   *
//*                                                                 *
//*******************************************************************
//* FUNCTION =                                                      *
//*    Sample JCL to run the Load Module Scanner component of the   *
//*    Interdependencies Utility (SUMMARY mode, DB2 output).        *
//*******************************************************************
//SCAN     EXEC PGM=IKJEFT1B,DYNAMNBR=20,
```

```
//             PARM=('%CIUDB2BT','SYS(DB2P)','PROG(CIULMS)',
//             'PLAN(CIUBTCH4)','PARM(''$SUMMARY,DETAILMODS,TABLE'')')
//STEPLIB  DD DSN=CIU.SCIULOAD,DISP=SHR
//         DD DSN=CIU.SCIULODE,DISP=SHR
//         DD DSN=DSN710.SDSNLOAD,DISP=SHR
//SYSPROC  DD DSN=REDBK23.MIG23T31.SCIUCLIS,DISP=SHR
//INPUT    DD DSN=REDBK23.APPL.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN  DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//INTMOD   DD DSN=REDBK23.APPL.DETMODS,DISP=(NEW,CATLG,DELETE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000),SPACE=(CYL,(1,1))
//DETAIL   DD DUMMY
//
```

The output from this job can be seen in Example 8-5.

We look at the populated DB2 tables in the section 8.3.3, “Identifying COBOL/VS programs” on page 224.

Example 8-5 IA Scanner summary output

```
CICS INTERDEPENDENCY ANALYZER  Version 2.1.0
LOAD MODULE SCANNER -  SUMMARY LISTING OF REDBK23.APPL.LOADLIB

Module       Module     Module       Language   Possible statements......
Name         Length     Language     Version    Affinities   Dependencies
--------    --------    ----------   --------   ----------   ------------
CCVSREMP    00003E28    ASSEMBLER                       70             72
CDCB001#    00001F38    COBOL         Non LE             0              2
CDCB0010    00002090    COBOL         Non LE             0              2
CDCB0020    000020D0    COBOL         Non LE             0              5
CDCB0510    00002090    COBOL         Non LE             0              2
CDCB0710    00002090    COBOL         Non LE             0              2
CICB0010    000020B8    COBOL         Non LE             0              7
CICB0020    00001EB0    COBOL         Non LE             0              8
CICB0030    00001EB0    COBOL         Non LE             0              8
CICB0050    00001A90    COBOL         Non LE             0              5
COBOLVS1    00001318    COBOL         Non LE             1              2
COBOLVS2    00001318    COBOL         Non LE             1              2
CSCB0010    00001358    COBOL         Non LE             0              1
CSCB0030    00004BC0    COBOL         Non LE             0              2
CSCB0200    00001250    COBOL         Non LE             0              1
REDBK1      00001630    C/370         LE                 4              6
REDBK1A     00001630    C/370         LE                 4              6
```

```
REDBK1B    00001630   C/370         LE                 4              6
REDBK1C    00001630   C/370         LE                 4              6
REDBK1D    00001630   C/370         LE                 4              6
REDBK1E    00001630   C/370         LE                 4              6
REDBK2     000026D8   PL/I          LE                 6              6
REDBK3     00001720   C/370         LE                 0              4
REDBK4     00001738   COBOL         Non LE             0              1
REDBK5     00001780   C/370         LE                 2              6
CICS INTERDEPENDENCY ANALYZER  Version 2.1.0
LOAD MODULE SCANNER -  SUMMARY LISTING OF REDBK23.APPL.LOADLIB

                    LOAD LIBRARY STATISTICS
=================================================================
Total modules in library                                =       25
Total modules scanned                                   =       25
Total CICS modules/tables (not scanned)                 =        0
Total modules in error (not scanned)                    =        0
Total modules containing possible MVS POSTs             =        0
Total modules containing possible Dependency commands =       25
Total modules containing possible Affinity commands   =       11
  Total ASSEMBLER modules                               =        1
  Total C/370 modules                                   =        8
  Total COBOL modules                                   =       15
  Total COBOL II modules                                =        0
  Total PL/I modules                                    =        1
```

To run the detailed report for the load module scanner we must edit and run the customized job CIUJCLTD. The job appears in Example 8-6. The values that require editing in this job are:

scan The load library to be scanned. We scan REDBK23.APPL.LOADLIB.

ciudet The input data set created by the summary job. We use REDBK23.APPL.DETMODS.

Example 8-6 CIUJCLTS - IA detailed scanner

```
//CIUJCLTD JOB USER=EYJ,NOTIFY=EYJ,
//         CLASS=A,MSGCLASS=Y,REGION=0M
//*********************************************************************
//* JCL NAME = CIUJCLTD                                               *
//*                                                                   *
//* DESCRIPTIVE NAME = IBM CICS INTERDEPENDENCIES UTILITY            *
//*                       RUN SCANNER IN DETAIL MODE WITH DB2 OUTPUT  *
//*                                                                   *
//* CHANGES TO BE MADE                                                *
//*                                                                   *
//*   1) CHANGE THE JOB CARD TO SUIT YOUR SYSTEM CONVENTIONS         *
//*   2) CHANGE THE FOLLOWING PARAMETERS:-                            *
```

```
//*   DB2P                                                              *
//*   THE DB2 ID                                                        *
//*   CIU                                                               *
//*   DATASET HLQ FOR CIU PRODUCT                                       *
//*   DSN710                                                            *
//*   DATASET HLQ FOR DB2 SDSNLOAD and RUNLIB.LOAD                      *
//*   _scan_                                                            *
//*   The load library to be scanned                                    *
//*   _ciudet_                                                          *
//*   Input dataset created from a SCANNER SUMMARY JOB                  *
//*                                                                     *
//*   3) EDIT THE MEMBER CIUDB2BT IN                                    *
//*      REDBK23.MIG23T31.SCIUCLIS                                      *
//*      AND CHANGE THE FOLLOWING:-                                     *
//*   CIU                                                               *
//*   DATASET HLQ FOR CIU PRODUCT                                       *
//*                                                                     *
//*********************************************************************
//* FUNCTION =                                                          *
//*                                                                     *
//*    Sample JCL to run the Load Module Scanner component of the       *
//*    Interdependencies Utility (DETAIL mode, DB2 output).             *
//*                                                                     *
//SCAN     EXEC PGM=IKJEFT1B,DYNAMNBR=20,
//             PARM=('%CIUDB2BT','SYS(DB2P)','PROG(CIULMS)',
//             'PLAN(CIUBTCH4)','PARM(''$DETAIL,TABLE'')')
//STEPLIB  DD DSN=CIU.SCIULOAD,DISP=SHR
//         DD DSN=CIU.SCIULODE,DISP=SHR
//         DD DSN=DSN710.SDSNLOAD,DISP=SHR
//SYSPROC  DD DSN=REDBK23.MIG23T31.SCIUCLIS,DISP=SHR
//INPUT    DD DSN=REDBK23.APPL.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN  DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//DETAIL   DD DSN=REDBK23.APPL.DETMODS,DISP=(OLD,DELETE)
//INTMOD   DD DUMMY
//
```

The output from this job can be seen in Example 8-7.

We look at the populated DB2 tables in the section 8.3.3, “Identifying COBOL/VS programs” on page 224.

Example 8-7 IA Scanner detailed output

```
CICS INTERDEPENDENCY ANALYZER  Version 2.1.0                                                06/27/06    Page     1
LOAD MODULE SCANNER - DETAILED LISTING OF REDBK23.APPL.LOADLIB

Module Name - REDBK1    /  Load Module Length - 00001630  /  Module Entry Point - 00000028
Offset    Storage Content (HEX)                              EDF DEBUG  Possible Command             Depcy   Affinity
--------  -------------------------------------------------  ---------  ----------------------------  ------  --------
000008A4  020880002F1F00000000000000000000000000000000        00007000   ASSIGN   APPLID               Yes
00000904  020280002F020000000000000000000000000000000         00007400   ADDRESS  CWA                  Yes     Trans
00000924  0A02E0002F00004100                                  00007800   WRITEQ   TS                   Yes     Trans
00000964  0604F0002F28004400                                  00010000   WRITE    FILE                 Yes
00000978  0A02E0002F00004100                                  00010600   WRITEQ   TS                   Yes     Trans
0000098C  0A02E0002F00004100                                  00011000   WRITEQ   TS                   Yes     Trans
Total possible Affinity commands   =        4
Total possible Dependency commands =        6
Total possible MVS POSTs           =        0

CICS INTERDEPENDENCY ANALYZER  Version 2.1.0                                                06/27/06    Page    10
LOAD MODULE SCANNER - DETAILED LISTING OF REDBK23.APPL.LOADLIB

                    LOAD LIBRARY STATISTICS
=================================================================
Total modules in DETAIL file                         =        25
Total modules scanned                                =        25
Total CICS modules/tables (not scanned)              =         0
Total modules in error (not scanned)                 =         0
Total modules containing possible MVS POSTs          =         0
Total modules containing possible Dependency commands =       25
Total modules containing possible Affinity commands  =        11
  Total ASSEMBLER modules                            =         1
  Total C/370 modules                                =         8
  Total COBOL modules                                =        15
  Total COBOL II modules                             =         0
  Total PL/I modules                                 =         1
```

Running the CSECT scanner

In order to use the CSECT scanner we must first populate the DB2 table CIU4_TRANSLATORS with a list of translator and compiler names. To do this we must edit and run the customized job CIUTLOAD. To run the CSECT scanner we must edit and run the customized job CIUJCLCS. The job appears in Example 8-8. The value that requires editing in this job is _scan_, the load library to be scanned. We scan REDBK23.APPL.LOADLIB.

Example 8-8 CIUJCLCS - IA CSECT Scanner JCL

```
//CIUJCLCS JOB USER=EYJ,NOTIFY=EYJ,
//         CLASS=A,MSGCLASS=Y,REGION=0M
//*********************************************************************
//* JCL NAME = CIUJCLCS                                               *
//* DESCRIPTIVE NAME = IBM CICS INTERDEPENDENCIES UTILITY             *
//*                    Sample JCL for running CSECT Scanner with      *
//*                    DB2 output.                                    *
//*                                                                   *
//* CHANGES TO BE MADE                                                *
```

```
//*                                                                      *
//*   1) CHANGE THE JOB CARD TO SUIT YOUR SYSTEM CONVENTIONS             *
//*   2) CHANGE THE FOLLOWING PARAMETERS:-                               *
//*   DB2P                                                               *
//*   THE DB2 ID                                                         *
//*   CIU                                                                *
//*   DATASET HLQ FOR CIU PRODUCT                                        *
//*   DSN710                                                             *
//*   DATASET HLQ FOR DB2 SDSNLOAD and RUNLIB.LOAD                       *
//*   _scan_                                                             *
//*   CICS LOAD DATASET TO BE SCANNED                                    *
//*                                                                      *
//*   3) EDIT THE MEMBER CIUDB2BT IN                                     *
//*      REDBK23.MIG23T31.SCIUCLIS                                       *
//*      AND CHANGE THE FOLLOWING:-                                      *
//*   CIU                                                                *
//*   DATASET HLQ FOR CIU PRODUCT                                        *
//*                                                                      *
//**********************************************************************
//SCAN     EXEC PGM=IKJEFT1B,DYNAMNBR=20,
//             PARM=('%CIUDB2BT','SYS(DB2P)','PROG(CIUCSS)',
//             'PLAN(CIUBTCH4)','PARM(''$TABLE'')')
//STEPLIB  DD DSN=CIU.SCIULOAD,DISP=SHR
//         DD DSN=CIU.SCIULODE,DISP=SHR
//         DD DSN=DSN710.SDSNLOAD,DISP=SHR
//SYSPROC  DD DSN=REDBK23.MIG23T31.SCIUCLIS,DISP=SHR
//LOADLIB  DD DSN=REDBK23.APPL.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN  DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//
```

The output from this job can be seen in Example 8-9.

We look at the populated DB2 tables in the section 8.3.3, “Identifying COBOL/VS programs” on page 224.

Example 8-9 IA CSECT scanner output

```
CICS INTERDEPENDENCY ANALYZER Version 2.1.0                                                  06/27/06    Page     1
CSECT SCANNER - LISTING OF: REDBK23.APPL.LOADLIB
REDBK4     00001738   00000020             5695PMB01    01.07   2006163104940    24    24
  DFHECI   1997256 569623400  01.02
  REDBK4   2006163 5740CB103  02.04
  ILBOCOM0 1983194 5734AS100  05.01                        1983194 RSI31940368
  ILBOSRV  1983194 5734AS100  05.01                        1983194 RSI31940563
  ILBOMSG  1983194 5734AS100  05.01                        1983194 RSI31940572
  ILBOBEG  1983194 5734AS100  05.01                        1983194 RSI31940346
```

8.3.2 Using the CICS IA Collector

The collector consists of:

- A control transaction, CINT
- An autosave transaction, CINB
- A number of global user exit programs

In this section we describe how to:

- Configure the collector.
- Start the collector.
- Stop the collector.
- Load the collected data into DB2.

To configure and run the collector we use transaction CINT.

Further configuration options are discussed in 11.5, “Reducing performance impact during IA collection” on page 387.

Configuring the collector

To configure the collector we use transaction CINT. Figure 8-4 shows the initial screen when CINT is entered.

```
CIU000        CICS Interdependency Analyzer for z/OS - V2R1M0        2006/06/09
                          Main Administration Menu                    11:25:23AM


Select one of the following.  Then press Enter.


    1   Operations Menu.
    2   Configure Region Options.
    3   Configure Global Options.









CICS Sysid:  RB23   CICS Applid:  REDBKV23   TermID:  CP51
CIU7000I 5697-J23 (C) Copyright IBM Corp. 2001,2005
F1=Help      F2=           F3=Exit      F4=          F5=          F6=
F7=          F8=           F9=          F10=         F11=         F12=Exit
```

Figure 8-4   CINT - Administration panel

Select option 3 to configure global options. Figure 8-5 shows the global options screen.

```
CIU300        CICS Interdependency Analyzer for z/OS - V2R1M0        2006/06/09
                           Global Options Menu                        11:26:09AM

Modify the options and press Enter to update, or press PF12 to cancel.

 Control options
   VSAM file sharing  . . . . . . : N  (Yes/No)
   High Level Trace . . . . . . . : N  (Yes/No)

 National Language Option . . . : E  Code: ENU

 Date and Time Formats
   Date . . . . . . . 4 1. MMDDYY 2. DDMMYY    Separator . . . . . . . . /
                        3. YYMMDD 4. YYYYMMDD

   Time . . . . . . . 1 1. 12 hrs 2. 24 hrs    Separator . . . . . . . . :



CICS Sysid:   RB23   CICS Applid:   REDBKV23   TermID:   CP51

F1=Help       F2=           F3=End        F4=           F5=Refresh    F6=
F7=           F8=           F9=           F10=          F11=          F12=Cancel
```

Figure 8-5 CINT - Global Options Menu

Select option 2 from the initial screen to configure region options. Figure 8-6 shows the region options screen.

```
 CIU200         CICS Interdependency Analyzer for z/OS - V2R1M0          2006/06/09
                            Region Configuration Menu                    11:26:35AM

 Type action code then press ENTER.                                  More :
 1=Add Region     3=Delete Region    5=DB2/IMS/MQ Options    7=Affinity Options
 2=Copy Region    4=CICS Options     6=General Options       8=Time/Date Options

        CICS       CICS    New        New
   Act  Applid     Sysid   Applid     Sysid   Status       Collecting
    4   DEFAULTS   DFTS    ________   ____
    _   REDBKV23   RB23    ________   ____    STOPPED
    _                      ________   ____
    _                      ________   ____
    _                      ________   ____
    _                      ________   ____
    _                      ________   ____
    _                      ________   ____
    _                      ________   ____
    _                      ________   ____

 CICS Sysid:  RB23   CICS Applid:  REDBKV23   TermID:  CP51

 F1=Help      F2=          F3=Exit      F4=          F5= Refresh  F6=
 F7=Page Up   F8=Page Down F9=          F10=         F11=         F12=
```

Figure 8-6 CINT - Region Configuration Menu

Select option 4 to configure the CICS default options. Figure 8-7 shows the CICS default options screen. To collect all dependencies make sure all options are set to Y for YES.

```
CIU240        CICS Interdependency Analyzer for z/OS - V2R1M0         2006/06/09
                          CICS Resources Options for                  11:27:07AM
          CICS Sysid   : DFTS     CICS Applid   : DEFAULTS

Modify the options and press Enter to update, or PF12 to Cancel.

Detect command types: Y=Yes, N=No

APIs
 START  . . . . . Y  XCTL . . . . . . Y  LOAD . . . . . . Y  LINK . . . . . . Y
 RETURN TRANSID Y  Handle Abend . Y  Task Control . Y  File Control . Y
 BMS  . . . . . . Y  TS Queues  . . Y  TD Queues  . . Y  Journals . . . Y
 DTP  . . . . . . Y  Counters . . . Y  FEPI . . . . . . Y  WEB Services . Y
 Others . . . . . Y

SPIs (Create/Inquire/Set/Discard/Perform)
 Programs . . . Y  Files  . . . . . Y  Transactions . Y  Temp Storage . Y
 Transient Data Y  DB2  . . . . . . Y  DJAR . . . . . . Y  BRFacility . . Y
 Corbaserver  . Y  TCPIPService . Y  FEPI . . . . . . Y  Journals . . . Y

CICS Sysid:  RB23   CICS Applid:  REDBKV23   TermID:  CP51

F1=           F2=            F3=Exit       F4=           F5=           F6=
F7=           F8=            F9=           F10=          F11=          F12=Cancel
```

Figure 8-7 CINT - CICS Resource Options

Starting the collector

To start the collector enter transaction CINT and choose option 1 for the operations menu. Figure 8-8 shows the Operations screen. Select 1 to start CICS IA.

```
CIU100        CICS Interdependency Analyzer for z/OS - V2R1M0         2006/06/09
                              Operations Menu                         11:48:14AM

Type action code then press ENTER.                                     More :

1= Start 2= Stop 3= Pause 4= Continue 5= Statistics

      CICS      CICS                  Start        Start
 Act  Applid    Sysid  Status         Date         Time         Collecting
  1   REDBKV23  RB23   STOPPED
  _
  _
  _
```

Figure 8-8 CINT - IA start

IA asks for you to confirm the start of the region. See Figure 8-9. Press Enter to confirm.

```
  _
  _

CICS Sysid:  RB23   CICS Applid:  REDBKV23   TermID:  CP51
CIU2120I Press Enter to confirm Start with data restore or PF12 to cancel
F1=Help      F2=           F3=End       F4=          F5=Refresh  F6=
F7=Page Up   F8=Page Down  F9=          F10=         F11=         F12= CANCEL
```

Figure 8-9 CINT - confirm start

The operations screen will then refresh to show IA running and collecting dependencies. See Figure 8-10.

```
CIU100       CICS Interdependency Analyzer for z/OS - V2R1M0        2006/06/09
                            Operations Menu                          11:50:38AM

Type action code then press ENTER.                              More :

1= Start 2= Stop 3= Pause 4= Continue 5= Statistics

      CICS      CICS                  Start        Start
 Act  Applid    Sysid Status          Date         Time         Collecting
  *   REDBKV23  RB23  RUNNING         2006/06/09   11:50:38AM   Dependencies
  _
```

Figure 8-10 CINT - collecting dependencies

Stopping the collector

To stop the collector enter transaction CINT and choose option 1 for the operations menu. Figure 8-11 shows the Operations screen. Select 2 to stop CICS IA.

```
CIU100       CICS Interdependency Analyzer for z/OS - V2R1M0        2006/06/09
                            Operations Menu                          11:51:57AM

Type action code then press ENTER.                              More :

1= Start 2= Stop 3= Pause 4= Continue 5= Statistics

      CICS      CICS                  Start        Start
 Act  Applid    Sysid Status          Date         Time         Collecting
  2   REDBKV23  RB23  RUNNING         2006/06/09   11:50:38AM   Dependencies
  _
```

Figure 8-11 CINT - IA stop

IA asks for you to confirm the stop of the region. See Figure 8-12. Press Enter to confirm.

```
CICS Sysid:  RB23   CICS Applid:  REDBKV23   TermID:  CP51
CIU2122I Press Enter to confirm Stop or PF12 to cancel
F1=Help      F2=           F3=End        F4=          F5=Refresh  F6=
F7=Page Up   F8=Page Down  F9=           F10=         F11=         F12= CANCEL
```

Figure 8-12   CINT - stop confirmation

IA shows the statistics screen for the region once it has stopped. See Figure 8-13.

```
CIU150        CICS Interdependency Analyzer for z/OS - V2R1M0        2006/06/09
                            Statistics Menu for                        11:52:38AM

         CICS Sysid   : RB23     CICS Applid   : REDBKV23

  CINT state . . . . . . . . . : STOPPED      Collecting Dependencies
  Number of pauses . . . . . . : 0
  Number of saves. . . . . . . : 1
  Records written last save. : 13
  Total records on file. . . : 46

  Date/time of last start. . : 2006/06/09 11:50:38AM
  Date/time of last save . . : 2006/06/09 11:52:37AM
  Date/time of last change . : 2006/06/09 11:51:47AM

  Total time RUNNING . . . . : 0000:01:41  (HHHH:MM:SS)
  Total time PAUSED. . . . . :             (HHHH:MM:SS)

  Table dataspace name . . . :                   % full

CICS Sysid:  RB23   CICS Applid:  REDBKV23   TermID:  CP51

F1=Help      F2=           F3=End        F4=          F5=Refresh  F6=
F7=          F8=           F9=           F10=         F11=         F12=
```

Figure 8-13   CINT - collection statistics

Loading the collected data into DB2

To load the collected data into DB2 we must edit and run the customized job CIUUPDB1.

The job appears in Example 8-10.

Example 8-10 CIUUPDB1 - DB2 update JCL

```
//CIUUPDB1 JOB USER=EYJ,NOTIFY=EYJ,
//         CLASS=A,MSGCLASS=Y,REGION=0M
//*********************************************************************
//*                                                                   *
//* JCL NAME = CIUUPDB1                                               *
//*                                                                   *
//* DESCRIPTIVE NAME = IBM CICS INTERDEPENDENCIES UTILITY             *
//*                    UPDATE THE DATABASE WITH CICS DEPENDENCIES     *
//*                                                                   *
//* CHANGES TO BE MADE                                                *
//* PLEASE CONSULT WITH YOUR DB2 ADMINISTRATOR                        *
//*                                                                   *
//*   1) CHANGE THE JOB CARD TO SUIT YOUR SYSTEM CONVENTIONS          *
//*                                                                   *
//*   2) CHANGE THE FOLLOWING PARAMETERS:-                            *
//*                                                                   *
//*  DB2P                                                             *
//*  THE DB2 ID                                                       *
//*                                                                   *
//*  CIU                                                              *
//*  THE HLQ FOR CIU PRODUCT                                          *
//*                                                                   *
//*  DSN710                                                           *
//*  THE DATASET HLQ FOR DB2 SDSNLOAD                                 *
//*                                                                   *
//*  REDBK23.MIG23T31                                                 *
//*  THE HLQ FOR THE CIU FILE RESOURCES. THESE SHOULD BE              *
//*  THE SAME AS THOSE DEFINED IN JOBS CIUJCLCA/CIUJCLCC              *
//*                                                                   *
//*   3) EDIT THE SCIUCLIS MEMBER CIUDB2BT IN                         *
//*                                                                   *
//*      REDBK23.MIG23T31.SCIUCLIS                                    *
//*                                                                   *
//*      AND CHANGE THE FOLLOWING:-                                   *
//*                                                                   *
//*  CIU                                                              *
//*  THE HLQ FOR CIU PRODUCT                                          *
//*                                                                   *
//*   4) IF YOU WISH TO UPDATE THE DATABASE WITH THE LAST USED        *
//*      TIME STAMP FOR EACH DB2 ROW THEN CHOOSE PARM(UPD)            *
//******************************************************************
//*----------------------------------------------------------------
//*        RUN THE BATCH PROGRAM CIUUREG
//*----------------------------------------------------------------
//STEP000  EXEC PGM=IKJEFT1B,
```

```
//              DYNAMNBR=20,
//              PARM=('%CIUDB2BT','SYS(DB2P)','PROG(CIUUREG)',
//              'PLAN(CIUBTCH4)')
//*-------------------------------------------------------------------
//*      IF YOU WISH TO UPDATE THE DATABASE WITH THE LAST USED
//*      TIME STAMP FOR EACH DB2 ROW THEN CHOOSE PARM(UPD)
//*-------------------------------------------------------------------
//STEPLIB  DD DSN=CIU.SCIULOAD,DISP=SHR
//         DD DSN=CIU.SCIULODE,DISP=SHR
//         DD DSN=DSN710.SDSNLOAD,DISP=SHR
//SYSPROC  DD DSN=REDBK23.MIG23T31.SCIUCLIS,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSTSIN  DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//CIUCNTL  DD DSN=REDBK23.MIG23T31.CIUCNTL,
//             DISP=SHR
//*-------------------------------------------------------------------
//*         CONVERT COLLECTED DATA TO QSAM FILE
//*-------------------------------------------------------------------
//STEP010  EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//IN       DD  DSN=REDBK23.MIG23T31.CIUINT1,DISP=SHR
//OUT      DD  DSN=&&DATA1,DISP=(,PASS),SPACE=(CYL,(5,5),RLSE),
//             UNIT=SYSDA,DCB=(RECFM=VB,LRECL=131,BLKSIZE=13100)
//SYSIN    DD  *
 REPRO IFILE(IN),OFILE(OUT)
//*********************************************************************
//*-------------------------------------------------------------------
//*         CONVERT COLLECTED DATA TO QSAM FILE - LONG FILE
//*-------------------------------------------------------------------
//STEP015  EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//IN       DD  DSN=REDBK23.MIG23T31.CIUINT5,DISP=SHR
//OUT      DD  DSN=&&DATA0,DISP=(,PASS),SPACE=(CYL,(5,5),RLSE),
//             UNIT=SYSDA,DCB=(RECFM=VB,LRECL=361,BLKSIZE=36100)
//SYSIN    DD  *
 REPRO IFILE(IN),OFILE(OUT)
//*-------------------------------------------------------------------
//*         REFORMAT INPUT FILE
//*-------------------------------------------------------------------
//STEP020  EXEC PGM=CIUU040
//STEPLIB  DD DSN=CIU.SCIULOAD,DISP=SHR
//         DD DSN=CIU.SCIULODE,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//INPUT    DD DSN=&&DATA1,DISP=(OLD,DELETE)
//INPUT2   DD DSN=&&DATA0,DISP=(OLD,DELETE)
//OUTPUT   DD DSN=&&DATA2,DISP=(,PASS),SPACE=(CYL,(5,5),RLSE),
```

```
//              UNIT=SYSDA,DCB=(RECFM=FB,LRECL=384,BLKSIZE=38400)
//*-----------------------------------------------------------------
//*        SORT THE INPUT FILE
//*-----------------------------------------------------------------
//STEP030  EXEC PGM=SORT,COND=(0,NE,STEP020)
//SORTLIB  DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SORTIN   DD DSN=&&DATA2,DISP=(OLD,DELETE)
//SORTOUT  DD DSN=&&DATA3,DISP=(,PASS),SPACE=(CYL,(5,5),RLSE),
//              UNIT=SYSDA,DCB=*.SORTIN
//SYSIN    DD *
 SORT FIELDS=(1,8,A,13,4,A,17,8,A,41,255,A),
 FORMAT=CH
 RECORD TYPE=F,LENGTH=(384)
/*
//*-----------------------------------------------------------------
//*        RUN THE BATCH PROGRAM CIUU050
//STEP040  EXEC PGM=IKJEFT1B,COND=(0,NE,STEP020),
//              DYNAMNBR=20,
//              PARM=('%CIUDB2BT','SYS(DB2P)','PROG(CIUU050)',
//              'PLAN(CIUBTCH4)','PARM(NOPARM)') <-- NO TIMESTAMP UPDATE
//*             'PLAN(CIUBTCH4)','PARM(UPD)')    <-- TIMESTAMP UPDATE
//*-----------------------------------------------------------------
//*     IF YOU WISH TO UPDATE THE DATABASE WITH THE LAST USED
//*     TIME STAMP FOR EACH DB2 ROW THEN CHOOSE PARM(UPD)
//*-----------------------------------------------------------------
//STEPLIB  DD DSN=CIU.SCIULOAD,DISP=SHR
//         DD DSN=CIU.SCIULODE,DISP=SHR
//         DD DSN=DSN710.SDSNLOAD,DISP=SHR
//SYSPROC  DD DSN=REDBK23.MIG23T31.SCIUCLIS,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSTSIN  DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//CIUINT1  DD DSN=&&DATA3,DISP=(OLD,DELETE)
//*-----------------------------------------------------------------
//*             REFRESH CIU4_CICS_CHAIN                            *
//*-----------------------------------------------------------------
//STEP050  EXEC PGM=IKJEFT1B,COND=(0,NE,STEP020),
//              DYNAMNBR=20,
//              PARM=('%CIUDB2BT','SYS(DB2P)','PROG(CIUU100)',
//              'PLAN(CIUBTCH4)')
//STEPLIB  DD DSN=CIU.SCIULOAD,DISP=SHR
//         DD DSN=CIU.SCIULODE,DISP=SHR
//         DD DSN=DSN710.SDSNLOAD,DISP=SHR
//SYSPROC  DD DSN=REDBK23.MIG23T31.SCIUCLIS,DISP=SHR
//SYSUDUMP DD SYSOUT=*
```

```
//SYSTSIN  DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//*------------------------------------------------------------------
//*             REFRESH CIU4_CICS_CHAINP
//*------------------------------------------------------------------
//STEP051  EXEC PGM=IKJEFT1B,COND=(0,NE,STEP020),
//             DYNAMNBR=20,
//             PARM=('%CIUDB2BT','SYS(DB2P)','PROG(CIUU200)',
//             'PLAN(CIUBTCH4)')
//STEPLIB  DD DSN=CIU.SCIULOAD,DISP=SHR
//         DD DSN=CIU.SCIULODE,DISP=SHR
//         DD DSN=DSN710.SDSNLOAD,DISP=SHR
//SYSPROC  DD DSN=REDBK23.MIG23T31.SCIUCLIS,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSTSIN  DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//SYSOUT   DD SYSOUT=* //*           REFRESH CIU4_CICS_CHAINP
```

The load job produces output to indicate how many records were extracted from the VSAM file and how many were added/updated in the DB2 table. See Example 8-11.

Example 8-11 CIUUPDB1 - sample output

```
********************************************
**** CICS records extracted =         8 ****
********************************************
14390710 CIU6003I LAST USE TIMESTAMPS WILL NOT BE UPDATED
14390715 CIU6005I NUMBER OF NEW ROWS ADDED TO CIU4_CICS_DATA = 000000008
14390715 CIU6006I NUMBER OF EXISTING ROWS IN CIU4_CICS_DATA = 000000000
14390715 CIU6007I NUMBER OF ROWS UPDATED IN CIU4_CICS_DATA = 000000000
```

We can now query the database to find which programs are compiled as COBOL/VS and which programs are non threadsafe.

8.3.3 Identifying COBOL/VS programs

To identify which programs were compiled with COBOL/VS we can query either the CIU4_SCAN_SUMMARY table populated by job CIUJCLTS (load module scanner) or the V_CIU4_CSECT_TRANS view populated by job CIUJCLCS (CSECT scanner).

All of the following SQL queries were performed using IBM SPUFI interface.

Querying the CIU4_SCAN _SUMMARY table

Example 8-12 shows all of the modules in REDBK23.APPL.LOADLIB that the scanner has identified as being COBOL/VS programs.

Example 8-12 COBOL query using the scan summary table

```
--Show me all possible programs that are COBOL/VS in dataset
--REDBK23.APPL.LOADLIB using the load module scanner summary.
  SELECT PROGRAM, LANGUAGE  FROM CIU4_SCAN_SUMMARY
  WHERE DSNAME='REDBK23.APPL.LOADLIB' AND LANGUAGE='COBOL';
---------+---------+---------+---------+---------+---------+---------+--------+
PROGRAM   LANGUAGE
---------+---------+---------+---------+---------+---------+---------+--------+
REDBK4    COBOL
CSCB0200  COBOL
CSCB0030  COBOL
CSCB0010  COBOL
COBOLVS2  COBOL
COBOLVS1  COBOL
CICB0050  COBOL
CICB0030  COBOL
CICB0020  COBOL
CICB0010  COBOL
CDCB0710  COBOL
CDCB0510  COBOL
CDCB0020  COBOL
CDCB0010  COBOL
CDCB001#  COBOL
DSNE610I NUMBER OF ROWS DISPLAYED IS 15
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
```

Example 8-13 shows the compiler language for all the programs with a prefix of RED. We can see that program REDBK4 is a COBOL/VS program.

Example 8-13 Language query using the scan summary

```
--Show me all possible programs that are COBOL/VS in data set
--REDBK23.APPL.LOADLIB using the load module scanner summary.
  SELECT PROGRAM, LANGUAGE  FROM CIU4_SCAN_SUMMARY
  WHERE DSNAME='REDBK23.APPL.LOADLIB' AND PROGRAM LIKE 'RED%'
---------+---------+---------+---------+---------+---------+---------+--------+
PROGRAM   LANGUAGE
---------+---------+---------+---------+---------+---------+---------+--------+
REDBK1    C/370
REDBK1A   C/370
REDBK1B   C/370
REDBK1C   C/370
REDBK1D   C/370
```

```
REDBK1E   C/370
REDBK2    PL/I
REDBK3    C/370
REDBK4    COBOL
REDBK5    C/370
DSNE610I NUMBER OF ROWS DISPLAYED IS 10
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
```

Querying the V_CIU4_CSECT_TRANS view

Example 8-14 shows a query against the data collected for the CSECT scanner. The query is performed against the DB2 view V_CIU_CSECT_TRANS.

Example 8-14 COBOL query using the CSECT view

```
--Show me all possible programs that are COBOL/VS in data set
--REDBK31.APPL.LOADLIB using the CSECT scanner view.
  SELECT DISTINCT PROGRAM, CSECT_NAME, DESCRIPTION , TRAN_1_NAME
  FROM V_CIU4_CSECT_TRANS
  WHERE DSNAME='REDBK23.APPL.LOADLIB'
  AND DESCRIPTION LIKE 'OS/VS COBOL%'
---------+---------+---------+---------+---------+---------+---------+--------+
PROGRAM   CSECT_NAME  DESCRIPTION                     TRAN_1_NAME
---------+---------+---------+---------+---------+---------+---------+--------+
CDCB001#  CDCB0010    OS/VS COBOL R2M3 R2M4 (VSR1)    5740CB103
CDCB0010  CDCB0010    OS/VS COBOL R2M3 R2M4 (VSR1)    5740CB103
CDCB0020  CDCB0020    OS/VS COBOL R2M3 R2M4 (VSR1)    5740CB103
CDCB0510  CDCB0510    OS/VS COBOL R2M3 R2M4 (VSR1)    5740CB103
CDCB0710  CDCB0710    OS/VS COBOL R2M3 R2M4 (VSR1)    5740CB103
CICB0010  CICB0010    OS/VS COBOL R2M3 R2M4 (VSR1)    5740CB103
CICB0020  CICB0020    OS/VS COBOL R2M3 R2M4 (VSR1)    5740CB103
CICB0030  CICB0030    OS/VS COBOL R2M3 R2M4 (VSR1)    5740CB103
CICB0050  CICB0050    OS/VS COBOL R2M3 R2M4 (VSR1)    5740CB103
COBOLVS1  COBOLVS1    OS/VS COBOL R2M3 R2M4 (VSR1)    5740CB103
COBOLVS2  COBOLVS2    OS/VS COBOL R2M3 R2M4 (VSR1)    5740CB103
CSCB0010  CSCB0010    OS/VS COBOL R2M3 R2M4 (VSR1)    5740CB103
CSCB0030  CSCB0030    OS/VS COBOL R2M3 R2M4 (VSR1)    5740CB103
CSCB0200  CSCB0010    OS/VS COBOL R2M3 R2M4 (VSR1)    5740CB103
REDBK4    REDBK4      OS/VS COBOL R2M3 R2M4 (VSR1)    5740CB103
DSNE610I NUMBER OF ROWS DISPLAYED IS 15
DSNE612I DATA FOR COLUMN HEADER DESCRIPTION COLUMN NUMBER 3 WAS TRUNCATED
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
```

8.3.4 Identifying non-threadsafe programs

To identify which programs are non threadsafe we can query either the CIU4_SCAN_DETAIL table populated by job CIUJCLTD (load module scanner) or the CIU4_CICS_DATA table populated by data from the collector.

Querying the CIU4_SCAN _DETAIL table

The following query tells us all programs that have possible commands that would cause the program to be non threadsafe (that is, the program executes a LOAD, EXTRACT, GETMAIN, or ADDRESS CWA). The query is restricted to the REDBK23.APPL.LOADLIB data set only. See Example 8-15.

Example 8-15 Threadsafe query using the scan detail table

```
--Show me all possible programs that are not threadsafe in data set
--REDBK23.APPL.LOADLIB using the load module scanner detail
  SELECT PROGRAM , COMMAND , RESOURCE_TYPE
  FROM CIU4_SCAN_DETAIL
  WHERE COMMAND IN ('LOAD    ', 'EXTRACT ', 'GETMAIN ', 'ADDRESS ')
  AND DSNAME='REDBK23.APPL.LOADLIB';
---------+---------+---------+---------+---------+---------+---------+--------+
PROGRAM   COMMAND   RESOURCE_TYPE
---------+---------+---------+---------+---------+---------+---------+--------+
COBOLVS1  ADDRESS   CWA
COBOLVS2  GETMAIN   SHARED
REDBK1    ADDRESS   CWA
REDBK1A   ADDRESS   CWA
REDBK1B   ADDRESS   CWA
REDBK1C   ADDRESS   CWA
REDBK1D   ADDRESS   CWA
REDBK1E   ADDRESS   CWA
REDBK5    ADDRESS   CWA
DSNE610I NUMBER OF ROWS DISPLAYED IS 9
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
---------+---------+---------+---------+---------+---------+---------+--------+
```

Querying the CIU4_CICS _DATA table

The query in Example 8-16 will show us all resources used in CICS region REDBKV23.

Example 8-16 All resources query for region REDBKV23 using CICS table

```
--Show me all resources in region REDBKV23
--from the collector
  SELECT DISTINCT PROGRAM , FUNCTION,  TYPE , OBJECT
  FROM CIU4_CICS_DATA
  WHERE APPLID='REDBKV23'
```

```
  ORDER BY 1;
---------+---------+---------+---------+---------+---------+---------+--------+
PROGRAM   FUNCTION  TYPE      OBJECT
---------+---------+---------+---------+---------+---------+---------+--------+
REDBK1    ADDRESS   CWA       CWA
REDBK1    ASSIGN    APPLID    REDBKV23
REDBK1    WRITE     FILE      REDBOOKF
REDBK1    WRITEQ    TD        CESE
REDBK1    WRITEQ    TSSHR     REDBOOKQ
REDBK2    START     TRANSID   RDBA
REDBK2    START     TRANSID   RDBB
REDBK2    START     TRANSID   RDBC
REDBK2    START     TRANSID   RDBD
REDBK2    START     TRANSID   RDBE
REDBK2    START     TRANSID   RDB1
REDBK3    ASSIGN    APPLID    REDBKV23
REDBK3    ENDBR     FILE      REDBOOKF
REDBK3    READNEXT  FILE      REDBOOKF
REDBK3    STARTBR   FILE      REDBOOKF
REDBK4    LINK      PROGRAM   REDBK3
REDBK5    ADDRESS   CWA       CWA
REDBK5    ASSIGN    APPLID    REDBKV23
REDBK5    ENDBR     FILE      REDBOOKF
REDBK5    READNEXT  FILE      REDBOOKF
REDBK5    STARTBR   FILE      REDBOOKF
DSNE610I NUMBER OF ROWS DISPLAYED IS 21
DSNE612I DATA FOR COLUMN HEADER OBJECT COLUMN NUMBER 4 WAS TRUNCATED
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
```

Note: The output from this query only shows programs that have actually been executed while the CICS IA collector was running. For example, program COBOLVS1 in the output in Example 8-15 on page 227 is not in the output in Example 8-16 on page 227 because it has not been executed.

The following query will tell us all programs that have possible commands that would cause the program to be non threadsafe (that is, the program executes a LOAD, EXTRACT, GETMAIN, or ADDRESS CWA). The query is restricted to the CICS region that is to be migrated, REDBKV23. See Example 8-17.

Example 8-17 Threadsafe query using CICS table

```
--Show me all programs that are not threadsafe in region REDBKV23
--from the collector
  SELECT DISTINCT PROGRAM , FUNCTION, OBJECT
  FROM CIU4_CICS_DATA
  WHERE FUNCTION IN ('LOAD    ', 'EXTRACT ', 'GETMAIN ', 'ADDRESS ')
  AND APPLID='REDBKV23'
```

```
  ORDER BY 1;
---------+---------+---------+---------+---------+---------+---------+--------+
PROGRAM   FUNCTION  OBJECT
---------+---------+---------+---------+---------+---------+---------+--------+
REDBK1    ADDRESS   CWA
REDBK5    ADDRESS   CWA
DSNE610I NUMBER OF ROWS DISPLAYED IS 2
DSNE612I DATA FOR COLUMN HEADER OBJECT COLUMN NUMBER 3 WAS TRUNCATED
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
```

Programs REDBK1 and REDBK5 contain EXEC ADDRESS CWA commands and therefore would need careful investigation prior to being defined as threadsafe. If the reference to the CWA is for read-only purposes then these programs could potentially be defined as an OPENAPI program, which allows them to run in their own OTE TCB from the start.

In Example 8-16 on page 227 we can see that program REDBK2 consists of only EXEC CICS STARTs and could be considered for being defined as threadsafe.

8.3.5 Identifying applications to be migrated

The CICS TS 2.3 region to be migrated contains two applications:

- The first application contains all programs with a prefix of RED. The CICS IA application code for this application will be RDB.
- The second application contains all programs with a prefix of COB, CDC, or CSC. This CICS IA application code for this application will be COB.

To define CICS IA applications see “Creating new applications” on page 415.

We can use the CICS IA query CINQ transaction to see which resources are used by the application. Figure 8-14 shows the first screen for transaction CINQ. Select option 1 for CICS resources.

```
CIU400        CICS Interdependency Analyzer for z/OS - V2R1M0        2006/06/13
                                   Query Menu                        12:37:36PM

Select one of the following.  Then press Enter.

1   1   Inquire on CICS Resources.
    2   Inquire on DB2 Resources.
    3   Inquire on MQ Resources.
    4   Inquire on IMS Resources.
    5   Additional inquire on DB2 Resources.
    6   Inquire on CICS Affinities.









CICS Sysid:  RB23   CICS Applid:  REDBKV23   TermID:  CP89

F1=Help      F2=          F3=Exit      F4=          F5=          F6=
F7=          F8=          F9=          F10=         F11=         F12=Exit
```

Figure 8-14 CINQ - inquire about CICS resources

Figure 8-15 shows the CICS resources screen. To show all the resources for Redbook Application 1 enter RDB in the application code field and select Y for details.

```
CIU500        CICS Interdependency Analyzer for z/OS - V2R1M0         2006/06/13
                               CICS Query Menu                        01:05:29PM

      Select the resource type to query:    _

      1. Transactions
      2. Programs
      3. TSQs
      4. TDQs
      5. Maps
      6. Files
      7. Applications
      8. Regions

      OR display all resources in application rdb   detailed y (Y/N)

      Enter the application's 3 character code or
      ? for a list of applications available.
      WARNING : Option 7 may take a long time.

CICS Sysid:  RB23   CICS Applid:  REDBKV23   TermID:  CP89

F1=Help      F2=          F3=End       F4=Exit      F5=          F6=
F7=          F8=          F9=          F10=         F11=         F12=End
```

Figure 8-15 CINQ - inquire for all resources in application RDB showing detailed information

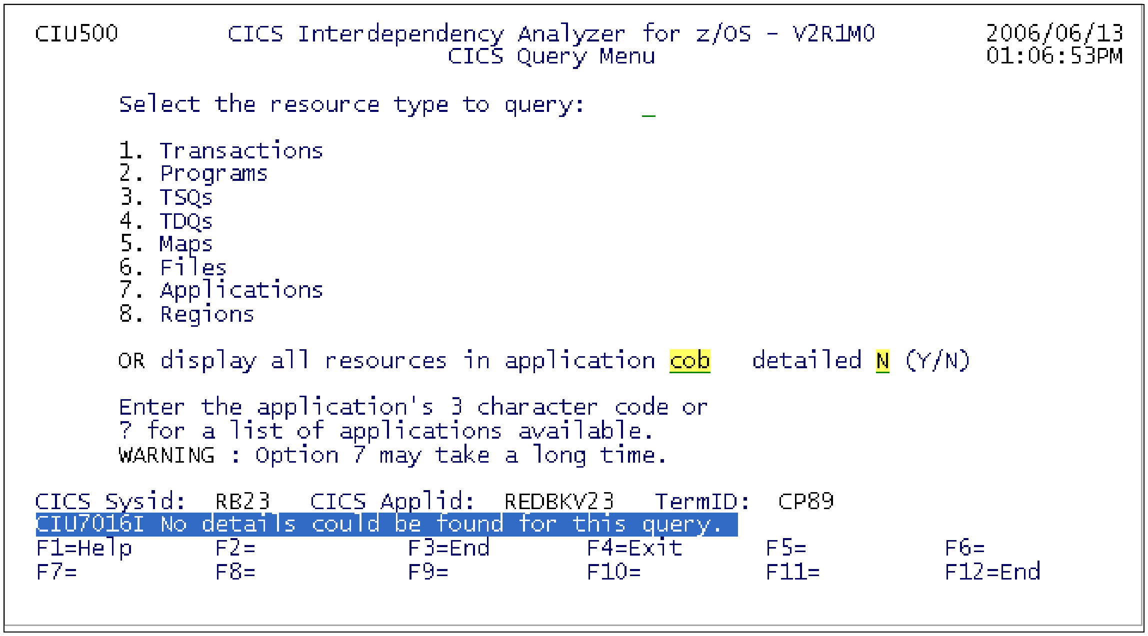
Figure 8-16 shows the first page of the output for query ‘Which resources are in Redbook Application 1”.

```
 CIU585       CICS Interdependency Analyzer for z/OS - V2R1M0        2006/06/13
  For your CICS Query                                                01:06:12PM
  WHICH RESOURCES ARE IN Redbook Application 1                 Page    1 of    5

  In  Tran Program  Resource Resource Resource
 Regn               Function Type     Name
 RB23 RDBA REDBK1   WRITEQ   TD       CESE
      RDBA REDBK1   ADDRESS  CWA      CWA
      RDBA REDBK1   ASSIGN   APPLID   REDBKV23
      RDBA REDBK1   WRITE    FILE     REDBOOKF
      RDBA REDBK1   WRITEQ   TSSHR    REDBOOKQ
      RDBB REDBK1   WRITEQ   TD       CESE
      RDBB REDBK1   ADDRESS  CWA      CWA
      RDBB REDBK1   ASSIGN   APPLID   REDBKV23
      RDBB REDBK1   WRITE    FILE     REDBOOKF
      RDBB REDBK1   WRITEQ   TSSHR    REDBOOKQ
      RDBC REDBK1   WRITEQ   TD       CESE
      RDBC REDBK1   ADDRESS  CWA      CWA
      RDBC REDBK1   ASSIGN   APPLID   REDBKV23

 CICS Sysid:  RB23   CICS Applid:  REDBKV23   TermID:  CP89

 F1=          F2=           F3=End        F4=Exit      F5=          F6=
 F7= Page Up  F8= Page Down F9=           F10=         F11=         F12=End
```

Figure 8-16 CINQ - Which Resources are in Redbook Application 1

> **Note:** The Redbook Application 1 uses a file resource of REDBOOKF. The DSNAME for this resource is REDBK23.REDBKV23.VSAM and is dependant on the CICS region it runs in. This DSNAME will need to be changed when migrating to the CICS TS 3.1 region.

To show all the resources for Redbook Application 2 enter COB in the application code field. Figure 8-17 shows that there are no resources used by Redbook Application 2.



```
CIU500        CICS Interdependency Analyzer for z/OS - V2R1M0        2006/06/13
                          CICS Query Menu                             01:06:53PM

      Select the resource type to query:     _

      1. Transactions
      2. Programs
      3. TSQs
      4. TDQs
      5. Maps
      6. Files
      7. Applications
      8. Regions

      OR display all resources in application cob   detailed N (Y/N)

      Enter the application's 3 character code or
      ? for a list of applications available.
      WARNING : Option 7 may take a long time.

CICS Sysid:  RB23   CICS Applid:  REDBKV23   TermID:  CP89
CIU7016I No details could be found for this query.
F1=Help      F2=          F3=End       F4=Exit      F5=          F6=
F7=          F8=          F9=          F10=         F11=         F12=End
```

Figure 8-17 CINQ - no resources found for application COB

8.4 Migrating CSD resources using CICS CM

This section describes the steps involved in migrating the CSD from CICS TS 2.3 to CICS TS 3.1 using CICS CM and information identified by CICS IA in the previous section. The following are discussed in more detail:

- Resources that need to be changed
- Resources not to be migrated
- Building CICS CM migration package
- Readying the package
- Migrating the package

We recommend that you review Chapter 3, “Overview of CICS CM” on page 37, to understand the functions that CM can provide.

8.4.1 Resources that need to be changed

In the previous section we identified that the following resource definitions need to be changed during the migration:

- The LANGUAGE definition for program REDBK4 will be changed from COBOL to LE370.

> **Note:** The source of the program has been recompiled using Enterprise COBOL for z/OS V3, and the load module is now stored in REBK31.APPL.LOADLIB.

- The DSNAME definition for file REDBOOKF will be changed from REDBK23.REDBKV23.VSAM to REDBK31.REDBKV31.VSAM.
- CICS TS 3.1 has introduced new attributes for the TCPIPSERVICE resource definition. A new MAXDATALEN attribute has been added. We will define a transformation rule to set this value to 1000 KB if it is found to be less.

8.4.2 Resources not to be migrated

In the previous section we identified that the resource definitions associated with Redbook Application 2 (COB) are no longer required. The programs associated with the application are also COBOL/VS. They will not be migrated to the CICS TS 3.1 region.

8.4.3 Building change packages using CICS CM

In this section we discuss building change packages using CICS CM.

Create a CICS CM configuration record

Figure 8-18 shows the CICS Configuration Manager Primary Menu.

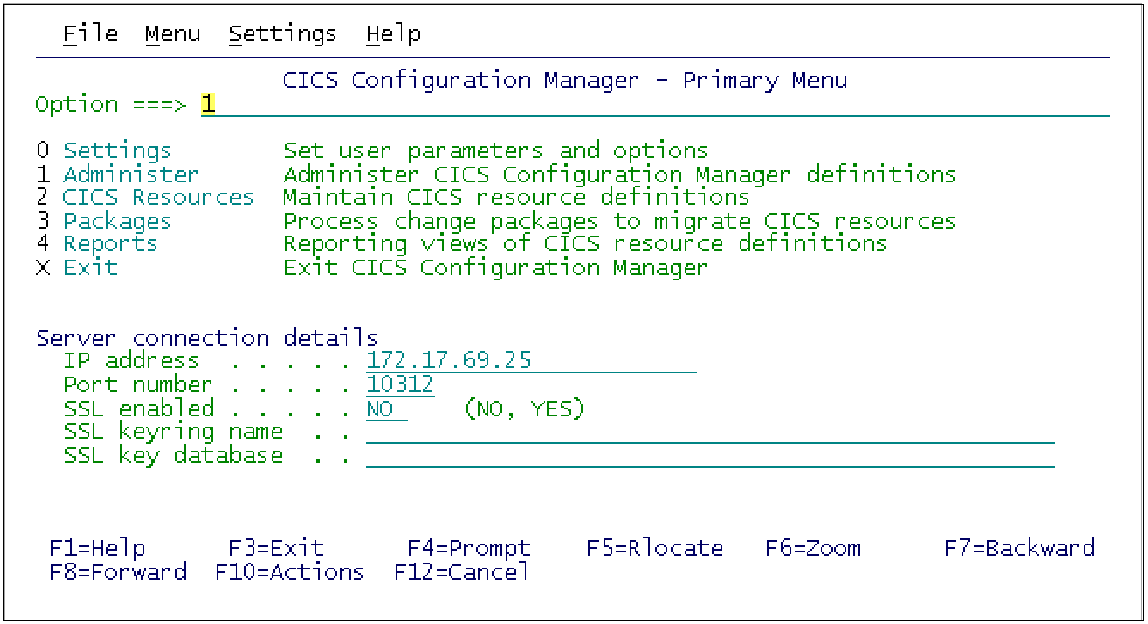


Figure 8-18 CICS CM Primary Menu

The steps are:

1. Select option **1. Administer**.

Figure 8-19 shows the Administration Menu.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
                            Administration Menu                    Enter option
Option ===> 2
 ------------------------------------------------------------------------------
1 System Options        Define system options and features
2 CICS Configurations   Define CICS systems or CICSPlex SM Contexts
3 Migration Schemes     Define migration paths between CICS configurations
4 Approval Profiles     Define change package approval profiles
5 Transform Rules       Define migration transformation rules
6 Exit Points           Manage Exit Points in the CICS CM Server









 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=Actions  F12=Cancel
```

Figure 8-19 CICS CM Administration Menu

2. Select option **2. CICS Configurations**.

We will define two CSD configurations, one for the CICS TS 2.3 region and one for the CICS TS 3.1 region. These configurations will enable CICS CM to locate the respective source CSD files.

Once the migration process has been successful, this configuration record can be deleted.

From the CICS configurations panel enter define redbk23, as shown in Figure 8-20.

```
   File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 Administer                    CICS Configurations                  No matches
 Command ===> define redbk23                                 Scroll ===> PAGE

 Filter R*                                                *            *

    /   Name      Prompt                                  --- Changed ----   ID
 ******************************* Bottom of data ********************************










  F1=Help       F3=Exit       F4=Prompt     F5=Rlocate    F6=Zoom       F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 8-20 CICS CM - define configuration REDBK23

Press Enter and the DEFINE pop-up screen is shown (Figure 8-21). Do not enter a model name. Press Enter and the CICS Configuration screen is displayed. Enter a description, select option **2. CSD File**, and enter the CSD file name for CICS TS 2.3 region, as in Figure 8-22 on page 239.

```
  File  Menu  Settings  Help
 -  ---------------------- CICS Configurations -----------------------  -------
 A |                         DEFINE                                    |
 C |                                                                   | => PAGE
   | Enter the name of the new definition and optional model.          |
 F |                                                                   |
   | Name . . . REDBK23_                                               |
   | Model  . . ________  +                                            |  ID
 * |                                                                   | ********
   |  F1=Help      F3=Exit      F4=Prompt     F6=Resize     F7=Backward |
   |  F8=Forward  F12=Cancel                                           |
   ---------------------------------------------------------------------




 F1=Help       F3=Exit      F4=Prompt     F5=Rlocate    F6=Zoom       F7=Backward
 F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 8-21   CICS CM - Confirm DEFINE

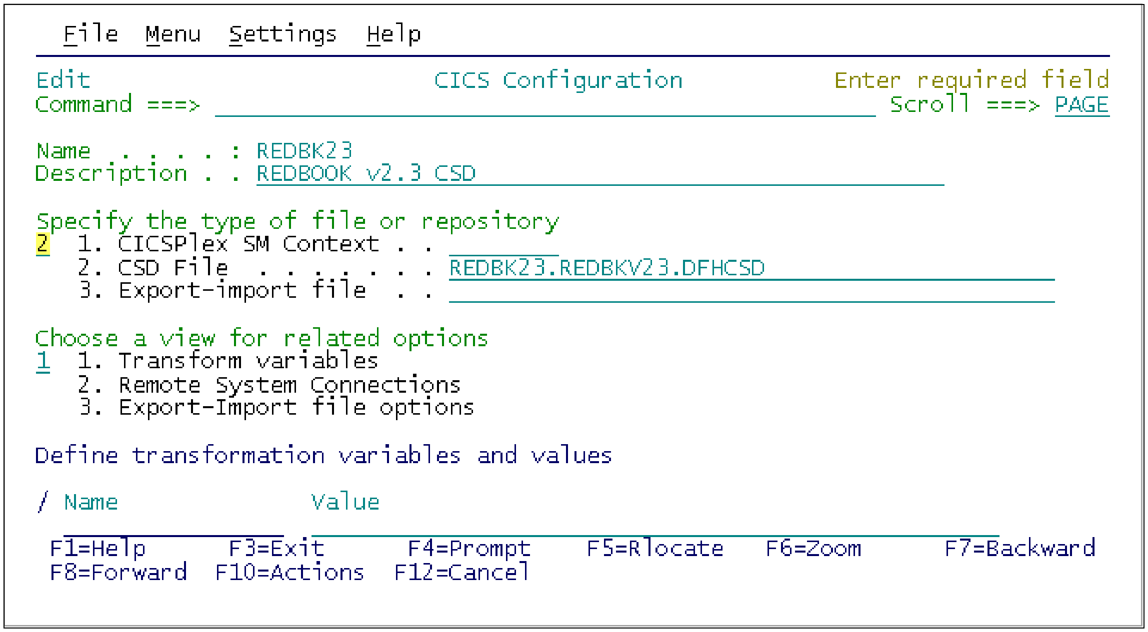


Figure 8-22 CICS CM - Define CSD name to the configuration

Repeat the process for the CICS TS 3.1 region.

Refresh the CICS configurations screen. It should now contain the new entries for REDBK23 and REDBK31. See Figure 8-23.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 Administer                    CICS Configurations                Row 1 to 2 of 2
 Command ===> ______________________________________________   Scroll ===> PAGE

 Filter R*______                                      *_______________  *________

    /   Name      Prompt                              --- Changed ----    ID
        REDBK23                                       2006/06/13 14:23 EYJ
   ___  REDBK31                                       2006/06/13 14:24 EYJ
 ******************************* Bottom of data ********************************

  F1=Help       F3=Exit       F4=Prompt     F5=Rlocate    F6=Zoom       F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 8-23 CICS CM - defined configurations for REDBK23 and REDBK31

Create a migration scheme

The next step is to create a migration scheme. We will create a scheme to migrate from the REDBK23 CSD to the REDBK31 CSD. We will call it REDBKCSD.

Select option **1.3** from the CM Primary Menu and enter define redbkcsd, as in Figure 8-24.

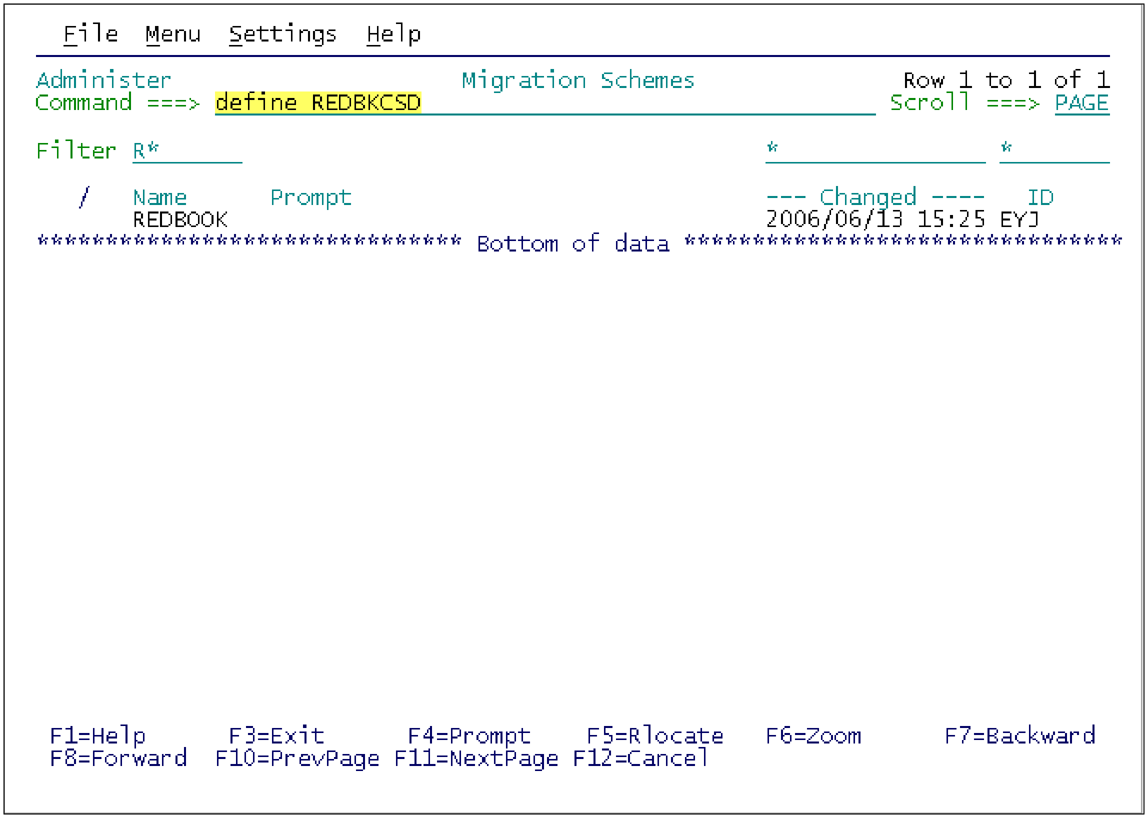


Figure 8-24 CICS CM - Define migration scheme

Do not select a model in the DEFINE pop-up screen. Press Enter and the Migration Scheme screen is displayed.

Enter a description, a source configuration, and a target configuration, as in Figure 8-25. At this stage we have not defined a transformation rule.

```
   File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 Edit                          Migration Scheme                   Row 1 to 1 of 1
 Command ===> ________________________________________________ Scroll ===> PAGE

 Name  . . . . . : REDBKCSD
 Description . . Migrate 2.3 CSD to 3.1 CSD__________________________________

 Approval Processing
 _   Activate

 Choose a view for related options
 1   1. Migration paths
     2. Transform variables

 Define the migration path source and target CICS Configurations

 / Source +  Target +  Transform Rule +
 i REDBK23_  REDBK31_  ________
 ******************************* Bottom of data ********************************







  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 8-25 CICS CM - Define source and target for migration

Create transformation rules

The next step is to create a transform rule for which rule sets are defined. In our scenario, a transform rule called REDBOOK is created, which contains rule sets for resource types *program*, *tcpipservice*, and *file*.

1. Select option **1.5** from the CM primary menu and enter define redbook, as shown in Figure 8-26.

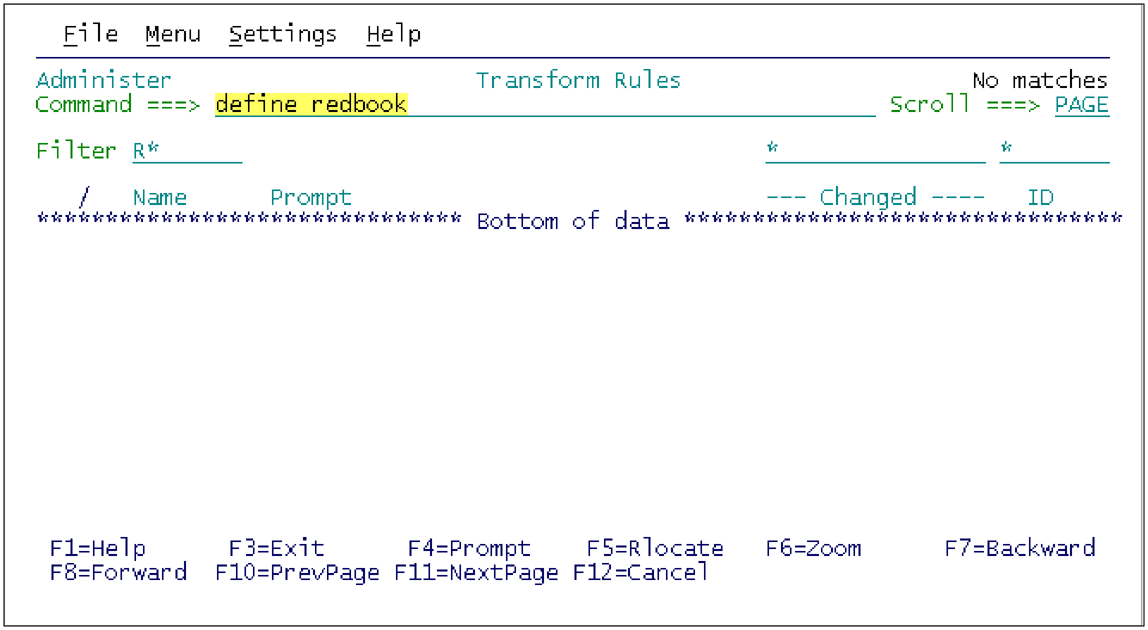


Figure 8-26 CICS CM - Define transformation rules

2. Do not select a model in the DEFINE pop-up screen. Press Enter and the Transform Rule screen is displayed.

3. Enter a description and select i to insert a new rule, as shown in Figure 8-27.

```
  File  Menu  Settings  Help
View                           Transform Rule                     Row 1 to 1 of 1
Command ===>                                                      Scroll ===> PAGE

Name . . . . : REDBOOK
Description . . 2.3 to 3.1 migration transformation


Define transformation rules

            Source     Target                Resource      Resource
/ Scheme    Config     Config     Group      Type          Name
i
******************************* Bottom of data ********************************









 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 8-27 CICS CM - Insert transformation rules

4. Enter a rule to change all programs defined as COBOL to LE/370, as shown in Figure 8-28.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 View                          Transform Rule
 Command ===> __________________________________________________________________

 Name  . . . . . : REDBOOK
 Description . . Change COBOL language to LE370_____________________

 Qualification criteria
   Scheme  . . . . . . REDBKCSD_   +
   Source Config . . REDBK23__   +
   Target Config . . REDBK31__   +
   Group . . . . . . REDBOOK_
   Resource Type . . PROGRAM_____     +
   Resource Name . . *
   Check Field . . . LANGUAGE____     +
   Check Operator    EQ  +
   Check Value . . . COBOL____________________________________________  +

 Choose a Processing Option
 1  1. Transform and continue
    2. Transform and lock field
    3. Transform and lock record
    4. Stop migration of this resource

 Transform Field and Values
   Change Field  . . LANGUAGE____     +
   Change From . . . COBOL____________________________________________  +
   Change To . . . . LE370____________________________________________  +

  F1=Help      F3=Exit      F4=Prompt     F5=Rlocate    F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 8-28 CICS CM - Define PROGRAM transformation rule

5. Enter a rule to rename the VSAM file from REDBK23.REBKV23 to REDBK31.REDBKV31, as in Figure 8-29.

```
  File  Menu  Settings  Help
 --------------------------------------------------------------------------------
 View                          Transform Rule
 Command ===> ____________________________________________________________

 Name  . . . . . : REDBOOK
 Description . . Change file from REDBK23 to REDBK31

 Qualification criteria
   Scheme  . . . . . . REDBKCSD   +
   Source Config . . REDBK23    +
   Target Config . . REDBK31    +
   Group . . . . . . REDBOOK
   Resource Type . . FILE           +
   Resource Name . . *
   Check Field . . . DSNAME         +
   Check Operator    EQ  +
   Check Value . . . *23*.*23*                                            +

 Choose a Processing Option
 1  1. Transform and continue
    2. Transform and lock field
    3. Transform and lock record
    4. Stop migration of this resource

 Transform Field and Values
   Change Field  . . DSNAME         +
   Change From . . . *23*.*23*                                            +
   Change To . . . . *31*.*31*                                            +

  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 8-29   CICS CM - Define FILE transformation rule

Similarly, Figure 8-30 shows the rules for the TCPIPSERVICE resource type. In this case, the value for attribute MAXDATALEN will be changed to 1000 KB if the value found in any of the TCPIPSERVICE definitions in group REDBOOK is found to be less than 1000 KB.

> **Note:** This example shows the potential scope that CICS CM has in terms of being able to introduce and fix a value to new attributes in a migrated CICS environment. MAXDATALEN is a new attribute in the tcpipservice resource definition in CICS TS 3.1.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 Edit                           Transform Rule
 Command ===> _________________________________________________________________

 Name  . . . . . : REDBOOK
 Description . . Make Maxdatalen a minimum of 1000k________________

 Qualification criteria
   Scheme  . . . . . . REDBOOK    +
   Source Config . . . REDBK23    +
   Target Config . . . CCVPLEXE   +
   Group . . . . . . . REDBOOK
   Resource Type . . . TCPIPSERVICE    +
   Resource Name . . . *
   Check Field . . . . MAXDATALEN      +
   Check Operator      LT   +
   Check Value . . . . 1000__________________________________________________  +

 Choose a Processing Option
 1  1. Transform and continue
    2. Transform and lock field
    3. Transform and lock record
    4. Stop migration of this resource

 Transform Field and Values
   Change Field  . . . MAXDATALEN      +
   Change From . . . . *_____________________________________________________  +
   Change To . . . . . 1000__________________________________________________  +

  F1=Help      F3=Exit      F4=Prompt     F5=Rlocate    F6=Zoom      F7=Backward
  F8=Forward  F12=Cancel
```

Figure 8-30 CICS CM - Define TCPIPSERVICE transformation rule

We have now created the required migration rules.

We need to associate the rule with the migration scheme. Select option **1.3** from the Primary CM Menu and edit the REDBKCSD scheme. Add the REDBOOK transformation rule, as shown in Figure 8-31.

```
  File  Menu  Settings  Help
 ─────────────────────────────────────────────────────────────────────────────
 Edit                        Migration Scheme                   REDBKCSD saved
 Command ===> ________________________________________________ Scroll ===> PAGE

 Name  . . . . : REDBKCSD
 Description . . Migrate 2.3 CSD to 3.1 CSD __________________________

 Approval Processing
 _   Activate

 Choose a view for related options
 1   1. Migration paths
     2. Transform variables

 Define the migration path source and target CICS Configurations

 / Source +  Target +  Transform Rule +
   REDBK23   REDBK31   REDBOOK
 ******************************* Bottom of data ********************************










  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 8-31   CICS CM - Add transformation rule REDBOOK to migration scheme

Create a change package

At this point we now define all the candidates or resource definitions into a change package, which will be used in the actual migration. We will create a package to migrate from the TS 2.3 CSD to the TS 3.1 CSD. We will call it REDBKCSD.

1. Select option **3. Packages** from the CM primary menu and enter `define redbkcsd`, as shown in Figure 8-32. Press Enter.

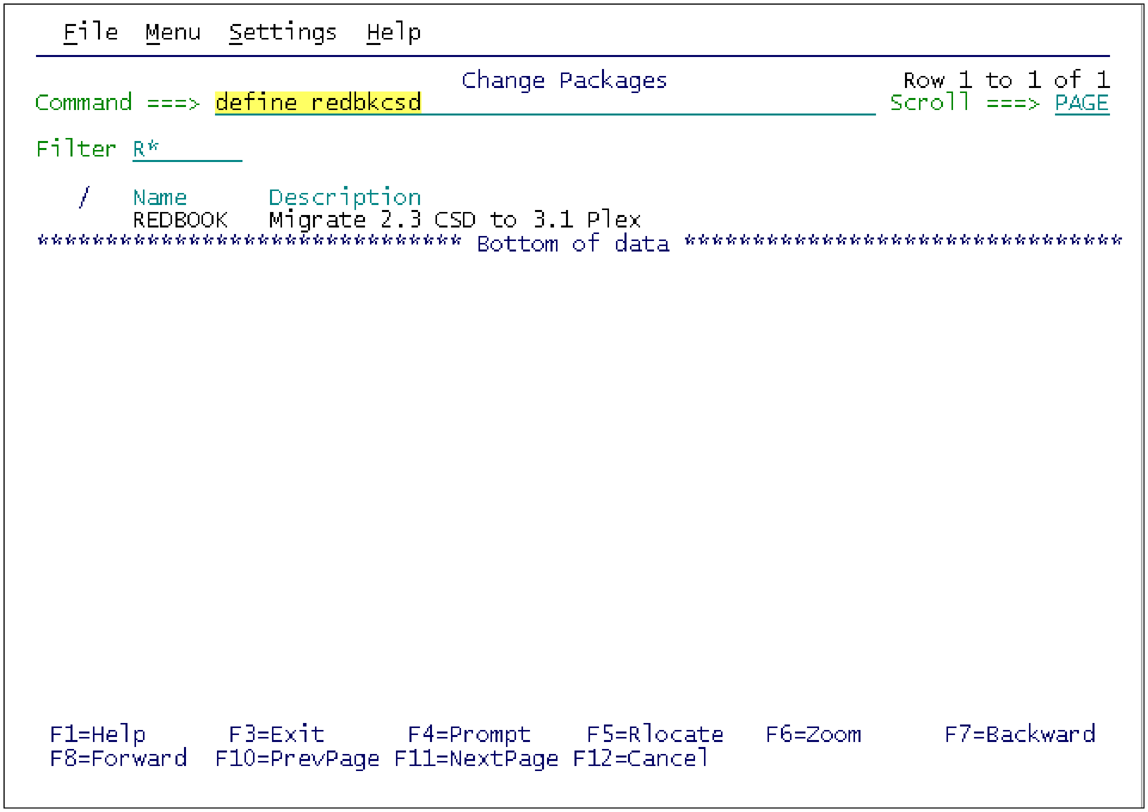


Figure 8-32 CICS CM - Define package REDBKCSD

2. Press Enter in the DEFINE pop-up screen, and the Change Package screen for REDBKCSD is displayed.

3. Enter a description, select option **1. Package**, and enter REDBKCSD for the migration scheme, as shown in Figure 8-33.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 Edit                        Change Package REDBKCSD
 Command ===> _________________________________________________________________

 Name  . . . . . : REDBKCSD
 Description . . Migrate 2.3 CSD to 3.1 CSD________________________

 Change package settings
   Approval profile . . . ________  +
   External reference . . ________

 Choose a processing command and press Enter
 1  1. Package   Package CICS resources into the change package
    2. List      List CICS resources assigned to the change package
    3. Ready     Ready or unready the package for processing
    4. Approve   Approve or disapprove the change package
    5. Migrate   Migrate the change package
    6. Install   Install the package's resources into CICS regions
    7. Newcopy   Newcopy the package's maps, partitionsets or programs
    8. Backout   Backout a previous migrate of the change package
    9. History   Display the change package processing history

      Migration scheme . . . redbkcsd  +  (Required for options 1-7)




  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 8-33 CICS CM - Create package REDBKCSD

4. Press Enter and the resource selection screen is displayed. In this scenario we are migrating one group called REDBOOK. Enter REDBOOK in the group filter field, as shown in Figure 8-34.

> **Note:** The process described here has shown CICS CM packaging up the resources for one group, REDBOOK. CICS CM can also perform this task in other ways, and the following steps show an alternative method for situations where more than one group is to be migrated at the same time (as would be the case in an upgrade).

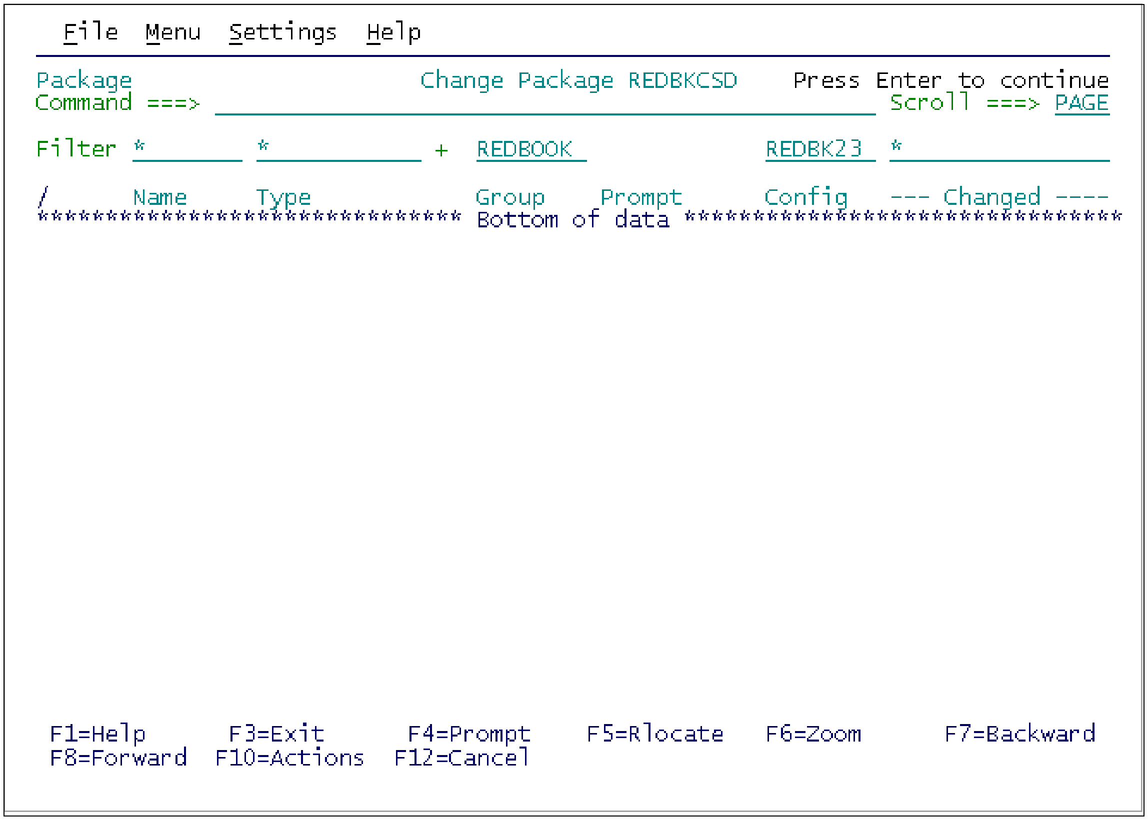


Figure 8-34 CICS CM - filter screen for change package REDBKCSD

5. Press Enter to continue. The resources for group REDBOOK is listed as shown in Figure 8-35.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
Package                     Change Package REDBKCSD               Row 1 to 22 of 41
Command ===> __________________________________________________  Scroll ===> PAGE

Filter * _____ * _________  + REDBOOK              REDBK23  * __________

/      Name     Type            Group    Prompt      Config   --- Changed ----
       CCVR     TRANSACTION     REDBOOK              REDBK23  2006/06/12 10:16
______ CCVSREMP PROGRAM         REDBOOK              REDBK23  2006/06/06 12:07
______ CDCB001# PROGRAM         REDBOOK              REDBK23  2006/06/06 11:53
______ CDCB0010 PROGRAM         REDBOOK              REDBK23  2006/06/12 15:42
______ CDCB0020 PROGRAM         REDBOOK              REDBK23  2006/06/06 11:53
______ CDCB0510 PROGRAM         REDBOOK              REDBK23  2006/06/06 11:53
______ CDCB0710 PROGRAM         REDBOOK              REDBK23  2006/06/06 11:53
______ CICB0010 PROGRAM         REDBOOK              REDBK23  2006/06/06 11:53
______ CICB0020 PROGRAM         REDBOOK              REDBK23  2006/06/06 11:53
______ CICB0030 PROGRAM         REDBOOK              REDBK23  2006/06/06 11:53
______ CICB0050 PROGRAM         REDBOOK              REDBK23  2006/06/06 11:53
______ COBOLVS1 PROGRAM         REDBOOK              REDBK23  2006/06/06 11:53
______ COBOLVS2 PROGRAM         REDBOOK              REDBK23  2006/06/06 11:53
______ CSCB0010 PROGRAM         REDBOOK              REDBK23  2006/06/06 11:53
______ CSCB0030 PROGRAM         REDBOOK              REDBK23  2006/06/06 11:53
______ CSCB0200 PROGRAM         REDBOOK              REDBK23  2006/06/06 11:53
______ DB7P     DB2CONN         REDBOOK              REDBK23  2006/06/06 16:15
______ RDBA     TRANSACTION     REDBOOK              REDBK23  2006/06/07 13:37
______ RDBB     TRANSACTION     REDBOOK              REDBK23  2006/06/07 13:37
______ RDBC     TRANSACTION     REDBOOK              REDBK23  2006/06/07 13:37
______ RDBD     TRANSACTION     REDBOOK              REDBK23  2006/06/07 13:37
______ RDBE     TRANSACTION     REDBOOK              REDBK23  2006/06/07 13:37
 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=Actions  F12=Cancel
```

Figure 8-35 CICS CM - resources in group REDBOOK

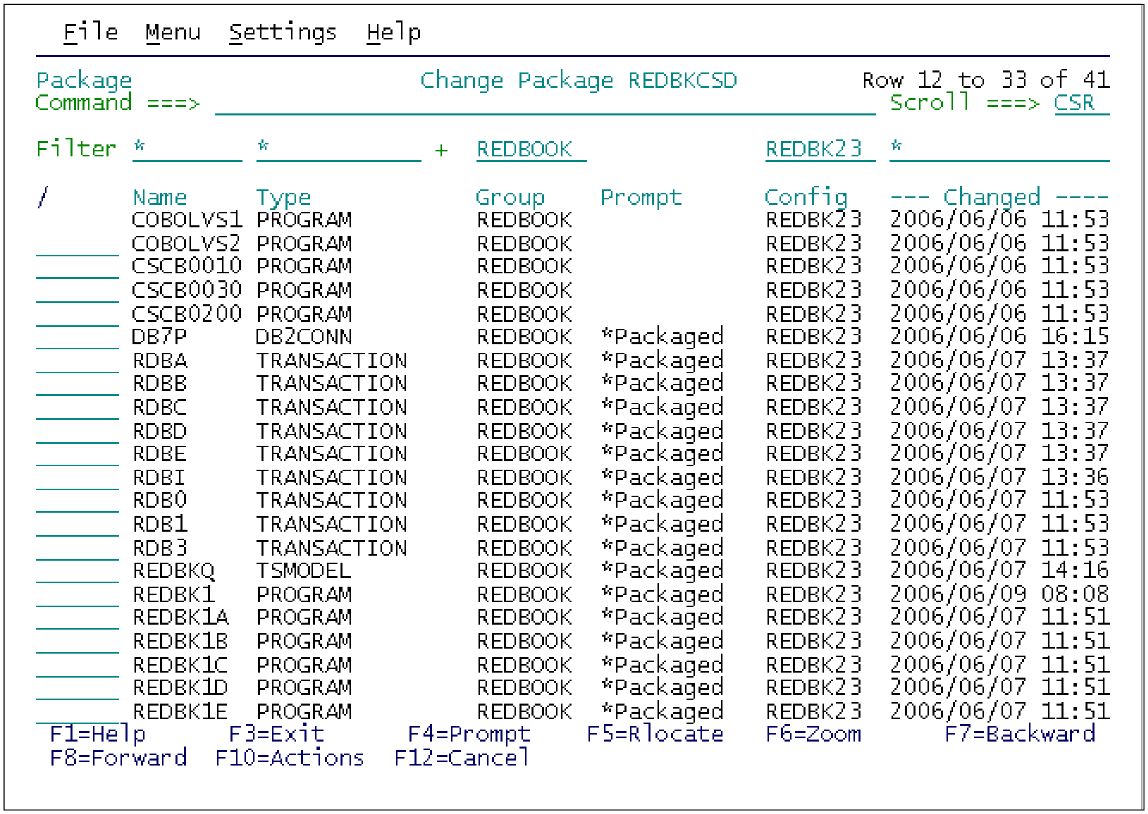
At this stage we can select the resources that we want to package for migration. In the previous section we identified, through the use of CICS IA, that the COBOL/VS programs with a program prefix of CSC, CDC, CIC, or COB are no longer required. Therefore we will not package them.

1. Enter a p in the line action field for the resources we want packaged, as shown in Figure 8-36.

```
  File  Menu  Settings  Help
Package                     Change Package REDBKCSD          Row 12 to 33 of 41
Command ===>                                                  Scroll ===> CSR

Filter *        *            +  REDBOOK            REDBK23  *

/       Name     Type           Group    Prompt     Config   --- Changed ----
        COBOLVS1 PROGRAM        REDBOOK             REDBK23  2006/06/06 11:53
        COBOLVS2 PROGRAM        REDBOOK             REDBK23  2006/06/06 11:53
        CSCB0010 PROGRAM        REDBOOK             REDBK23  2006/06/06 11:53
        CSCB0030 PROGRAM        REDBOOK             REDBK23  2006/06/06 11:53
        CSCB0200 PROGRAM        REDBOOK             REDBK23  2006/06/06 11:53
        DB7P     DB2CONN        REDBOOK             REDBK23  2006/06/06 16:15
P       RDBA     TRANSACTION    REDBOOK             REDBK23  2006/06/07 13:37
P       RDBB     TRANSACTION    REDBOOK             REDBK23  2006/06/07 13:37
P       RDBC     TRANSACTION    REDBOOK             REDBK23  2006/06/07 13:37
P       RDBD     TRANSACTION    REDBOOK             REDBK23  2006/06/07 13:37
P       RDBE     TRANSACTION    REDBOOK             REDBK23  2006/06/07 13:37
P       RDBI     TRANSACTION    REDBOOK             REDBK23  2006/06/07 13:36
P       RDB0     TRANSACTION    REDBOOK             REDBK23  2006/06/07 11:53
P       RDB1     TRANSACTION    REDBOOK             REDBK23  2006/06/07 11:53
P       RDB3     TRANSACTION    REDBOOK             REDBK23  2006/06/07 11:53
P       REDBKQ   TSMODEL        REDBOOK             REDBK23  2006/06/07 14:16
P       REDBK1   PROGRAM        REDBOOK             REDBK23  2006/06/09 08:08
P       REDBK1A  PROGRAM        REDBOOK             REDBK23  2006/06/07 11:51
P       REDBK1B  PROGRAM        REDBOOK             REDBK23  2006/06/07 11:51
P       REDBK1C  PROGRAM        REDBOOK             REDBK23  2006/06/07 11:51
P       REDBK1D  PROGRAM        REDBOOK             REDBK23  2006/06/07 11:51
P       REDBK1E  PROGRAM        REDBOOK             REDBK23  2006/06/07 11:51
 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=Actions  F12=Cancel
```

Figure 8-36 CICS CM - Select resources to be packaged in REDBKCSD

2. Press Enter. The change package screen is refreshed to show which resources have been packaged, as shown in Figure 8-37.



```
   File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 Package                     Change Package REDBKCSD           Row 12 to 33 of 41
 Command ===> ______________________________________________  Scroll ===> CSR

 Filter *       *           + REDBOOK          REDBK23  *

 /      Name      Type            Group    Prompt     Config   --- Changed ----
        COBOLVS1  PROGRAM         REDBOOK             REDBK23  2006/06/06 11:53
 ______ COBOLVS2  PROGRAM         REDBOOK             REDBK23  2006/06/06 11:53
 ______ CSCB0010  PROGRAM         REDBOOK             REDBK23  2006/06/06 11:53
 ______ CSCB0030  PROGRAM         REDBOOK             REDBK23  2006/06/06 11:53
 ______ CSCB0200  PROGRAM         REDBOOK             REDBK23  2006/06/06 11:53
 ______ DB7P      DB2CONN         REDBOOK  *Packaged  REDBK23  2006/06/06 16:15
 ______ RDBA      TRANSACTION     REDBOOK  *Packaged  REDBK23  2006/06/07 13:37
 ______ RDBB      TRANSACTION     REDBOOK  *Packaged  REDBK23  2006/06/07 13:37
 ______ RDBC      TRANSACTION     REDBOOK  *Packaged  REDBK23  2006/06/07 13:37
 ______ RDBD      TRANSACTION     REDBOOK  *Packaged  REDBK23  2006/06/07 13:37
 ______ RDBE      TRANSACTION     REDBOOK  *Packaged  REDBK23  2006/06/07 13:37
 ______ RDBI      TRANSACTION     REDBOOK  *Packaged  REDBK23  2006/06/07 13:36
 ______ RDB0      TRANSACTION     REDBOOK  *Packaged  REDBK23  2006/06/07 11:53
 ______ RDB1      TRANSACTION     REDBOOK  *Packaged  REDBK23  2006/06/07 11:53
 ______ RDB3      TRANSACTION     REDBOOK  *Packaged  REDBK23  2006/06/07 11:53
 ______ REDBKQ    TSMODEL         REDBOOK  *Packaged  REDBK23  2006/06/07 14:16
 ______ REDBK1    PROGRAM         REDBOOK  *Packaged  REDBK23  2006/06/09 08:08
 ______ REDBK1A   PROGRAM         REDBOOK  *Packaged  REDBK23  2006/06/07 11:51
 ______ REDBK1B   PROGRAM         REDBOOK  *Packaged  REDBK23  2006/06/07 11:51
 ______ REDBK1C   PROGRAM         REDBOOK  *Packaged  REDBK23  2006/06/07 11:51
 ______ REDBK1D   PROGRAM         REDBOOK  *Packaged  REDBK23  2006/06/07 11:51
 ______ REDBK1E   PROGRAM         REDBOOK  *Packaged  REDBK23  2006/06/07 11:51
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 8-37   CICS CM - resources selected for packaging in REDBKCSD

8.4.4 Ready the package

We are now in a position to *ready* the package for migration.

1. Select option **3. Ready** from the Change Package action menu, as shown in Figure 8-38.

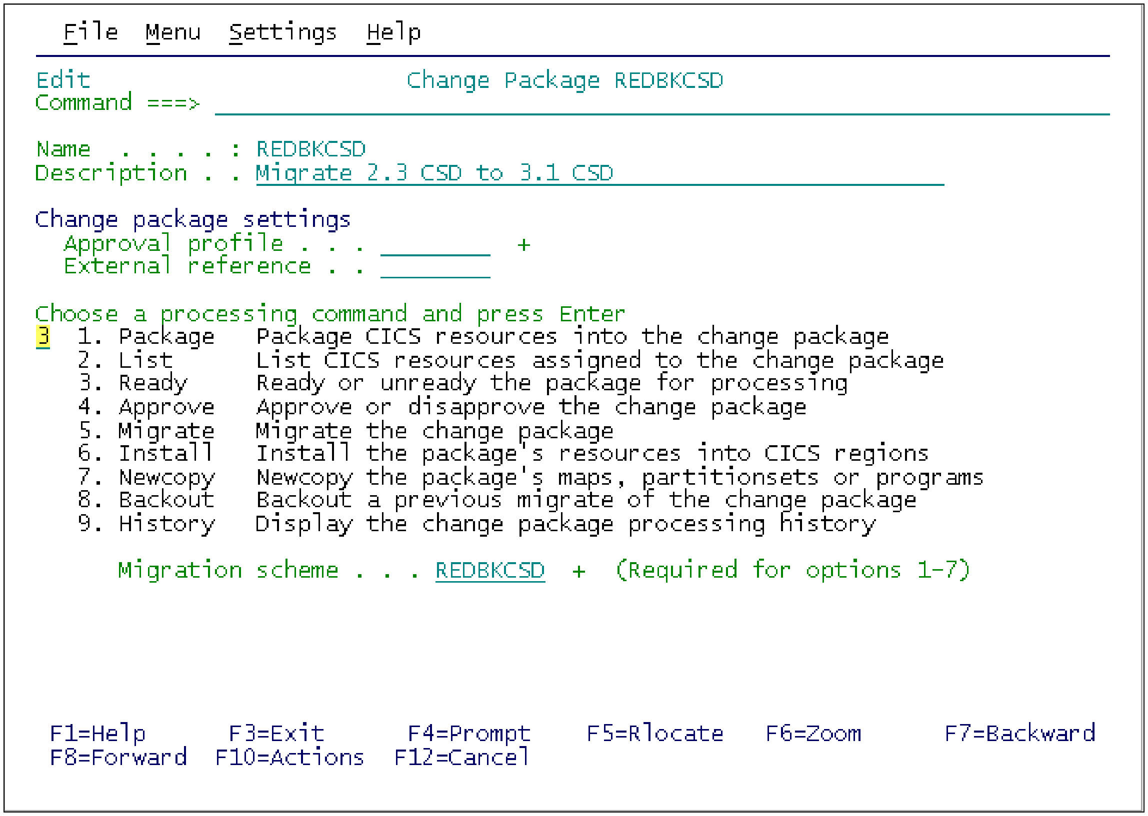


Figure 8-38 CICS CM - Select ready package REDBKCSD

2. Select processing option **1. Ready** and execution mode **1. Foreground** processing, as shown in Figure 8-39. Press Enter.

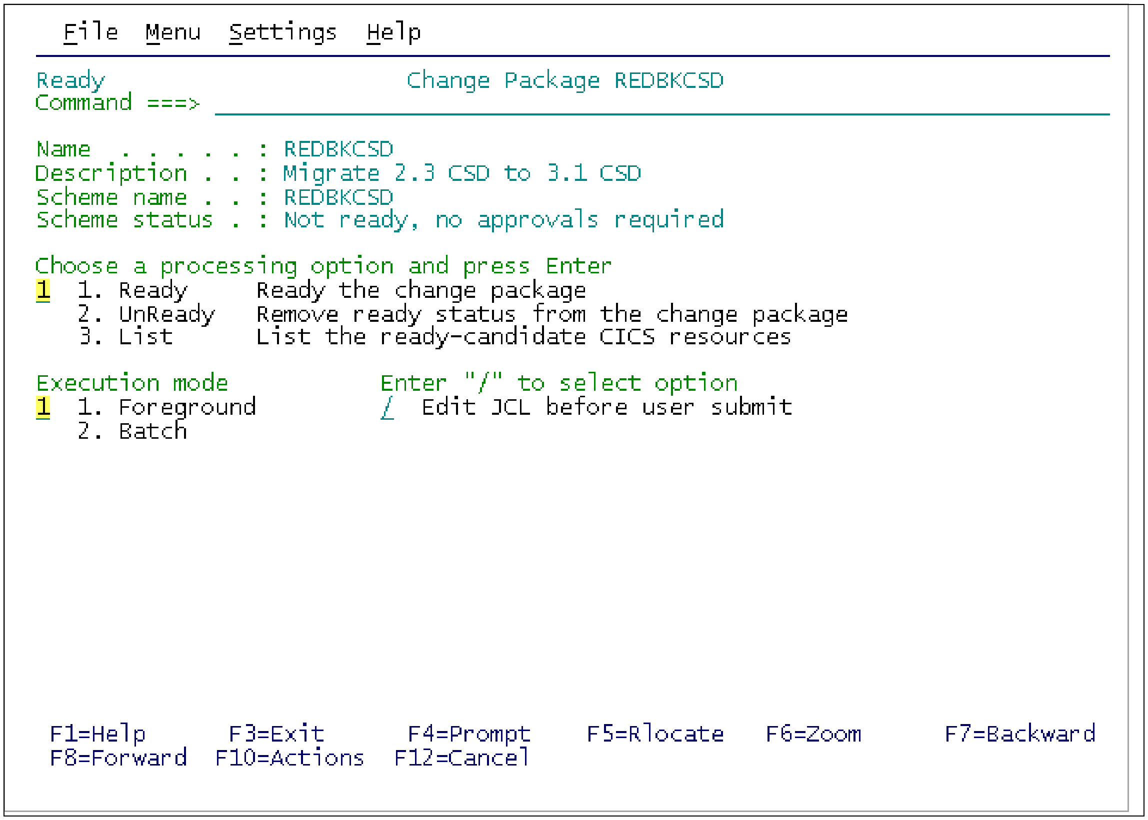


Figure 8-39 CICS CM - Ready package REDBKCSD

3. Press Enter. The package has now been *readied*.

To review the list of resources in the readied package select option **3. List** in the Change Package screen. Figure 8-40 shows the list of readied resources.

```
   File  Menu  Settings  Help
------------------------------------------------------------------------------
List                         Change Package REDBKCSD              Row 1 to 22 of 23
Command ===> ________________________________________________  Scroll ===> CSR

Filter *_______  *____________  +  *_______

/      Name      Type            Group     Status                        Config
       DB7P      DB2CONN         REDBOOK   Ready                         REDBK23
_____  RDBA      TRANSACTION     REDBOOK   Ready                         REDBK23
_____  RDBB      TRANSACTION     REDBOOK   Ready                         REDBK23
_____  RDBC      TRANSACTION     REDBOOK   Ready                         REDBK23
_____  RDBD      TRANSACTION     REDBOOK   Ready                         REDBK23
_____  RDBE      TRANSACTION     REDBOOK   Ready                         REDBK23
_____  RDBI      TRANSACTION     REDBOOK   Ready                         REDBK23
_____  RDB0      TRANSACTION     REDBOOK   Ready                         REDBK23
_____  RDB1      TRANSACTION     REDBOOK   Ready                         REDBK23
_____  RDB3      TRANSACTION     REDBOOK   Ready                         REDBK23
_____  REDBKQ    TSMODEL         REDBOOK   Ready                         REDBK23
_____  REDBK1    PROGRAM         REDBOOK   Ready                         REDBK23
_____  REDBK1A   PROGRAM         REDBOOK   Ready                         REDBK23
_____  REDBK1B   PROGRAM         REDBOOK   Ready                         REDBK23
_____  REDBK1C   PROGRAM         REDBOOK   Ready                         REDBK23
_____  REDBK1D   PROGRAM         REDBOOK   Ready                         REDBK23
_____  REDBK1E   PROGRAM         REDBOOK   Ready                         REDBK23
_____  REDBK2    PROGRAM         REDBOOK   Ready                         REDBK23
_____  REDBK3    PROGRAM         REDBOOK   Ready                         REDBK23
_____  REDBK4    PROGRAM         REDBOOK   Ready                         REDBK23
_____  REDBK5    PROGRAM         REDBOOK   Ready                         REDBK23
_____  REDBOOKF  FILE            REDBOOK   Ready                         REDBK23
 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate    F6=Zoom      F7=Backward
 F8=Forward  F10=Actions  F12=Cancel
```

Figure 8-40 CICS CM - resources for change package REDBKCSD

8.4.5 Migrate the package

We are now ready to migrate the selected resources using the migration scheme REDBKCSD.

1. Select option **5. Migrate** from the Change Package action menu, as shown in Figure 8-41.

```
  File  Menu  Settings  Help
Edit                          Change Package REDBKCSD
Command ===> ________________________________________________

Name  . . . . : REDBKCSD
Description . . Migrate 2.3 CSD to 3.1 CSD

Change package settings
  Approval profile . . . ________  +
  External reference . . ________

Choose a processing command and press Enter
5  1. Package   Package CICS resources into the change package
   2. List      List CICS resources assigned to the change package
   3. Ready     Ready or unready the package for processing
   4. Approve   Approve or disapprove the change package
   5. Migrate   Migrate the change package
   6. Install   Install the package's resources into CICS regions
   7. Newcopy   Newcopy the package's maps, partitionsets or programs
   8. Backout   Backout a previous migrate of the change package
   9. History   Display the change package processing history

      Migration scheme . . . REDBKCSD  +  (Required for options 1-7)



 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=Actions  F12=Cancel
```

Figure 8-41 CICS CM - Select migrate package REDBKCSD

2. Select processing option **1. Migrate** and execution mode **1. Foreground** processing, as shown in Figure 8-42. Press Enter.

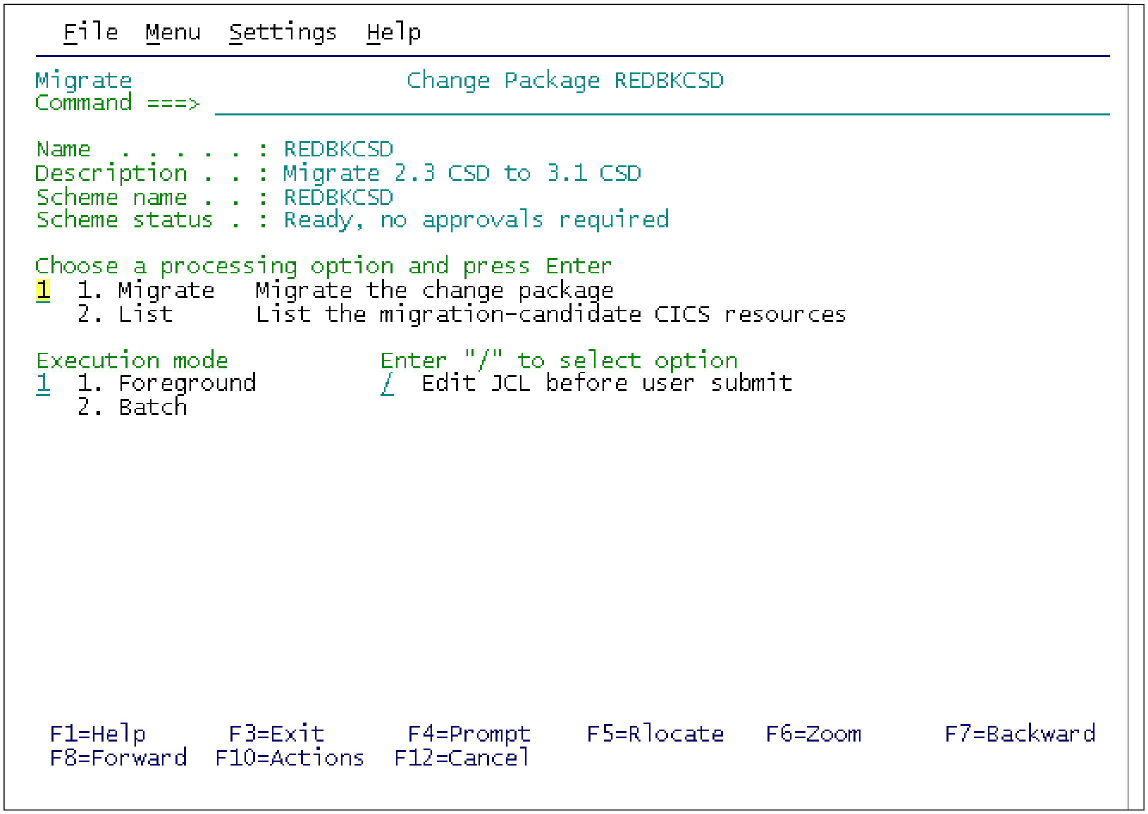


Figure 8-42 CICS CM - migrating package REDBKCSD

We have now migrated the required resources with modifications from group REDBOOK in TS 2.3 to the same group in TS 3.1 using CICS CM.

8.4.6 Using CM’s COPY function and EXIT to migrate

So far we have shown the preparation and execution of a change package, transformation rules, migration scheme, and migrate function to move resource definitions from a source CSD to a target CSD in a migrated environment. There is another CM function that can be used to achieve the same result. This is through the use of the COPY command and a resource attribute migration exit. A further discussion on this alternate approach and example EXIT code is found in 8.4.6, “Using CM’s COPY function and EXIT to migrate” on page 259.

8.5 Using CICS CM and CICS IA to verify migration

This section describes the steps involved in reviewing the migration to CICS TS 3.1 using CICS CM and CICS IA. The following approaches are discussed in more detail:

- Review CICS CM transformation rule changes.
- Use CICS CM to compare source and target CSDs.
- Review COBOL/VS program usage using CICS IA.

8.5.1 Review CICS CM transformation rule changes

We can now review that the CM rules we created for the migration actually worked. We can do this using CICS CM.

1. Select option **2. CICS Resources** from the CM Primary Menu. Select the TS 3.1 region configuration, REDBK31, as shown in Figure 8-43.

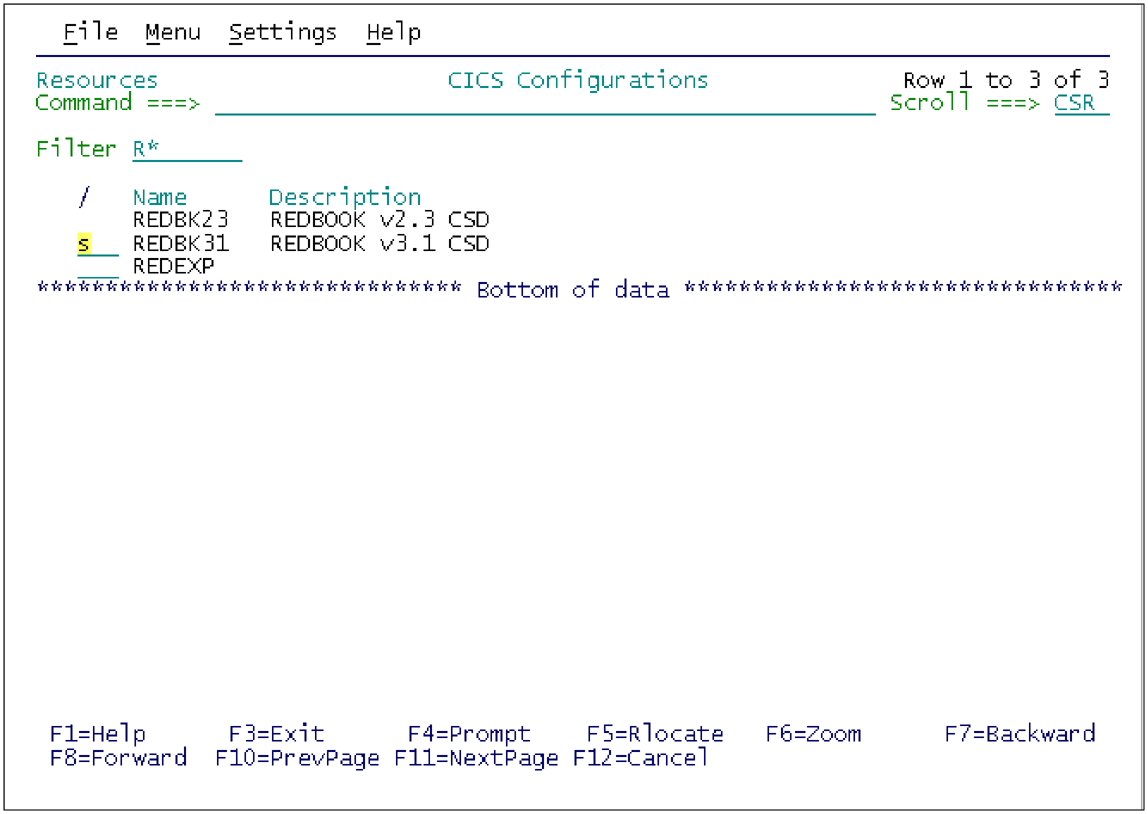


Figure 8-43 Select REDBK31 configuration

2. Press Enter and the REDBK31 **CICS Resources** screen is displayed. Enter redbook in the group filter field, as shown in Figure 8-44.

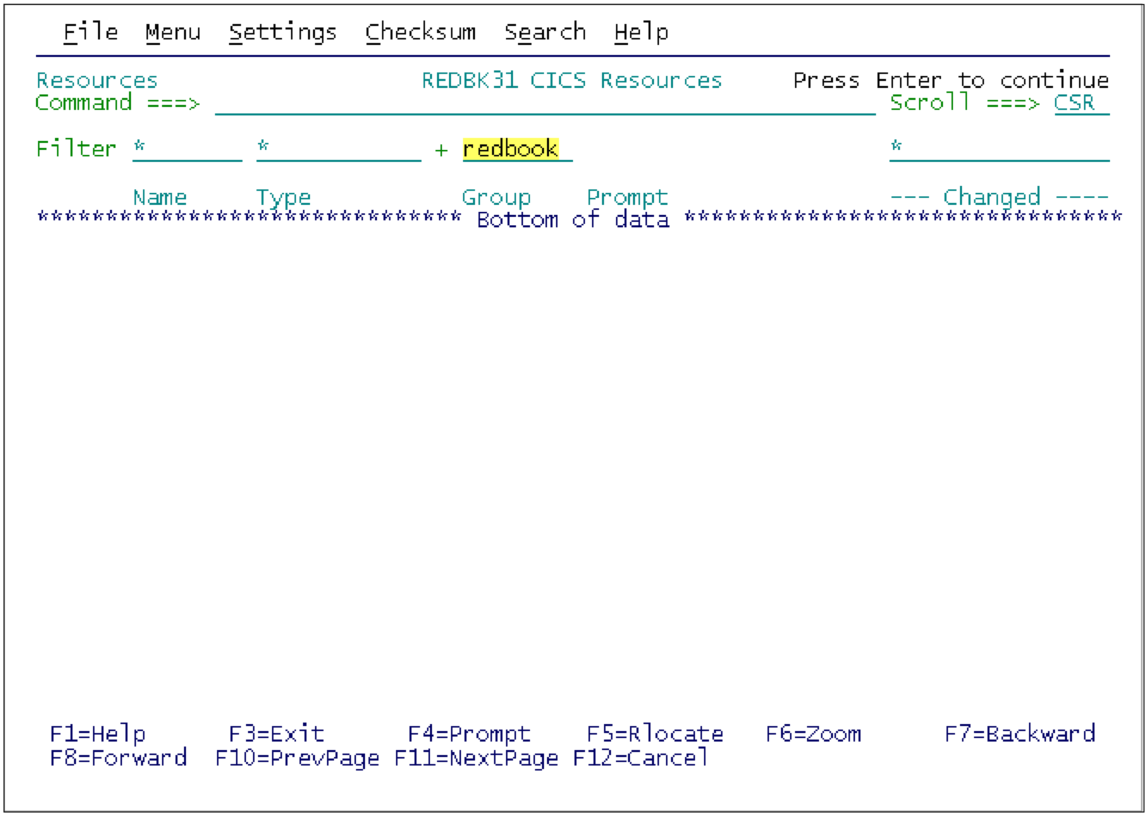
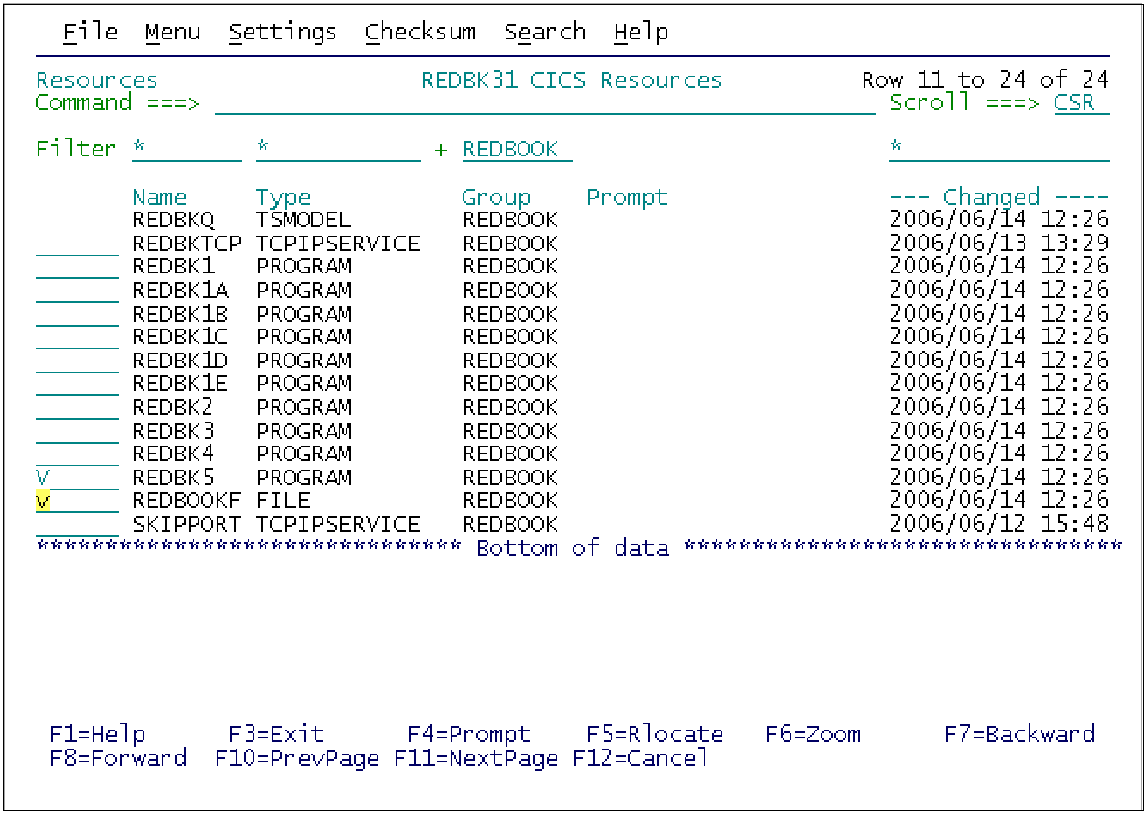


Figure 8-44 Select REDBOOK group

3. Press Enter and the list of CICS resources is displayed. Select program resource **REDBK5**, as shown in Figure 8-45.



```
  File  Menu  Settings  Checksum  Search  Help
-----------------------------------------------------------------------------
Resources                    REDBK31 CICS Resources              Row 11 to 24 of 24
Command ===> ______________________________________________________ Scroll ===> CSR

Filter * _______ * __________ + REDBOOK_                       * _______________

       Name     Type         Group    Prompt               --- Changed ----
       REDBKQ   TSMODEL      REDBOOK                       2006/06/14 12:26
______ REDBKTCP TCPIPSERVICE REDBOOK                       2006/06/13 13:29
______ REDBK1   PROGRAM      REDBOOK                       2006/06/14 12:26
______ REDBK1A  PROGRAM      REDBOOK                       2006/06/14 12:26
______ REDBK1B  PROGRAM      REDBOOK                       2006/06/14 12:26
______ REDBK1C  PROGRAM      REDBOOK                       2006/06/14 12:26
______ REDBK1D  PROGRAM      REDBOOK                       2006/06/14 12:26
______ REDBK1E  PROGRAM      REDBOOK                       2006/06/14 12:26
______ REDBK2   PROGRAM      REDBOOK                       2006/06/14 12:26
______ REDBK3   PROGRAM      REDBOOK                       2006/06/14 12:26
______ REDBK4   PROGRAM      REDBOOK                       2006/06/14 12:26
V_____ REDBK5   PROGRAM      REDBOOK                       2006/06/14 12:26
V_____ REDBOOKF FILE         REDBOOK                       2006/06/14 12:26
______ SKIPPORT TCPIPSERVICE REDBOOK                       2006/06/12 15:48
******************************** Bottom of data ********************************

 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 8-45 CICS resource list for group REDBOOK

4. Press Enter and the resource definition for program REDBK5 is displayed, as shown in Figure 8-46.

> **Note:** The language definition is now LE370.

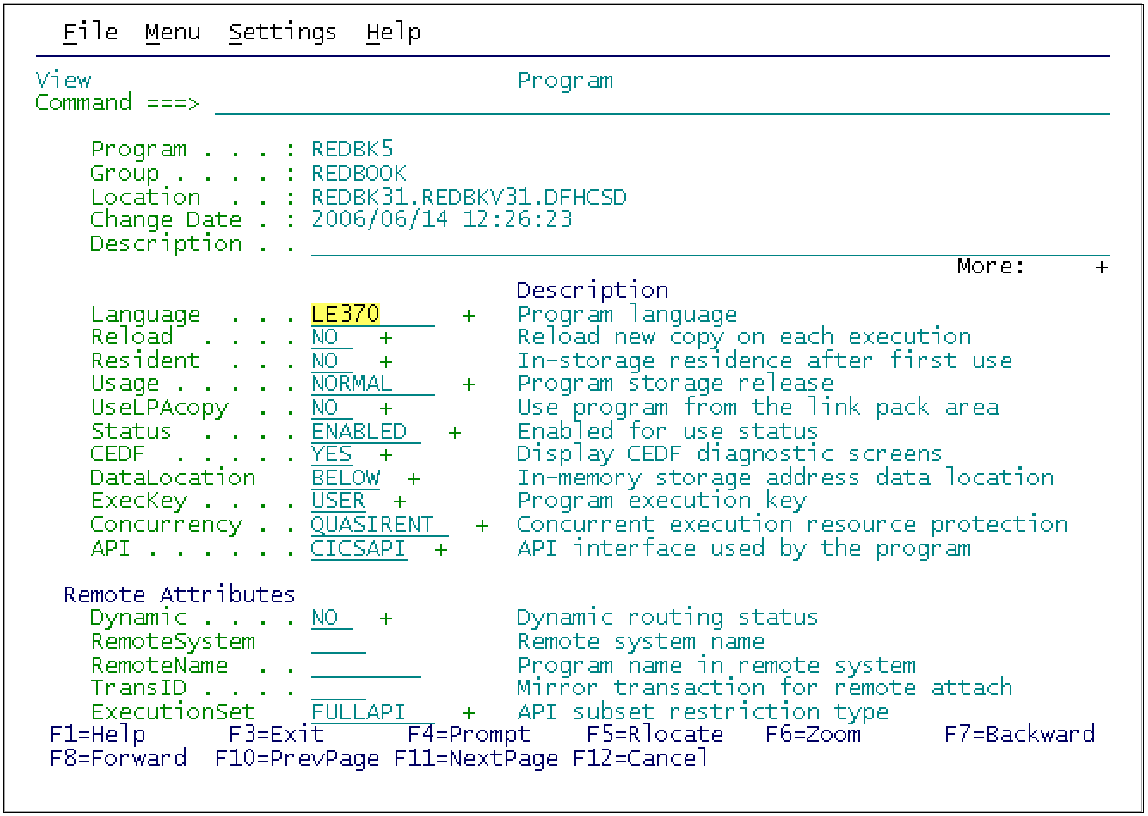


Figure 8-46 CICS CM view for REDBK5 program definition

Figure 8-47 on page 264 shows that DSNAME for file resource REDBOOKF is now REDBK31.REDBKV31.VSAM.

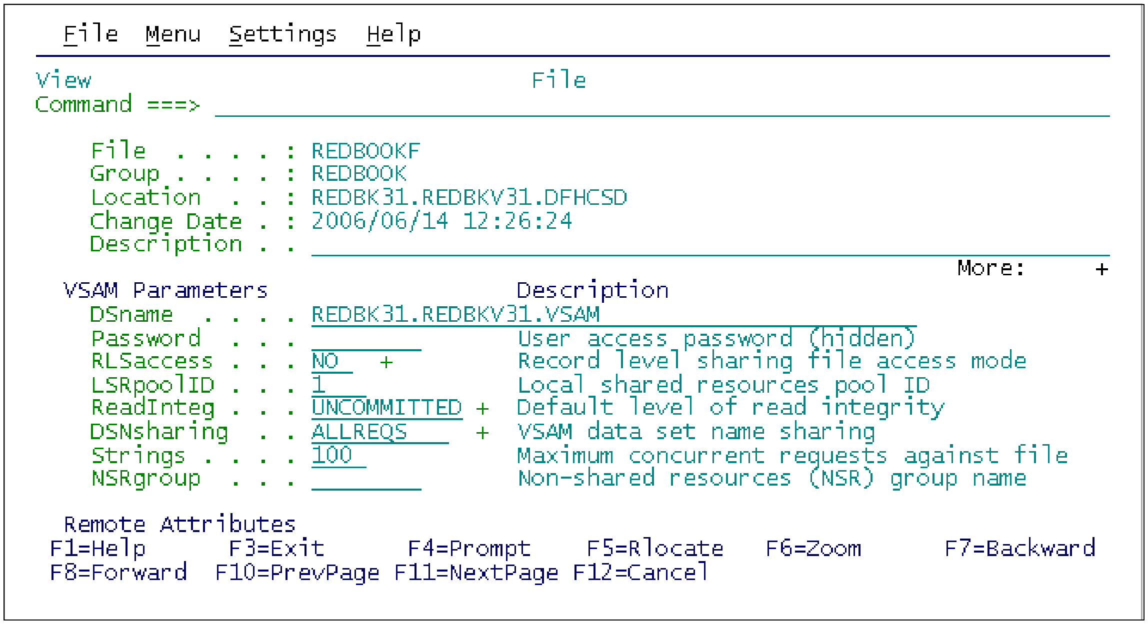


Figure 8-47 CICS CM view for VSAM file REDBOOKF definition

8.5.2 Using the CICS CM COMPARE function

The CM compare function enables the comparison of source and target CSD contents. This can be particularly useful if resources are changed, removed, or added during or post migration using change packages, transformation rules, or a migration exit. The compare provides a verification that changes have occurred.

The use of this function is demonstrated in detail in 9.4.5, “Using CICS CM to compare source and target repositories” on page 314.

8.5.3 Review COBOL/VS program usage using CICS IA

Before installing our definitions we can now use CICS IA to scan our TS 3.2 application load library, REDBK31.APPL.LOADLIB.

> **Note:** All COBOL/VS programs have been recompiled using Enterprise COBOL for z/OS V3 and are stored in data set REBK31.APPL.LOADLIB.

We will use the CSECT scanner to identify the compiler used for all load modules in REDBK31.APPL.LOADLIB. To use the CSECT scanner refer to “Running the CSECT scanner” on page 212.

We can now run some queries against the V_CIU4_CSECT_TRANS view.

Example 8-18 shows the compilers for all modules with a prefix of RED in our TS 3.1 load library, REDBK31.APPL.LOADLIB.

Example 8-18 Query to show program compilers for Redbook Application 1

```
---------+---------+---------+---------+---------+---------+---------+---------+
--Show me the Compiler used for program with a prefix of 'RED' in
--REDBK31.APPL.LOADLIB using the CSECT scanner view.
  SELECT DISTINCT PROGRAM, DESCRIPTION , TRAN_1_NAME
  FROM V_CIU4_CSECT_TRANS
  WHERE DSNAME='REDBK31.APPL.LOADLIB'
  AND CSECT_NAME LIKE 'RED%';
---------+---------+---------+---------+---------+---------+---------+---------+
PROGRAM   DESCRIPTION                     TRAN_1_NAME
---------+---------+---------+---------+---------+---------+---------+---------+
REDBK1    C/C++ z/OS R5                   5694A01
REDBK1A   C/C++ z/OS R5                   5694A01
REDBK1B   C/C++ z/OS R5                   5694A01
REDBK1C   C/C++ z/OS R5                   5694A01
REDBK1D   C/C++ z/OS R5                   5694A01
REDBK1E   C/C++ z/OS R5                   5694A01
REDBK2    PL/I for MVS AND VM V1          5688-235
REDBK3    C/C++ z/OS R5                   5694A01
REDBK4    Enterprise COBOL for z/OS V3    5655G5300
REDBK5    C/C++ z/OS R5                   5694A01
DSNE610I NUMBER OF ROWS DISPLAYED IS 10
DSNE612I DATA FOR COLUMN HEADER DESCRIPTION COLUMN NUMBER 2 WAS TRUNCATED
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
```

We can see that program REDBK4 has now been compiled with Enterprise COBOL for z/OS V3.

Example 8-19 shows us all COBOL program in all scanned load libraries.

Example 8-19 Query to show all COBOL compiled programs in all scanned load libraries

```
--Show me the COBOL compiled programs in all load libraries
--using the CSECT scanner view.
  SELECT DISTINCT DSNAME, PROGRAM, DESCRIPTION
  FROM V_CIU4_CSECT_TRANS
  WHERE DESCRIPTION LIKE '%COBOL%'
  ORDER BY 2;
---------+---------+---------+---------+---------+---------+---------+---------+
DSNAME                          PROGRAM   DESCRIPTION
---------+---------+---------+---------+---------+---------+---------+---------+
REDBK23.APPL.LOADLIB            CDCB001#  OS/VS COBOL R2M3 R2M4 (VSR1)
REDBK23.APPL.LOADLIB            CDCB0010  OS/VS COBOL R2M3 R2M4 (VSR1)
REDBK23.APPL.LOADLIB            CDCB0020  OS/VS COBOL R2M3 R2M4 (VSR1)
REDBK23.APPL.LOADLIB            CDCB0510  OS/VS COBOL R2M3 R2M4 (VSR1)
REDBK23.APPL.LOADLIB            CDCB0710  OS/VS COBOL R2M3 R2M4 (VSR1)
REDBK23.APPL.LOADLIB            CICB0010  OS/VS COBOL R2M3 R2M4 (VSR1)
REDBK23.APPL.LOADLIB            CICB0020  OS/VS COBOL R2M3 R2M4 (VSR1)
REDBK23.APPL.LOADLIB            CICB0030  OS/VS COBOL R2M3 R2M4 (VSR1)
REDBK23.APPL.LOADLIB            CICB0050  OS/VS COBOL R2M3 R2M4 (VSR1)
REDBK23.APPL.LOADLIB            COBOLVS1  OS/VS COBOL R2M3 R2M4 (VSR1)
REDBK23.APPL.LOADLIB            COBOLVS2  OS/VS COBOL R2M3 R2M4 (VSR1)
REDBK23.APPL.LOADLIB            CSCB0010  OS/VS COBOL R2M3 R2M4 (VSR1)
REDBK23.APPL.LOADLIB            CSCB0030  OS/VS COBOL R2M3 R2M4 (VSR1)
REDBK23.APPL.LOADLIB            CSCB0200  OS/VS COBOL R2M3 R2M4 (VSR1)
REDBK23.APPL.LOADLIB            REDBK4    OS/VS COBOL R2M3 R2M4 (VSR1)
REDBK31.APPL.LOADLIB            REDBK4    Enterprise COBOL for z/OS V3
DSNE610I NUMBER OF ROWS DISPLAYED IS 17
DSNE612I DATA FOR COLUMN HEADER DSNAME COLUMN NUMBER 1 WAS TRUNCATED
DSNE612I DATA FOR COLUMN HEADER DESCRIPTION COLUMN NUMBER 3 WAS TRUNCATED
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
```

We can see that program REDBK4 has been recompiled in the TS 3.1 load library and is OK to install.

8.6 Installing CSD resources using CICS CM

This section describes how to use CICS CM to install CSD resource definitions.

8.6.1 Using CM to install definitions into the target CSD

Having confirmed that the migration was successful using the CICS CM compare function and reviewed the rule changes, these migrated resource definitions are now able to be installed into our CICS TS 3.1 region, REDBK31. This can be achieved using the CICS CM Install function.

1. Select option **6. Install** from the Change Package action panel and then choose processing option **1. Install**, shown in Figure 8-48.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 Install                    Change Package REDBKCSD
 Command ===> _________________________________________________________________

 Name  . . . . . . : REDBKCSD
 Description . . : Migrate 2.3 CSD to 3.1 CSD
 Scheme name . . : REDBKCSD

 Choose a processing option and press Enter
 1  1. Install   Install the change package
    2. List      List the install-candidate CICS resources

 Execution mode              Enter "/" to select option
 1  1. Foreground            /  Edit JCL before user submit
    2. Batch









  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 8-48 CICS CM - Installing the resource definitions

2. The pop-up window shown in Figure 8-49 appears, in which install parameters can be entered if required. Then press Enter to install.

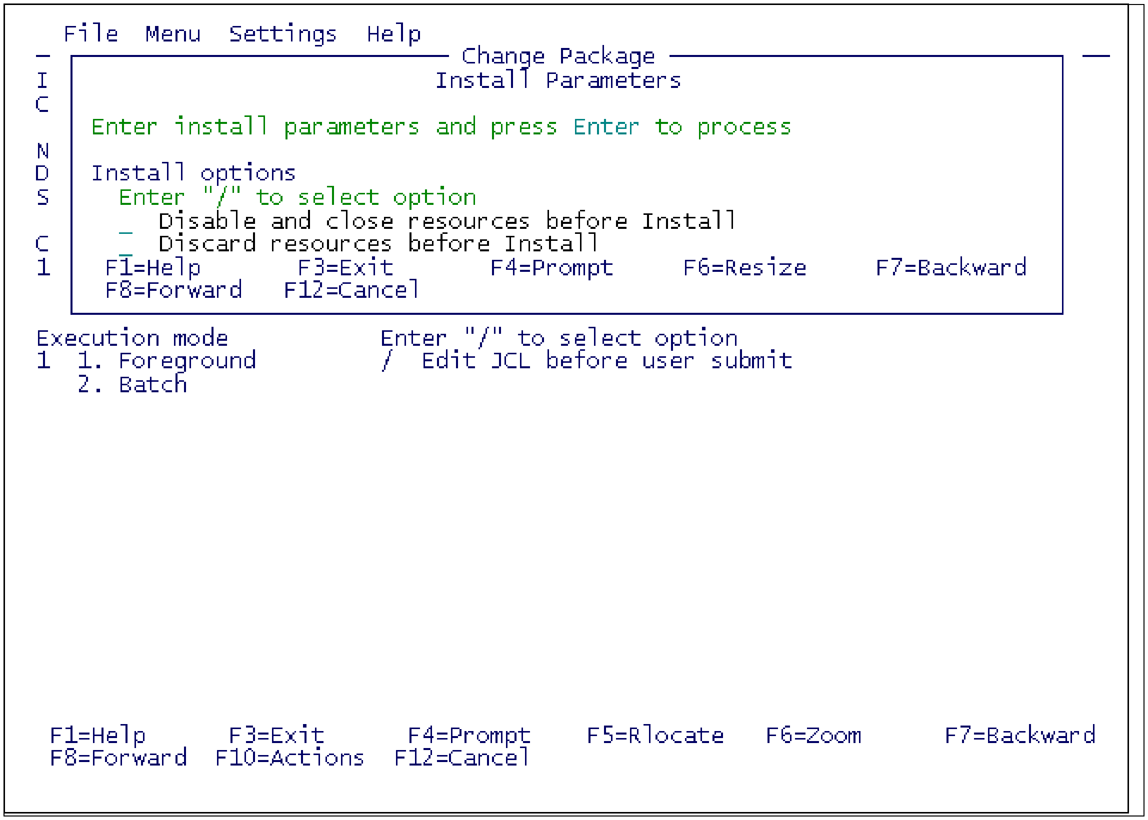


Figure 8-49 CICS CM - Install Parameters

9

Migrating CICS TS 2.3 CSD to CPSM 3.1 BAS

This chapter describes the migration of CICS resources from a CICS TS 2.3 CSD to a CICSPlex SM V3.1 data repository in a CICS TS 3.1 CPSM environment. In particular we are going to look at scenarios where the use of the CICS Tools Interdependency Analyzer and CICS Configuration Manager can be used to facilitate migration.

This chapter does not discuss the use of CICS PA for performance analysis on the CICS TS regions before or after migration. This is discussed in Chapter 2, “Overview of CICS PA” on page 29.

9.1 The environment

In this migration scenario we use the environment shown in Figure 9-1. In this chapter we demonstrate how to migrate CSD resource definitions from a CICS TS 2.3 region to a CICS TS 3.1 region managed by CPSM.

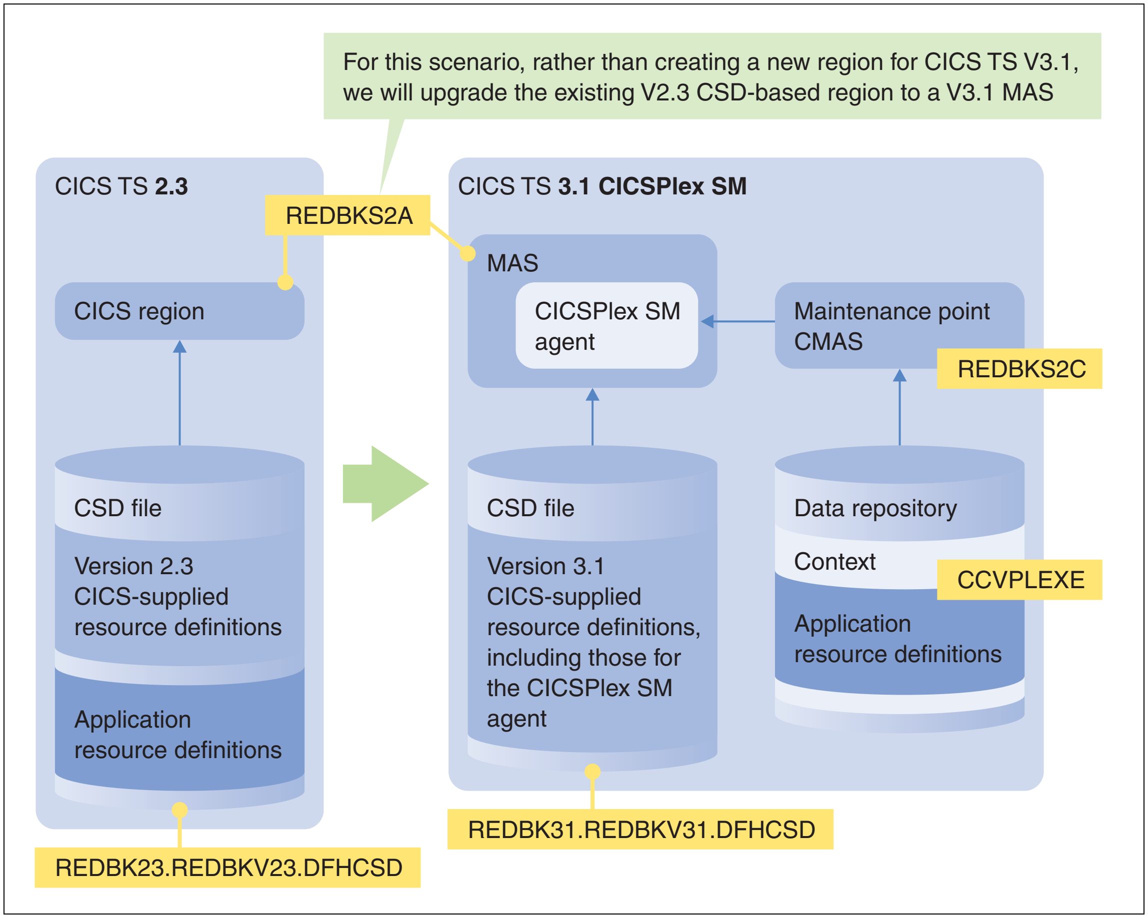


Figure 9-1 The migration scenario

The migration path we demonstrate in this scenario is:

1. Migrate the resource definitions using:
   - CPSM with the IBM-supplied extract routine and BATCHREP utility
   - CICS CM with change packages and transform rules

   - CICS CM COPY function with a supplied user exit to perform resource attribute changes
2. Migrate the CICS Region from V2.3 to V3.1.

   This is described in Chapter 8, “Migrating CICS TS 2.3 CSD to CICS TS 3.1 CSD” on page 197, and is not discussed further in this chapter.
3. Define the CICS Region to CPSM.
   - Changes required to convert the CICS TS 3.1 region REDBKS2A to a MAS
   - Configuring the CICSPlex SM V3.1 CMAS in CCVPLEXE to include REDBKS2A

Assumptions

In order to use CICS CM to migrate and change definitions, the target CMAS environment has already been migrated to a CICS TS 3.1 level. For the purposes of this book, we do not show the actual migration steps of the CMAS from CICSPlex SM V2.3 to CICSPlex SM 3.1. This process is fully described in Chapter 2 of the IBM Redbook *Using the Web User Interface in CICSPlex SM*, SG24-6793.

As in migration scenario 1, we have two applications, Redbook Application 1 and Redbook Application 2, and the associated resource inventory will be used in this CSD to BAS migration. It is also assumed that all of these resources are unique in name to the CPSM repository (that is, there are no resources of the same name already in the data repository).

CICS CM does have the capability of recognizing whether there is a resource already defined in the target repository—a common scenario in a CICSPlex environment. In Chapter 11, “Advanced features of CICS IA and CICS CM” on page 347 we show a case where a resource already exists on the target repository. Refer to 11.1, “Post-migration cleanup using CICS CM” on page 349.

9.2 Migrating the resource definitions using BATCHREP

In this section we discuss migrating the resource definitions using BATCHREP.

9.2.1 The CPSM-supplied extract routine

CPSM supplies an extract routine EYU9BCSD to generate CICSPlex SM resource definition records for each CSD record identified in the input file. This output is then used to populate the data repository in the target environment.

EYU9BCSD is supplied in the CICSTS31.CPSM.SEYUAUTH library. For each record identified in the input file, EYU9BCSD generates an equivalent CICSPlex SM resource definition record. For example, a CSD PROGRAM record is used to build a CPSM PROGDEF resource definition. Each field in the CSD record is used to assign the appropriate attribute value to the resource definition.

In addition to generating individual resource definitions, EYU9BCSD also generates CPSM resource group definitions (RESGROUP). It uses the RESGROUP keyword of the xxxxDEF resource definitions to maintain the relationship to the resource group. That means that when a PROGDEF resource definition is generated from a CSD PROGRAM record, it can automatically be associated with an appropriate resource group.

Output from EYU9BCSD is in the form of batched repository update facility CREATE commands. When you submit those commands, the BATCHREP update facility creates the appropriate resource definition records in the data repository.

If multiple CSD records are found for the same resource type and name, from different GROUPS, multiple CREATE commands are generated in EYUOUT. Once this is submitted to the repository update utility, and the CPSM definitions are created, there will be multiple definitions, each with a different version number.

Note: EYU9BCSD will not build BATCHREP output for CSD resources stored in the CSD groups with names beginning with DFH or EYU. It is not intended that these types of system resources should be defined/migrated using BAS. System resources for a target environment are supplied in the target system libraries and do not need to be migrated.

9.2.2 The EYU9BCSD job

Submitting a job to run the CPSM extract exit EYU9BCSD is done through the DFHSCDUP EXTRACT command by specifying the following in the JCL:

```
EXTRACT LIST(listname) | Group(group name)
   USERPROGRAM(EYU9BCSD) OBJECTS
```

In our migration scenario there are two application groups in our V2.3 CSD called REDGROUP and UTILITY, both within a LIST called REDLIST. Example 9-1 on page 273 shows the sample JCL that we have used to run the extract. This job extracts the resource definitions of all the resource types from the CSD in the REDLIST LIST. The output file of BATCHREP input commands, EYUOUT, is directed to a data set called JEN.EYUOUT.REDBOOK1.

Note: EYU9BCSD must be invoked from the USERPROGRAM keyword. It cannot be called on the entry linkage to DFHCSDUP using the EXIT parameter. In addition, the OBJECTS keyword is required.

Example 9-1 Sample DFHCSDUP with EYU9BCSD exit

```
//BATCHREP JOB (KSM),'JEN',MSGCLASS=X,CLASS=A,
// NOTIFY=&SYSUID
//*-------------------------------------------------------------
//*
//* Delete the extract output file for a rerun of this job
//*
//*-------------------------------------------------------------
//BR14OUT  EXEC PGM=IEFBR14
//EYUOUT    DD  DISP=(MOD,DELETE,DELETE),
//             DSN=JEN.EYUOUT.REDBOOK1,
//             SPACE=(TRK,(1,1)),
//             UNIT=SYSDA
//*-------------------------------------------------------------
//*
//* Extract the CSD Resource Definitions
//*
//*-------------------------------------------------------------
//CSDXTRCT EXEC PGM=DFHCSDUP,
//             COND=(0,NE),
//             PARM='CSD(READONLY)'
//STEPLIB   DD  DISP=SHR,DSN=CICS.V630.CICS.SDFHLOAD
//          DD  DISP=SHR,DSN=CICS.V630.CPSM.SEYUAUTH
//DFHCSD    DD  DISP=SHR,DSN=REDBK23.REDBKV23.DFHCSD
//EYUOUT    DD  DISP=(,CATLG,DELETE),
//             DSN=JEN.EYUOUT.REDBOOK1,
//             SPACE=(TRK,(1,5)),
//             UNIT=SYSDA
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
 EXTRACT USERPROGRAM(EYU9BCSD) OBJECTS LIST(REDLIST)
/*
//EYUIN     DD  *
 CONTEXT(CCVPLEXE)
 RESGROUP(CSDGROUP)
RESINGRP(CSDGROUP)
CONNECTION(*)
CORBASERVER(*)
DB2CONN(*)
DB2ENTRY(*)
DB2TRAN(*)
DJAR(*)
DOCTEMPLATE(*)
ENQMODEL(*)
FILE(*)
JOURNAL(*)
JOURNALMODEL(*)
LSRPOOL(*)
MAPSET(*)
PARTITIONSET(*)
PARTNER(*)
PROCESSTYPE(*)
PROFILE(*)
PROGRAM(*)
REQUESTMODEL(*)
SESSIONS(*)
TCPIPSERVICE(*)
TDQUEUE(*)
```

```
TERMINAL(*)
TRANCLASS(*)
TRANSACTION(*)
TSMODEL(*)
TYPETERM(*)
/*
//*---------------------------------------------------------
//*
//* List EYUOUT to view errors
//*
//*---------------------------------------------------------
//LISTOUT  EXEC PGM=IEBGENER
//SYSUT1    DD  DISP=SHR,DSN=JYN.EYUOUT.REDBOOK1
//SYSUT2    DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  DUMMY
```

Further information about how to extract CSD records is given in *CICSPlex System Manager Managing Business Applications,* SC34-6467.

9.2.3 Submitting EYUOUT to BATCHREP update facility

The output from the job in Example 9-1 on page 273 can now be used as input to the CPSM BATCHREP utility.

Example 9-2 shows some output of EYUOUT in which the CSD definitions having been transformed into BATCHREP input statements. Notice that at the end of each resource definition the RESGROUP has been included.

Example 9-2 Sample EYUOUT output from DFHCSDUP extract job

```
*/
  CONTEXT CCVPLEXE            ;
  CREATE RESGROUP     RESGROUP(REDBOOK)
                   DESCRIPTION( )
                              ;
  CREATE CONNDEF           NAME(T23M)
                   DESCRIPTION(CONNECTION TO CCVLINKS)
                       NETNAME(CCVT23M)
                        INDSYS( )
                  REMOTESYSTEM( )
                    REMOTENAME( )
                     RESGROUP(REDBOOK)
                             ;
 CREATE FILEDEF           NAME(REDBOOKF)
                  DESCRIPTION( )
                       DSNAME(REDBK23.REDBKV23.VSAM)
                     PASSWORD( )
                    RLSACCESS(NO)
                    LSRPOOLID(1)
                            |||
                  JNLSYNCWRITE(YES)
                      RECOVERY(BACKOUTONLY)
                   FWDRECOVLOG(NO)
                    BACKUPTYPE(STATIC)
                      RESGROUP(REDBOOK)
                              ;
  CREATE PROGDEF           NAME(CDCB001#)
                   DESCRIPTION( )
                      LANGUAGE(COBOL)
```

```
                       RELOAD(NO)
                     RESIDENT(NO)
                        USAGE(NORMAL)
                           |||
                     JVMCLASS( )
                   JVMPROFILE(DFHJVMPR)
                      HOTPOOL(NO)
                     RESGROUP(REDBOOK)
                            ;
CREATE TRANDEF           NAME(RDBA)
                  DESCRIPTION( )
                      PROGRAM(REDBK1)
                      TWASIZE(0)
                      PROFILE(DFHCICST)
                 PARTITIONSET( )
                       STATUS(ENABLED)
                  TASKDATALOC(BELOW)
                  TASKDATAKEY(USER)
                           |||
                       CMDSEC(NO)
                     RESGROUP(REDBOOK)
                            ;
```

Considerations for EYUOUT

You may need to make some changes to this file if any of the following are true in your environment:

- CONTEXT

  The batched repository-update facility needs to know the CICSPlex SM context for the resource definitions being processed. You must insert a CONTEXT statement at the beginning of the file to identify the CICSplex to which the updates apply.

- PASSWORDS

  The CSD records extracted by DFHCSDUP do not include passwords. Any resource definitions that include passwords are generated with blanks (X'40') in the password fields, unless you add the passwords manually.

  You can edit individual CREATE commands in the file to add the appropriate password fields. The passwords are then included in the resource definitions that CICSPlex SM generates in the data repository. Be aware, however, that the batched repository-update facility output will include a visible record of the passwords that you entered.

- OBSOLETE FIELDS

  The CSD records extracted by DFHCSDUP do not include fields that are considered obsolete, but that are retained for compatibility.

Further details regarding the use of the batched repository-update facility are in *CICSPlex System Manager Administration,* SC34-6462.

9.2.4 Using the CPSM WUI to perform the BATCHREP

The output of the DFHSCDUP EXTRACT - EYUOUT is now ready to be used by the batched repository update facility to perform the migration of the CSD definitions into the CPSM data repository. The following figures from the WUI demonstrate the steps taken.

1. From the WUI main menu, define the context of your CMAS and click **Set**, as in Figure 9-2.

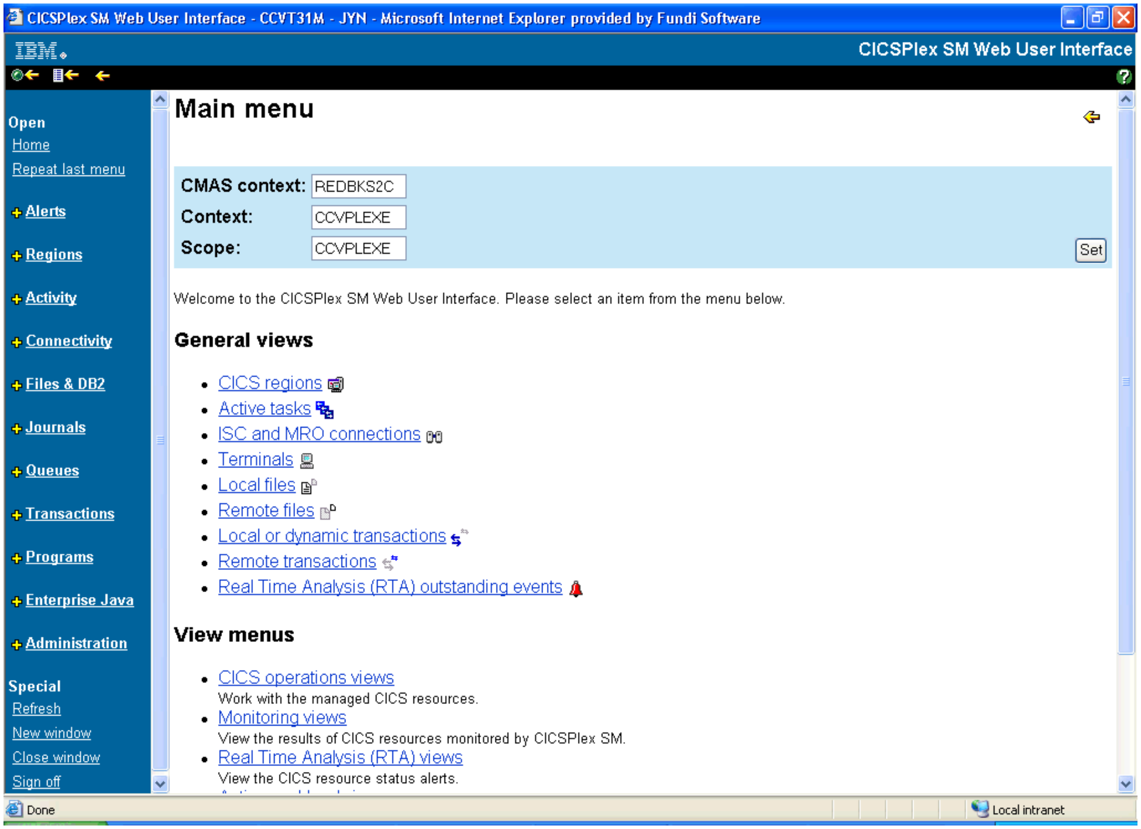


Figure 9-2 Defining the context

2. From the Administration menu, select the **Batched Repository Update Job** and check that the CMAS context is correct. This is shown in Figure 9-3.

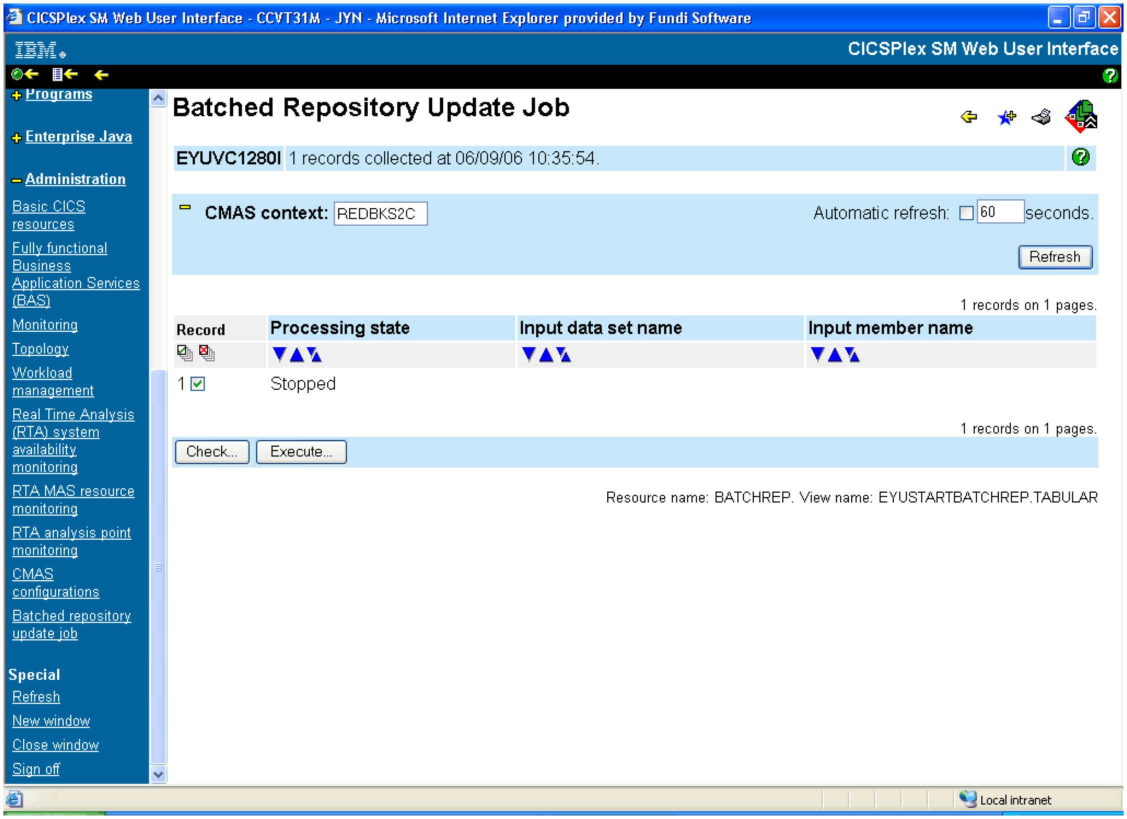


Figure 9-3 Batched Repository Update Job initial screen

3. Tick the box to select Record 1 then click **Execute** and the screen shown in Figure 9-4 appears. Enter the name of the EYUOUT input file that was created in “The EYU9BCSD job” on page 272 and then click **Yes** to confirm.

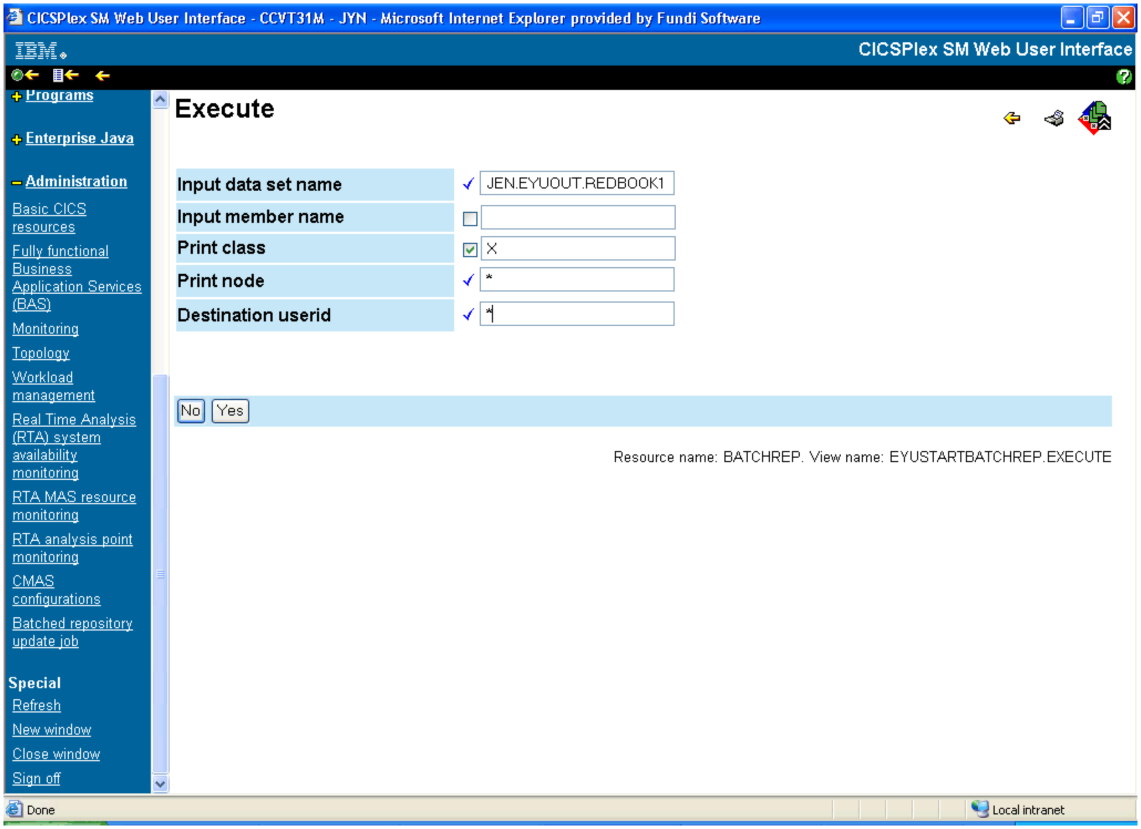


Figure 9-4 Execute the update

4. Now execute the batched repository update job. If you refresh the screen, you will see that the process has been started, as in Figure 9-5.

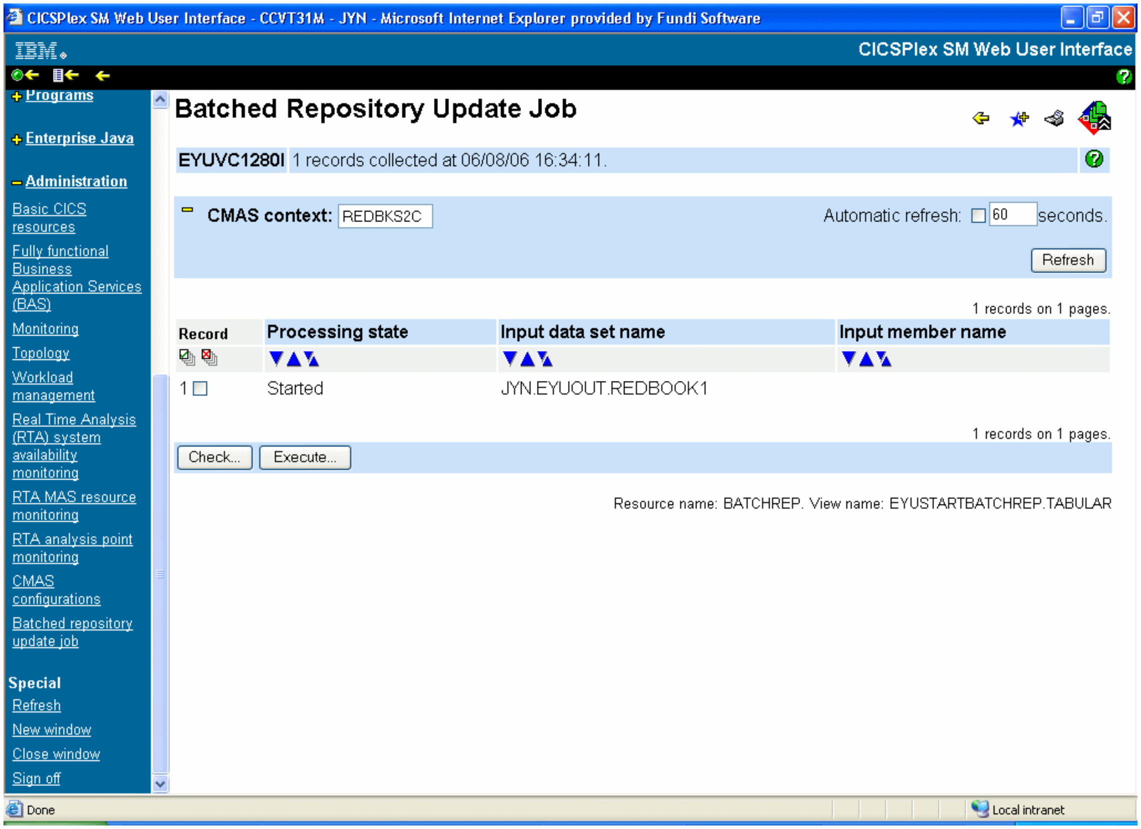


Figure 9-5 Execute the Batched Repository Update Job

5. To check that the job is successful and to verify the upgraded definitions, the output of the BATCHREP will be in the JES log of the CMAS to which the WUI is connected. Alternatively, use the WUI to check the Basic CICS resources from the Administration drop-down menu to check that the group has been added and to verify the contents. This is seen in Figure 9-6.

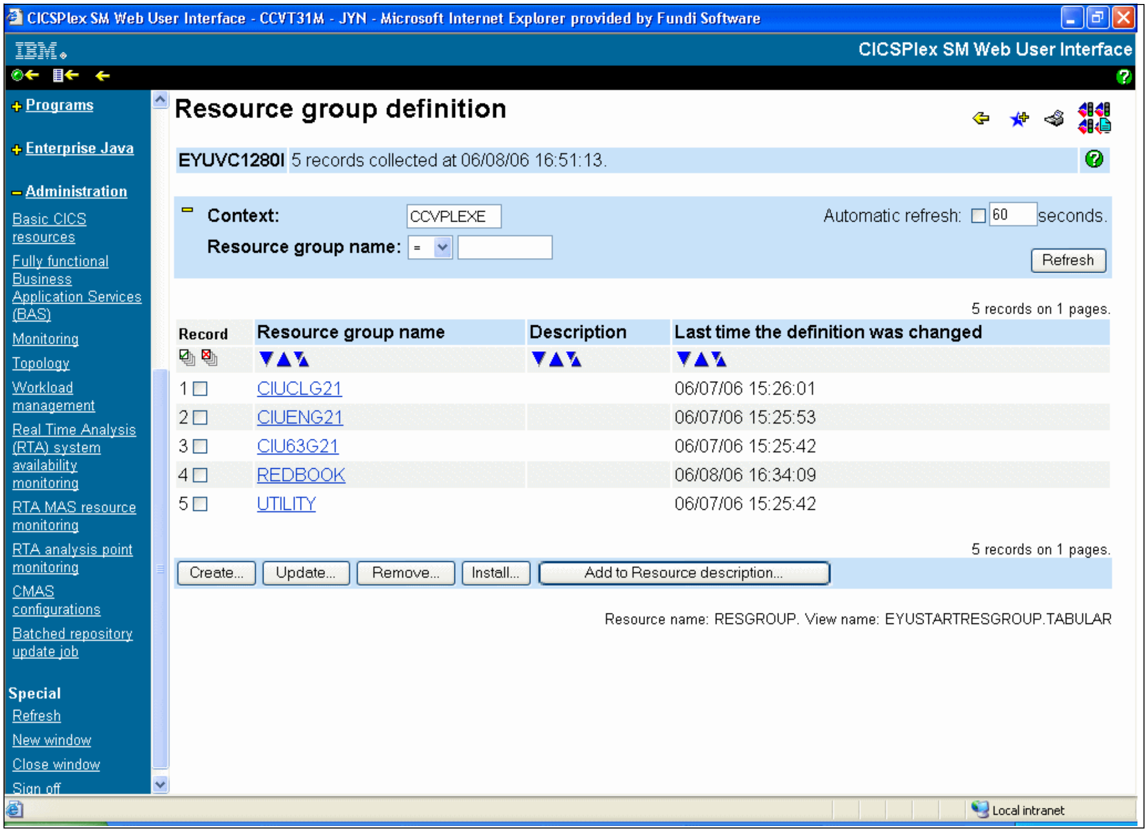


Figure 9-6 Resource group definitions created

9.3 Use input from CICS Interdependency Analyzer

CICS IA identifies resources for each set of applications that needs to be migrated (in runtime and via the load library scanner). This section demonstrates the use of CICS IA showing output reports and examples to aid in migration. This information can be used to identify:

- Non-LE and OS/VS Cobol programs. If any are found, the IBM Debug Tools and Advanced Functions can be used to convert these programs. For further information see “Debug Tool Utilities and Advanced Functions” on page 105.

- Applications that do not conform to threadsafe standards. In CICS TS 3.1 such applications could impact performance.
- A group of resources for each application that needs to be migrated from the current CICS TS environment to the upgraded CICS TS environment. This information will be used by CICS CM to create the CICSPlex SM BAS definitions to be stored in the CICSPlex SM repository.
- Affinities and use this information to build CICSPlex SM definitions. CICS IA creates and enables you to manage affinity groups. Alternatively, this information can be used to eliminate affinities prior to CICSPlex SM enablement.

To install and customize CICS IA V2R1 refer to Appendix A, “CICS IA installation and customization” on page 405.

To use the CICS IA scanners see 8.3.1, “Using the CICS IA Scanners” on page 207.

To run the IA collector see 8.3.2, “Using the CICS IA Collector” on page 214.

9.3.1 Identifying COBOL/VS programs

To identify which programs that were compiled with COBOL/VS we can query either the CIU4_SCAN_SUMMARY table populated by job CIUJCLTS (load module scanner) or the V_CIU4_CSECT_TRANS view populated by job CIUJCLCS (CSECT scanner).

All of the following SQL queries were performed using IBM SPUFI interface.

Querying the CIU4_SCAN _SUMMARY table

Example 9-3 shows all the modules in REDBK23.APPL.LOADLIB that the scanner has identified as being COBOL/VS programs.

Example 9-3 COBOL query using the scan summary table

```
--Show me all possible programs that are COBOL/VS in dataset
--REDBK23.APPL.LOADLIB using the load module scanner summary.
  SELECT PROGRAM, LANGUAGE  FROM CIU4_SCAN_SUMMARY
  WHERE DSNAME='REDBK23.APPL.LOADLIB' AND LANGUAGE='COBOL';
---------+---------+---------+---------+---------+---------+---------+---------+
PROGRAM   LANGUAGE
---------+---------+---------+---------+---------+---------+---------+---------+
REDBK4    COBOL
CSCB0200  COBOL
CSCB0030  COBOL
CSCB0010  COBOL
COBOLVS2  COBOL
COBOLVS1  COBOL
CICB0050  COBOL
CICB0030  COBOL
CICB0020  COBOL
CICB0010  COBOL
CDCB0710  COBOL
CDCB0510  COBOL
CDCB0020  COBOL
CDCB0010  COBOL
CDCB001#  COBOL
DSNE610I NUMBER OF ROWS DISPLAYED IS 15
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
```

9.3.2 Identifying non-threadsafe programs

To identify which programs are non threadsafe we can query either the CIU4_SCAN_DETAIL table populated by job CIUJCLTD (load module scanner) or the CIU4_CICS_DATA table populated by data from the collector.

Querying the CIU4_SCAN _DETAIL table

The following query will tell us all programs that have possible commands that would cause the program to be non threadsafe (that is, the program executes a LOAD, EXTRACT, GETMAIN, or ADDRESS CWA). The query is restricted to the REDBK23.APPL.LOADLIB data set only. See Example 9-4.

Example 9-4 Threadsafe query using the scan detail table

```
--Show me all possible programs that are not threadsafe in dataset
--REDBK23.APPL.LOADLIB using the load module scanner detail
  SELECT PROGRAM , COMMAND , RESOURCE_TYPE
  FROM CIU4_SCAN_DETAIL
  WHERE COMMAND IN ('LOAD    ', 'EXTRACT ', 'GETMAIN ', 'ADDRESS ')
  AND DSNAME='REDBK23.APPL.LOADLIB';
---------+---------+---------+---------+---------+---------+---------+---------+
PROGRAM   COMMAND   RESOURCE_TYPE
---------+---------+---------+---------+---------+---------+---------+---------+
COBOLVS1  ADDRESS   CWA
COBOLVS2  GETMAIN   SHARED
REDBK1    ADDRESS   CWA
REDBK1A   ADDRESS   CWA
REDBK1B   ADDRESS   CWA
REDBK1C   ADDRESS   CWA
REDBK1D   ADDRESS   CWA
REDBK1E   ADDRESS   CWA
REDBK5    ADDRESS   CWA
DSNE610I NUMBER OF ROWS DISPLAYED IS 9
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
---------+---------+---------+---------+---------+---------+---------+---------+
```

9.3.3 Identifying applications to be migrated

The CICS TS 2.3 region to be migrated contains two applications:

- The first application contains all programs with a prefix of RED. The CICS IA application code for this application will be RDB.
- The second application contains all programs with a prefix of COB, CDC, or CSC. This CICS IA application code for this application will be COB.

To define CICS IA applications see “Creating new applications” on page 415.

We can use the CICS IA query CINQ transaction to see which resources are used by the application.

To show all resources used by application RDB select option 1 for CICS resources in the main Query menu. See Figure 8-14 on page 230.

Enter `RDB` in the application code field and select Y for details in the CICS Query menu. See Figure 8-15 on page 231.

Figure 9-7 shows the first page of the output for the IA query: *Which resources are in Redbook Application 1?*

```
 CIU585       CICS Interdependency Analyzer for z/OS - V2R1M0        2006/06/13
  For your CICS Query                                                01:06:12PM
  WHICH RESOURCES ARE IN Redbook Application 1                      Page   1 of   5

 In  Tran Program  Resource Resource Resource
Regn              Function Type     Name
RB23 RDBA REDBK1   WRITEQ   TD       CESE
     RDBA REDBK1   ADDRESS  CWA      CWA
     RDBA REDBK1   ASSIGN   APPLID   REDBKV23
     RDBA REDBK1   WRITE    FILE     REDBOOKF
     RDBA REDBK1   WRITEQ   TSSHR    REDBOOKQ
     RDBB REDBK1   WRITEQ   TD       CESE
     RDBB REDBK1   ADDRESS  CWA      CWA
     RDBB REDBK1   ASSIGN   APPLID   REDBKV23
     RDBB REDBK1   WRITE    FILE     REDBOOKF
     RDBB REDBK1   WRITEQ   TSSHR    REDBOOKQ
     RDBC REDBK1   WRITEQ   TD       CESE
     RDBC REDBK1   ADDRESS  CWA      CWA
     RDBC REDBK1   ASSIGN   APPLID   REDBKV23

 CICS Sysid:  RB23   CICS Applid:  REDBKV23   TermID:  CP89

 F1=          F2=           F3=End       F4=Exit      F5=          F6=
 F7= Page Up  F8= Page Down F9=          F10=         F11=         F12=End
```

Figure 9-7 IA Query - Which resources are used by application RDB

> **Note:** Redbook Application 1 uses a file resource of REDBOOKF. The dsname for this resource is REDBK23.REDBKV23.VSAM and is dependant upon the CICS region it runs in. This DSNAME will need to be changed when migrating to the CICS TS 3.1 region.

9.4 Migration using CICS CM

This section shows how CICS CM is used to aid in the migration of resources to the CICS TS 3.1 CPSM environment. There are two approaches that we use to demonstrate the effectiveness of performing this migration using CM functions:

- Using CM transformation rules and building a change package

  The transform rules are defined to enable the changing of resource attributes during migration to the target repository. We show the steps taken to build transform rules and a change package, and then how CM can be used to perform the migration of definitions to a target BAS data repository.

While this can be a useful technique for some resource attributes, consideration should be made as to the implication of changing an attribute as part of the migration process. There are some resource attributes such as CONCURRENCY(THREADSAFE) that should be specified in a transform rule for migration with caution due to the potential impact on integrity and performance.

- Using the CM COPY command and a user exit to change resource attributes

  There is a supplied user exit that can be modified to define changes to resource attributes that are applied during execution of the CM COPY command.

We recommend that you review Chapter 3, “Overview of CICS CM” on page 37, to understand the functions that CM can provide. Conceptually, the role that CICS CM plays in the environment and to our scenario is shown in Figure 9-8.

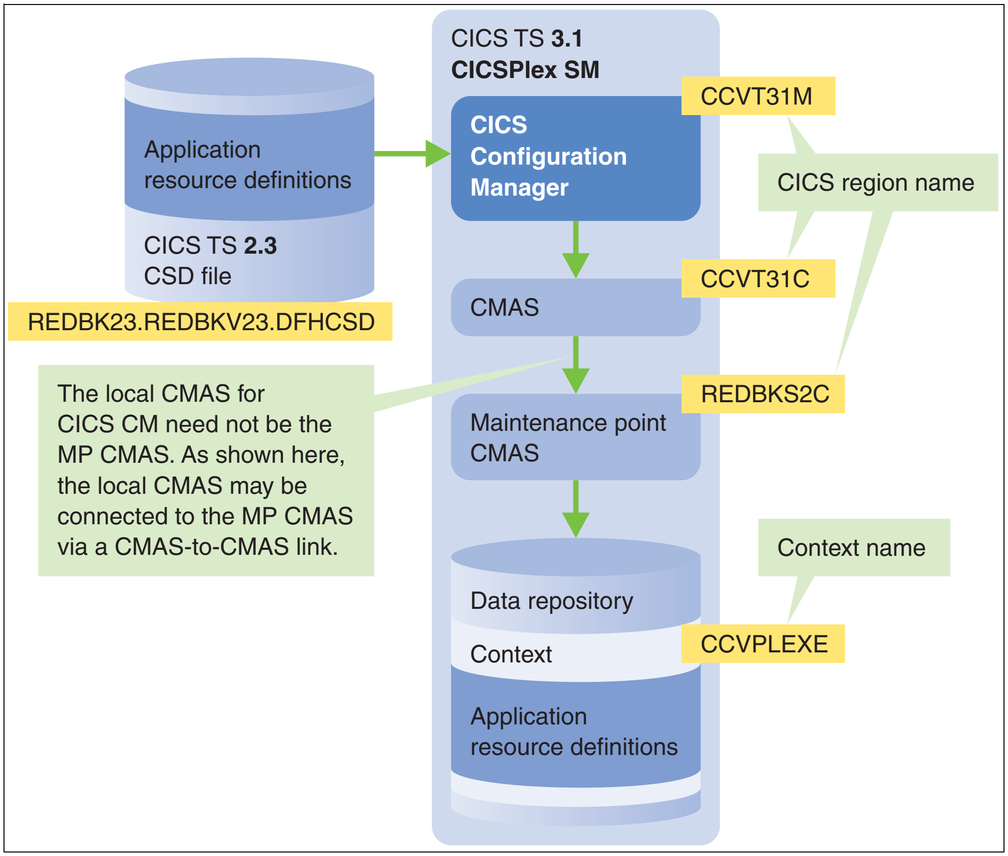


Figure 9-8 Using CICS CM in our scenario

9.4.1 Building change packages using CICS CM

In this section we discuss building change packages using CICS CM.

Create a CICS CM configuration record

From the CICS CM Primary Menu, select option **1. Administration** and then option **2. CICS Configurations** to create a configuration record and associate it with the source CSD. This enables CICS CM to locate the source CSD.

Once the migration process has been successful, this configuration record can be deleted. Figure 9-9 shows the CICS Configuration panel for CM.

```
  File  Menu  Settings  Help
------------------------------------------------------------------------------
Edit                         CICS Configuration                 Row 1 to 1 of 1
Command ===> ________________________________________________  Scroll ===> PAGE

Name . . . . : REDBK23
Description . . REDBOOK v2.3 CSD

Specify the type of file or repository
2   1. CICSPlex SM Context . .
    2. CSD File . . . . . . . REDBK23.REDBKV23.DFHCSD
    3. Export-import file  . .

Choose a view for related options
1   1. Transform variables
    2. Remote System Connections
    3. Export-Import file options

Define transformation variables and values

/ Name                 Value
******************************** Bottom of data ******************************




 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F12=Cancel
```

Figure 9-9 Defining a configuration record

Create a transform rule

The next step is to create a transform rule set for which rules are defined. In our scenario, a transform rule set called REDBOOK is created that contains rules s for resource types *program*, *tcpipservice,* and *file*.

Option 5. Transform Rules from the Administration menu allows you to view the transform rules for the rule set. The rules for our example are shown in Figure 9-10.

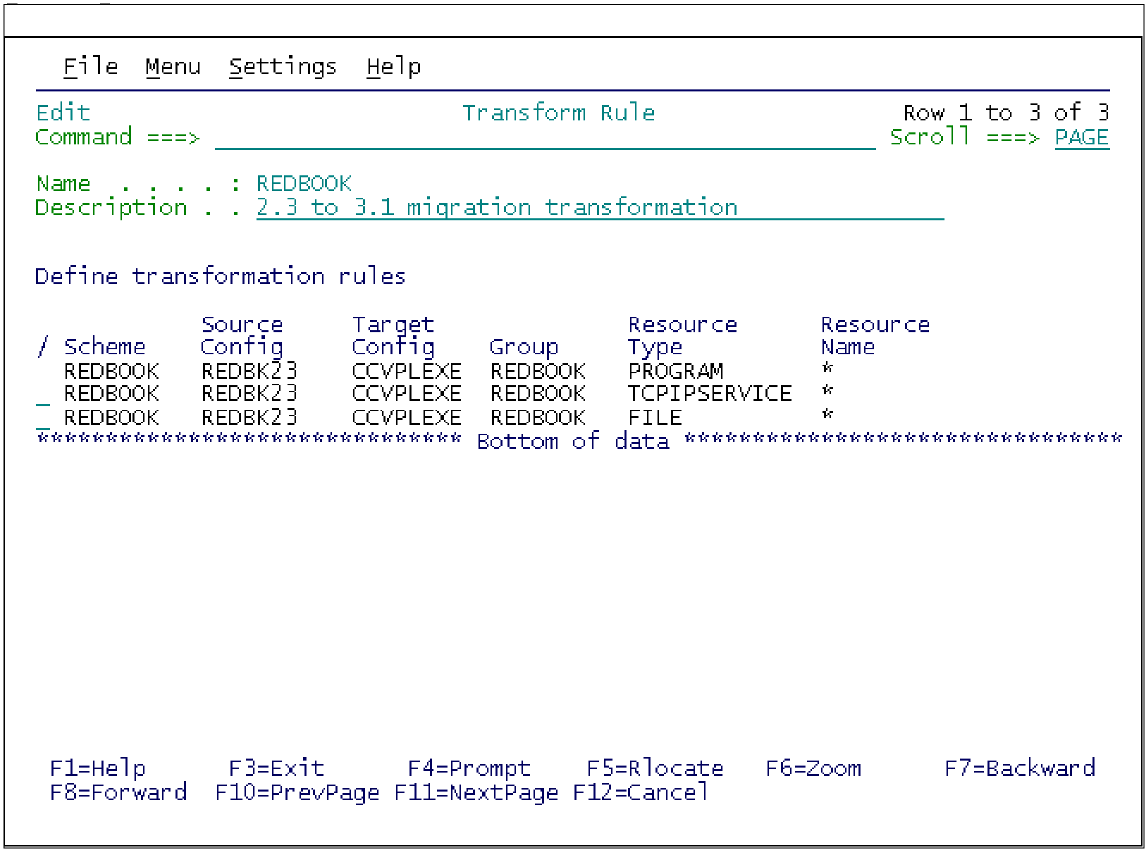


Figure 9-10 Transform rules for three resource types

Define rules for transformation of individual resources

By selecting a scheme for the resource type, the attributes can now be set with directives to indicate how the attribute is to be changed during the transformation.

For example, the rule for the *program* resource type shown in Figure 9-11 will change any PROGRAM resource type in group REDBOOK that has a LANGUAGE attribute equals COBOL to a LANGUAGE equal LE370 in the transformed data repository.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 Edit                           Transform Rule
 Command ===> _________________________________________________________________

 Name  . . . . : REDBOOK
 Description . . Change COBOL language to LE370

 Qualification criteria
   Scheme  . . . . . . REDBOOK   +
   Source Config . . . REDBK23   +
   Target Config . . . CCVPLEXE  +
   Group . . . . . . . REDBOOK
   Resource Type . . . PROGRAM       +
   Resource Name . . . *
   Check Field . . . . LANGUAGE      +
   Check Operator      EQ  +
   Check Value . . . . COBOL                                              +

 Choose a Processing Option
 1  1. Transform and continue
    2. Transform and lock field
    3. Transform and lock record
    4. Stop migration of this resource

 Transform Field and Values
   Change Field  . . . LANGUAGE      +
   Change From . . . . COBOL                                              +
   Change To . . . . . LE370                                              +

  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F12=Cancel
```

Figure 9-11 The transform rule for the PROGRAM resource type

Similarly, Figure 9-12 shows the rules for the *tcpipservice* resource type. In this case, the value for attribute MAXDATALEN will be changed to 1000 KB if the values found in any of the TCPIPSERVICE definitions in group REDBOOK are found to be less than 1000 KB.

> **Note:** This example shows the potential scope that CICS CM has in terms of being able to introduce and fix a value to new attributes in a migrated CICS environment. MAXDATALEN is a new attribute in the *tcpipservice* resource definition in CICS TS 3.1

```
  File  Menu  Settings  Help
--------------------------------------------------------------------------------
Edit                           Transform Rule
Command ===> ____________________________________________________________________

Name  . . . . : REDBOOK
Description . . Make Maxdatalen a minimum of 1000k________________

Qualification criteria
  Scheme  . . . . . . REDBOOK_   +
  Source Config . . REDBK23_   +
  Target Config . . CCVPLEXE   +
  Group . . . . . . REDBOOK_
  Resource Type . . TCPIPSERVICE   +
  Resource Name . . *_______
  Check Field . . . MAXDATALEN__   +
  Check Operator    LT   +
  Check Value . . . 1000__________________________________________________  +

Choose a Processing Option
1  1. Transform and continue
   2. Transform and lock field
   3. Transform and lock record
   4. Stop migration of this resource

Transform Field and Values
  Change Field  . . MAXDATALEN__   +
  Change From . . . *_____________________________________________________  +
  Change To . . . . 1000__________________________________________________  +

 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F12=Cancel
```

Figure 9-12 The transform rule for the TCPIPSERVICE resource type

Tip: In a more complex scenario, the use of CM’s transform rules can be of great benefit. Consider the migration scenario where there is a requirement to change the SYSID of the target regions, and there are many PROGRAM resource definitions that contain a REMOTESYSTEM attribute and need to be changed.

Traditionally, these definitions would be changed individually or via multiple DFHCSDUP ALTER commands to reflect the new SYSIDs as a post migration task. The CM transform rule can address this task in a more efficient manner and can either be incorporated as part of the migration or as a post-migration activity.

Create a migration scheme

The Migration Scheme panel from option 3 on the Administration menu is where the source and target configurations are specified, as well as the name of the transform rule to use.

This is shown in Figure 9-13 and will tell CICS CM to take our CSD definitions that are in the REDBK23 source, apply the rules from the schemes defined in the transform rule set REDBOOK, and place the new/changed definitions in the CICSPlex SM repository for CCVPLEXE.

```
  File  Menu  Settings  Help
--------------------------------------------------------------------------------
Edit                          Migration Scheme                   Row 1 to 1 of 1
Command ===> ________________________________________________ Scroll ===> PAGE

Name  . . . . : REDBOOK
Description . . Migrate 2.3 CSD TO 3.1 Plex

Approval Processing
_   Activate

Choose a view for related options
1   1. Migration paths
    2. Transform variables

Define the migration path source and target CICS Configurations

/ Source +  Target +  Transform Rule +
  REDBK23   CCVPLEXE  REDBOOK
******************************* Bottom of data ********************************



 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F12=Cancel
```

Figure 9-13 The migration scheme

Create a change package

At this point we now add all the candidates or resource definitions into a change package that will be used in the actual migration. Option *3. Packages* from the CM primary menu manages the change packages in the CICS CM environment. Figure 9-14 shows the action menu and the processing commands that can now be performed on our package REDBOOK.

```
  File  Menu  Settings  Help
------------------------------------------------------------------------------
Edit                       Change Package REDBOOK
Command ===> ________________________________________________________________

Name  . . . . : REDBOOK
Description . . Migrate 2.3 CSD to 3.1 Plex__________________________

Change package settings
  Approval profile . . . ________  +
  External reference . . ________

Choose a processing command and press Enter
1  1. Package   Package CICS resources into the change package
   2. List      List CICS resources assigned to the change package
   3. Ready     Ready or unready the package for processing
   4. Approve   Approve or disapprove the change package
   5. Migrate   Migrate the change package
   6. Install   Install the package's resources into CICS regions
   7. Newcopy   Newcopy the package's maps, partitionsets or programs
   8. Backout   Backout a previous migrate of the change package
   9. History   Display the change package processing history

      Migration scheme . . . REDBOOK_  +  (Required for options 1-8)




 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F12=Cancel
```

Figure 9-14 The Change Package action menu for REDBOOK

The *1. Package* processing command from Figure 9-14 on page 293 is selected from where the resources or candidates are nominated for inclusion in this migration change package that we are using. Figure 9-15 shows this selection.

```
  File  Menu  Settings  Help
------------------------------------------------------------------------------
Package                  Change Package REDBOOK              Row 1 to 22 of 41
Command ===> ________________________________________________ Scroll ===> PAGE

Filter *_______ *___________ + REDBOOK_         REDBK23_ *________________

/      Name     Type            Group    Prompt    Config   --- Changed ----
P_____ CCVR     TRANSACTION     REDBOOK            REDBK23  2006/06/06 12:07
P_____ CCVSREMP PROGRAM         REDBOOK            REDBK23  2006/06/06 12:07
P_____ CDCB001# PROGRAM         REDBOOK            REDBK23  2006/06/06 11:53
P_____ CDCB0010 PROGRAM         REDBOOK            REDBK23  2006/06/06 11:53
P_____ CDCB0020 PROGRAM         REDBOOK            REDBK23  2006/06/06 11:53
P_____ CDCB0510 PROGRAM         REDBOOK            REDBK23  2006/06/06 11:53
P_____ CDCB0710 PROGRAM         REDBOOK            REDBK23  2006/06/06 11:53
P_____ CICB0010 PROGRAM         REDBOOK            REDBK23  2006/06/06 11:53
P_____ CICB0020 PROGRAM         REDBOOK            REDBK23  2006/06/06 11:53
P_____ CICB0030 PROGRAM         REDBOOK            REDBK23  2006/06/06 11:53
P_____ CICB0050 PROGRAM         REDBOOK            REDBK23  2006/06/06 11:53
P_____ COBOLVS1 PROGRAM         REDBOOK            REDBK23  2006/06/06 11:53
P_____ COBOLVS2 PROGRAM         REDBOOK            REDBK23  2006/06/06 11:53
P_____ CSCB0010 PROGRAM         REDBOOK            REDBK23  2006/06/06 11:53
P_____ CSCB0030 PROGRAM         REDBOOK            REDBK23  2006/06/06 11:53
P_____ CSCB0200 PROGRAM         REDBOOK            REDBK23  2006/06/06 11:53
P_____ DB7P     DB2CONN         REDBOOK            REDBK23  2006/06/06 16:15
P_____ RDBA     TRANSACTION     REDBOOK            REDBK23  2006/06/07 13:37
P_____ RDBB     TRANSACTION     REDBOOK            REDBK23  2006/06/07 13:37
P_____ RDBC     TRANSACTION     REDBOOK            REDBK23  2006/06/07 13:37
P_____ RDBD     TRANSACTION     REDBOOK            REDBK23  2006/06/07 13:37
P_____ RDBE     TRANSACTION     REDBOOK            REDBK23  2006/06/07 13:37
 F1=Help      F3=Exit       F4=Prompt     F5=Rlocate    F6=Zoom       F7=Backward
 F8=Forward  F12=Cancel
```

Figure 9-15 Selecting the candidates for the change package

Once all the candidates have been selected, the output in Figure 9-16 shows the included items in the package.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 Package                    Change Package REDBOOK                Row 1 to 22 of 41
 Command ===> ________________________________________________ Scroll ===> PAGE

 Filter * ____  * _________  + REDBOOK _      REDBK23  * ________________

 /      Name     Type           Group    Prompt       Config     --- Changed ----
        CCVR     TRANSACTION    REDBOOK  *Packaged    REDBK23    2006/06/06 12:07
 _____  CCVSREMP PROGRAM        REDBOOK  *Packaged    REDBK23    2006/06/06 12:07
 _____  CDCB001# PROGRAM        REDBOOK  *Packaged    REDBK23    2006/06/06 11:53
 _____  CDCB0010 PROGRAM        REDBOOK  *Packaged    REDBK23    2006/06/06 11:53
 _____  CDCB0020 PROGRAM        REDBOOK  *Packaged    REDBK23    2006/06/06 11:53
 _____  CDCB0510 PROGRAM        REDBOOK  *Packaged    REDBK23    2006/06/06 11:53
 _____  CDCB0710 PROGRAM        REDBOOK  *Packaged    REDBK23    2006/06/06 11:53
 _____  CICB0010 PROGRAM        REDBOOK  *Packaged    REDBK23    2006/06/06 11:53
 _____  CICB0020 PROGRAM        REDBOOK  *Packaged    REDBK23    2006/06/06 11:53
 _____  CICB0030 PROGRAM        REDBOOK  *Packaged    REDBK23    2006/06/06 11:53
 _____  CICB0050 PROGRAM        REDBOOK  *Packaged    REDBK23    2006/06/06 11:53
 _____  COBOLVS1 PROGRAM        REDBOOK  *Packaged    REDBK23    2006/06/06 11:53
 _____  COBOLVS2 PROGRAM        REDBOOK  *Packaged    REDBK23    2006/06/06 11:53
 _____  CSCB0010 PROGRAM        REDBOOK  *Packaged    REDBK23    2006/06/06 11:53
 _____  CSCB0030 PROGRAM        REDBOOK  *Packaged    REDBK23    2006/06/06 11:53
 _____  CSCB0200 PROGRAM        REDBOOK  *Packaged    REDBK23    2006/06/06 11:53
 _____  DB7P     DB2CONN        REDBOOK  *Packaged    REDBK23    2006/06/06 16:15
 _____  RDBA     TRANSACTION    REDBOOK  *Packaged    REDBK23    2006/06/07 13:37
 _____  RDBB     TRANSACTION    REDBOOK  *Packaged    REDBK23    2006/06/07 13:37
 _____  RDBC     TRANSACTION    REDBOOK  *Packaged    REDBK23    2006/06/07 13:37
 _____  RDBD     TRANSACTION    REDBOOK  *Packaged    REDBK23    2006/06/07 13:37
 _____  RDBE     TRANSACTION    REDBOOK  *Packaged    REDBK23    2006/06/07 13:37
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom       F7=Backward
  F8=Forward  F12=Cancel
```

Figure 9-16 Candidates have been selected in the change package

Note: The process described here has shown CICS CM packaging up the resources for 1 group, REDBOOK. CICS CM can also perform this task in other ways, and the following steps show an alternative method for situations where more than one group is to be migrated at the same time (as would be the case in an upgrade).

Packaging objects from more than 1 group

The following figures highlight an alternative way of adding objects to a change package, which is useful when there is more than one group to migrate simultaneously.

Using Option **4. Reports** from the Primary menu and then option **1. Multiple Configs**, multiple groups can be chosen and their resources selected to package in the one operation.

The panel shown in Figure 9-17 is first presented, in which the line command **g** is used to list the groups found in the source CSD REDBK23.

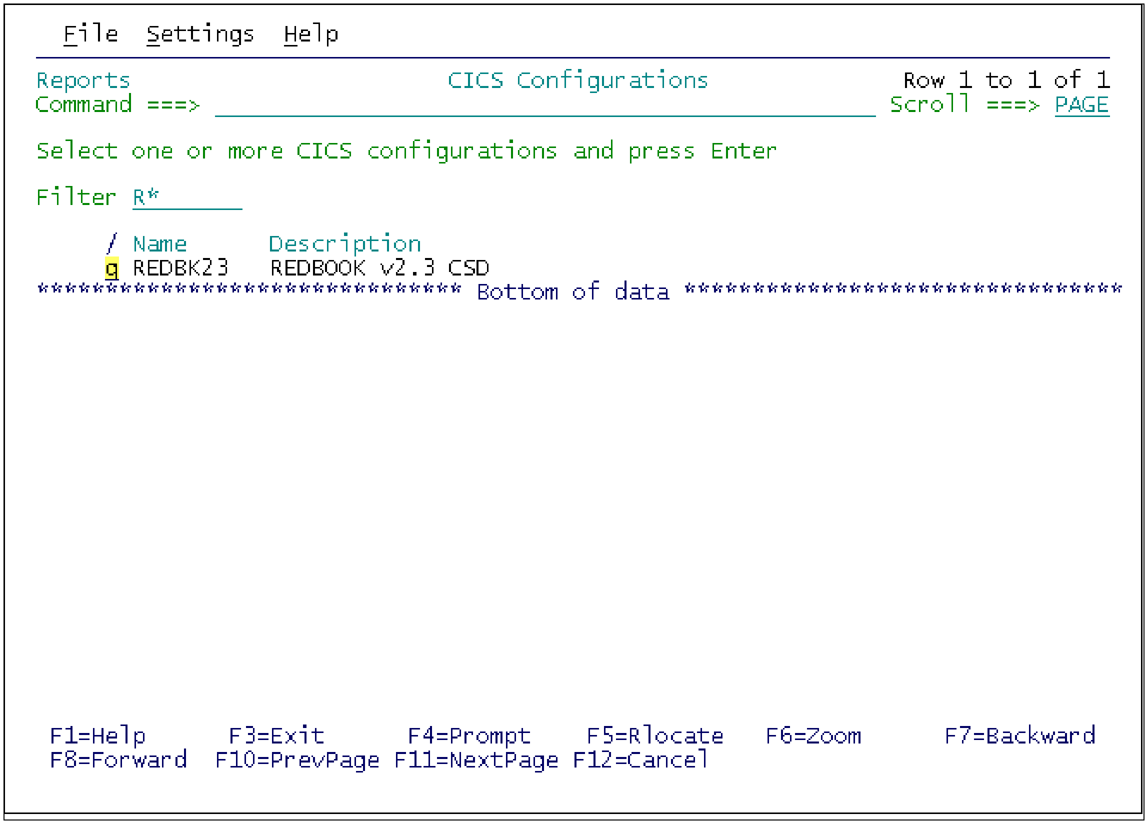


Figure 9-17 Selecting the source for copy

The interim panel shown in Figure 9-18 allows you to filter out certain groups from the source for the subsequent display and packaging. In our example, there are no groups to filter, so press Enter to continue.

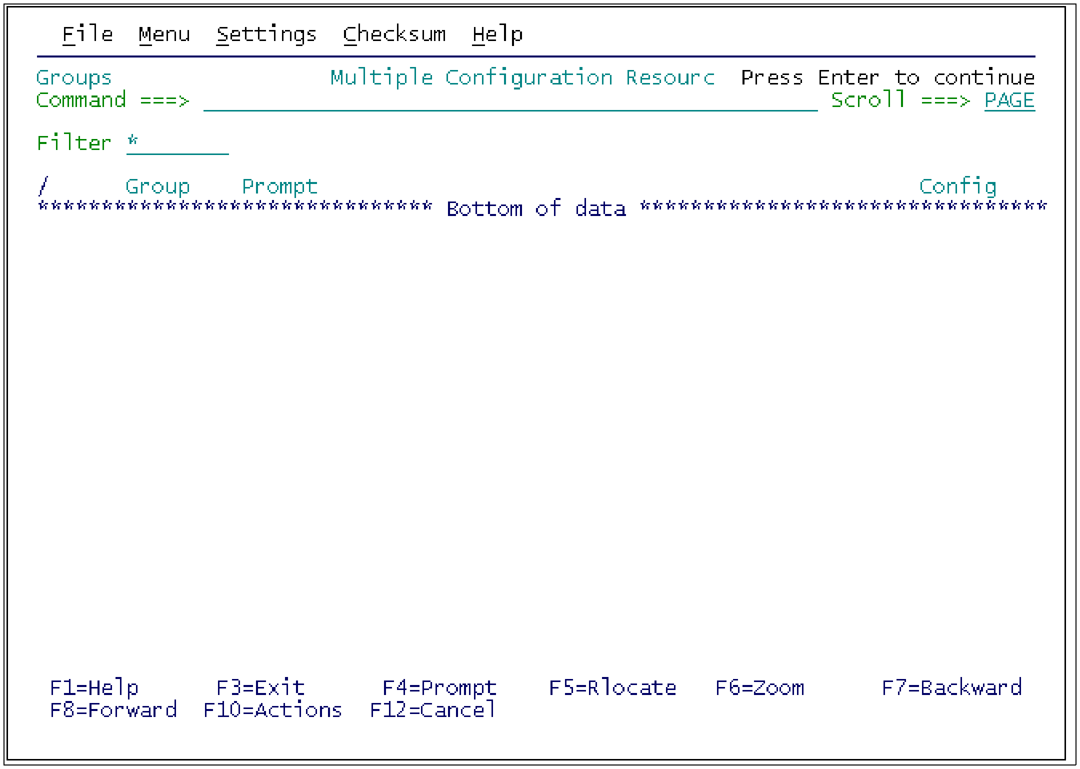


Figure 9-18 Filtering the groups for package selection

Now we are ready to select the groups to migrate using the **s** line command. In our example, we are not migrating any of the CICS system groups, only the application groups REDBOOK and UTILITY. This is shown in Figure 9-19.

```
   File  Menu  Settings  Checksum  Help
 ------------------------------------------------------------------------------
 Groups                 Multiple Configuration Resources  Row 116 to 136 of 136
 Command ===> ________________________________________________ Scroll ===> CSR

 Filter *_______

 /      Group    Prompt                                                Config
        DFHPSSGN                                                       REDBK23
 ______ DFHRMI                                                         REDBK23
 ______ DFHRP                                                          REDBK23
 ______ DFHRPC                                                         REDBK23
 ______ DFHRPCF                                                        REDBK23
 ______ DFHRQS                                                         REDBK23
 ______ DFHRSEND                                                       REDBK23
 ______ DFHSDAP                                                        REDBK23
 ______ DFHSIGN                                                        REDBK23
 ______ DFHSO                                                          REDBK23
 ______ DFHSPI                                                         REDBK23
 ______ DFHSTAND                                                       REDBK23
 ______ DFHTCL                                                         REDBK23
 ______ DFHTERM                                                        REDBK23
 ______ DFHTERMC                                                       REDBK23
 ______ DFHTYPE                                                        REDBK23
 ______ DFHVTAM                                                        REDBK23
 ______ DFHVTAMP                                                       REDBK23
 ______ DFHWEB                                                         REDBK23
 s_____ REDBOOK                                                        REDBK23
 s_____ UTILITY                                                        REDBK23
 ******************************* Bottom of data ********************************
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 9-19 Selection of application groups to be packaged

The resulting panel (Figure 9-20) shows a list of all the resources from all groups selected, which are now selected for packaging.

> **Tip:** Use the command **p *** as a shortcut to mark all the resources in the display and subsequent pages. In our example, there are 43 resources in the groups that we want to package.

```
  File  Menu  Settings  Checksum  Search  Help
 ------------------------------------------------------------------------------
 Group Resources       Multiple Configuration Resources       Row 1 to 22 of 43
 Command ===> ________________________________________________ Scroll ===> CSR

 Filter *_______  *____________  +

        Name     Type           Group    Prompt                       Config
 P_____ CCVR     TRANSACTION    REDBOOK                               REDBK23
 P_____ CCVSREMP PROGRAM        REDBOOK                               REDBK23
 P_____ CDCB001# PROGRAM        REDBOOK                               REDBK23
 P_____ CDCB0010 PROGRAM        REDBOOK                               REDBK23
 P_____ CDCB0020 PROGRAM        REDBOOK                               REDBK23
 P_____ CDCB0510 PROGRAM        REDBOOK                               REDBK23
 P_____ CDCB0710 PROGRAM        REDBOOK                               REDBK23
 P_____ CICB0010 PROGRAM        REDBOOK                               REDBK23
 P_____ CICB0020 PROGRAM        REDBOOK                               REDBK23
 P_____ CICB0030 PROGRAM        REDBOOK                               REDBK23
 P_____ CICB0050 PROGRAM        REDBOOK                               REDBK23
 P_____ COBOLVS1 PROGRAM        REDBOOK                               REDBK23
 P_____ COBOLVS2 PROGRAM        REDBOOK                               REDBK23
 P_____ CONS     TERMINAL       UTILITY                               REDBK23
 P_____ CSCB0010 PROGRAM        REDBOOK                               REDBK23
 P_____ CSCB0030 PROGRAM        REDBOOK                               REDBK23
 P_____ CSCB0200 PROGRAM        REDBOOK                               REDBK23
 P_____ DB7P     DB2CONN        REDBOOK                               REDBK23
 P_____ LSRPOOL1 LSRPOOL        UTILITY                               REDBK23
 P_____ RDBA     TRANSACTION    REDBOOK                               REDBK23
 P_____ RDBB     TRANSACTION    REDBOOK                               REDBK23
 P_____ RDBC     TRANSACTION    REDBOOK                               REDBK23
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 9-20   Resources from multiple groups selected for packaging

The interim screen appears to specify the name of the package now that all the resources are selected in the migration. This is shown in Figure 9-21.

```
   File  Menu  Settings  Checksum  Search  Help
 -  ----------------- CICS Definitions -------------------     ----------------------
 G                      Packaging                                     Row 1 to 22 of 43
 C                                                                      Scroll ===> CSR
    Package ID  . . redbook    +
 F
    Press Enter to confirm
    Press Cancel or Exit to cancel                                           Config
 P   F1=Help          F3=Exit          F4=Prompt                              REDBK23
 P   F6=Resize        F7=Backward      F8=Forward                             REDBK23
 P                                                                            REDBK23
 P      CDCB0010 PROGRAM       REDBOOK                                       REDBK23
 P      CDCB0020 PROGRAM       REDBOOK                                       REDBK23
 P      CDCB0510 PROGRAM       REDBOOK                                       REDBK23
 P      CDCB0710 PROGRAM       REDBOOK                                       REDBK23
 P      CICB0010 PROGRAM       REDBOOK                                       REDBK23
 P      CICB0020 PROGRAM       REDBOOK                                       REDBK23
 P      CICB0030 PROGRAM       REDBOOK                                       REDBK23
 P      CICB0050 PROGRAM       REDBOOK                                       REDBK23
 P      COBOLVS1 PROGRAM       REDBOOK                                       REDBK23
 P      COBOLVS2 PROGRAM       REDBOOK                                       REDBK23
 P      CONS     TERMINAL      UTILITY                                       REDBK23
 P      CSCB0010 PROGRAM       REDBOOK                                       REDBK23
 P      CSCB0030 PROGRAM       REDBOOK                                       REDBK23
 P      CSCB0200 PROGRAM       REDBOOK                                       REDBK23
 P      DB7P     DB2CONN       REDBOOK                                       REDBK23
 P      LSRPOOL1 LSRPOOL       UTILITY                                       REDBK23
 P      RDBA     TRANSACTION   REDBOOK                                       REDBK23
 P      RDBB     TRANSACTION   REDBOOK                                       REDBK23
 P      RDBC     TRANSACTION   REDBOOK                                       REDBK23
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate    F6=Zoom       F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 9-21 Specifying the package name

The panel shown in Figure 9-22 shows that the packaging was successful, and we are now able to ready the package and perform the migration. This process is continued in 9.4.2, “Ready the package” on page 303.

```
   File  Menu  Settings  Checksum  Search  Help
 ------------------------------------------------------------------------------
 Group Resources         Multiple Configuration Resources        Row 1 to 22 of 43
 Command ===> ________________________________________________  Scroll ===> CSR

 Filter *_______ *___________ +

        Name      Type           Group     Prompt                       Config
        CCVR      TRANSACTION    REDBOOK   *Packaged                    REDBK23
 ______ CCVSREMP  PROGRAM        REDBOOK   *Packaged                    REDBK23
 ______ CDCB001#  PROGRAM        REDBOOK   *Packaged                    REDBK23
 ______ CDCB0010  PROGRAM        REDBOOK   *Packaged                    REDBK23
 ______ CDCB0020  PROGRAM        REDBOOK   *Packaged                    REDBK23
 ______ CDCB0510  PROGRAM        REDBOOK   *Packaged                    REDBK23
 ______ CDCB0710  PROGRAM        REDBOOK   *Packaged                    REDBK23
 ______ CICB0010  PROGRAM        REDBOOK   *Packaged                    REDBK23
 ______ CICB0020  PROGRAM        REDBOOK   *Packaged                    REDBK23
 ______ CICB0030  PROGRAM        REDBOOK   *Packaged                    REDBK23
 ______ CICB0050  PROGRAM        REDBOOK   *Packaged                    REDBK23
 ______ COBOLVS1  PROGRAM        REDBOOK   *Packaged                    REDBK23
 ______ COBOLVS2  PROGRAM        REDBOOK   *Packaged                    REDBK23
 ______ CONS      TERMINAL       UTILITY   *Packaged                    REDBK23
 ______ CSCB0010  PROGRAM        REDBOOK   *Packaged                    REDBK23
 ______ CSCB0030  PROGRAM        REDBOOK   *Packaged                    REDBK23
 ______ CSCB0200  PROGRAM        REDBOOK   *Packaged                    REDBK23
 ______ DB7P      DB2CONN        REDBOOK   *Packaged                    REDBK23
 ______ LSRPOOL1  LSRPOOL        UTILITY   *Packaged                    REDBK23
 ______ RDBA      TRANSACTION    REDBOOK   *Packaged                    REDBK23
 ______ RDBB      TRANSACTION    REDBOOK   *Packaged                    REDBK23
 ______ RDBC      TRANSACTION    REDBOOK   *Packaged                    REDBK23
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 9-22 Packaging of selected resources was successful

Removal of object resources from a package

There may be object resources within the groups selected that you do not want to migrate. For example, the use of Interdependent Analyzer may have highlighted resources that can be excluded from the migration. To remove objects from a package, the line command **r** is used against the object displayed in the list obtained from choosing processing command **2. List** in the Change Package action menu. The use of this command is shown in Figure 9-23, in which four programs have been selected for removal.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 List                      Change Package REDBOOK                Row 1 to 22 of 43
 Command ===> ________________________________________________  Scroll ===> CSR

 Filter *_______ *___________ + *_______

 /      Name     Type            Group    Status                        Config
        CCVR     TRANSACTION     REDBOOK  Not ready, new object         REDBK23
 ______ CCVSREMP PROGRAM         REDBOOK  Not ready, new object         REDBK23
 ______ CDCB001# PROGRAM         REDBOOK  Not ready, new object         REDBK23
 ______ CDCB0010 PROGRAM         REDBOOK  Not ready, new object         REDBK23
 ______ CDCB0020 PROGRAM         REDBOOK  Not ready, new object         REDBK23
 ______ CDCB0510 PROGRAM         REDBOOK  Not ready, new object         REDBK23
 ______ CDCB0710 PROGRAM         REDBOOK  Not ready, new object         REDBK23
 r_____ CICB0010 PROGRAM         REDBOOK  Not ready, new object         REDBK23
 r_____ CICB0020 PROGRAM         REDBOOK  Not ready, new object         REDBK23
 r_____ CICB0030 PROGRAM         REDBOOK  Not ready, new object         REDBK23
 r_____ CICB0050 PROGRAM         REDBOOK  Not ready, new object         REDBK23
 ______ COBOLVS1 PROGRAM         REDBOOK  Not ready, new object         REDBK23
 ______ COBOLVS2 PROGRAM         REDBOOK  Not ready, new object         REDBK23
 ______ CONS     TERMINAL        UTILITY  Not ready, new object         REDBK23
 ______ CSCB0010 PROGRAM         REDBOOK  Not ready, new object         REDBK23
 ______ CSCB0030 PROGRAM         REDBOOK  Not ready, new object         REDBK23
 ______ CSCB0200 PROGRAM         REDBOOK  Not ready, new object         REDBK23
 ______ DB7P     DB2CONN         REDBOOK  Not ready, new object         REDBK23
 ______ LSRPOOL1 LSRPOOL         UTILITY  Not ready, new object         REDBK23
 ______ RDBA     TRANSACTION     REDBOOK  Not ready, new object         REDBK23
 ______ RDBB     TRANSACTION     REDBOOK  Not ready, new object         REDBK23
 ______ RDBC     TRANSACTION     REDBOOK  Not ready, new object         REDBK23
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 9-23 Removing resources from a package

9.4.2 Ready the package

The package is now readied using option **3. Ready** from the Change Package action menu. The screen shown in Figure 9-24 appears and then the **1. Ready** processing option is chosen.

```
   File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 Ready                          Change Package REDBOOK
 Command ===> ________________________________________________________________

 Name  . . . . . : REDBOOK
 Description . . : Migrate 2.3 CSD to 3.1 Plex
 Scheme name . . : REDBOOK
 Scheme status . : Ready, no approvals required

 Choose a processing option and press Enter
 1  1. Ready      Ready the change package
    2. UnReady    Remove ready status from the change package
    3. List       List the ready-candidate CICS resources

 Execution mode               Enter "/" to select option
 2  1. Foreground             /  Edit JCL before user submit
    2. Batch





  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F12=Cancel
```

Figure 9-24   Ready the package

9.4.3 Migrating the resources

In this scenario, the change package has now been created, resources have been packaged from the source CSD, transformation rules have been applied for some of the resource types, a migration scheme has been created, and the package has been made ready. Now the package can be migrated.

Option *5. Migrate* from the Change Package action menu is shown in Figure 9-25. Choose the **1. Migrate** processing option and press Enter.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 Migrate                     Change Package REDBOOK
 Command ===> _______________________________________________________________

 Name  . . . . . . : REDBOOK
 Description . . : Migrate 2.3 CSD to 3.1 Plex
 Scheme name . . : REDBOOK
 Scheme status . : Ready, no approvals required

 Choose a processing option and press Enter
 1  1. Migrate   Migrate the change package
    2. List      List the migration-candidate CICS resources

 Execution mode              Enter "/" to select option
 1  1. Foreground            c  Edit JCL before user submit
    2. Batch









  F1=Help      F3=Exit      F4=Prompt     F5=Rlocate    F6=Zoom      F7=Backward
  F8=Forward  F12=Cancel
```

Figure 9-25 Migrate resources using change packages

Migration of resources from CSD to BAS is now complete using a change package and transformation rules.

9.4.4 Using CM's COPY function and EXIT to migrate

The previous discussion has shown the preparation of change packages in CICS CM to use as the input for the migration of resources to a target data repository. There is an alternative method that can be employed using the COPY function, which will achieve the same result.

We also demonstrate the addition of a user exit into this scenario, which changes resource attributes during the COPY, just as transform rules did with the change package scenario.

The use of transformation rules in change packages has been discussed in "Create a transform rule" on page 287. Transformation rules cannot be used in conjunction with COPY. Instead, a sample user exit is provided, which produces the same outcome as when using a transformation rule.

The following figures demonstrate the use of the COPY command and a sample user exit CCV9RDBK.

Enable the user exit

CICS CM supplies a sample user exit to perform COPY, UPDATE, or CREATE actions on resource definitions during a migration. This is distributed in the CICS CM sample library as program CCVXRANC, which is a sample resource attribute migration exit.

The modified version of this module that we use in this scenario is called CCV9RDBK and is shown in Example 9-5. This exit performs the same operations on the resource attributes as the transformation rules we created in "Define rules for transformation of individual resources" on page 288.

Example 9-5 Sample resource attribute CICS TS 3.1 migration exit

```
*-----------------------------------------------------------------
      *
      * Module Name:  CCV9RDBK
      * Product:      CICS Configuration Manager
      * Version:      V1R2M0
      * Author:       Fundi Software (c) 2006.
      *
      * Copyright:    IBM CICS Configuration Manager for z/OS
      *               Licensed Materials - Property of Fundi Software
      *               5697-I78 (C) Copyright Fundi Software 2006.
      *                        (C) Copyright IBM Corp 2006.
      *               All Rights Reserved. Use, duplication or
      *               disclosure restricted.
      *
      * Description:  Sample Resource Attribute TS 3.1 Migration exit.
      *
      * Input:        Via COMMAREA, a parameter list described by
      *               CCVXRAC
      * Output:       An accepted or rejected request, indicated by
      *               status information in the supplied parameter
      *               list; the Object IO Area may also be modified
      *
      * Function:
      *               1.  For Program Objects in the REDBOOK Group,
      *                   ensure that Language Cobol is set to LE370.
      *               2.  For Tcpipservice Objects in the REDBOOK Group,
      *                   ensure that MAXDATALEN is set to 1000K.
      *               3.  For File Objects in the REDBOOK Group,
      *                   ensure that the DSN is altered from
      *                   REDBK23.REDBKV23 to REDBK31.REDBKV31.
      *
      * Notes:        This code is for TS 3.1 attributes
      *
      * Attributes:   AMODE 31, RMODE ANY, re-entrant
      *
      *-----------------------------------------------------------------
      * Modification Log (most recent first) :
```

```
*
* Level   Date      Who Comment
* ------- --------- --- -----------------------------------------
* Base120   Apr2006 SRP Initial creation
*
*-----------------------------------------------------------------

 IDENTIFICATION DIVISION.
 PROGRAM-ID. CCVXRDBK.
 AUTHOR. Fundi Software.
 DATE-COMPILED.

 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.

 DATA DIVISION.

 WORKING-STORAGE SECTION.

 01  C-CONSTANTS.
     03  C-RC4                   PIC S9(8) COMP VALUE +4.
     03  C-RC8                   PIC S9(8) COMP VALUE +8.
     03  C-RC16                  PIC S9(8) COMP VALUE +16.
     03  C-R8000                 PIC X(02)      VALUE X'8000'.
     03  C-R8001                 PIC X(02)      VALUE X'8001'.
     03  C-PROGOBJ               PIC X(08)      VALUE 'PROGDEF'.
     03  C-TCPOBJ                PIC X(08)      VALUE 'TCPDEF'.
     03  C-FILEOBJ               PIC X(08)      VALUE 'FILEDEF'.
     03  C-1000K                 PIC X(06)      VALUE '001000'.
     03  C-SOURCE-DSN            PIC X(17)      VALUE
             'REDBK23.REDBKV23.'.
     03  C-TARGET-DSN            PIC X(17)      VALUE
             'REDBK31.REDBKV31.'.
     03  C-TGTIOA                PIC X(32)      VALUE
             'Target IO Area'.
     03  C-REDBOOK               PIC X(08)      VALUE 'REDBOOK'.

 01  W-VARIABLES.
     03  W-POMITTED.
         05  FILLER              PIC X(35)      VALUE
             'RANC: Required parameter omitted - '.
         05  W-POMITTED-REASON   PIC X(32).
     03  W-DSNAME.
         05  DSN-HLQ             PIC X(17).
         05  FILLER              PIC X(43).

*.   Transaction Object layout

     COPY PROGDEF.
     COPY TCPDEF.
     COPY FILEDEF.

 LINKAGE SECTION.

*.   Parameter list copy book to map COMMAREA

 01  CCVXRAC-PLIST.
     COPY CCVXRAC.

*.   Dummy structure addressing the Object IO Area

 01  OBJECT-IOAREA.
     03  FILLER                  PIC X(2048).

 PROCEDURE DIVISION.

*    -----------------------------------------------
```

```
*    Check that Parameter List is the correct length
*    -----------------------------------------------

     IF  EIBCALEN NOT = LENGTH OF CCVXRAC-PLIST
         EXEC CICS ABEND
                   ABCODE('CCIP')
         END-EXEC
     ELSE
         SET ADDRESS OF CCVXRAC-PLIST TO ADDRESS OF DFHCOMMAREA
     END-IF

*  --------------------------
*  Initialise response fields
*  --------------------------

     MOVE ZERO       TO RANA-EXIT-RC
     MOVE LOW-VALUES TO RANA-EXIT-RSN-MODID
                        RANA-EXIT-RSN-CODE
     MOVE SPACES     TO RANA-MSGTEXT-1
                        RANA-MSGTEXT-2


*  ------------------------------------------------------------
*  For the REDBOOK Group only:
*
*  Make Program Language COBOL = LE370.
*  Make TCPIPSERVICE Maxdatalen = 1000k.
*  Make FILE Dsname = REDBK31.REDBKV31.*
*  ------------------------------------------------------------

     IF  RANA-OBJGROUP = C-REDBOOK   AND
         RANA-OBJTYPE  = C-PROGOBJ
         IF  RANA-TARGET-IOAREA EQUAL NULLS
             MOVE C-TGTIOA   TO W-POMITTED-REASON
             MOVE W-POMITTED TO RANA-MSGTEXT-1
             MOVE C-RC16     TO RANA-EXIT-RC
             MOVE C-R8000    TO RANA-EXIT-RSN-CODE
             EXEC CICS RETURN NOHANDLE END-EXEC
         ELSE
             SET ADDRESS OF OBJECT-IOAREA TO RANA-TARGET-IOAREA
             MOVE OBJECT-IOAREA TO PROGDEF
         END-IF

         IF  LANGUAGE  = EYUVALUE(COBOL)
             MOVE C-RC4              TO RANA-EXIT-RC
             MOVE C-R8001            TO RANA-EXIT-RSN-CODE
             MOVE EYUVALUE(LE370)    TO LANGUAGE
         END-IF


         MOVE PROGDEF TO OBJECT-IOAREA(1:LENGTH OF PROGDEF)

     END-IF

     IF  RANA-OBJGROUP = C-REDBOOK   AND
         RANA-OBJTYPE  = C-TCPOBJ
         IF  RANA-TARGET-IOAREA EQUAL NULLS
             MOVE C-TGTIOA   TO W-POMITTED-REASON
             MOVE W-POMITTED TO RANA-MSGTEXT-1
             MOVE C-RC16     TO RANA-EXIT-RC
             MOVE C-R8000    TO RANA-EXIT-RSN-CODE
             EXEC CICS RETURN NOHANDLE END-EXEC
         ELSE
             SET ADDRESS OF OBJECT-IOAREA TO RANA-TARGET-IOAREA
             MOVE OBJECT-IOAREA TO TCPDEF
         END-IF
```

```
    MOVE C-RC4              TO RANA-EXIT-RC
    MOVE C-R8001            TO RANA-EXIT-RSN-CODE
    MOVE C-1000K            TO MAXDATALEN


    MOVE TCPDEF TO OBJECT-IOAREA(1:LENGTH OF TCPDEF)

END-IF

IF  RANA-OBJGROUP = C-REDBOOK   AND
    RANA-OBJTYPE  = C-FILEOBJ
    IF  RANA-TARGET-IOAREA EQUAL NULLS
        MOVE C-TGTIOA   TO W-POMITTED-REASON
        MOVE W-POMITTED TO RANA-MSGTEXT-1
        MOVE C-RC16     TO RANA-EXIT-RC
        MOVE C-R8000    TO RANA-EXIT-RSN-CODE
        EXEC CICS RETURN NOHANDLE END-EXEC
    ELSE
        SET ADDRESS OF OBJECT-IOAREA TO RANA-TARGET-IOAREA
        MOVE OBJECT-IOAREA TO FILEDEF
    END-IF

    MOVE DSNAME-A               TO W-DSNAME
    IF  DSN-HLQ        = C-SOURCE-DSN
        MOVE C-RC4              TO RANA-EXIT-RC
        MOVE C-R8001            TO RANA-EXIT-RSN-CODE
        MOVE C-TARGET-DSN       TO DSN-HLQ
        MOVE W-DSNAME           TO DSNAME-A
    END-IF

    MOVE FILEDEF TO OBJECT-IOAREA(1:LENGTH OF FILEDEF)

END-IF

EXEC CICS RETURN NOHANDLE END-EXEC.
```

To activate the exit so that it will be executed during the COPY in migration of our resource definitions, the following figures show the steps that need to be performed.

Select option **6. Exit Points** from the Administration menu. Specify the name of the exit program and then activate the exit with the `save` command. This is shown in Figure 9-26.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
                               Exit Programs          21 exit points inactive
Command ===> save
 ------------------------------------------------------------------------------
     Enter "/" to activate exit point program
     Press END to save updates, or CANCEL to cancel request.
                                                                 More:     +
                                   Exit      Exit      PPT      Program   Execution
     CICS Resource Exits          Point     Program   Resolved  Loadable   Status
     ----------------------      --------  --------  --------  --------  --------
  /  CPSM contexts               RANACP01  ccv9rdbk     N         N      Inactive
  _  CSD files                   RANACS01  NOTINUSE     N         N      Inactive
  _  Export files                RANAEX01  NOTINUSE     N         N      Inactive

     Change Package                Exit      Exit      PPT      Program   Execution
     Pre-Processing Exits         Point     Program   Resolved  Loadable   Status
     ----------------------      --------  --------  --------  --------  --------
  _  APPROVE                     CPPRAP01  NOTINUSE     N         N      Inactive
  _  BACKOUT                     CPPRBO01  NOTINUSE     N         N      Inactive
  _  DISAPPROVE                  CPPRDA01  NOTINUSE     N         N      Inactive
  _  IMPORT                      CPPRIM01  NOTINUSE     N         N      Inactive
  _  INSTALL                     CPPRIN01  NOTINUSE     N         N      Inactive
  _  MIGRATE                     CPPRMG01  NOTINUSE     N         N      Inactive
  _  NEWCOPY                     CPPRNC01  NOTINUSE     N         N      Inactive
  _  READY                       CPPRRD01  NOTINUSE     N         N      Inactive
  _  UNREADY                     CPPRUR01  NOTINUSE     N         N      Inactive

     Change Package               Exit      Exit      PPT      Program   Execution
     Post-Processing Exits        Point     Program   Resolved  Loadable   Status
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate    F6=Zoom      F7=Backward
  F8=Forward   F10=Actions  F12=Cancel
```

Figure 9-26 Specifying the exit name

At this point notice that the execution status still says `Inactive`. The screen needs to be refreshed to reflect the correct status as `Active`. This is seen in Figure 9-27.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
                                Exit Programs
 Command ===> ____________________________________________________________

      Enter "/" to activate exit point program
      Press END to save updates, or CANCEL to cancel request.
                                                                    More:    +
                                     Exit      Exit      PPT      Program  Execution
      CICS Resource Exits            Point     Program   Resolved Loadable  Status
      ----------------------        --------  --------  -------- --------  --------
   /  CPSM contexts                 RANACP01  CCV9RDBK     Y        Y      Active
   _  CSD files                     RANACS01  NOTINUSE     N        N      Inactive
   _  Export files                  RANAEX01  NOTINUSE     N        N      Inactive

      Change Package                 Exit      Exit      PPT      Program  Execution
      Pre-Processing Exits           Point     Program   Resolved Loadable  Status
      ----------------------        --------  --------  -------- --------  --------
   _  APPROVE                       CPPRAP01  NOTINUSE     N        N      Inactive
   _  BACKOUT                       CPPRBO01  NOTINUSE     N        N      Inactive
   _  DISAPPROVE                    CPPRDA01  NOTINUSE     N        N      Inactive
   _  IMPORT                        CPPRIM01  NOTINUSE     N        N      Inactive
   _  INSTALL                       CPPRIN01  NOTINUSE     N        N      Inactive
   _  MIGRATE                       CPPRMG01  NOTINUSE     N        N      Inactive
   _  NEWCOPY                       CPPRNC01  NOTINUSE     N        N      Inactive
   _  READY                         CPPRRD01  NOTINUSE     N        N      Inactive
   _  UNREADY                       CPPRUR01  NOTINUSE     N        N      Inactive

      Change Package                 Exit      Exit      PPT      Program  Execution
      Post-Processing Exits          Point     Program   Resolved Loadable  Status
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate    F6=Zoom       F7=Backward
  F8=Forward   F10=Actions  F12=Cancel
```

Figure 9-27 The exit program is active

Now we need to check that system-wide exit processing has been activated. Otherwise the exit program will not be executed, even though it has an execution status of *active*.

Select option **1. System Options** from the Administration menu and ensure that “Exit Point processing is active in the CICS CM Server” has been selected. This is seen in Figure 9-28.

```
  File  Settings  Help
 ------------------------------------------------------------------------------
                                System Options
Command ===> __________________________________________________________________

Options
  Enter "/" to select option
  /  Approvals checking for change packages
  _  Automatic change package numbering
  _  Security checking for server API commands
  /  Transformation processing for resource migrations
  /  Exit Point processing in the CICS CM Server

Security Controls
  SAF resource class . . . #CCV
  Security key prefix  . . DUMMY










 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=Actions  F12=Cancel
```

Figure 9-28 System option to activate exit point processing

COPY the resources

As with the package scenario for multiple groups, choose option **1. Multiple Configs** from the Reporting menu. Figure 9-29 shows a list of all the resources from all groups selected, which are now selected for copy.

> **Tip:** Use the command **c *** as a shortcut to mark all the resources in the display and subsequent pages. In our example, there are 43 resources in the groups that we want to copy.

```
  File  Menu  Settings  Checksum  Search  Help
 ------------------------------------------------------------------------------
 Group Resources       Multiple Configuration Resources        Row 1 to 22 of 43
 Command ===> ________________________________________________ Scroll ===> CSR

 Filter *_______ *____________ +

        Name     Type           Group    Prompt                         Config
 C_____ CCVR     TRANSACTION    REDBOOK                                 REDBK23
 C_____ CCVSREMP PROGRAM        REDBOOK                                 REDBK23
 C_____ CDCB001# PROGRAM        REDBOOK                                 REDBK23
 C_____ CDCB0010 PROGRAM        REDBOOK                                 REDBK23
 C_____ CDCB0020 PROGRAM        REDBOOK                                 REDBK23
 C_____ CDCB0510 PROGRAM        REDBOOK                                 REDBK23
 C_____ CDCB0710 PROGRAM        REDBOOK                                 REDBK23
 C_____ CICB0010 PROGRAM        REDBOOK                                 REDBK23
 C_____ CICB0020 PROGRAM        REDBOOK                                 REDBK23
 C_____ CICB0030 PROGRAM        REDBOOK                                 REDBK23
 C_____ CICB0050 PROGRAM        REDBOOK                                 REDBK23
 C_____ COBOLVS1 PROGRAM        REDBOOK                                 REDBK23
 C_____ COBOLVS2 PROGRAM        REDBOOK                                 REDBK23
 C_____ CONS     TERMINAL       UTILITY                                 REDBK23
 C_____ CSCB0010 PROGRAM        REDBOOK                                 REDBK23
 C_____ CSCB0030 PROGRAM        REDBOOK                                 REDBK23
 C_____ CSCB0200 PROGRAM        REDBOOK                                 REDBK23
 C_____ DB7P     DB2CONN        REDBOOK                                 REDBK23
 C_____ LSRPOOL1 LSRPOOL        UTILITY                                 REDBK23
 C_____ RDBA     TRANSACTION    REDBOOK                                 REDBK23
 C_____ RDBB     TRANSACTION    REDBOOK                                 REDBK23
 C_____ RDBC     TRANSACTION    REDBOOK                                 REDBK23
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 9-29 Copy the resources to be migrated

The interim screen appears to specify the name of the group and confirms the name of the target environment in which the objects will be copied. This is shown in Figure 9-30.

```
  File  Menu  Settings  Checksum  Search  Help
 - ------------------- CICS Definitions -------------           ----------------
 G                   Copy Confirmation                          Row 1 to 22 of 43
 C                                                               Scroll ===> CSR
   Enter details and press Enter to copy resources.
 F
   Copy target details
     Group  . . . . . . . . .  *_______  +                          Config
 C   CICS Configuration . . .  CCVPLEXE  +                          REDBK23
 C                                                                  REDBK23
 C Processing options                                               REDBK23
 C   Replace definitions   . . NO_   (No, Yes)                      REDBK23
 C                                                                  REDBK23
 C  F1=Help      F3=Exit      F4=Prompt     F6=Resize               REDBK23
 C  F7=Backward  F8=Forward  F12=Cancel                             REDBK23
 C                                                                  REDBK23
 C      CICB0020 PROGRAM        REDBOOK                             REDBK23
 C      CICB0030 PROGRAM        REDBOOK                             REDBK23
 C      CICB0050 PROGRAM        REDBOOK                             REDBK23
 C      COBOLVS1 PROGRAM        REDBOOK                             REDBK23
 C      COBOLVS2 PROGRAM        REDBOOK                             REDBK23
 C      CONS     TERMINAL       UTILITY                             REDBK23
 C      CSCB0010 PROGRAM        REDBOOK                             REDBK23
 C      CSCB0030 PROGRAM        REDBOOK                             REDBK23
 C      CSCB0200 PROGRAM        REDBOOK                             REDBK23
 C      DB7P     DB2CONN        REDBOOK                             REDBK23
 C      LSRPOOL1 LSRPOOL        UTILITY                             REDBK23
 C      RDBA     TRANSACTION    REDBOOK                             REDBK23
 C      RDBB     TRANSACTION    REDBOOK                             REDBK23
 C      RDBC     TRANSACTION    REDBOOK                             REDBK23
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 9-30 Copy Confirmation

Figure 9-31 shows that the resources have been successfully copied. It is during the execution of this COPY that the exit program CCV9RDBK has been invoked, which has changed all the resource attributes of the object definitions that were specified in the module.

```
  File  Menu  Settings  Checksum  Search  Help
 --------------------------------------------------------------------------
 Group Resources        Multiple Configuration Resources     Row 1 to 22 of 43
 Command ===> ______________________________________________  Scroll ===> CSR

 Filter *_______ *____________ +

        Name     Type           Group     Prompt                        Config
        CCVR     TRANSACTION    REDBOOK   *Copied                       REDBK23
 ______ CCVSREMP PROGRAM        REDBOOK   *Copied                       REDBK23
 ______ CDCB001# PROGRAM        REDBOOK   *Copied                       REDBK23
 ______ CDCB0010 PROGRAM        REDBOOK   *Copied                       REDBK23
 ______ CDCB0020 PROGRAM        REDBOOK   *Copied                       REDBK23
 ______ CDCB0510 PROGRAM        REDBOOK   *Copied                       REDBK23
 ______ CDCB0710 PROGRAM        REDBOOK   *Copied                       REDBK23
 ______ CICB0010 PROGRAM        REDBOOK   *Copied                       REDBK23
 ______ CICB0020 PROGRAM        REDBOOK   *Copied                       REDBK23
 ______ CICB0030 PROGRAM        REDBOOK   *Copied                       REDBK23
 ______ CICB0050 PROGRAM        REDBOOK   *Copied                       REDBK23
 ______ COBOLVS1 PROGRAM        REDBOOK   *Copied                       REDBK23
 ______ COBOLVS2 PROGRAM        REDBOOK   *Copied                       REDBK23
 ______ CONS     TERMINAL       UTILITY   *Copied                       REDBK23
 ______ CSCB0010 PROGRAM        REDBOOK   *Copied                       REDBK23
 ______ CSCB0030 PROGRAM        REDBOOK   *Copied                       REDBK23
 ______ CSCB0200 PROGRAM        REDBOOK   *Copied                       REDBK23
 ______ DB7P     DB2CONN        REDBOOK   *Copied                       REDBK23
 ______ LSRPOOL1 LSRPOOL        UTILITY   *Copied                       REDBK23
 ______ RDBA     TRANSACTION    REDBOOK   *Copied                       REDBK23
 ______ RDBB     TRANSACTION    REDBOOK   *Copied                       REDBK23
 ______ RDBC     TRANSACTION    REDBOOK   *Copied                       REDBK23
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate    F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 9-31 Resources successfully copied

The migration from CSD to BAS using CICS CM copy and a user exit is now complete.

9.4.5 Using CICS CM to compare source and target repositories

This section shows another function of CICS CM that is useful as a post migration task. The use of the COMPARE function enables the comparison of before and after images of the source and target data repositories, whether they are CSD or BAS owned.

To highlight this function, we first remove four programs from the REDBOOK package on the source CSD. The remove function has already been discussed in

“Removal of object resources from a package” on page 302, and Figure 9-32 shows the resources removed.

```
  File  Menu  Settings  Help
-------------------------------------------------------------------------------
List                       Change Package REDBOOK                Row 1 to 22 of 43
Command ===> ________________________________________________  Scroll ===> CSR

Filter *        *            +  *

/     Name     Type           Group    Status                       Config
      CCVR     TRANSACTION    REDBOOK  Not ready, new object        REDBK23
      CCVSREMP PROGRAM        REDBOOK  Not ready, new object        REDBK23
      CDCB001# PROGRAM        REDBOOK  Not ready, new object        REDBK23
      CDCB0010 PROGRAM        REDBOOK  Not ready, new object        REDBK23
      CDCB0020 PROGRAM        REDBOOK  Not ready, new object        REDBK23
      CDCB0510 PROGRAM        REDBOOK  Not ready, new object        REDBK23
      CDCB0710 PROGRAM        REDBOOK  Not ready, new object        REDBK23
r     CICB0010 PROGRAM        REDBOOK  Not ready, new object        REDBK23
r     CICB0020 PROGRAM        REDBOOK  Not ready, new object        REDBK23
r     CICB0030 PROGRAM        REDBOOK  Not ready, new object        REDBK23
r     CICB0050 PROGRAM        REDBOOK  Not ready, new object        REDBK23
      COBOLVS1 PROGRAM        REDBOOK  Not ready, new object        REDBK23
      COBOLVS2 PROGRAM        REDBOOK  Not ready, new object        REDBK23
      CONS     TERMINAL       UTILITY  Not ready, new object        REDBK23
      CSCB0010 PROGRAM        REDBOOK  Not ready, new object        REDBK23
      CSCB0030 PROGRAM        REDBOOK  Not ready, new object        REDBK23
      CSCB0200 PROGRAM        REDBOOK  Not ready, new object        REDBK23
      DB7P     DB2CONN        REDBOOK  Not ready, new object        REDBK23
      LSRPOOL1 LSRPOOL        UTILITY  Not ready, new object        REDBK23
      RDBA     TRANSACTION    REDBOOK  Not ready, new object        REDBK23
      RDBB     TRANSACTION    REDBOOK  Not ready, new object        REDBK23
      RDBC     TRANSACTION    REDBOOK  Not ready, new object        REDBK23
 F1=Help      F3=Exit       F4=Prompt     F5=Rlocate    F6=Zoom       F7=Backward
 F8=Forward  F10=Actions   F12=Cancel
```

Figure 9-32 Removal of programs from the REDBOOK package

Once removed, the package is readied and migrated using the same transform rules from our earlier scenario in “Define rules for transformation of individual resources” on page 288. The result is that there are only 39 resources in the group representing all but the four programs that we have removed.

Now, using option *1. Multiple Configs* from the Reports menu, the source CSD and the target data repository CCVPLEXE are selected using the **g** line command. This is shown in Figure 9-33.

```
   File  Settings  Help
 ------------------------------------------------------------------------------
 Reports                     CICS Configurations                Row 1 to 20 of 21
 Command ===> ________________________________________________  Scroll ===> CSR

 Select one or more CICS configurations and press Enter

 Filter *_______

      / Name      Description
        CCMPLEXD  Redbook 3.1 Plex MP CCVT31C
      _ CCVPLEXC
      _ CCVPLEXD  cpsm ts 3.1 plexd
      g CCVPLEXE  Redbook 3.1 Plex MP CCVT32C
      _ CCVT22M   cics ts 2.2
      _ CCVT23M   cics ts 2.3
      _ CCVT31A   ccvt31a - csd
      _ CCVT31M   cics ts 3.1
      _ CCVT31T   cics ts 3.1
      _ CCVT32A   Redbook 2.3 CSD
      _ CCVWDTR   CICS Configuration - CCVWSRP
      _ CCVWSRP   CICS Configuration - CCVWSRP
      _ CCVWTST   CICS Configuration - CCVWTST
      _ KXWCONFA  KXWCONFA (2.2)
      _ KXWCONFB  kxwconfb (2.2)
      _ KXWCONFE  kxwconfe (2.2)
        PLEXD     CICS Configuration - CPSM31 - CCVD
      g REDBK23   REDBOOK v2.3 CSD
        REDBK31   REDBOOK v3.1 CSD
      _ REDEXP
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 9-33 Selecting the target and source repositories

The resulting screen (Figure 9-34) shows the names of all the groups from each data repository chosen. Our group REDBOOK appears twice as it is represented in both repositories. To compare the contents of REDBOOK from both repositories use the compare multiple **cm** line command.

```
  File  Menu  Settings  Checksum  Help
------------------------------------------------------------------------------
Groups                 Multiple Configuration Resources  Row 133 to 138 of 138
Command ===> ________________________________________________ Scroll ===> CSR

Filter *_______

/      Group    Prompt                                                 Config
       DFHVTAMP                                                        REDBK23
______ DFHWEB                                                          REDBK23
cm____ REDBOOK                                                         CCVPLEXE
cm____ REDBOOK                                                         REDBK23
______ UTILITY                                                         CCVPLEXE
______ UTILITY                                                         REDBK23
******************************* Bottom of data ********************************















 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=Actions  F12=Cancel
```

Figure 9-34 Comparing the group REDBOOK from chosen data repositories

Figure 9-35 shows the resulting in-line report of the compare, listing all the resources found in both of the repositories selected. The figure shows the CCVPLEXE results on the left hand of the report and the DFHCSD on the right. We can see that the four programs that we removed earlier are missing from CCVPLEXE.

```
  File  Menu  Settings  Checksum  Help
------------------------------------------------------------------------------
Compare - All Resources                     Program          Row 1 to 12 of 41
Command ===> ________________________________________________ Scroll ===> CSR

  To prepare other reports, activate Checksum

  ResGroup  . . : REDBOOK                       REDBOOK
  Location  . . : CCVPLEXE                      REDBK23.REDBKV23.DFHCSD
  Change Date . : 2006/06/08 16:34              2006/06/06 11:53
  Anomaly flags : Missing, Checksum

                              All Group Resources
       ----------------------------------------------------------------

Filter *_______ *_____________ +
                                                Flags
/      Name     Type         ResGroup            M C         Group      /
       CCVR     TRANSACTION  REDBOOK                         REDBOOK    ______
______ CCVSREMP PROGRAM      REDBOOK                         REDBOOK    ______
______ CDCB001# PROGRAM      REDBOOK                         REDBOOK    ______
______ CDCB0010 PROGRAM      REDBOOK                         REDBOOK    ______
______ CDCB0020 PROGRAM      REDBOOK                         REDBOOK    ______
______ CDCB0510 PROGRAM      REDBOOK                         REDBOOK    ______
______ CDCB0710 PROGRAM      REDBOOK                         REDBOOK    ______
______ CICB0010 PROGRAM               *Missing   M           REDBOOK    ______
______ CICB0020 PROGRAM               *Missing   M           REDBOOK    ______
______ CICB0030 PROGRAM               *Missing   M           REDBOOK    ______
______ CICB0050 PROGRAM               *Missing   M           REDBOOK    ______
______ COBOLVS1 PROGRAM      REDBOOK                         REDBOOK    ______
 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 9-35 Resulting in-line report with compare results

To compare the attributes of each object in the group, a checksum can be applied to the resource. To turn this on, drop down the Checksum panel option and select **checksum FULL**. This is shown in Figure 9-36.

```
  File  Menu  Settings  Checksum  Help
 _____________________ ____________________________________________________  _
 Compare - All Resourc| 2  1. Checksum OFF                                 | 1
 Command ===>         |    2. Checksum FULL for all fields                 |
                      |    3. Checksum PARTIAL for CICS CM selected fields |
   To prepare other re|    4. Checksum LIST for specified user fields      |
                      |____________________________________________________|
   Group . . . . . : REDBOOK                        REDBOOK
   Location   . . : REDBK23.REDBKV23.DFHCSD         CCVPLEXE
   Change Date . : 2006/06/06 11:53                 2006/06/08 16:34
   Anomaly flags : Missing, Checksum

                         All Group Resources
        ________________________________________________________________

 Filter *        *            +
                                          Flags
 /      Name      Type        Group        M C               ResGroup /
        CCVR      TRANSACTION REDBOOK                        REDBOOK
        CCVSREMP  PROGRAM     REDBOOK                        REDBOOK
        CDCB001#  PROGRAM     REDBOOK                        REDBOOK
        CDCB0010  PROGRAM     REDBOOK                        REDBOOK
        CDCB0020  PROGRAM     REDBOOK                        REDBOOK
        CDCB0510  PROGRAM     REDBOOK                        REDBOOK
        CDCB0710  PROGRAM     REDBOOK                        REDBOOK
        CICB0010  PROGRAM     REDBOOK          M    *Missing
        CICB0020  PROGRAM     REDBOOK          M    *Missing
        CICB0030  PROGRAM     REDBOOK          M    *Missing
        CICB0050  PROGRAM     REDBOOK          M    *Missing
        COBOLVS1  PROGRAM     REDBOOK                        REDBOOK
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 9-36   Turn on checksum to compare resource attributes

Figure 9-37 shows the result of the checksum. The C in the Flag column in the report indicates that the checksums are different for the resources listed. To further examine the differences—to check that these were intended during the migration—use the **cm** line command against each object in the two locations. This is also demonstrated in Figure 9-37.

```
  File  Menu  Settings  Checksum  Help
 ------------------------------------------------------------------------------
 Compare - All Resources              Group                    Row 1 to 12 of 41
 Command ===> ________________________________________________ Scroll ===> CSR

   Scroll right (NextPage) to view other reports

   Group . . . . : REDBOOK                      REDBOOK
   Location  . . : REDBK23.REDBKV23.DFHCSD      CCVPLEXE
   Change Date . : 2006/06/06 11:53             2006/06/08 16:34
   Anomaly flags : Missing, Checksum

                            All Group Resources
        ----------------------------------------------------------------

 Filter *        *            +
                                      Checksum Flags Checksum
 /      Name     Type         Group     Full    M C    Full    ResGroup /
        CCVR     TRANSACTION  REDBOOK  6F97D3F4        6F97D3F4 REDBOOK
 ______ CCVSREMP PROGRAM      REDBOOK  115D5AAD        115D5AAD REDBOOK  ______
 cm____ CDCB001# PROGRAM      REDBOOK  4AA4E7D1      C 63B96314 REDBOOK  cm____
 ______ CDCB0010 PROGRAM      REDBOOK  19410BBF      C C526CBE3 REDBOOK  ______
 ______ CDCB0020 PROGRAM      REDBOOK  716658BE      C 587BDC7B REDBOOK  ______
 ______ CDCB0510 PROGRAM      REDBOOK  3E80C3EE      C 179D472B REDBOOK  ______
 ______ CDCB0710 PROGRAM      REDBOOK  9606727F      C BF1BF6BA REDBOOK  ______
 ______ CICB0010 PROGRAM      REDBOOK  3A2CED5A   M    *Missing          ______
 ______ CICB0020 PROGRAM      REDBOOK  464DC881   M    *Missing          ______
 ______ CICB0030 PROGRAM      REDBOOK  DB4229F7   M    *Missing          ______
 ______ CICB0050 PROGRAM      REDBOOK  23806241   M    *Missing          ______
 ______ COBOLVS1 PROGRAM      REDBOOK  6FAD66B4      C 46B0E271 REDBOOK  ______
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 9-37 Checksum results

The comparison shows that for this PROGRAM resource, the transformation rule has successfully changed the program LANGUAGE attribute from COBOL to LE370. The definitions from both repositories are displayed. This can be seen in Figure 9-38.

```
  File  Menu  Settings  Hilite  Help
 ------------------------------------------------------------------------------
Compare                           Program
Command ===> __________________________________________________________________

    Program . . . : CDCB001#                     CDCB001#
    ResGroup  . . : REDBOOK                      REDBOOK
==> Location  . . : REDBK23.REDBKV23.DFHCSD      CCVPLEXE
==> Change Date . : 2006/06/06 11:53:47          2006/06/14 10:36:34
    Description . :
                                                                 More:    +
==> Language  . . : COBOL                        LE370
    Reload  . . . : NO                           NO
    Resident  . . : NO                           NO
    Usage . . . . : NORMAL                       NORMAL
    UseLPAcopy  . : NO                           NO
    Status  . . . : ENABLED                      ENABLED
    CEDF  . . . . : YES                          YES
    DataLocation  : ANY                          ANY
    ExecKey . . . : USER                         USER
    Concurrency . : QUASIRENT                    QUASIRENT
    API . . . . . : CICSAPI                      CICSAPI

   Remote Attributes
    Dynamic . . . : NO                           NO
    RemoteSystem  :
    RemoteName  . :
    TransID . . . :
    ExecutionSet  : FULLAPI                      FULLAPI

 F1=Help      F3=Exit      F4=Prompt     F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=Actions  F12=Cancel
```

Figure 9-38 Comparison of a changed program resource

Another example showing the compare of our changed TCPIPSERVICE resource definition is shown in Figure 9-39 with the change to the attribute MAXDATALEN.

```
  File  Menu  Settings  Hilite  Help
 ------------------------------------------------------------------------------
 Compare                         TCPIPservice
 Command ===> _________________________________________________________________

     TCPIPservice   : SKIPPORT                     SKIPPORT
     Group . . . . : REDBOOK                       REDBOOK
 ==> Location   . . : REDBK23.REDBKV23.DFHCSD      CCVPLEXE
 ==> Change Date . : 2006/06/08 15:26:35           2006/06/14 10:36:38
     Description . : CIU Client - TCPIPSERVICE     CIU Client - TCPIPSERVICE
                                                                    More:     +
     URM . . . . . : DFHWBADX                      DFHWBADX
     PortNumber   . . : 10329                      10329
     Status   . . . : OPEN                         OPEN
     Protocol   . . : HTTP                         HTTP
     Transaction . : CWXN                          CWXN
     Backlog . . . : 5                             5
     TSQprefix . . :
     IPaddress . . :
     SocketClose . : NO                            NO
 ==> MaxDataLen   . : 32                           1000

    Security
     SSL . . . . . : NO                            NO
     Certificate . :
     Ciphers . . . :
     Authenticate  : NO                            NO
     AttachSec . . : NOTAPPLIC                     NOTAPPLIC
     Privacy . . . : N_A                           N_A

  F1=Help      F3=Exit      F4=Prompt     F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 9-39 Comparison of a changed TCPIPSERVICE resource definition

9.5 Adding the CICS TS 3.1 MAS to the CICSPlex

Now that the resources from our CICS region have been migrated to the CICSPlex SM data repository, we need to make a few changes to the CICS TS 3.1 region to convert it to a MAS, thus enabling the region to participate in the plex and access its resources.

There are three steps to perform to achieve this:

1. The following statements were inserted into the EYUPARM member of the REDBKS2A to nominate which CMAS the new MAS will connect to at startup:

   ```
   NAME(REDBOOK)
   CICSPLEX(CCVPLEXE)
   CMASSYSID(T32C)
   ```

2. The job in Example 9-6 was used to add the CPSM group EYU310G1 to the REDBKS2A CSD and to append the group to the startup group list.

Example 9-6 Adding a CPSM group to the REDBKS2A CSD

```
//DEFMAS31 JOB ,REGION=0M,NOTIFY=&SYSUID
//CSDUP    EXEC PGM=DFHCSDUP
//STEPLIB  DD DISP=SHR,DSN=CICS.V640B40.CICS.SDFHLOAD
//         DD DISP=SHR,DSN=CICS.V640B40.CPSM.SEYULOAD
//DFHCSD   DD DISP=SHR,DSN=REDBK31.REDBKV31.DFHCSD
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
UPGRADE USING(EYU964G1)
ADD GROUP(EYU310G1) LIST(REDLIST)
/*
```

3. The following DD cards were added to the CICS REDBKS2A startup JCL:

   ```
   //STEPLIB  DD DISP=SHR,DSN=CICS.V640B40.CPSM.SEYUAUTH
   //DFHRPL   DD DISP=SHR,DSN=CICS.V640B40.CPSM.SEYULOAD
   //EYUPARM  DD DISP=SHR,DSN=REDBK31.REDBKV31.SYSIN(RED31MAS)
   ```

9.6 Using CM to install definitions into the MAS

Having confirmed that the migration was successful using the CICS CM compare function, these migrated resource definitions are now able to be installed into our migrated CICS TS 3.1 MAS, REDBKS2A. This can be achieved using the CICS CM Install function.

1. Select option **6. Install** from the change package action panel, shown in Figure 9-40.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
Edit                           Change Package REDBOOK
Command ===> __________________________________________________________________

Name . . . . : REDBOOK
Description . . Migrate 2.3 CSD to 3.1 Plex______________________

Change package settings
  Approval profile . . . ________  +
  External reference . . ________

Choose a processing command and press Enter
6   1. Package   Package CICS resources into the change package
    2. List      List CICS resources assigned to the change package
    3. Ready     Ready or unready the package for processing
    4. Approve   Approve or disapprove the change package
    5. Migrate   Migrate the change package
    6. Install   Install the package's resources into CICS regions
    7. Newcopy   Newcopy the package's maps, partitionsets or programs
    8. Backout   Backout a previous migrate of the change package
    9. History   Display the change package processing history

      Migration scheme . . . REDBOOK_  +  (Required for options 1-7)




 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=Actions  F12=Cancel
```

Figure 9-40 The CM Install option

2. From the Install action panel (Figure 9-41), select the processing option **1. Install**.

```
  File  Menu  Settings  Help
-----------------------------------------------------------------------------
Install                    Change Package REDBOOK
Command ===> _________________________________________________________________

Name  . . . . . . : REDBOOK
Description . . . : Migrate 2.3 CSD to 3.1 Plex
Scheme name . . . : REDBOOK

Choose a processing option and press Enter
1  1. Install    Install the change package
   2. List       List the install-candidate CICS resources

Execution mode              Enter "/" to select option
1  1. Foreground            /  Edit JCL before user submit
   2. Batch









 F1=Help       F3=Exit       F4=Prompt     F5=Rlocate    F6=Zoom       F7=Backward
 F8=Forward   F10=Actions   F12=Cancel
```

Figure 9-41 The Install action panel

The interim panel specifying the install parameters (Figure 9-42) appears, which allows the CPSM scope to be input as the target. Also, we choose to select the option to “Perform a CPSM unconditional (Force) install”, which will overwrite the definitions regardless of whether they exist.

Tip: If the group to be installed contains any CONNECTION resource definitions, you must provide the CPSM resource assignment RASGNDEF name (in our scenario REDBKRAS), which nominates the resource type SESSDEF that is associated with the definitions that represent the sessions to be installed with the connections.

Press Enter to perform the install.

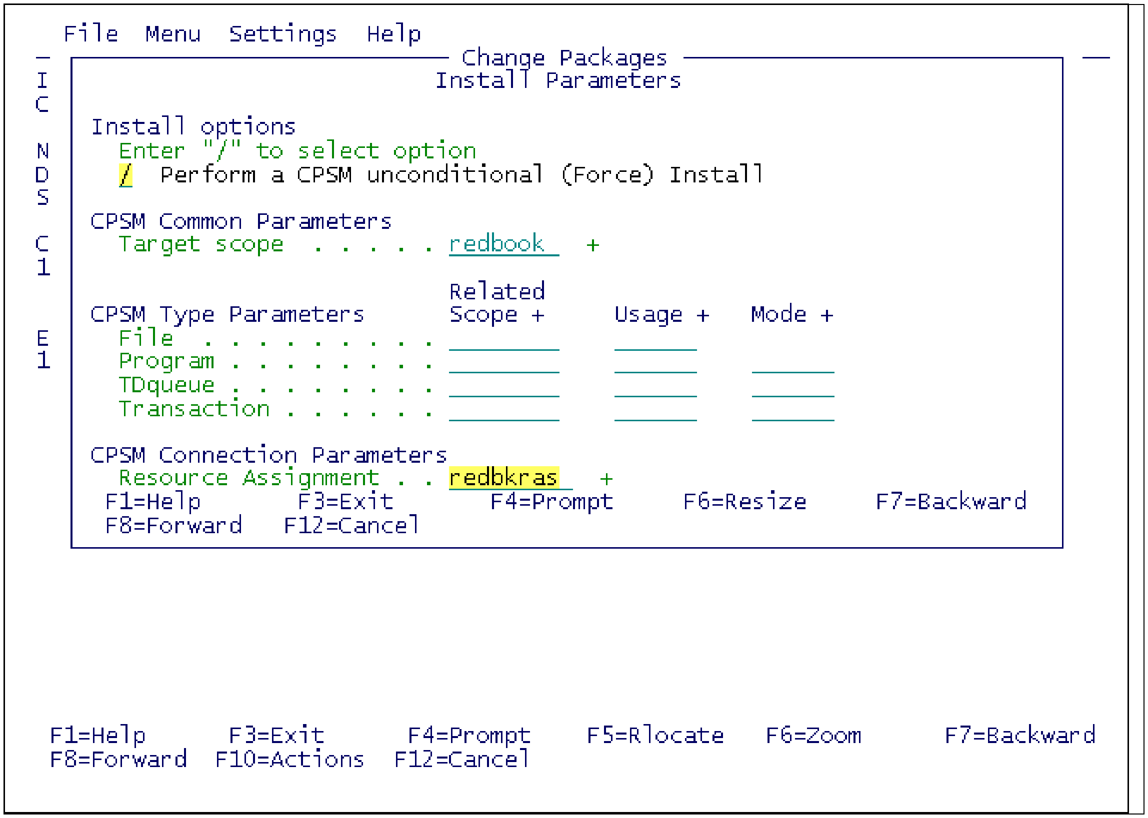


Figure 9-42 Install Parameters

Installation of the resources into our MAS is now completed.

10

Application migration

This chapter discusses the possibility of exploiting CICS Web Services capabilities after migration to CICS TS 3.1, in particular, how CICS Interdependency Analyzer (CICS IA) and CICS Configuration Manager (CICS CM) can assist in this process.

We provide an overview of CICS Web Services followed by details of our sample application and the steps we performed to convert the application to use CICS Web Services.

10.1 CICS Web Services

Web Services make it possible for applications to be integrated more rapidly, easily, and cheaply than ever before.

CICS Transaction Server provides comprehensive support for Web Services:

- A CICS application can participate in a heterogeneous Web Services environment as a service requester, as a service provider, or both.
- Support for HTTP and MQ.
- CICS Transaction Server for z/OS includes the CICS Web Services assistant, a set of utility programs that help you map WSDL service descriptions into high-level programming language data structures, and vice versa. The utility programs support these programming languages:
  - COBOL
  - PL/I
  - C
  - C++
- The CICS support for Web Services conforms to open standards including:
  - SOAP 1.1 and 1.2
  - HTTP 1.1
  - WSDL 1.1
- CICS support for Web Services ensures maximum interoperability with other Web Services implementations by conforming with the Web Services Interoperability Organization (WS-I) Basic Profile 1.1 (`http://www.ws-i.org/Profiles/BasicProfile-1.1.html`) and the WS-I Simple SOAP Binding Profile 1.0 (`http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0.html`). The profiles are a set of non-proprietary Web Services specifications, along with clarifications and amendments to those specifications, which, taken together, promote interoperability between different implementations of Web Services. Conformance with both profiles is equivalent to conforming with the WS-Basic Profile 1.0.

10.1.1 Web Service defined

A Web Service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically, Web Service Definition Language, or WSDL).

Web Services fulfill a specific task or a set of tasks. A Web Service is described using a standard, formal XML notion, called its service description, that provides all of the details necessary to interact with the service, including message formats (that detail the operations), transport protocols, and location.

The nature of the interface hides the implementation details of the service so that it can be used independently of the hardware or software platform on which it is implemented and independently of the programming language in which it is written.

This allows and encourages Web Service based applications to be loosely coupled, component-oriented, cross-technology implementations. Web Services can be used alone or in conjunction with other Web Services to carry out a complex aggregation or a business transaction.

10.1.2 How Web Services can help your business

Web Services is a technology for deploying, and providing access to, business functions over the World Wide Web. Web Services make it possible for applications to be integrated more rapidly, easily, and cheaply than ever before.

Web Services can help your business by:

- Reducing the cost of doing business
- Making it possible to deploy solutions more rapidly
- Opening up new opportunities

The key to achieving all these things is a common program-to-program communication model, built on existing and emerging standards such as HTTP, XML, SOAP, and WSDL.

The support that CICS provides for Web Services makes it possible for your existing applications to be deployed in new ways, with the minimum amount of reprogramming.

10.2 Our sample application

A sample application was created for the purposes of this and the other migration scenarios. Components of this application have been identified in previous chapters, but of interest to this scenario is the RDB3 transaction. This transaction invokes a program REDBK4, which has previously been identified by CICS IA as COBOL OS/VS (see 10.3, “CICS Interdendency Analyzer” on page 331). REDBK4 is conceptually the presentation logic for the application (see Figure 10-1 on page 330) it links to program REDBK3 passing a COMMAREA, which is populated with the last 10 updates made to a VSAM file. The COMMAREA structure is shown in Figure 10-3 on page 331, and this needs to be defined separately from the source program if the CICS Web Services assistant is to be used.

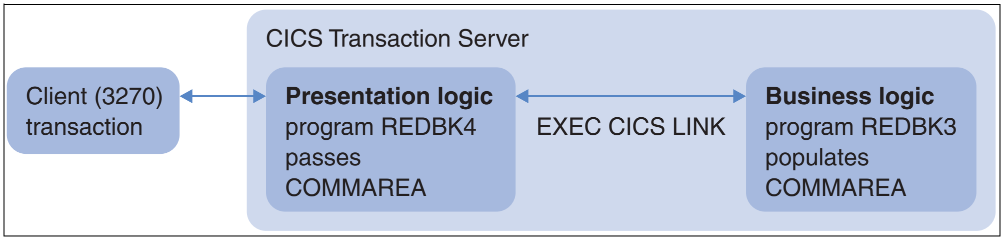


Figure 10-1 Presentation and business logic separation

Program REDBK4 could be re-compiled using a supported COBOL compiler. However, since it is the presentation logic of the application it is a candidate for CICS Web Services (see Figure 10-2).

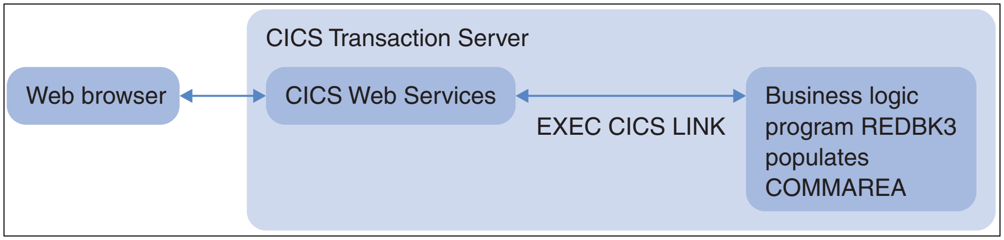


Figure 10-2 CICS Web Services enabled

```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       AYS.REDBOOK.COPY(COMMAREA) - 01.00              Columns 00001 00072
Command ===> ________________________________________________ Scroll ===> CSR
****** ***************************** Top of Data ******************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001            03  OUTMSG.
000002              05  OUTMSG1             PIC X(80).
000003              05  OUTMSG2             PIC X(80).
000004              05  OUTMSG3             PIC X(80).
000005              05  OUTMSG4             PIC X(80).
000006              05  OUTMSG5             PIC X(80).
000007              05  OUTMSG6             PIC X(80).
000008              05  OUTMSG7             PIC X(80).
000009              05  OUTMSG8             PIC X(80).
000010              05  OUTMSG9             PIC X(80).
000011              05  OUTMSG10            PIC X(80).
****** **************************** Bottom of Data ****************************
```

Figure 10-3 COBOL copybook used as COMMAREA between REDBK3 and REDBK4

10.3 CICS Interdendency Analyzer

Previously CICS IA has identified program REDBK4 as being a NON LE level of COBOL (see Figure 10-4). Through application knowledge we knew that REDBK4 was a candidate for CICS Web Services, but this was also confirmed by the CICS IA report shown in Figure 10-5 on page 332. This report shows that REDBK4 only LINKS to program REDBK3, which then performs the file operations.

CICS INTERDEPENDENCY ANALYZER Version 2.1.0
LOAD MODULE SCANNER - SUMMARY LISTING OF REDBK23.APPL.LOADLIB

Module Name	Module Length	Module Language	Language Version	Possible statements... Affinities	Dependencies	MVS POSTs
CSCB0200	00001250	COBOL	Non LE	0	1	0
REDBK1	00001630	C/370	LE	4	6	0
REDBK1A	00001630	C/370	LE	4	6	0
REDBK1B	00001630	C/370	LE	4	6	0
REDBK1C	00001630	C/370	LE	4	6	0
REDBK1D	00001630	C/370	LE	4	6	0
REDBK1E	00001630	C/370	LE	4	6	0
REDBK2	000026D8	PL/I	LE	6	6	0
REDBK3	00001720	C/370	LE	0	4	0
REDBK4	**00001738**	**COBOL**	**Non LE**	**0**	**1**	**0**
REDBK5	00001780	C/370	LE	2	6	0

Figure 10-4 CICS IA report showing REDBK4 as NON LE

```
---------+---------+---------+---------+---------+---------+---------+---------+
--SHOW ALL RESOURCES USED BY TRANSACTION RDB3 IN REGION REDBKV23
--FROM THE COLLECTOR
SELECT APPLID, TRANSID, PROGRAM, FUNCTION, TYPE, OBJECT
  FROM CIU4_CICS_DATA WHERE TRANSID='RDB3'
  AND APPLID='REDBKV23';
---------+---------+---------+---------+---------+---------+---------+---------+
APPLID    TRANSID  PROGRAM   FUNCTION  TYPE      OBJECT
---------+---------+---------+---------+---------+---------+---------+---------+
REDBKV23  RDB3     REDBK3    ASSIGN    APPLID    REDBKV23
REDBKV23  RDB3     REDBK3    ENDBR     FILE      REDBOOKF
REDBKV23  RDB3     REDBK3    READNEXT  FILE      REDBOOKF
REDBKV23  RDB3     REDBK3    STARTBR   FILE      REDBOOKF
REDBKV23  RDB3     REDBK4    LINK      PROGRAM   REDBK3
DSNE610I NUMBER OF ROWS DISPLAYED IS 5
DSNE612I DATA FOR COLUMN HEADER OBJECT COLUMN NUMBER 6 WAS TRUNCATED
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
```

Figure 10-5 CICS IA output showing transaction RDB3 resource usage

10.4 The CICS Web Services assistant

The CICS Web Services assistant simplifies the task of deploying your CICS applications in a service provider setting. It is described fully in the *CICS TS 3.1 Web Service Guide*, SC34-6458. The steps we performed in order to create our Web Service application are:

1. Create the HFS directories.
2. Create and execute JCL to invoke the DFHLS2WS utility.

10.4.1 Created the HFS directories

The DFHLS2WS utility writes files to the HFS directories identified in the JCL. These directories will not be created by the utility and hence need to be created prior to execution. The directories we created are shown in Example 10-1.

Example 10-1 HFS directories

```
/u/REDBK31/CIWS/R3C1/config
/u/REDBK31/CIWS/R3C1/shelf
/u/REDBK31/CIWS/R3C1/wsbind
/u/REDBK31/CIWS/R3C1/wsdl
```

10.4.2 Created JCL to invoke the DFHLS2WS utility

DFHLS2WS generates a Web Service description and a Web Service binding file from a high-level language data structure. A sample job was taken from the CICS TS 3.1 Web Services guide and modified as shown in Example 10-2.

Example 10-2 Sample DFHLS2WS JCL

```
//DFHLS2WS JOB (APC),'AYS',MSGCLASS=X,CLASS=A,NOTIFY=AYS,
//         REGION=0M
//JLIB     JCLLIB  ORDER=(CICS.V640B40.CICS.SDFHINST)
//*
//LS2WS     EXEC DFHLS2WS,
//          USSDIR='cics.v640b40',
//          PATHPREF='/Z17',
//          JAVADIR='java/J1.4'
//*         TMPDIR='/u/REDBK31'
//INPUT.SYSUT1 DD *
LOGFILE=/u/REDBK31/REDBOOK.LOG
PDSLIB=//AYS.REDBOOK.COPY
REQMEM=REDBK4C
RESPMEM=REDBK4C
LANG=COBOL
PGMNAME=REDBK3
URI=REDBOOK/WEBSERVICE/EXAMPLE
PGMINT=COMMAREA
WSBIND=/u/REDBK31/CIWS/R3C1/wsbind/provider/redbook.wsbind
WSDL=/u/REDBK31/CIWS/R3C1/wsdl/redbook.wsdl
*/
```

The parameters of significance are:

PDSLIB This is the data set containing the COBOL copybook, which describes the commarea being passed from the presentation to the business logic programs.

REQMEM This specifies the name of the partitioned data set member that contains the high-level language structure for the Web Service request (that is, the COBOL copybook included by REDBK3 and REDBK4 and *passed* in the COMMAREA).

RESPMEM This specifies the name of the partitioned data set member that contains the high-level language structure for the Web Service response.

PGMNAME This is the name of the program to be invoked by the CICS Web Services on receipt of the appropriate inbound request (that is, the business logic program).

WSBIND This is the fully qualified HFS name of the Web Service binding file.

WSDL This is the fully qualified HFS name of the Web Service binding file.

This JOB was submitted and completed with a zero return code. The two files identified by the WSBIND and WSDL parameters were created.

10.5 CICS resource definitions

We used CICS CM to make a PIPELINE and a TCPIPSERVICE definition. See Figure 10-6 and Figure 10-7 on page 335. Rather than making explicit URIMAP and WEBSERVICE definitions we let CICS dynamically create these.

Having installed the TCPIPSERVICE and PIPELINE definitions, using CICS CM, we used CEMT to validate the resource definitions created by CICS, as shown in Figure 10-8 on page 336 to Figure 10-11 on page 337.

```
  File  Menu  Settings  Help
Edit                              Pipeline
Command ===>

    Pipeline  . . : REDPIPE0
    Group . . . . : REDBOOK
    Location  . . : REDBK31.REDBKV31.DFHCSD
    Change Date . :
    Description . . PIPELINE definition for REDBOOK Web Service application
                                                                  More:     +
                                  Description
    Status  . . . . ENABLED    +    Initial install status
    ConfigFile  . . /Z17/usr/lpp/cicsts/cics.v640b40/samples/pipelines/basi  >
                . . csoap11provider.xml
                . .
                . .
                . .

    Shelf . . . . . /u/REDBK31/CIWS/R3C1/shelf/                              >
          . . . . .
          . . . . .
          . . . . .
          . . . . .

    WSdir . . . . . /u/REDBK31/CIWS/R3C1/wsbind/provider/                    >
          . . . . .
          . . . . .
          . . . . .
          . . . . .
```

Figure 10-6 CICS CM PIPELINE definition

```
  File  Menu  Settings  Help
Edit                            TCPIPservice
Command ===>

    TCPIPservice  : REDBKTCP
    Group . . . . : REDBOOK
    Location  . . : REDBK31.REDBKV31.DFHCSD
    Change Date . : 2006/06/19 11:53:38
    Description . . TCPIPSERVICE definition for redbook Web Serive appl.
                                                                 More:     +
                                             Description
    URM . . . . . . DFHWBADX                 User-replaceable module name
    PortNumber  . . 3010                     Port number
    Status  . . . . OPEN     +               TCP/IP service status
    Protocol  . . . HTTP        +            Application level protocol on TCP/IP port
    Transaction . . CWXN                     CICS transaction ID
    Backlog . . . . 5                        Queue backlog limit
    TSQprefix . . .                          Temporary storage queue prefix
    IPaddress . . .
    SocketClose . . NO                       Timeout for socket close (HHMMSS)
    MaxDataLen  . . 32                       Maximum data length

  Security
    SSL . . . . . . NO           +           Secure sockets layer (SSL) type
    Certificate . .
    Ciphers . . . .
    Authenticate    BASIC         +          Authentication level
    AttachSec . . . NOTAPPLIC   +            Attach-time security
    Privacy . . . . N_A           +          Not supported for CICS release 0310

  DNS Connection Balancing
    DNSgroup  . . .
    GRPcritical . . NO          +            Critical domain name service group member
```

Figure 10-7 CICS CM TCPIPSERVICE definition

```
 INQUIRE TCPIPSERVICE
 RESULT - OVERTYPE TO MODIFY
   Tcpipservice(REDBKTCP)
   Openstatus( Open )
   Port(03010)
   Protocol(Http)
   Ssltype(Nossl)
   Transid(CWXN)
   Authenticate(Basic)
   Connections(00000)
   Backlog( 00005 )
   Maxdatalen( 000032 )
   Urm( DFHWBADX )
   Privacy(Notsupported)
   Ciphers()
   Ipaddress(172.17.69.25)
   Socketclose(Wait)
   Closetimeout(000000)
   Dnsgroup()
+  Dnsstatus(              )

                                                         SYSID=RB31 APPLID=REDBKV31
                                                   TIME:  08.53.02  DATE: 06.20.06
PF 1 HELP 2 HEX 3 END       5 VAR        7 SBH 8 SFH        10 SB 11 SF
```

Figure 10-8   CEMT INQUIRE TCPIPSERVICE

```
 INQUIRE PIPELINE
 RESULT - OVERTYPE TO MODIFY
   Pipeline(REDPIPE0)
   Enablestatus( Enabled )
   Configfile(/Z17/usr/lpp/cicsts/cics.v640b40/samples/pipelines/basicsoap1)
   Configfile(1provider.xml)
   Shelf(/u/REDBK31/CIWS/R3C1/shelf/)
   Wsdir(/u/REDBK31/CIWS/R3C1/wsbind/provider/)

                                                         SYSID=RB31 APPLID=REDBKV31
                                                   TIME:  08.51.10  DATE: 06.20.06
PF 1 HELP 2 HEX 3 END       5 VAR        7 SBH 8 SFH        10 SB 11 SF
```

Figure 10-9   CEMT INQUIRE PIPELINE

```
 INQUIRE URIMAP
 RESULT - OVERTYPE TO MODIFY
   Urimap($355450)
   Usage(Pipe)
   Enablestatus( Enabled )
   Analyzerstat(Noanalyzer)
   Scheme(Http)
   Redirecttype( None )
   Tcpipservice()
   Host(*)
   Path(/REDBOOK/WEBSERVICE/EXAMPLE)
   Transaction(CPIH)
   Converter()
   Program()
   Pipeline(REDPIPE0)
   Webservice(redbook)
   Userid()
   Certificate()
   Ciphers()
+  Templatename()

                                                          SYSID=RB31 APPLID=REDBKV31
                                                   TIME:  08.51.56  DATE: 06.20.06
PF 1 HELP 2 HEX 3 END       5 VAR        7 SBH 8 SFH       10 SB 11 SF
```

Figure 10-10 CEMT INQUIRE URIMAP

```
 INQUIRE WEBSERVICE
 RESULT - OVERTYPE TO MODIFY
   Webservice(redbook)
   Pipeline(REDPIPE0)
   Validationst( Novalidation )
   State(Inservice)
   Urimap($355450)
   Program(REDBK3)
   Pgminterface(Commarea)
   Container()
   Datestamp(20060614)
   Timestamp(03:55:45)
   Wsdlfile(/u/REDBK31/CIWS/R3C1/wsbind/provider/redbook.wsdl)
   Wsbind(/u/REDBK31/CIWS/R3C1/wsbind/provider/redbook.wsbind)
   Endpoint()
   Binding(REDBK3HTTPSoapBinding)

                                                          SYSID=RB31 APPLID=REDBKV31
                                                   TIME:  08.52.27  DATE: 06.20.06
PF 1 HELP 2 HEX 3 END       5 VAR        7 SBH 8 SFH       10 SB 11 SF
```

Figure 10-11 CEMT INQUIRE WEBSERVICE

10.6 VBScript Web Service client

The WSDL file generated by the DFHLS2WS utility (identified by the statement WSDL=/u/REDBK31/CIWS/R3C1/wsdl/redbook.wsdl) was manually scanned to create an XML request file that could be sent, as a SOAP request, to our CICS TS 3.1 region. The request file that was created is shown in Figure 10-12.

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header SOAP-ENV:mustUnderstand="no"/>
<SOAP-ENV:Body>
<REDBK3Operation>
<outmsg>
<outmsg1>Message 1</outmsg1>
<outmsg2>Message 2</outmsg2>
<outmsg3>Message 3</outmsg3>
<outmsg4>Message 4</outmsg4>
<outmsg5>Message 5</outmsg5>
<outmsg6>Message 6</outmsg6>
<outmsg7>Message 7</outmsg7>
<outmsg8>Message 8</outmsg8>
<outmsg9>Message 9</outmsg9>
<outmsg10>Message 10</outmsg10>
</outmsg>
</REDBK3Operation>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 10-12 XML Request file

A VBScript client was written to send the request to CICS and receive the response. This code is shown in Example 10-3.

Example 10-3 VBScript client code

```
Option Explicit

' Rudimentary Web Service client
' Sends a SOAP request message to a Web Service
' Saves the response to an XML file,
' and then displays the file in Internet Explorer

' Constants

' Web Service URI, as defined in the WSDL file by
```

```
' <soap:address location="uri"/>
Const conURI = "http://is390:3010/REDBOOK/WEBSERVICE/EXAMPLE"
' User ID and password required?
Const conAuthenticate = True
' Request message file (must be in same folder as this script)
Const conRequestFileName = "request.xml"
' Response message file (saved to same folder as this script)
Const conResponseFileName = "response.xml"
' Miscellaneous scripting constants
Const conForReading = 1, conOverWrite = True, conWaitForResponse = False

' Variables

' File system object required to open files
Dim fso
' Path of the folder containing this script
Dim strScriptFolderPath
' Object used to send request
Dim xmlhttp
' Absolute paths of request and response files
Dim strRequestFilePath, strResponseFilePath
' Contents of the request file
Dim strRequest
' User ID and password, if authentication is required
Dim strUserID, strPassword
' File object for the request
Dim fileResponse

' Functions and subprocedures

Function Base64Encode(strInput)

    ' Converts an ASCII string into a base64-encoded string
    ' according to the Base64 Content-Transfer-Encoding process
    ' described in RFC1521 (http://rfc.net/rfc1521.html)

    Dim i, lngByte1, lngByte2, lngByte3, strOutput
    Dim lng24BitGroup, oct24BitGroup, str8OctalChars
    Dim strBase64Char1, strBase64Char2, strBase64Char3, strBase64Char4

    ' Initialize output string
    strOutput = ""

    ' Convert the input string three bytes at a time
    For i = 1 To Len(strInput) Step 3

        ' Form a 24-bit (3-byte) input group
        ' by concatenating 3 8-bit (1-byte) input groups
        ' To do this, store the concatenated input group as
        ' an integer (of type Long)
        lngByte1 = &H10000 * Asc(Mid(strInput, I, 1))
        lngByte2 = &H100 * AscOrZeroIfNull(Mid(strInput, I + 1, 1))
        lngByte3 = AscOrZeroIfNull(Mid(strInput, I + 2, 1))
        lng24BitGroup = lngByte1 + lngByte2 + lngByte3

        ' Split the 24-bit input group into 4 6-bit groups
        ' To do this, convert the Long number to octal
        ' (each octal digit represents 3 bits;
        ' each pair of octal digits is a 6-bit group)
        oct24BitGroup = Oct(lng24BitGroup)

        ' Convert the octal number into a string of 8 octal digits
        ' (we may need to add a leading zero)
        str8OctalChars = String(8 - Len(CStr(oct24BitGroup)), "0") & _
            CStr(oct24BitGroup)

        ' Translate each 6-bit group (that is, each pair of octal digits)
```

```
        ' into a base64 character
        strBase64Char1 = OctalCharPairToBase64Char(str8OctalChars, 1)
        strBase64Char2 = OctalCharPairToBase64Char(str8OctalChars, 2)
        strBase64Char3 = OctalCharPairToBase64Char(str8OctalChars, 3)
        strBase64Char4 = OctalCharPairToBase64Char(str8OctalChars, 4)

        ' Append the base64 characters to the output
        strOutput = strOutput & _
        strBase64Char1 & strBase64Char2 & strBase64Char3 & strBase64Char4

    Next

    ' Set final padding characters, if necessary
    Select Case Len(strInput) Mod 3
        Case 1:
            strOutput = Left(strOutput, Len(strOutput) - 2) & "=="
        Case 2:
            strOutput = Left(strOutput, Len(strOutput) - 1) & "="
    End Select

    ' Return the output string
    Base64Encode = strOutput

End Function

Function AscOrZeroIfNull(strSingleChar)

    ' Enhanced version of built-in VBScript function Asc
    ' Returns 0 if character is empty string, instead of
    ' causing a run-time error

    If strSingleChar = "" Then
        AscOrZeroIfNull = 0
    Else
        AscOrZeroIfNull = Asc(strSingleChar)
    End If

End Function

Function OctalCharPairToBase64Char(str24BitGroup, intOctalPairIndex)

    ' Translates a pair of octal digit characters
    ' (representing a 6-bit number) into a single base64 character

    Const conBase64Alphabet = _
        "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"

    Dim intOctalPairStartPos, intBase64CharIndex

    ' intOctalPairIndex can be 1, 2, 3, or 4
    ' (first, second, third, or fourth pair of octal characters)
    intOctalPairStartPos = (intOctalPairIndex - 1) * 2 + 1

    ' Use the pair of octal digits (that is, the 6-bit group)
    ' as an index into the base64 alphabet (array of 64 characters)
    intBase64CharIndex = _
        CLng("&o" & Mid(str24BitGroup, intOctalPairStartPos, 2)) + 1

    ' Return the appropriate base64 character
    OctalCharPairToBase64Char = Mid(conBase64Alphabet, intBase64CharIndex, 1)

End Function

Sub ShowIE(strFilePath)

    ' Opens a file in Internet Explorer
```

```
    Dim IE

    ' Initialize Internet Explorer window
    On Error Resume Next
    Set IE = CreateObject("InternetExplorer.Application")
    If Err.Number Then
        MsgBox "Could not create Internet Explorer window. " & _
            "Close any open Internet Explorer windows, and then try again."
        Exit Sub
    End If
    On Error Goto 0
    With IE
        .Navigate strFilePath
        .MenuBar = False
        .ToolBar = False
        .AddressBar = False
        .StatusBar = False
        .Visible = 1
    End With

End Sub

' Main procedure

' Create the object for file access
Set fso = CreateObject("Scripting.FileSystemObject")

' Get the parent folder of this script
strScriptFolderPath = fso.GetParentFolderName(WScript.ScriptFullName)

' Create the object that will send the request and receive the response
Set xmlhttp = CreateObject("Microsoft.XMLHTTP")
' Initialize the request
xmlhttp.open "POST", conURI, conWaitForResponse

' Set the message content type in the request header
xmlhttp.setRequestHeader "Content-Type", "text/xml; charset=""UTF-8"""

' If authentication is required, then prompt for user ID and password
' and add them to the request header, encoded in base64
If conAuthenticate Then
    strUserID = InputBox("Enter your user ID:")
    strPassword = InputBox("Enter your password:")
    xmlhttp.setRequestHeader "Authorization", "basic " & _
        Base64Encode(strUserID & ":" & strPassword)
End If

' Build path of file containing the request message
strRequestFilePath = fso.BuildPath(strScriptFolderPath, conRequestFileName)

' Get request message
strRequest = fso.OpenTextFile(strRequestFilePath, conForReading).ReadAll

' Send the request to the Web Service
On Error Resume Next
xmlhttp.send strRequest
If Err.Number Then
    MsgBox "Could not send request.", vbExclamation
    WScript.Quit
End If
On Error Goto 0

' Build path of file to save the SOAP response message
strResponseFilePath = fso.BuildPath(strScriptFolderPath, conResponseFileName)

' Write response to file
Set fileResponse = fso.CreateTextFile(strResponseFilePath, conOverwrite)
```

```
fileResponse.WriteLine CStr(xmlhttp.responseText)
fileResponse.Close
Set fileResponse = Nothing

' Show response file in Internet Explorer
ShowIE strResponseFilePath
```

10.7 Test the application

When we first tested the application using the VBScript Web Service client we encountered the error shown in Figure 10-13.

```
DFHPI0500 06/14/2006 18:56:49 REDBKV31 CPIH The CICS Pipeline
Manager DFHPIPM encountered an error while trying to link to program
DFHPITP. The program was not defined. PIPELINE: REDPIPE0.
```

Figure 10-13 DFHPI0550 error message

We used CICS CM to confirm that program DFHPITP was not defined in our CICS TS 3.1 region CSD (see Figure 10-14).

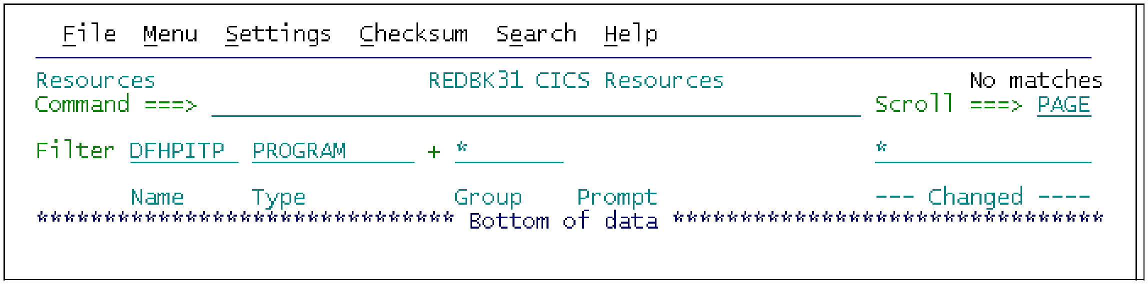


Figure 10-14 No DFHPITP definition

We then used CICS CM to check another CICS TS 3.1 CSD for the existence of program DFHPITP (see Figure 10-15). Having found program DFHPITP in group DFHPIPE we used CICS CM to compare group DFHPIPE in the two CSDs (see Figure 10-16).

```
  File  Menu  Settings  Checksum  Search  Help

Resources                   CICSTS31 CICS Resources                Row 1 to 1 of 1
Command ===>                                                      Scroll ===> PAGE

Filter DFHPITP  PROGRAM     + *                                   *

       Name     Type           Group     Prompt                   --- Changed ----
       DFHPITP  PROGRAM        DFHPIPE                            2006/06/15 14:12
******************************* Bottom of data ********************************
```

Figure 10-15   DFHPITP definition

```
  File  Menu  Settings  Checksum  Help

Compare - All Resources            Group                          Row 13 to 18 of 19
Command ===>                                                       Scroll ===> PAGE

  To prepare other reports, activate Checksum

  Group . . . . : DFHPIPE                         DFHPIPE
  Location  . . : AYS.REDBKV31.DFHCSD             REDBK31.REDBKV31.DFHCSD
  Change Date . : 2006/06/16 11:35                2006/06/15 14:12
  Anomaly flags : Missing, Checksum

                                 All Group Resources

Filter *        *            +
                                                 Flags
/      Name     Type         Group                M C             Group     /
       DFHPISN1 PROGRAM      DFHPIPE                              DFHPIPE
       DFHPISN2 PROGRAM      DFHPIPE                              DFHPIPE
       DFHPITL  PROGRAM      DFHPIPE              M    *Missing
       DFHPITP  PROGRAM               *Missing    M               DFHPIPE
       DFHPITQ1 PROGRAM               *Missing    M               DFHPIPE
       DFHPIVAL PROGRAM               *Missing    M               DFHPIPE
```

Figure 10-16   GROUP compare showing missing definitions

Having identified that the groups were different, we discovered that the CSD INITIALIZE step performed in “DFHCOMDS” on page 201 had used a pre-release version of the CICS TS 3.1 SDFHLOAD data set. To rectify this error we ran a DFHCSDUP UPGRADE job, as shown in Figure 10-17.

```
//DFHCSDUP JOB USER=AYS,NOTIFY=AYS,
//            CLASS=A,MSGCLASS=Y,REGION=0M
//STEPLIB   DD DSN=CICS.V640B40.CICS.SDFHLOAD,DISP=SHR
//DFHCSD    DD DSN=REDBK31.REDBKV31.DFHCSD,DISP=SHR
//SYSUT1    DD UNIT=SYSDA,SPACE=(1024,(100,100))
//SYSPRINT DD SYSOUT=*
//SYSIN     DD *
  UPGRADE
/*
```

Figure 10-17 DFHCSDUP UPGRADE job

Once the CSD was upgraded we used CICS CM to re-install group DFHPIPE and ran our test again. This time the data shown in Figure 10-18 was returned.

Figure 10-18 VBScript Web Service client output

By way of comparison, Figure 10-19 shows the data displayed by transaction RDB3, having first recompiled program REDBK4 with a compatible COBOL compiler.

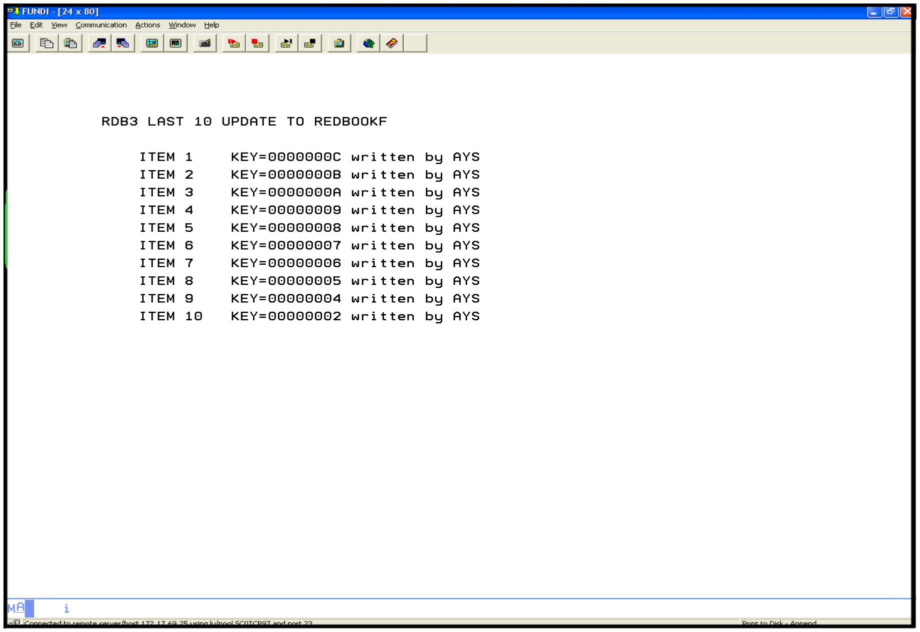


Figure 10-19 Transaction RDB3 3270 output

11

Advanced features of CICS IA and CICS CM

This chapter describes advanced uses of the CICS CM and CICS IA tools that can be used to assist in CICS TS 3.1 migration and CICS TS 3.1 exploitation in more complex scenarios. It demonstrates how the tools can be used to understand your existing environment better and how this knowledge can reduce migration times.

This chapter comprises the following sections:

- Post migration cleanup using CICS CM

  Identifying duplicate resource definition versions
- The power of comparing using CICS CM

  Comparing the CICS TS 2.3 CSD and the migrated CICS TS 3.1 CSD
- The power of searching using CICS CM
  - Searching for all programs defined as threadsafe, openapi, and CICS key
  - Searching for TCPIPSERVICE port numbers used in the TS 3.1 configuration
- How IA can assist during migration test and deployment
  - What resources have been tested
  - Which transactions/programs we have run

- Which CSD data sets and lists are used during IA collection
- Reducing performance impact during IA collection
- Making a program threadsafe and how IA can help
  - Identifying non threadsafe programs
  - Using IA to display TCB modes
  - Showing the dangers of defining programs as threadsafe in CICS
  - Making a program threadsafe using the ENQ/DEQ technique

11.1 Post-migration cleanup using CICS CM

Our scenario has assumed that the names of the resources from the source CSD are unique. A CICSPlex will more commonly have resources that have names that will already exist on the target repository. CICS CM can assist in recognizing this situation, and the following figures highlight this function. We look for duplication of program CDBC0010 in group APPL01 and group REDBOOK.

From the CICS CM Primary Menu, select option **4. Reports** and then option **1. Multiple Configs** to list the configurations. Select configuration CCVPLEXE using the **g** line command, as shown in Figure 11-1.

```
  File  Settings  Help
 ------------------------------------------------------------------------------
 Reports                      CICS Configurations              Row 1 to 20 of 22
 Command ===> ________________________________________________ Scroll ===> CSR

 Select one or more CICS configurations and press Enter

 Filter *_______

     / Name      Description
       ADRIAN
     _ CCMPLEXD  Redbook 3.1 Plex MP CCVT31C
     _ CCVPLEXC
     _ CCVPLEXD  cpsm ts 3.1 plexd
     g CCVPLEXE  Redbook 3.1 Plex MP CCVT32C
     _ CCVT22M   cics ts 2.2
     _ CCVT23M   cics ts 2.3
     _ CCVT31A   ccvt31a - csd
     _ CCVT31M   cics ts 3.1
     _ CCVT31T   cics ts 3.1
     _ CCVT32A   Redbook 2.3 CSD
     _ CCVWDTR   CICS Configuration - CCVWSRP
     _ CCVWSRP   CICS Configuration - CCVWSRP
     _ CCVWTST   CICS Configuration - CCVWTST
     _ KXWCONFA  KXWCONFA (2.2)
     _ KXWCONFB  kxwconfb (2.2)
     _ KXWCONFE  kxwconfe (2.2)
     _ PLEXD     CICS Configuration - CPSM31 - CCVD
     _ REDBK23   REDBOOK v2.3 CSD
     _ REDBK31   REDBOOK v3.1 CSD
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 11-1 List resource groups for configuration CCVPLEXE

The resulting screen shows the names of all the groups in configuration CCVPLEXE. Select the groups to be inspected by typing `s` on the line command, as shown in Figure 11-2. In this case we select all groups.

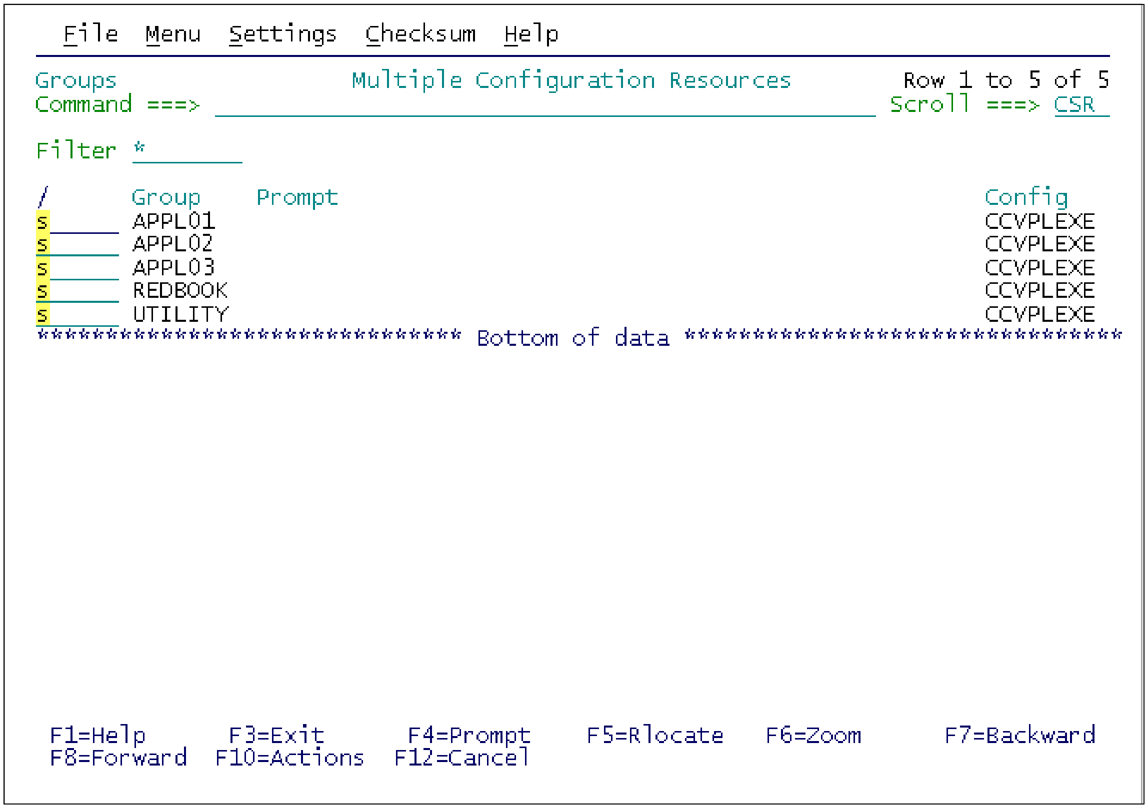


Figure 11-2 Multiple Configuration Resources screen - Select groups to be inspected

Figure 11-3 shows the resulting list of resources for configuration CCVPLEXE.

```
 File  Menu  Settings  Checksum  Search  Help
--------------------------------------------------------------------------------
Group Resources       Multiple Configuration Resources        Row 1 to 22 of 40
Command ===> ________________________________________________  Scroll ===> CSR

Filter *_______  *____________  +
                                                             Checksum
       Name     Type          Group    Prompt                  Full     Config
       CCVR     TRANSACTION   REDBOOK                        6F97D3F4 CCVPLEXE
______ CCVSREMP PROGRAM       REDBOOK                        115D5AAD CCVPLEXE
______ CDCB001# PROGRAM       REDBOOK                        63B96314 CCVPLEXE
______ CDCB0010 PROGRAM       APPL01                         241AF9A0 CCVPLEXE
______ CDCB0010 PROGRAM       REDBOOK                        241AF9A0 CCVPLEXE
______ CDCB0020 PROGRAM       REDBOOK                        587BDC7B CCVPLEXE
______ CDCB0510 PROGRAM       REDBOOK                        179D472B CCVPLEXE
______ CDCB0710 PROGRAM       REDBOOK                        BF1BF6BA CCVPLEXE
______ COBOLVS1 PROGRAM       REDBOOK                        46B0E271 CCVPLEXE
______ COBOLVS2 PROGRAM       REDBOOK                        FB7A8EBF CCVPLEXE
______ CONS     TERMINAL      UTILITY                        52AE6938 CCVPLEXE
______ CSCB0010 PROGRAM       REDBOOK                        9F4284A7 CCVPLEXE
______ CSCB0030 PROGRAM       REDBOOK                        7E2C400A CCVPLEXE
______ CSCB0200 PROGRAM       REDBOOK                        AACBD440 CCVPLEXE
______ DB7P     DB2CONN       REDBOOK                        C40354CE CCVPLEXE
______ LSRPOOL1 LSRPOOL       UTILITY                        DCA64601 CCVPLEXE
______ RDBA     TRANSACTION   REDBOOK                        5DEEB7F5 CCVPLEXE
______ RDBB     TRANSACTION   REDBOOK                        4C93DD8C CCVPLEXE
______ RDBC     TRANSACTION   REDBOOK                        F5680664 CCVPLEXE
______ RDBD     TRANSACTION   REDBOOK                        6E69097E CCVPLEXE
______ RDBE     TRANSACTION   REDBOOK                        D792D296 CCVPLEXE
______ RDBI     TRANSACTION   REDBOOK                        956ED3AE CCVPLEXE
 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate    F6=Zoom      F7=Backward
 F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 11-3 List of resources for configuration CCVPLEXE

We now look for duplicates of name and type that exist in more than one group with a matching *checksum full*. Program CDCB0010 is duplicated in groups APPL01 and REDBOOK. We now expand the resource CDCB0010 to show all the Resgroups to which it belongs. To do this we enter an x on the line command next to program CDBC0010 in group APPL01, as shown in Figure 11-4.

```
  File  Menu  Settings  Checksum  Search  Help
------------------------------------------------------------------------------
Group Resources        Multiple Configuration Resources        Row 1 to 22 of 40
Command ===> ________________________________________________ Scroll ===> CSR

Filter *_______  *____________  +
                                                          Checksum
       Name      Type           Group    Prompt             Full   Config
       CCVR      TRANSACTION    REDBOOK                   6F97D3F4 CCVPLEXE
______ CCVSREMP  PROGRAM        REDBOOK                   115D5AAD CCVPLEXE
______ CDCB001#  PROGRAM        REDBOOK                   63B96314 CCVPLEXE
x_____ CDCB0010  PROGRAM        APPL01                    241AF9A0 CCVPLEXE
______ CDCB0010  PROGRAM        REDBOOK                   241AF9A0 CCVPLEXE
______ CDCB0020  PROGRAM        REDBOOK                   587BDC7B CCVPLEXE
______ CDCB0510  PROGRAM        REDBOOK                   179D472B CCVPLEXE
______ CDCB0710  PROGRAM        REDBOOK                   BF1BF6BA CCVPLEXE
______ COBOLVS1  PROGRAM        REDBOOK                   46B0E271 CCVPLEXE
______ COBOLVS2  PROGRAM        REDBOOK                   FB7A8EBF CCVPLEXE
______ CONS      TERMINAL       UTILITY                   52AE6938 CCVPLEXE
______ CSCB0010  PROGRAM        REDBOOK                   9F4284A7 CCVPLEXE
______ CSCB0030  PROGRAM        REDBOOK                   7E2C400A CCVPLEXE
______ CSCB0200  PROGRAM        REDBOOK                   AACBD440 CCVPLEXE
______ DB7P      DB2CONN        REDBOOK                   C40354CE CCVPLEXE
______ LSRPOOL1  LSRPOOL        UTILITY                   DCA64601 CCVPLEXE
______ RDBA      TRANSACTION    REDBOOK                   5DEEB7F5 CCVPLEXE
______ RDBB      TRANSACTION    REDBOOK                   4C93DD8C CCVPLEXE
______ RDBC      TRANSACTION    REDBOOK                   F5680664 CCVPLEXE
______ RDBD      TRANSACTION    REDBOOK                   6E69097E CCVPLEXE
______ RDBE      TRANSACTION    REDBOOK                   D792D296 CCVPLEXE
______ RDBI      TRANSACTION    REDBOOK                   956ED3AE CCVPLEXE
 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 11-4   Expand resource CSCB0010 to show all ResGroups to which it belongs

Figure 11-5 shows that program CDCB0010 belongs to both the APPL01 resource group and the REDBOOK resource group. It also shows us that there are two different versions of the resource definition in the repository.

```
  File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 Related ResGroups                CICS Resources                  Row 1 to 2 of 2
 Command ===> ________________________________________________ Scroll ===> CSR

 /      Name     Type            Group    VV Prompt            --- Changed ----
        CDCB0010 PROGRAM         APPL01   2                    2006/06/16 12:42
 ______ CDCB0010 PROGRAM         REDBOOK  3                    2006/06/16 12:42
 ******************************* Bottom of data ********************************











  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 11-5 List of all resource groups to which resource definition CDCB0010 belongs

Figure 11-4 on page 352 shows us that these two definitions are duplicates, so one of them can be deleted. We will delete the version 3 definition. To do this enter a `d` in the line command next to version 3 resource, as shown in Figure 11-6. Press Enter, and enter again to confirm the delete.

```
   File  Menu  Settings  Help
 ------------------------------------------------------------------------------
 Related ResGroups                CICS Resources                 Row 1 to 2 of 2
 Command ===> ________________________________________________  Scroll ===> CSR

 /      Name     Type             Group    VV Prompt             --- Changed ----
        CDCB0010 PROGRAM          APPL01   2                     2006/06/16 12:42
 d_____ CDCB0010 PROGRAM          REDBOOK  3                     2006/06/16 12:42
 ******************************* Bottom of data ********************************




  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 11-6 Deleting a version of program resource CDCB0010

We now need to associate version 2 of the program resource definition for CDBC0010 with group REDBOOK. To do this enter `a` (for add association) in the line command, as shown in Figure 11-7.

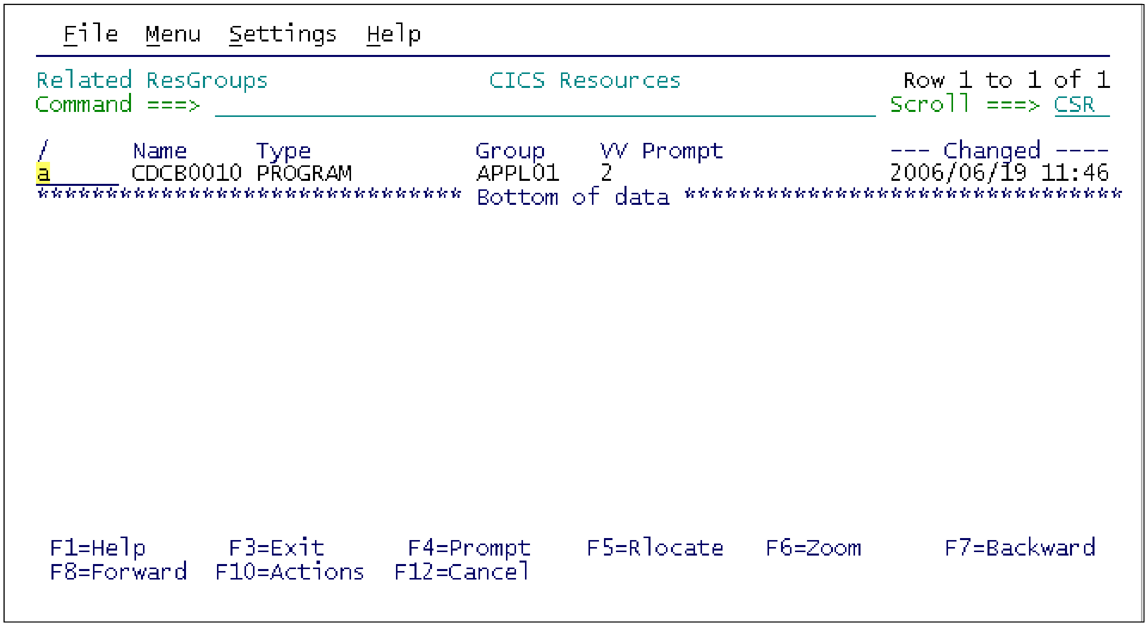


Figure 11-7 Adding a resource association for program CDCB0010

Press Enter, and enter the resource group to which the resource is to be added, as shown in Figure 11-8. In this case the resource group is REDBOOK.

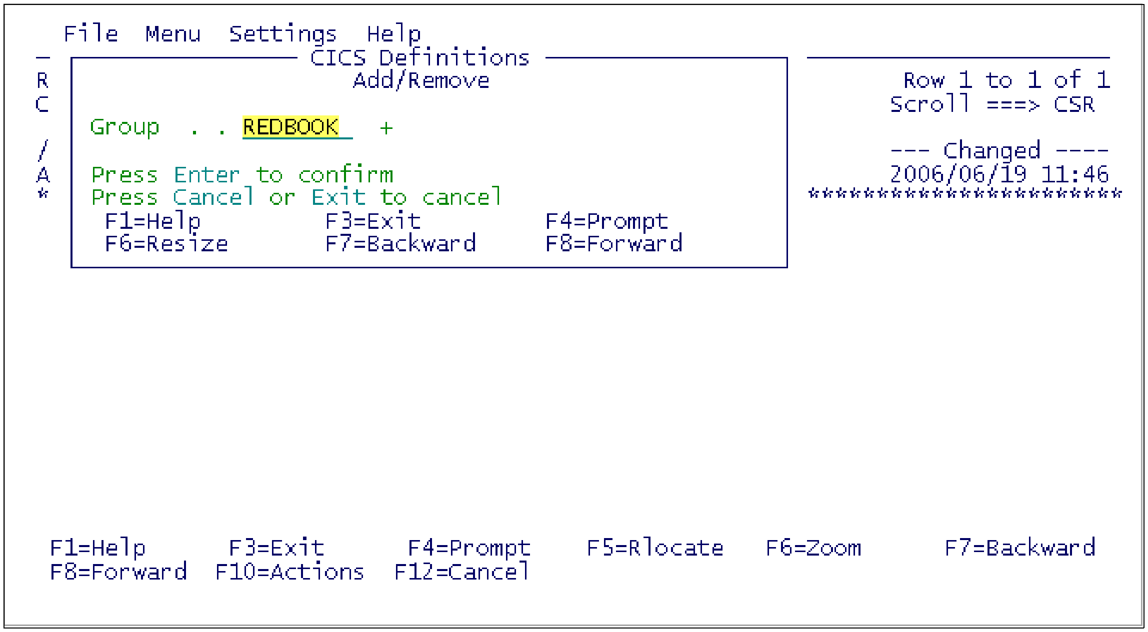


Figure 11-8 Adding CICS definitions to group REDBOOK

Press Enter. Figure 11-9 now shows that resource definitions in both groups are at the same version.

```
  File  Menu  Settings  Help
------------------------------------------------------------------------------
Related ResGroups                CICS Resources                  Row 1 to 2 of 2
Command ===> _______________________________________________  Scroll ===> CSR

/      Name     Type            Group    VV Prompt             --- Changed ----
       CDCB0010 PROGRAM         APPL01   2                     2006/06/16 12:42
______ CDCB0010 PROGRAM         REDBOOK  2                     2006/06/16 12:42
******************************* Bottom of data ********************************










 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=Actions  F12=Cancel
```

Figure 11-9 List of all resource groups to which resource definition CDCB0010 belongs

11.2 The power of comparing using CICS CM

You can use CICS CM to compare configurations, CSD lists, groups, and resources. In the following example we show how you can drill down from CSD lists to CSD groups and to a specific resource. If you are using CICS IA then you can see what CSD and lists were used during a particular collection (see Example 11-7 on page 386).

For more information about CICS CM and the compare feature see 3.8, “Comparing objects in a CICS configuration” on page 57.

Compare CSD lists

From the CICS CM Primary Menu, select option **4. Reports** and then option **1. Multiple Configs** to list the configurations. Select the configurations you wish to compare using the **x** line command, as shown in Figure 11-1 on page 349. In our case we are comparing two CSD configurations, REDBKV23 and REDBKV31.

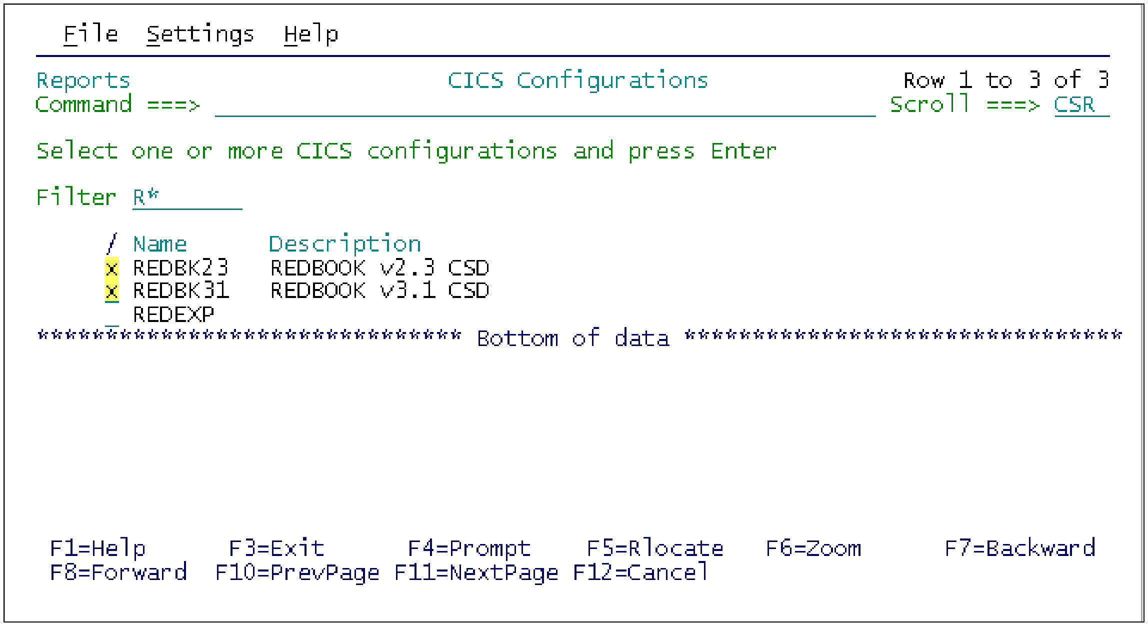


Figure 11-10 CICS CM - Select configurations for comparison

Using x expands the configurations selected by hierarchical lists, as shown in Figure 11-11.

```
  File  Menu  Settings  Checksum  Help
 ------------------------------------------------------------------------------
 Lists                   Multiple Configuration Resources           Row 1 to 7 of 7
 Command ===> ________________________________________________ Scroll ===> CSR

 Filter *_______
                                                             Checksum
 /      List     Prompt                                       Group    Config
        CCVT32A                                              F01C3C4B REDBK23
 ______ DFH$IVPL                                             DA116E1E REDBK31
 ______ DFH$IVPL                                             5F4750E9 REDBK23
 ______ DFHLIST                                              1BDEB8E8 REDBK31
 ______ DFHLIST                                              D1CFAF22 REDBK23
 ______ REDLIST                                              3FA26816 REDBK31
 ______ REDLIST                                              ECD12A61 REDBK23
 ******************************* Bottom of data ********************************




  F1=Help      F3=Exit      F4=Prompt     F5=Rlocate    F6=Zoom       F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 11-11 CICS CM - Multiple Configuration Resources

We can see that the checksum groups for list REDLIST are different in the selected configurations, indicating that there are differences in the CSD lists.

Enter `cm` against both REDLIST lists, as shown in Figure 11-12.

```
  File  Menu  Settings  Checksum  Help
 ------------------------------------------------------------------------------
 Lists                  Multiple Configuration Resources           Row 1 to 7 of 7
 Command ===> ________________________________________________   Scroll ===> CSR

 Filter *_______

 /      List     Prompt                                                   Config
        CCVT32A                                                           REDBK23
 ______ DFH$IVPL                                                          REDBK31
 ______ DFH$IVPL                                                          REDBK23
 ______ DFHLIST                                                           REDBK31
 ______ DFHLIST                                                           REDBK23
 cm____ REDLIST                                                           REDBK31
 cm____ REDLIST                                                           REDBK23
 ******************************* Bottom of data ********************************




  F1=Help      F3=Exit      F4=Prompt     F5=Rlocate    F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 11-12 CICS CM - Select CSDs to compare

Press Enter and the List screen is displayed, as shown in Figure 11-13. This screen shows us two main things:

- The groups are displayed in processing order.
  - The first three groups are in the same order.
  - REDBOOK is the sixth group in the TS 3.1 CSD, while it is the fourth in the TS2.3 CSD.
- Missing groups

  Group LINKS is in the TS 3.1 CSD but not in the TS 2.3 CSD.

```
  File  Menu  Settings  Checksum  Help
 --------------------------------------------------------------------------------
 Compare - Processing order           List                      Row 1 to 10 of 10
 Command ===> ______________________________________________     Scroll ===> CSR

   To prepare other reports, activate Checksum

   List  . . . . : REDLIST                       REDLIST
   Location  . . : REDBK31.REDBKV31.DFHCSD       REDBK23.REDBKV23.DFHCSD
   Change Date . : 2006/06/27 13:21              2006/06/27 13:18

                     All Groups aligned in processing order
               --------------------------------------------------------
         /         Groups   Notes                    Notes     Groups        /
               1   APPL01                                      APPL01    1  ___
        ___    2   APPL02                                      APPL02    2  ___
        ___    3   APPL03                                      APPL03    3  ___
        ___    4   CIUENG21                          *See 5                 ___
        ___    5   CIU64G21                          *Missing               ___
        ___    6   REDBOOK                                     REDBOOK   4  ___
        ___                 *See 4                             CIUENG21  5  ___
        ___    7   EYU310G1                          *Missing               ___
        ___                 *Missing                           CIU63G21  6  ___
        ___    8   LINKS                             *Missing               ___
 ******************************* Bottom of data ********************************



  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 11-13 CSD comparison of resource list

Compare groups within CSD lists

We now compare groups within the CSD lists. We do this by turning on the checksum feature for the name and type fields. Select the **checksum** action bar and select option **4**, as shown in Figure 11-14.

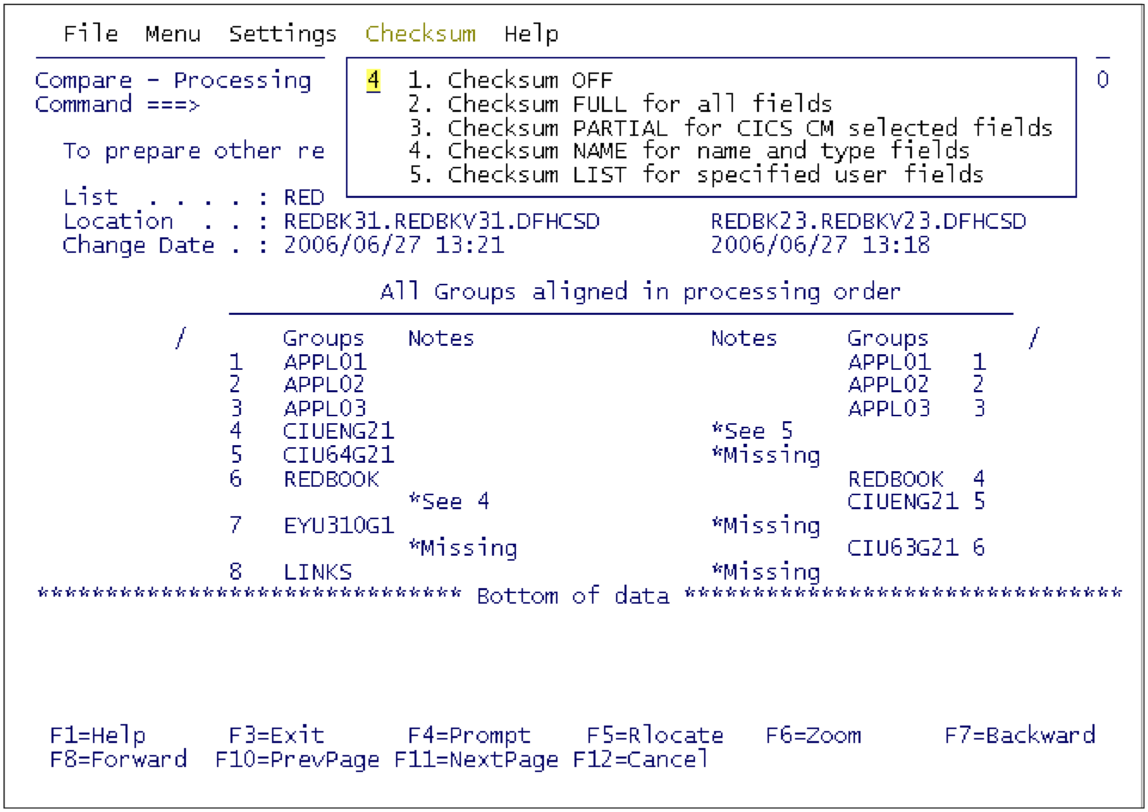


Figure 11-14 Switch on checksum for name and type fields

Press Enter and then PF11 to scroll right until we get to the All details report, as shown in Figure 11-15. We can see that the groups REDBOOK are different. To further compare the group we can enter `cm` against both, as shown.

```
  File  Menu  Settings  Checksum  Help
 ------------------------------------------------------------------------------
 Compare - All details                List                     Row 1 to 10 of 10
 Command ===> ________________________________________________ Scroll ===> CSR

   Scroll right (NextPage) to view other reports

   List  . . . . . : REDLIST                        REDLIST
   Location  . . . : REDBK31.REDBKV31.DFHCSD         REDBK23.REDBKV23.DFHCSD
   Change Date . . : 2006/06/27 13:21                2006/06/27 13:18
   Anomaly flags : Name, Order, Checksum

                         All Groups in processing order with Checksums
                     -----------------------------------------------------
                         REDLIST   Checksum   Flags  Checksum  REDLIST
                /        Groups      Name     N O C    Name    Groups        /
                     1   APPL01    E8FDFC1A           E8FDFC1A APPL01    1   ___
               ___   2   APPL02    FA4853F4           FA4853F4 APPL02    2   ___
               ___   3   APPL03    42F43491           42F43491 APPL03    3   ___
               ___   4   CIUENG21  B8A05257    O      B8A05257 *See 5        ___
               ___   5   CIU64G21  6C300F1B   N O     *Missing *Missing      ___
               cm    6   REDBOOK   3F001AAE    O C    8AB03A93 REDBOOK   4   cm
               ___       *See 4    B8A05257    O      B8A05257 CIUENG21  5   ___
               ___   7   EYU310G1  BFFC246F   N O     *Missing *Missing      ___
               ___       *Missing  *Missing   N O     65E1B95F CIU63G21  6   ___
               ___   8   LINKS     24995D1D   N O     *Missing *Missing      ___
 ******************************** Bottom of data ********************************


  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 11-15 All groups in list with checksums

Figure 11-17 on page 365 shows us all resources in group REDBOOK for both the TS 2.3 CSD and the TS 3.1 CSD. It shows us that some resources are missing from the TS 3.1 CSD. For example, program COBOLVS2 is not in the TS 3.1 CSD, as it was identified by CICS IA as a COBOL/VS program that was no longer required and it was therefore not migrated.

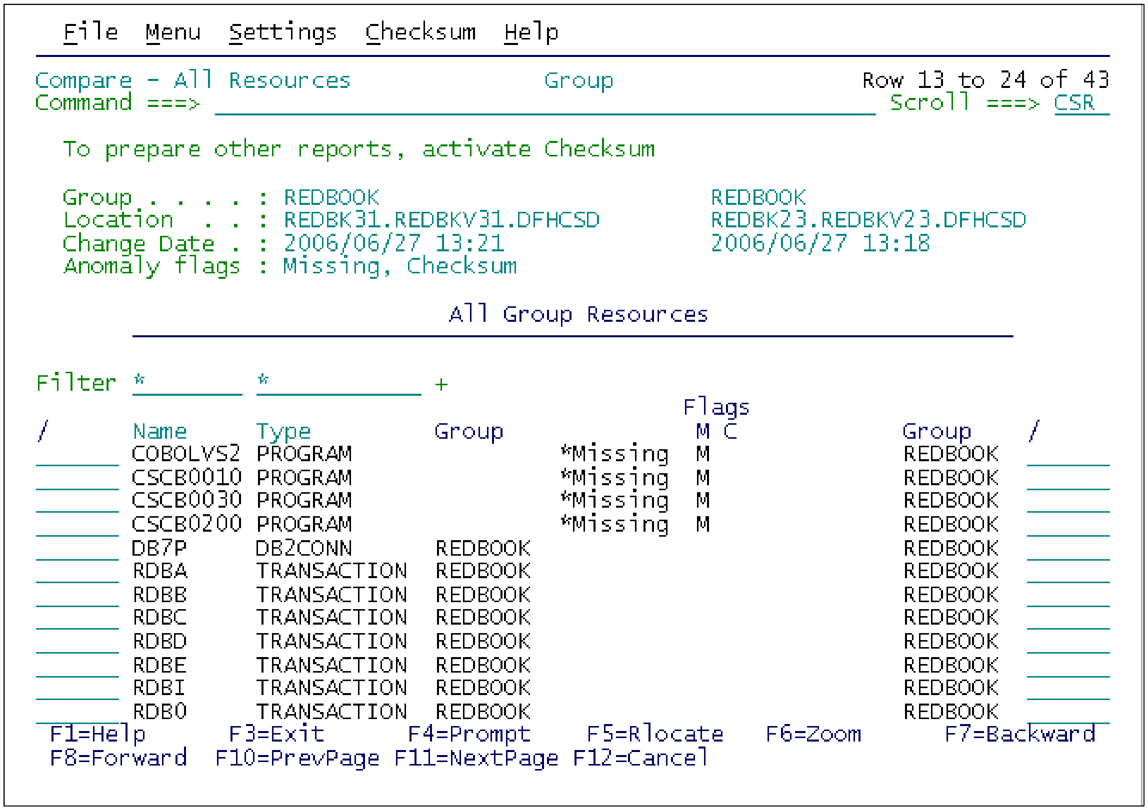


Figure 11-16 All resources in group REDBOOK for both CSDs

We can reduce the amount of data listed by the use of filters. In Figure 11-17 we have filtered the list to all program resources starting with the letter R.

```
  File  Menu  Settings  Checksum  Help
 ------------------------------------------------------------------------------
 Compare - All Resources             Group                      Row 1 to 10 of 10
 Command ===> ________________________________________________  Scroll ===> CSR

   To prepare other reports, activate Checksum

   Group . . . . : REDBOOK                        REDBOOK
   Location  . . : REDBK31.REDBKV31.DFHCSD        REDBK23.REDBKV23.DFHCSD
   Change Date . : 2006/06/27 13:21               2006/06/27 13:18
   Anomaly flags : Missing, Checksum

                           All Group Resources
        ----------------------------------------------------------------

 Filter r*______ PROGRAM_____ +
                                                 Flags
 /      Name     Type         Group               M C            Group    /
 ______ REDBK1   PROGRAM      REDBOOK                            REDBOOK  ______
 ______ REDBK1A  PROGRAM      REDBOOK                            REDBOOK  ______
 ______ REDBK1B  PROGRAM      REDBOOK                            REDBOOK  ______
 ______ REDBK1C  PROGRAM      REDBOOK                            REDBOOK  ______
 ______ REDBK1D  PROGRAM      REDBOOK                            REDBOOK  ______
 ______ REDBK1E  PROGRAM      REDBOOK                            REDBOOK  ______
 ______ REDBK2   PROGRAM      REDBOOK                            REDBOOK  ______
 ______ REDBK3   PROGRAM      REDBOOK                            REDBOOK  ______
 ______ REDBK4   PROGRAM      REDBOOK                            REDBOOK  ______
 ______ REDBK5   PROGRAM      REDBOOK                            REDBOOK  ______
 ******************************* Bottom of data ********************************

  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 11-17 List of all programs in both the TS 2.3 CSD and the TS 3.1 CSD

Compare resources within CSD lists

We can see that program REDBK1 is defined in group REDBOOK in both CSDs, but we cannot see if they are the same. To do this we need to activate the Checksum Full feature, as shown in Figure 11-18.

```
  File  Menu  Settings  Checksum  Help
 --------------------
Compare - All Resourc  | 2  1. Checksum OFF                               |  0
Command ===>           |    2. Checksum FULL for all fields               |
                       |    3. Checksum PARTIAL for CICS CM selected fields|
  To prepare other re  |    4. Checksum LIST for specified user fields    |

  Group . . . . : REDBOOK                         REDBOOK
  Location  . . : REDBK31.REDBKV31.DFHCSD         REDBK23.REDBKV23.DFHCSD
  Change Date . : 2006/06/27 13:21                2006/06/27 13:18
  Anomaly flags : Missing, Checksum

                          All Group Resources
        ----------------------------------------------------------------

Filter r*       PROGRAM      +
                                             Flags
/      Name     Type         Group           M C              Group    /
       REDBK1   PROGRAM      REDBOOK                          REDBOOK
       REDBK1A  PROGRAM      REDBOOK                          REDBOOK
       REDBK1B  PROGRAM      REDBOOK                          REDBOOK
       REDBK1C  PROGRAM      REDBOOK                          REDBOOK
       REDBK1D  PROGRAM      REDBOOK                          REDBOOK
       REDBK1E  PROGRAM      REDBOOK                          REDBOOK
       REDBK2   PROGRAM      REDBOOK                          REDBOOK
       REDBK3   PROGRAM      REDBOOK                          REDBOOK
       REDBK4   PROGRAM      REDBOOK                          REDBOOK
       REDBK5   PROGRAM      REDBOOK                          REDBOOK
******************************* Bottom of data ********************************

 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 11-18 Set checksum LIST on for all programs

Figure 11-19 shows us that the checksums for program REDBK1 are different. We can compare the differences by entering `cm` against both entries, as shown.

```
  File  Menu  Settings  Checksum  Help
 ------------------------------------------------------------------------------
 Compare - Checksum Differences       Group                     Row 1 to 8 of 8
 Command ===> ________________________________________________ Scroll ===> CSR

   Scroll right (NextPage) to view other reports

   Group . . . . . : REDBOOK                       REDBOOK
   Location  . . . : REDBK31.REDBKV31.DFHCSD       REDBK23.REDBKV23.DFHCSD
   Change Date . . : 2006/06/27 13:21              2006/06/27 13:18
   Anomaly flags : Missing, Checksum

                     Same resource names, different Checksums
        ----------------------------------------------------------------

 Filter r*______  PROGRAM______  +
                                          Checksum Flags Checksum
 /      Name      Type         Group        Full    M C    Full     Group      /
 cm____ REDBK1    PROGRAM      REDBOOK    8E747019    C  28EECF88  REDBOOK   cm____
 ______ REDBK1A   PROGRAM      REDBOOK    EF7D34B6    C  768649AE  REDBOOK   ______
 ______ REDBK1B   PROGRAM      REDBOOK    ABDC11AE    C  4F0B756B  REDBOOK   ______
 ______ REDBK1C   PROGRAM      REDBOOK    97BCF2A6    C  58706128  REDBOOK   ______
 ______ REDBK1D   PROGRAM      REDBOOK    229E5B9E    C  3C110CE1  REDBOOK   ______
 ______ REDBK1E   PROGRAM      REDBOOK    1EFEB896    C  2B6A18A2  REDBOOK   ______
 ______ REDBK2    PROGRAM      REDBOOK    01267B7A    C  EC9FE6A1  REDBOOK   ______
 ______ REDBK4    PROGRAM      REDBOOK    E18196E6    C  C89C1223  REDBOOK   ______
 ******************************* Bottom of data ********************************


  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 11-19 Compare program definition REDBK1

Figure 11-20 shows us the differences between the resource definition for program REDBK1 in group REDBOOK in the TS 2.3 CSD and the TS 3.1 CSD.

```
   File  Menu  Settings  Hilite  Help
 ------------------------------------------------------------------------------
 Compare                             Program
 Command ===> ________________________________________________________________

     Program . . . : REDBK1                          REDBK1
     ResGroup  . . : REDBOOK                         REDBOOK
 ==> Location  . . : REDBK31.REDBKV31.DFHCSD         REDBK23.REDBKV23.DFHCSD
 ==> Change Date . : 2006/06/23 12:15:41             2006/06/09 08:08:39
     Description . :
                                                                More:     +
     Language  . . : LE370                           LE370
     Reload  . . . : NO                              NO
     Resident  . . : NO                              NO
     Usage . . . . : NORMAL                          NORMAL
     UseLPAcopy  . : NO                              NO
     Status  . . . : ENABLED                         ENABLED
     CEDF  . . . . : YES                             YES
     DataLocation  : BELOW                           BELOW
     ExecKey . . . : CICS                            CICS
 ==> Concurrency . : THREADSAFE                      QUASIRENT
 ==> API . . . . . : OPENAPI                         CICSAPI

    Remote Attributes
     Dynamic . . . : NO                              NO
     RemoteSystem  :
     RemoteName  . :
     TransID . . . :
     ExecutionSet  : FULLAPI                         FULLAPI

  F1=Help      F3=Exit      F4=Prompt     F5=Rlocate    F6=Zoom       F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 11-20   Attribute differences for program definition REDBK1 in the TS 2.3 CSD and the TS 3.1 CSD

11.3 The power of searching using CICS CM

You can use CICS CM to search CICS resource definitions for specific attributes. In this section we will demonstrate 2 searches :

- Search for all programs in the REDBK31 configuration with the following attributes:
  - Running in the OPENAPI environment
  - With a concurrency of threadsafe
  - And running under the CICS key
- All TCPIPPORT numbers used in the REDBK31 configuration.

From the CICS CM Primary Menu, select option **2. CICS Resources** and then select configuration REDBK31 as shown in Figure 11-21.

```
  File  Menu  Settings  Help
------------------------------------------------------------------------------
Resources                     CICS Configurations              Row 1 to 22 of 23
Command ===> ________________________________________________ Scroll ===> CSR

Filter *______

   /    Name       Description
        ADRIAN
  ___   CCMPLEXD   Redbook 3.1 Plex MP CCVT31C
  ___   CCVPLEXC
  ___   CCVPLEXD   cpsm ts 3.1 plexd
  ___   CCVPLEXE   Redbook 3.1 Plex MP CCVT32C
  ___   CCVT22M    cics ts 2.2
  ___   CCVT23M    cics ts 2.3
  ___   CCVT31A    ccvt31a - csd
  ___   CCVT31M    cics ts 3.1
  ___   CCVT31T    cics ts 3.1
  ___   CCVT32A    Redbook 2.3 CSD
  ___   CCVWDTR    CICS Configuration - CCVWSRP
  ___   CCVWSRP    CICS Configuration - CCVWSRP
  ___   CCVWTST    CICS Configuration - CCVWTST
  ___   KXWCONFA   KXWCONFA (2.2)
  ___   KXWCONFB   kxwconfb (2.2)
  ___   KXWCONFE   kxwconfe (2.2)
  ___   N-IMPORT   cics ts 3.1 - import file
  ___   PLEXD      CICS Configuration - CPSM31 - CCVD
  ___   REDBK23    REDBOOK v2.3 CSD
  s__   REDBK31    REDBOOK v3.1 CSD
  ___   REDEXP
 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 11-21 Select REDBK31 configuration

This lists all resources in the REDBK31 configuration. We can only search on specific types, so we need to filter this list to a specific type. We are going to search on programs. We filter the list as shown in Figure 11-22.

```
  File  Menu  Settings  Checksum  Search  Help
 ------------------------------------------------------------------------------
 Resources                     REDBK31 CICS Resources       Row 1 to 22 of 4,108
 Command ===> ________________________________________________ Scroll ===> CSR

 Filter *_______ PROGRAM____ + *_______                     *____________

        Name     Type        Group    Prompt                --- Changed ----
 ______ ACCT00   PROGRAM     DFH$ACCT                       2006/06/15 14:12
 ______ ACCT01   PROGRAM     DFH$ACCT                       2006/06/15 14:12
 ______ ACCT02   PROGRAM     DFH$ACCT                       2006/06/15 14:12
 ______ ACCT03   PROGRAM     DFH$ACCT                       2006/06/15 14:12
 ______ ACCT04   PROGRAM     DFH$ACCT                       2006/06/15 14:12
 ______ CAUCAFBE PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUCAFB1 PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUCAFB2 PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUCAFFE PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUCAFF1 PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUCAFF2 PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUCAFF3 PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUCAFF4 PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUCAFF5 PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUCAFF6 PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUCAFF7 PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUCAFP  PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUMSGCS PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUTABM  PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUXDUMM PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUXITB1 PROGRAM     DFHCOMPB                       2006/06/15 14:12
 ______ CAUXITF1 PROGRAM     DFHCOMPB                       2006/06/15 14:12
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 11-22 Filter the list to show all program resources

We can now activate the search by switching the feature on, as shown in Figure 11-23.

```
  File  Menu  Settings  Checksum  Search  Help
 ------------------------------  ----------------------------------  ---------
 Resources                  RED | 2  1. Search OFF                  |  of 4,108
 Command ===>                   |    2. Search ON and set criteria  | ===> CSR
                                 ----------------------------------
 Filter *         PROGRAM      + *                            *

        Name      Type          Group     Prompt              --- Changed ----
        ACCT00    PROGRAM       DFH$ACCT                      2006/06/15 14:12
        ACCT01    PROGRAM       DFH$ACCT                      2006/06/15 14:12
        ACCT02    PROGRAM       DFH$ACCT                      2006/06/15 14:12
        ACCT03    PROGRAM       DFH$ACCT                      2006/06/15 14:12
        ACCT04    PROGRAM       DFH$ACCT                      2006/06/15 14:12
        CAUCAFBE  PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUCAFB1  PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUCAFB2  PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUCAFFE  PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUCAFF1  PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUCAFF2  PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUCAFF3  PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUCAFF4  PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUCAFF5  PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUCAFF6  PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUCAFF7  PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUCAFP   PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUMSGCS  PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUTABM   PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUXDUMM  PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUXITB1  PROGRAM       DFHCOMPB                      2006/06/15 14:12
        CAUXITF1  PROGRAM       DFHCOMPB                      2006/06/15 14:12
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 11-23 Activating a search

The Program Search Criteria panel is displayed, as shown in Figure 11-24.

```
  File  Menu  Settings  Hilite  Help
                            Program Search Criteria
Command ===> 

Enter Search options. Press End to continue, or Cancel to cancel the request.

Choose the search method
   1. New search             (Search all CICS resources)
_  2. Search within results  (Search the 4108 CICS resources already listed)

Enter "/" to select one or more sets of Search Criteria (Boolean OR)

_  Search Criteria Set 1   Attribute +     Op +        Value +
     Condition 1 . . . .   _______________  __  _________________________
           AND 2 . . . .   _______________  __  _________________________
           AND 3 . . . .   _______________  __  _________________________
           AND 4 . . . .   _______________  __  _________________________

_  Search Criteria Set 2   Attribute +     Op +        Value +
     Condition 1 . . . .   _______________  __  _________________________
           AND 2 . . . .   _______________  __  _________________________
           AND 3 . . . .   _______________  __  _________________________
           AND 4 . . . .   _______________  __  _________________________

_  Search Criteria Set 3   Attribute +     Op +        Value +
     Condition 1 . . . .   _______________  __  _________________________
           AND 2 . . . .   _______________  __  _________________________
           AND 3 . . . .   _______________  __  _________________________
           AND 4 . . . .   _______________  __  _________________________
 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=Actions  F12=Cancel
```

Figure 11-24 Program Search Criteria

We need to search for all programs defined as threadsafe running in the openapi and CICS key. In order to define our search we found that using the F4 Prompt key was useful in assisting us to remember the name of the attributes relating to a program resource and their values.

To prompt for program attributes, tab to the attribute input field and press F4. A list of program attributes is displayed, as shown in Figure 11-25. We are interested in the API attribute, so select it by entering an s against it.

```
   File  Menu  Settings  Hilite  Help
 -  ------------------------ CICS Definitions -----------------   -------
   |                         Attribute Values       Row 1 to 12 of 28 |
 C | Command ===> ______________________________________________      |
   |                                                                  |
 E | Select a value for Attribute                                     |  quest.
   |                                                                  |
 C |     Value                                                        |
 1 | .   API                                                          |
   | .   CEDF                                                         |  sted)
   | .   CHANGETIME                                                   |
 E | .   CONCURRENCY                                                  |
   | .   DATALOCATION                                                 |
 / | .   DESCRIPTION                                                  |
   | .   DYNAMIC                                                      |
   | .   EXECKEY                                                      |
   | .   EXECUTIONSET                                                 |
   | .   GROUP                                                        |
   | .   HOTPOOL                                                      |
   | .   JVM                                                          |
   |  F1=Help      F3=Exit      F6=Resize    F7=Backward  F8=Forward  |
   | F12=Cancel                                                       |
    ------------------------------------------------------------------
           AND 4 . . . .

    Search Criteria Set 3   Attribute +    Op +       Value +
      Condition 1 . . . .
           AND 2 . . . .
           AND 3 . . . .
           AND 4 . . . .
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward   F10=Actions  F12=Cancel
```

Figure 11-25 Prompt for program attributes

You can also prompt for attribute values once you have selected an attribute. Tab to the value input field and press F4. Figure 11-26 shows us the values for the API attribute.

```
   File  Menu  Settings  Hilite  Help
 -  ------------------------ CICS Definitions ---------------  -------
  |                              API Values          Row 1 to 2 of 2 |
 C| Command ===> _______________________________________________     |
  |                                                                  |
 E| Select a value for API                                           |  quest.
  |                                                                  |
 C|    Value                                                         |
 1| .  CICSAPI                                                       |
  | .  OPENAPI                                                       |  sted)
  |    **End**                                                       |
 E|                                                                  |
  |                                                                  |
 /|                                                                  |
  |                                                                  |
  |                                                                  |
  |  F1=Help      F3=Exit      F6=Resize    F7=Backward  F8=Forward  |
  | F12=Cancel                                                       |
  --------------------------------------------------------------------
               AND 4 . . . .

       Search Criteria Set 3   Attribute +    Op +       Value +
         Condition 1 . . . .
               AND 2 . . . .
               AND 3 . . . .
               AND 4 . . . .
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 11-26 Acceptable values for the API attribute

The complete search criteria is shown in Figure 11-27.

> **Note:** We have selected option 2 to search within the resources already selected. Also, we have entered a slash (/) against the Search Criteria Set 1. The attributes within a criteria set use the logical AND. If you wish to have a logical OR, then use a second criteria set.

```
  File  Menu  Settings  Hilite  Help
 ------------------------------------------------------------------------------
                            Program Search Criteria
 Command ===> ____________________________________________________________

 Enter Search options. Press End to continue, or Cancel to cancel the request.

 Choose the search method
 2  1. New search             (Search all CICS resources)
    2. Search within results  (Search the 4108 CICS resources already listed)

 Enter "/" to select one or more sets of Search Criteria (Boolean OR)

 /  Search Criteria Set 1   Attribute +    Op +       Value +
      Condition 1 . . . . API____________  EQ  OPENAPI_____________________
            AND 2 . . . . CONCURRENCY____  EQ  THREADSAFE__________________
            AND 3 . . . . EXECKEY________  EQ  CICS________________________
            AND 4 . . . . _______________  __  ____________________________

 _  Search Criteria Set 2   Attribute +    Op +       Value +
      Condition 1 . . . . _______________  __  ____________________________
            AND 2 . . . . _______________  __  ____________________________
            AND 3 . . . . _______________  __  ____________________________
            AND 4 . . . . _______________  __  ____________________________

 _  Search Criteria Set 3   Attribute +    Op +       Value +
      Condition 1 . . . . _______________  __  ____________________________
            AND 2 . . . . _______________  __  ____________________________
            AND 3 . . . . _______________  __  ____________________________
            AND 4 . . . . _______________  __  ____________________________
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 11-27 Search criteria for all programs defined as OPENAPI, THREADSAFE and key CICS

Figure 11-28 shows the result of our search. Note that it displays only the programs that match our criteria and tells us that 4102 programs were excluded.

```
  File  Menu  Settings  Checksum  Search  Help
 ------------------------------------------------------------------------------
Search Resources            REDBK31 CICS Resources                Row 1 to 6 of 6
Command ===> ________________________________________________   Scroll ===> CSR
4102 rows excluded by the Search Criteria.
Filter * _______ PROGRAM____ + * _______                          * _______________

       Name     Type           Group    Prompt                    --- Changed ----
______ DFHPIEP  PROGRAM        DFHPIPE                            2006/06/15 14:12
______ DFHPISN1 PROGRAM        DFHPIPE                            2006/06/15 14:12
______ DFHPISN2 PROGRAM        DFHPIPE                            2006/06/15 14:12
______ DFHPITP  PROGRAM        DFHPIPE                            2006/06/15 14:12
______ DFHPIXE  PROGRAM        DFHPIPE                            2006/06/15 14:12
______ REDBK1   PROGRAM        REDBOOK                            2006/06/23 12:15
******************************* Bottom of data ********************************

 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 11-28 All programs that match the search criteria

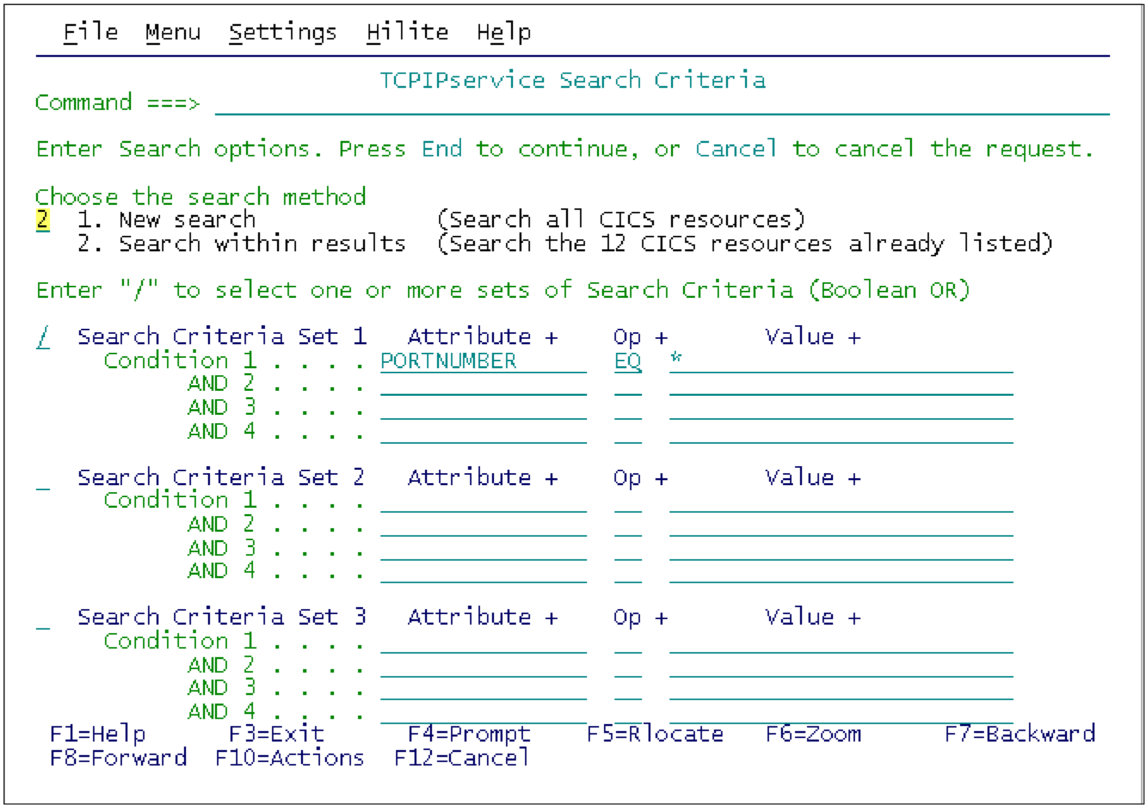
You could now copy one of these resource definitions to create a new definition with similar attributes.

Our second example is a search for TCP/IP port numbers used in the REDBK31 configuration. Change the filter type to TCPIPSERVICE, as shown in Figure 11-29.

```
  File  Menu  Settings  Checksum  Search  Help
 ------------------------------------------------------------------------------
 Resources                   REDBK31 CICS Resources               Row 1 to 12 of 12
 Command ===> ________________________________________________  Scroll ===> CSR

 Filter *_______ TCPIPSERVICE + *_______                           *_______________

        Name     Type            Group     Prompt                  --- Changed ----
 ______ DFHADTCP TCPIPSERVICE    DFHCOMP9                          2006/06/15 14:12
 ______ ECI      TCPIPSERVICE    DFH$SOT                           2006/06/15 14:12
 ______ EJBTCP1  TCPIPSERVICE    DFH$EJB                           2006/06/15 14:12
 ______ EJBTCP1  TCPIPSERVICE    DFH$EJB2                          2006/06/15 14:12
 ______ EXMPPORT TCPIPSERVICE    DFH$EXWS                          2006/06/15 14:12
 ______ HTTPNSSL TCPIPSERVICE    DFH$SOT                           2006/06/15 14:12
 ______ HTTPSSL  TCPIPSERVICE    DFH$SOT                           2006/06/15 14:12
 ______ IIOPNSSL TCPIPSERVICE    DFH$IIOP                          2006/06/15 14:12
 ______ IIOPSSL  TCPIPSERVICE    DFH$IIOP                          2006/06/15 14:12
 ______ REDBKTCP TCPIPSERVICE    ADRIAN                            2006/06/19 11:53
 ______ REDBKTCP TCPIPSERVICE    REDBOOK                           2006/06/19 12:04
 ______ SKIPPORT TCPIPSERVICE    REDBOOK                           2006/06/12 15:48
 ******************************* Bottom of data ********************************

  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 11-29 All TCPIPSERVICE definitions in REDBK31

To show all the port numbers used, activate the search and enter the search criteria, as shown in Figure 11-30.



```
   File  Menu  Settings  Hilite  Help
 ------------------------------------------------------------------------------
                         TCPIPservice Search Criteria
 Command ===> ____________________________________________________________

 Enter Search options. Press End to continue, or Cancel to cancel the request.

 Choose the search method
 2  1. New search              (Search all CICS resources)
    2. Search within results   (Search the 12 CICS resources already listed)

 Enter "/" to select one or more sets of Search Criteria (Boolean OR)

 /  Search Criteria Set 1   Attribute +     Op +       Value +
      Condition 1 . . . . . PORTNUMBER____  EQ  *________________________
            AND 2 . . . . . ______________  __  _________________________
            AND 3 . . . . . ______________  __  _________________________
            AND 4 . . . . . ______________  __  _________________________

 _  Search Criteria Set 2   Attribute +     Op +       Value +
      Condition 1 . . . . . ______________  __  _________________________
            AND 2 . . . . . ______________  __  _________________________
            AND 3 . . . . . ______________  __  _________________________
            AND 4 . . . . . ______________  __  _________________________

 _  Search Criteria Set 3   Attribute +     Op +       Value +
      Condition 1 . . . . . ______________  __  _________________________
            AND 2 . . . . . ______________  __  _________________________
            AND 3 . . . . . ______________  __  _________________________
            AND 4 . . . . . ______________  __  _________________________
  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=Actions  F12=Cancel
```

Figure 11-30 Search for all TCPIPSERVICE PORTNUMBERs

The same list as in Figure 11-29 on page 377 is shown with a message that no rows have been excluded. To show the PORTNUMBERs, press F11 to show the next page.

```
  File  Menu  Settings  Checksum  Search  Help
 ------------------------------------------------------------------------------
 Search Resources            REDBK31 CICS Resources               Row 1 to 12 of 12
 Command ===> ________________________________________________    Scroll ===> CSR

 Filter *_______ TCPIPSERVICE + *_______

        Name     Type            Group    PortNumber
 ______ DFHADTCP TCPIPSERVICE    DFHCOMP9 7673
 ______ ECI      TCPIPSERVICE    DFH$SOT  1435
 ______ EJBTCP1  TCPIPSERVICE    DFH$EJB  687
 ______ EJBTCP1  TCPIPSERVICE    DFH$EJB2 687
 ______ EXMPPORT TCPIPSERVICE    DFH$EXWS 12345
 ______ HTTPNSSL TCPIPSERVICE    DFH$SOT  80
 ______ HTTPSSL  TCPIPSERVICE    DFH$SOT  443
 ______ IIOPNSSL TCPIPSERVICE    DFH$IIOP 683
 ______ IIOPSSL  TCPIPSERVICE    DFH$IIOP 684
 ______ REDBKTCP TCPIPSERVICE    ADRIAN   3010
 ______ REDBKTCP TCPIPSERVICE    REDBOOK  3010
 ______ SKIPPORT TCPIPSERVICE    REDBOOK  10329
******************************* Bottom of data ********************************







  F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
  F8=Forward  F10=PrevPage F11=NextPage F12=Cancel
```

Figure 11-31 REDBK31 CICS Resources

11.4 Using CICS IA to assist during testing and deployment

Once you have used CICS IA to collect resource data for a given application in a given region you can then use that data in the future. There are a number of ways in which you could use this during testing and deployment. For example, if you have collected data in your TS 2.3 region you can then compare the resources used in the migrated TS 3.1 region. If you have automated test processing for your applications that can be run in both the TS 2.3 and the TS 3.1 region then collecting IA data in both should produce the same results. We use SQL queries against the collected data to demonstrate how IA can help during testing and deployment. You can also use the timestamps recorded by IA to ensure that all the resources were used during a specific automated test run. We use SQL queries against the Redbook Application 1 in the REDBKV23 and REDBKV31 regions to demonstrate:

- A comparison of transactions used
- A comparison of all resources and commands used all resources
- The use of the last used date to see which resources in a region were not tested during the last automated test
- Which CSD and resource lists were used during IA collection

Compare transactions used

The SQL query in Example 11-1 shows us all the transactions used by Redbook Application 1. There are nine distinct transactions.

Example 11-1 Show all transactions used in Application Redbook 1 in region REDBKV23

```
---------+---------+---------+---------+---------+---------+---------+---------+
--SHOW ME WHICH TRANSACTIONS HAVE BEEN USED BY REDBOOK APPLICATION 1    00010001
-- IN REGION REDBKV23                                                   00011001
SELECT DISTINCT TRANSID                                                 00012002
  FROM CIU4_CICS_DATA                                                   00014000
  WHERE APPLID='REDBKV23'                                               00015001
    AND PROGRAM LIKE 'RED%';                                            00016001
---------+---------+---------+---------+---------+---------+---------+---------+
TRANSID
---------+---------+---------+---------+---------+---------+---------+---------+
RDBA
RDBB
RDBC
RDBD
RDBE
RDBI
RDBO
RDB1
RDB3
DSNE610I NUMBER OF ROWS DISPLAYED IS 9
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
---------+---------+---------+---------+---------+---------+---------+---------+
```

Once we have migrated the application to the TS 3.1 region we can rerun the test with IA running. Example 11-2 shows us all the transactions used in region REDBKV31. There are only eight distinct transactions. It is clear from these two examples that transaction RDB3 has not been tested in the migrated TS3.1 region.

Example 11-2 Show all transactions used in Application Redbook 1 in region REDBKV31

```
---------+---------+---------+---------+---------+---------+---------+---------+
--SHOW ME WHICH TRANSACTIONS HAVE BEEN USED BY REDBOOK APPLICATION 1
-- IN REGION REDBKV31
SELECT DISTINCT TRANSID
  FROM CIU4_CICS_DATA
  WHERE APPLID='REDBKV31'
    AND PROGRAM LIKE 'RED%';
---------+---------+---------+---------+---------+---------+---------+---------+
TRANSID
---------+---------+---------+---------+---------+---------+---------+---------+
RDBA
RDBB
RDBC
RDBD
RDBE
RDBI
RDBO
RDB1
DSNE610I NUMBER OF ROWS DISPLAYED IS 8
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
---------+---------+---------+---------+---------+---------+---------+---------+
```

In a more complex environment we could use the compare query, as shown in Example 11-3.

Example 11-3 Compare transactions used by region

```
---------+---------+---------+---------+---------+---------+---------+---------+
-- SHOW ME WHAT TCBMODE THAT ALL THE RESOURCES IN REDBKV23
--LECT APPLID, TRANSID, PROGRAM, FUNCTION, TYPE, OBJECT, TCBMODE
SELECT DISTINCT TRANSID
  FROM CIU4_CICS_DATA
  WHERE APPLID='REDBKV23'
    AND TRANSID NOT IN (SELECT DISTINCT TRANSID FROM CIU4_CICS_DATA
                               WHERE APPLID='REDBKV31');
---------+---------+---------+---------+---------+---------+---------+---------+
TRANSID
---------+---------+---------+---------+---------+---------+---------+---------+
RDB3
DSNE610I NUMBER OF ROWS DISPLAYED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
```

We therefore need to test transaction RDB3 before deploying.

Compare resources and commands used

The SQL query in Example 11-4 shows us all resources used by Redbook Application 1. The test was last run on 2006-06-09. There are 46 distinct EXEC CICS calls.

Example 11-4 Show us all dependencies used by Application Redbook 1 in REDBKV23 region

```
---------+---------+---------+---------+---------+---------+---------+---------+
-- SHOW US ALL RESOURCES USED BY APPLICATION REDBOOK 1
-- DURING THE LASR TEST RUN IN REGION REDBKV23
SELECT APPLID, TRANSID, PROGRAM,
  FUNCTION, TYPE, OBJECT, DATE(LAST_RUN) AS LAST_RUN
  FROM CIU4_CICS_DATA
  WHERE APPLID='REDBKV23'
    AND PROGRAM LIKE ‘RED%’;
---------+---------+---------+---------+---------+---------+---------+---------+
APPLID    TRANSID  PROGRAM   FUNCTION  TYPE      OBJECT            LAST_RUN
---------+---------+---------+---------+---------+---------+---------+---------+
REDBKV23  RDBA     REDBK1    ADDRESS   CWA       CWA               2006-06-09
REDBKV23  RDBA     REDBK1    ASSIGN    APPLID    REDBKV23          2006-06-09
REDBKV23  RDBA     REDBK1    WRITE     FILE      REDBOOKF          2006-06-09
REDBKV23  RDBA     REDBK1    WRITEQ    TD        CESE              2006-06-09
REDBKV23  RDBA     REDBK1    WRITEQ    TSSHR     REDBOOKQ          2006-06-09
```

```
REDBKV23  RDBB    REDBK1   ADDRESS   CWA      CWA                 2006-06-09
REDBKV23  RDBB    REDBK1   ASSIGN    APPLID   REDBKV23            2006-06-09
REDBKV23  RDBB    REDBK1   WRITE     FILE     REDBOOKF            2006-06-09
REDBKV23  RDBB    REDBK1   WRITEQ    TD       CESE                2006-06-09
REDBKV23  RDBB    REDBK1   WRITEQ    TSSHR    REDBOOKQ            2006-06-09
REDBKV23  RDBC    REDBK1   ADDRESS   CWA      CWA                 2006-06-09
REDBKV23  RDBC    REDBK1   ASSIGN    APPLID   REDBKV23            2006-06-09
REDBKV23  RDBC    REDBK1   WRITE     FILE     REDBOOKF            2006-06-09
REDBKV23  RDBC    REDBK1   WRITEQ    TD       CESE                2006-06-09
REDBKV23  RDBC    REDBK1   WRITEQ    TSSHR    REDBOOKQ            2006-06-09
REDBKV23  RDBD    REDBK1   ADDRESS   CWA      CWA                 2006-06-09
REDBKV23  RDBD    REDBK1   ASSIGN    APPLID   REDBKV23            2006-06-09
REDBKV23  RDBD    REDBK1   WRITE     FILE     REDBOOKF            2006-06-09
REDBKV23  RDBD    REDBK1   WRITEQ    TD       CESE                2006-06-09
REDBKV23  RDBD    REDBK1   WRITEQ    TSSHR    REDBOOKQ            2006-06-09
REDBKV23  RDBE    REDBK1   ADDRESS   CWA      CWA                 2006-06-09
REDBKV23  RDBE    REDBK1   ASSIGN    APPLID   REDBKV23            2006-06-09
REDBKV23  RDBE    REDBK1   WRITE     FILE     REDBOOKF            2006-06-09
REDBKV23  RDBE    REDBK1   WRITEQ    TD       CESE                2006-06-09
REDBKV23  RDBE    REDBK1   WRITEQ    TSSHR    REDBOOKQ            2006-06-09
REDBKV23  RDBI    REDBK5   ADDRESS   CWA      CWA                 2006-06-09
REDBKV23  RDBI    REDBK5   ASSIGN    APPLID   REDBKV23            2006-06-09
REDBKV23  RDBI    REDBK5   ENDBR     FILE     REDBOOKF            2006-06-09
REDBKV23  RDBI    REDBK5   READNEXT  FILE     REDBOOKF            2006-06-09
REDBKV23  RDBI    REDBK5   STARTBR   FILE     REDBOOKF            2006-06-09
REDBKV23  RDB0    REDBK2   START     TRANSID  RDBA                2006-06-09
REDBKV23  RDB0    REDBK2   START     TRANSID  RDBB                2006-06-09
REDBKV23  RDB0    REDBK2   START     TRANSID  RDBC                2006-06-09
REDBKV23  RDB0    REDBK2   START     TRANSID  RDBD                2006-06-09
REDBKV23  RDB0    REDBK2   START     TRANSID  RDBE                2006-06-09
REDBKV23  RDB0    REDBK2   START     TRANSID  RDB1                2006-06-09
REDBKV23  RDB1    REDBK1   ADDRESS   CWA      CWA                 2006-06-09
REDBKV23  RDB1    REDBK1   ASSIGN    APPLID   REDBKV23            2006-06-09
REDBKV23  RDB1    REDBK1   WRITE     FILE     REDBOOKF            2006-06-09
REDBKV23  RDB1    REDBK1   WRITEQ    TD       CESE                2006-06-09
REDBKV23  RDB1    REDBK1   WRITEQ    TSSHR    REDBOOKQ            2006-06-09
REDBKV23  RDB3    REDBK3   ASSIGN    APPLID   REDBKV23            2006-06-09
REDBKV23  RDB3    REDBK3   ENDBR     FILE     REDBOOKF            2006-06-09
REDBKV23  RDB3    REDBK3   READNEXT  FILE     REDBOOKF            2006-06-09
REDBKV23  RDB3    REDBK3   STARTBR   FILE     REDBOOKF            2006-06-09
REDBKV23  RDB3    REDBK4   LINK      PROGRAM  REDBK3              2006-06-09
DSNE610I NUMBER OF ROWS DISPLAYED IS 46
```

Once we have migrated the application to the TS 3.1 region we can rerun the test with IA running. Example 11-5 shows us all resources for the application in region REDBKV31. Data for transaction RDB3 has now been collected.

Example 11-5 Show us all dependencies used by Application Redbook 1 in REDBKV31 region

```
---------+---------+---------+---------+---------+---------+---------+---------+
-- SHOW US ALL RESOURCES USED BY APPLICATION REDBOOK 1
-- DURING THE LASR TEST RUN IN REGION REDBKV31
SELECT APPLID, TRANSID, PROGRAM,
  FUNCTION, TYPE, OBJECT, DATE(LAST_RUN) AS LAST_RUN
  FROM CIU4_CICS_DATA
  WHERE APPLID='REDBKV31'
    AND PROGRAM LIKE 'RED%';
---------+---------+---------+---------+---------+---------+---------+---------+
APPLID    TRANSID  PROGRAM   FUNCTION  TYPE      OBJECT            LAST_RUN
---------+---------+---------+---------+---------+---------+---------+---------+
REDBKV31  RDBA     REDBK1    ADDRESS   CWA       CWA               2006-06-22
REDBKV31  RDBA     REDBK1    ASSIGN    APPLID    REDBKV31          2006-06-22
REDBKV31  RDBA     REDBK1    WRITE     FILE      REDBOOKF          2006-06-22
REDBKV31  RDBB     REDBK1    ADDRESS   CWA       CWA               2006-06-22
REDBKV31  RDBB     REDBK1    ASSIGN    APPLID    REDBKV31          2006-06-22
REDBKV31  RDBB     REDBK1    WRITE     FILE      REDBOOKF          2006-06-22
REDBKV31  RDBC     REDBK1    ADDRESS   CWA       CWA               2006-06-22
REDBKV31  RDBC     REDBK1    ASSIGN    APPLID    REDBKV31          2006-06-22
REDBKV31  RDBC     REDBK1    WRITE     FILE      REDBOOKF          2006-06-22
REDBKV31  RDBD     REDBK1    ADDRESS   CWA       CWA               2006-06-22
REDBKV31  RDBD     REDBK1    ASSIGN    APPLID    REDBKV31          2006-06-22
REDBKV31  RDBD     REDBK1    WRITE     FILE      REDBOOKF          2006-06-22
REDBKV31  RDBE     REDBK1    ADDRESS   CWA       CWA               2006-06-22
REDBKV31  RDBE     REDBK1    ASSIGN    APPLID    REDBKV31          2006-06-22
REDBKV31  RDBE     REDBK1    WRITE     FILE      REDBOOKF          2006-06-22
REDBKV31  RDBI     REDBK5    ADDRESS   CWA       CWA               2006-06-22
REDBKV31  RDBI     REDBK5    ASSIGN    APPLID    REDBKV31          2006-06-22
REDBKV31  RDBI     REDBK5    ENDBR     FILE      REDBOOKF          2006-06-22
REDBKV31  RDBI     REDBK5    READNEXT  FILE      REDBOOKF          2006-06-22
REDBKV31  RDBI     REDBK5    STARTBR   FILE      REDBOOKF          2006-06-22
REDBKV31  RDBO     REDBK2    START     TRANSID   RDBA              2006-06-22
REDBKV31  RDBO     REDBK2    START     TRANSID   RDBB              2006-06-22
REDBKV31  RDBO     REDBK2    START     TRANSID   RDBC              2006-06-22
REDBKV31  RDBO     REDBK2    START     TRANSID   RDBD              2006-06-22
REDBKV31  RDBO     REDBK2    START     TRANSID   RDBE              2006-06-22
REDBKV31  RDBO     REDBK2    START     TRANSID   RDB1              2006-06-22
REDBKV31  RDB1     REDBK1    ADDRESS   CWA       CWA               2006-06-22
REDBKV31  RDB1     REDBK1    ASSIGN    APPLID    REDBKV31          2006-06-22
```

```
REDBKV31  RDB1     REDBK1    WRITE     FILE      REDBOOKF          2006-06-22
REDBKV31  RDB3     REDBK3    ASSIGN    APPLID    REDBKV31          2006-06-22
REDBKV31  RDB3     REDBK3    ENDBR     FILE      REDBOOKF          2006-06-22
REDBKV31  RDB3     REDBK3    READNEXT  FILE      REDBOOKF          2006-06-22
REDBKV31  RDB3     REDBK3    STARTBR   FILE      REDBOOKF          2006-06-22
REDBKV31  RDB3     REDBK4    LINK      PROGRAM   REDBK3            2006-06-22
DSNE610I NUMBER OF ROWS DISPLAYED IS 34
DSNE612I DATA FOR COLUMN HEADER OBJECT COLUMN NUMBER 6 WAS TRUNCATED
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
---------+---------+---------+---------+---------+---------+---------+---------+
```

There are only 34 distinct commands this time. This indicates that not all tests have been completed. We know from our transaction query that all transactions have been tested. We can now use an SQL query to show us which functions have been performed in the REDBKV23 region and not in the REDBKV31 region. See Example 11-6. We can see that all the TDQUEUE and TSQUEUE processing has only been performed in region REDBKV23. From looking at the REDBK1 program source we can see that the WRITEQ processing is part of error processing. To complete the test we need to drive this program through the error processing code.

Example 11-6 SQL command to show which commands have run in REDBKV23 and not in REDBKV31

```
---------+---------+---------+---------+---------+---------+---------+---------+
SELECT Y.APPLID ,
       Y.TRANSID, Y.PROGRAM, Y.FUNCTION, Y.TYPE, Y.OBJECT
  FROM (
    (SELECT DISTINCT * FROM CIU4_CICS_DATA WHERE
       APPLID = 'REDBKV31'
       AND TRANSID LIKE 'RD%') AS X
   RIGHT OUTER JOIN
    (SELECT DISTINCT * FROM CIU4_CICS_DATA WHERE
       APPLID = 'REDBKV23'
       AND TRANSID LIKE 'RD%') AS Y
   ON  X.TRANSID = Y.TRANSID
   AND X.PROGRAM = Y.PROGRAM
   AND X.FUNCTION= Y.FUNCTION
   AND X.TYPE    = Y.TYPE
  )
  WHERE X.APPLID IS NULL
---------+---------+---------+---------+---------+---------+---------+---------+
APPLID    TRANSID  PROGRAM   FUNCTION  TYPE      OBJECT
---------+---------+---------+---------+---------+---------+---------+---------+
REDBKV23  RDBA     REDBK1    WRITEQ    TD        CESE
REDBKV23  RDBA     REDBK1    WRITEQ    TSSHR     REDBOOKQ
```

```
REDBKV23  RDBB     REDBK1    WRITEQ    TD        CESE
REDBKV23  RDBB     REDBK1    WRITEQ    TSSHR     REDBOOKQ
REDBKV23  RDBC     REDBK1    WRITEQ    TD        CESE
REDBKV23  RDBC     REDBK1    WRITEQ    TSSHR     REDBOOKQ
REDBKV23  RDBD     REDBK1    WRITEQ    TD        CESE
REDBKV23  RDBD     REDBK1    WRITEQ    TSSHR     REDBOOKQ
REDBKV23  RDBE     REDBK1    WRITEQ    TD        CESE
REDBKV23  RDBE     REDBK1    WRITEQ    TSSHR     REDBOOKQ
REDBKV23  RDB1     REDBK1    WRITEQ    TD        CESE
REDBKV23  RDB1     REDBK1    WRITEQ    TSSHR     REDBOOKQ
DSNE610I NUMBER OF ROWS DISPLAYED IS 12
```

Which CSD data sets and lists are used during IA collection

CICS IA 2.1 stores CSD information for each region in a new CIU4_REGION_INFO table.

The SQL query in Example 11-7 was used to generate the QMF™ report in Figure 11-32 on page 387. It shows us the CSD data set names, the CSD resource lists used, and the IA collection start time and last save time for all regions.

Example 11-7 IA query for region information

```
SELECT APPLID, CSD_NAME, CSD_GROUP_LIST1,
       CSD_GROUP_LIST2,
       DATE(DEP_COLL_LASTSTART) AS STARTED,
       DATE(DEP_COLL_LASTSAVE) AS SAVED
FROM CIU4_REGION_INFO;
```

You can now use CICS CM to compare CSD data sets and resource lists. See 11.2, “The power of comparing using CICS CM” on page 358.

```
REPORT                                              LINE 1      POS 1      79


                                                       CSD        CSD
   APPLID     CSD                                      GROUP      GROUP
              NAME                                     LIST1      LIST2
   --------   ---------------------------------------- --------   --------
   STARTED      SAVED
   ----------   ----------
   CCVT32A    CCVCICS.CCVT32A.DFHCSD                   DFHLIST    CCVT32A
   2006-06-07   2006-06-07
   REDBKV23   REDBK23.REDBKV23.DFHCSD                  DFHLIST    REDLIST
   2006-06-09   2006-06-09
   REDBKV31   REDBK31.REDBKV31.DFHCSD                  DFHLIST    REDLIST
   2006-06-27   2006-06-27


*** END ***







1=Help         2=           3=End       4=Print       5=Chart        6=Query
7=Backward     8=Forward    9=Form     10=Left       11=Right       12=
OK, REPORT is displayed.
COMMAND ===>                                                 SCROLL ===> PAGE
```

Figure 11-32 QMF report showing the CSD and group lists used by CICS region

11.5 Reducing performance impact during IA collection

CICS IA has a number of ways to reduce the small performance impact of having the CICS IA collector on. In this section we show you how to:

- Collect on transaction prefix.
- Exclude programs or transactions from collection.
- Use the IA timer facility to balance collections across regions.

Collect on transaction prefix

We know that the Redbook Application 1 consists of transactions starting RD. We can therefore turn the IA on to collect data for only resources used by transactions with a prefix of RD.

To do this we select option **2. Configure Region Options** from the Main Administration Menu of transaction CINT. Then we select option **6. General Options** in the line command against region REDBKV31, as shown in Figure 11-33.

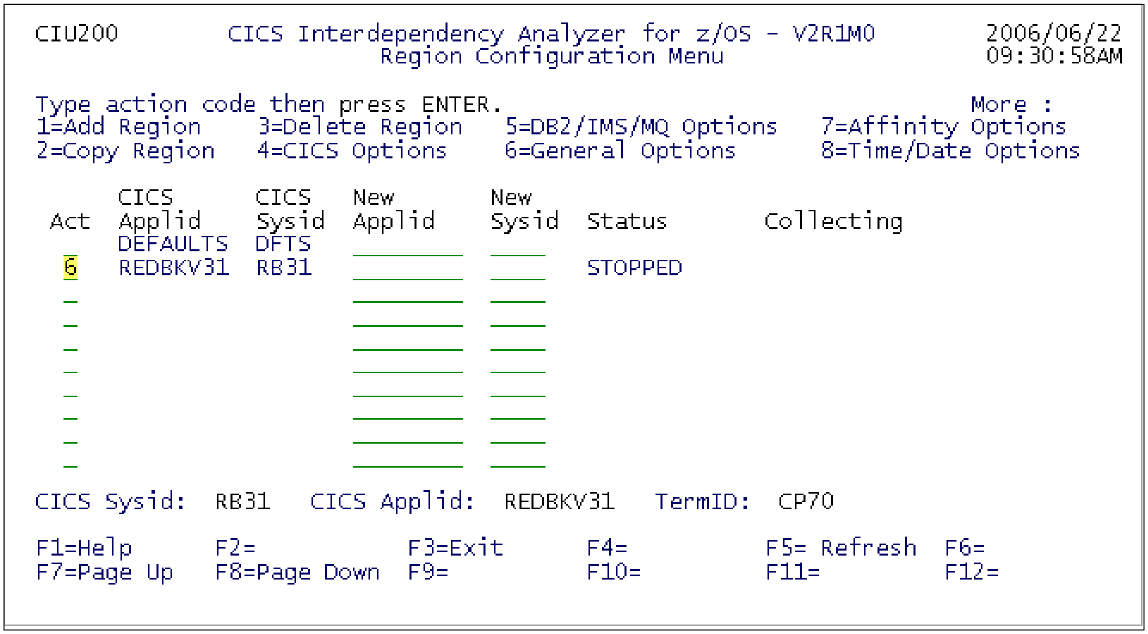


Figure 11-33 CINT - Region Configuration Menu

To configure IA to collect for transactions with a prefix of RD only, simply enter `RD` in the Transaction prefix option, as shown in Figure 11-34.

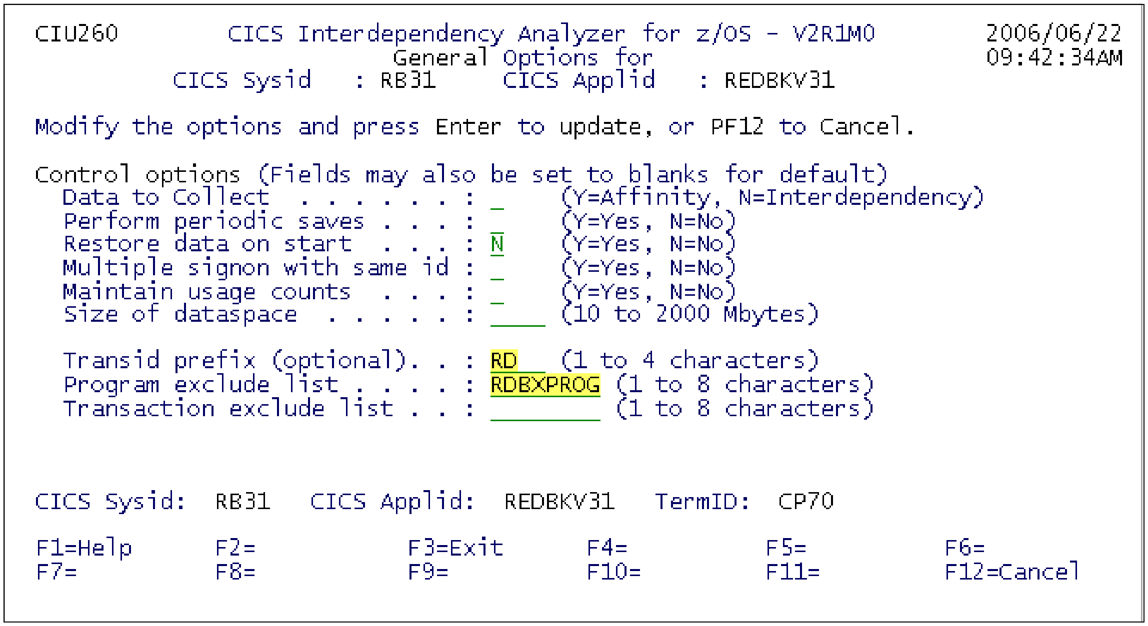


Figure 11-34 CINT - General Options for region REDBKV31

Exclude programs or transactions from collection

The Redbook Application 1 could have programs other than ones starting with a prefix of RED. For example, there could be a common suit of programs for printing used by many applications. We can remove these programs from the collection by adding them to the list of excluded programs supplied by CICS IA.

The default list of excluded programs supplied by CICS IA is contained in module CIUXPROG. This includes IBM prefix program names, for example, DFH.

To create our own exclude list it is recommended that we add program prefixes to the list supplied in CIU.SCIUSRCE(CIUXPROG).

First we must copy CIUXPROG into a new member called RDBXPROG. Then we edit this member to include program prefixes that you do not wish to collect data for. In this case we have added the prefix PRT (see Example 11-8). We have also changed the CSECT name from CIUXPROG to RDBXPROG.

Example 11-8 CICS IA - Program exclude list

```
         TITLE '** RDBXPROG - CIU Program Exclude List **'
         EJECT
RDBXPROG CSECT
RDBXPROG AMODE 31
RDBXPROG RMODE ANY
         DS    0F
* Add user prefixes here
         DC    AL1(3),C'PRT'          Redbook Print Interface
* Predefined prefixes
         DC    AL1(3),C'DFH'          CICS
         DC    AL1(3),C'CEE'
         DC    AL1(3),C'EQA'
         DC    AL1(3),C'IBM'
         DC    AL1(3),C'EDC'
         DC    AL1(3),C'IGZ'
         DC    AL1(3),C'CAU'          Affinities utility
         DC    AL1(3),C'CIU'          Interdependency utility
         DC    AL1(4),C'DSNC'         DB2
         DC    AL1(4),C'DSN2'         DB2
         DC    AL1(3),C'EYU'          CICSPlex SM
         DC    AL1(3),C'CSQ'          MQ
         DC    AL1(3),C'CMZ'          CICS PM
         DC    AL1(3),C'CPA'          CICS PA
         DC    AL1(3),C'ABL'          OTTO
         DC    AL1(3),C'CBM'          CICS BEP
         DC    AL1(3),C'DWW'          CICS VR
         DC    AL1(3),C'ISZ'          Session manager
         DC    AL1(3),C'VID'          VT
         DC    AL1(4),C'IN25'
         DC    AL1(0)                 End of list
         END   RDBXPROG
```

We must now assemble and link this module. This can be done by running the sample configured job CIUJCLXP. Review the SYSIN data set name for the assembler step and the SYSLMOD data set for the linkedit step, as in Example 11-9.

Example 11-9 Assemble and link program exclude list

```
.
.
//SYSIN    DD DSN=REDBK31.SCIUSRCE(RDBXPROG),DISP=SHR
.
.
//SYSLMOD  DD DSN=REDBK31.APPL.LOADLIB(RDBXPROG),DISP=SHR
.
.
```

We now need to configure CICS IA to use this new program exclude list in region RDBKV31. This is done by overriding the default list name in the General Options for region REDBKV31, as shown in Figure 11-34 on page 389.

Using the IA Timer facility

We can configure IA to collect data in a region at certain times of day, days of the week, days of the month, and months of the year. This facility is extremely useful when collecting in a production environment pre or post migration. There are several reasons for doing this:

- Reduce performance impact for certain times of day. For example, you know that your application peak time is between 11 a.m. and 2 p.m. Do not collect data at this time.
- If you have cloned CICS regions, you would set CICS IA to run at a certain time in one region then in the cloned region and compare results.
- Make sure IA is running at month end and year end to capture resources that might only be used during these periods.

The default timer options are set, so IA is to run at all times. To override this we can configure the timer options for each region. To do this we select option **8. Timer Options** in the line command against region REDBKV31 in the Configuration Menu.

Figure 11-35 shows an example that stops the collector between 10 a.m. and 2 p.m. each day and all day Monday.

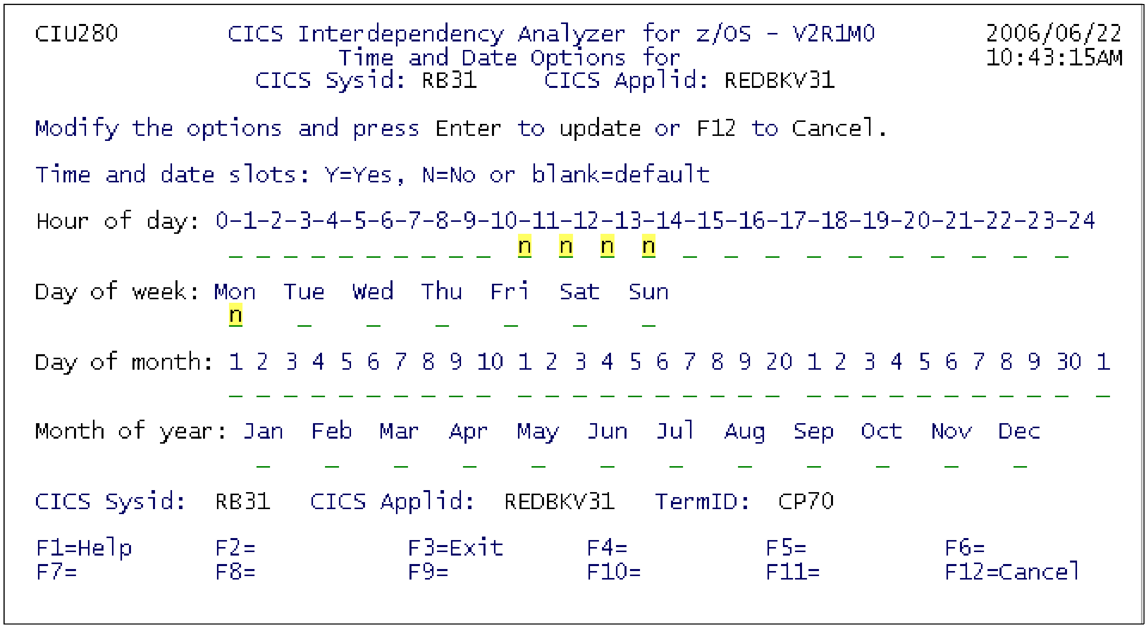


Figure 11-35 CINT - Timer and Date Options

11.6 Threadsafe and how IA can help

In this section we take the two application programs that we identified as non threadsafe in Chapter 8, “Migrating CICS TS 2.3 CSD to CICS TS 3.1 CSD” on page 197 and make them threadsafe. We use the ENQ/DEQ method to serialize any use of shared storage.

In this section we demonstrate the following:

- Making a sample application threadsafe
- Identifying non threadsafe programs using IA
- Displaying TCB modes using IA
- Showing what can go wrong if you do not take threadsafe seriously

Prior to enabling any application program to be defined as threadsafe, a review of the application code must be performed. This cannot be emphasized strongly enough. It is necessary for two reasons.

First application data integrity must be maintained. Prior to CICS transaction Server 2.2 user applications and exits ran on the QR TCB, which is a restricted or closed environment. CICS provided the serialization needed to ensure that application data integrity was never compromised. In this environment programs could be sure that no more than one quasi reentrant program could run at the same time. Now, for applications that have DB2 calls (or calls to other TRUEs that have been enabled as OPENAPI), it is possible for two or more programs to be running concurrently on different open TCBs and the QR TCB. Therefore it becomes very important that shared resources used by an application are serialized to prevent any application integrity problems due to more than one program accessing the same resource at the same time.

The second reason for conducting a review of your application code is to ensure that once CICS moves an application over to an open TCB it remains there for as long as possible after the DB2 call has been completed. CICS will switch the application program back to the QR TCB in order to execute CICS API or SPI commands that are non threadsafe. CICS must do this to maintain the integrity of such things as the CSA and other control blocks used by the commands.

In order to demonstrate the potential problems with defining an application as threadsafe we wrote a simple file update application. The following sections describe this application plus the various tests we performed.

We also recommend that CICS PA is used to compare performance before and after converting a program to threadsafe. For more information about the use of CICS PA see Chapter 2, “Overview of CICS PA” on page 29.

The application

In 8.3.4, “Identifying non-threadsafe programs” on page 227, we identified that programs REDBK1 and REDBK5 were non threadsafe. We can also see this from Example 11-4 on page 382. Both programs issue an EXEC CICS ADDRESS CWA.

The sample application program, REDBK5, initializes the key in the CWA from the last key stored in VSAM file REDBOOKF.

The sample application program, REDBK1, shown in outline in Figure 11-36, simply addresses the CWA and uses an integer value in the CWA as the next key to use in an EXEC CICS WRITE command. In a non threadsafe environment (that is, with the program running on the QR TCB) we would not expect there to be any duplicate file records (DUPREC) since there is only one instance of this program executing at the time of addressing the CWA, incrementing its value and using the increment value as the key in the subsequent WRITE command.

In order to test this application we started 75 instances of the invoking transaction RDB1.

```
int *CWAptr;                        /* Pointer to CWA                   */
int RidFld;                         /* Rid field for EXEC CICS WRITE    */
                                    /* request                          */
struct CWA_INFO {
  int  Counter;                     /* Just a number                    */
  char Userid[9];                   /* This userid                      */
  char Date[9];                     /* The current Date                 */
  char Time[9];                     /* the current Time                 */
};
CWA_INFO  *CWA;
EXEC CICS ADDRESS CWA(CWAptr) RESP(resp);
CWA = (CWA_INFO *)CWAptr;
---> missing code section to loop for a period of time <---
CWA->Counter++;
RidFld = CWA->Counter;
EXEC CICS WRITE FILE(File) RIDFLD(RidFld)
     FROM(FileRec) LENGTH(sizeof(FILEINFO)) RESP(resp);
---> missing code section to write any errors to a TS queue <---
EXEC CICS RETURN ;
```

Figure 11-36 Extract of sample application program REDBK1

Non threadsafe output (QR TCB)

When running in non threadsafe mode (that is, program REDBK1 being defined as shown in Figure 11-37) we had no error messages in our error message TS QUEUE plus the last 10 records written to the file had sequential key values, as shown in Figure 11-39 on page 396.

Example 11-10 on page 396 shows us the EXEC CICS calls issued by program REDBK1 running in QR mode. The SQL output shows us the TCBMODE for each command (in this case QR).

```
  File  Menu  Settings  Help
--------------------------------------------------------------------------
Edit                              Program
Command ===> _____________________________________________________________

    Program . . . : REDBK1
    Group . . . . : REDBOOK
    Location  . . : REDBK31.REDBKV31.DFHCSD
    Change Date . : 2006/06/21 16:00:05
    Description . . __________________________________________________________
                                                                More:     +

    Language  . . . LE370      +
    Reload  . . . . NO   +
    Resident  . . . NO   +
    Usage . . . . . NORMAL     +
    UseLPAcopy  . . NO   +
    Status  . . . . ENABLED   +
    CEDF  . . . . . YES  +
    DataLocation    BELOW  +
    ExecKey . . . . CICS  +
    Concurrency . . QUASIRENT   +
    API . . . . . . CICSAPI  +
```

Figure 11-37 Program REDBK1 definition - non threadsafe

```
CEBR  TSQ REDBOOKQ          SYSID RB31 REC     1 OF      0    COL      1 OF      0
ENTER COMMAND ===>
     **************************  TOP OF QUEUE  ******************************
     *************************  BOTTOM OF QUEUE  ****************************
```

Figure 11-38 CEBR display of error message queue

```
RDB3 LAST 10 UPDATES TO REDBOOKF

    ITEM 1    KEY=0000006B written by EYJ
    ITEM 2    KEY=0000006A written by EYJ
    ITEM 3    KEY=00000069 written by EYJ
    ITEM 4    KEY=00000068 written by EYJ
    ITEM 5    KEY=00000067 written by EYJ
    ITEM 6    KEY=00000066 written by EYJ
    ITEM 7    KEY=00000065 written by EYJ
    ITEM 8    KEY=00000064 written by EYJ
    ITEM 9    KEY=00000063 written by EYJ
    ITEM 10   KEY=00000062 written by EYJ
```

Figure 11-39   Transaction RDB3 output

Example 11-10   IA report for program REDBK1 in QR mode

```
---------+---------+---------+---------+---------+---------+---------+---------+
-- SHOW ME ALL RESOURCES USED BY PROGRAM REDBK1 IN REGION REDBV31
SELECT DISTINCT PROGRAM, FUNCTION, TYPE, OBJECT, TCBMODE
  FROM CIU4_CICS_DATA
  WHERE PROGRAM='REDBK1' AND APPLID='REDBKV31';
---------+---------+---------+---------+---------+---------+---------+---------+
PROGRAM   FUNCTION  TYPE      OBJECT                  TCBMODE
---------+---------+---------+---------+---------+---------+---------+---------+
REDBK1    ASSIGN    APPLID    REDBKV31                QR
REDBK1    ADDRESS   CWA       CWA                     QR
REDBK1    WRITE     FILE      REDBOOKF                QR
DSNE610I NUMBER OF ROWS DISPLAYED IS 3
```

Threadsafe output with unchanged program

We changed the REDBK1 program definition to be THREADSAFE and OPENAPI, as shown in Figure 11-40.

> **Note:** During the testing of this program it was identified that program REDBK1 was not compiled and linked with the RENT option. All programs defined as threadsafe must be reentrant.

```
  File  Menu  Settings  Help
-----------------------------------------------------------------------------
Edit                                   Program
Command ===> ________________________________________________________________

    Program . . . : REDBK1
    Group . . . . : REDBOOK
    Location  . . : REDBK31.REDBKV31.DFHCSD
    Change Date . : 2006/06/21 16:00:05
    Description . . __________________________________________________________
                                                                  More:     +

    Language  . . . LE370____  +
    Reload  . . . . NO___  +
    Resident  . . . NO___  +
    Usage . . . . . NORMAL____  +
    UseLPAcopy  . . NO___  +
    Status  . . . . ENABLED__  +
    CEDF  . . . . . YES__  +
    DataLocation    BELOW  +
    ExecKey . . . . CICS_  +
    Concurrency . . THREADSAFE  +
    API . . . . . . OPENAPI  +
```

Figure 11-40 Program REDBK1 definition - threadsafe

We again ran 75 instances of transaction RDB1. This time since multiple instances of program REDBK1 were executing concurrently on L8 TCBs there was the potential for the same CWA key value to be used more than once. This did in fact happen, as shown in Figure 11-41.

```
 CEBR  TSQ REDBOOKQ         SYSID RB31 REC     1 OF    12    COL     1 OF    80
  ENTER COMMAND ===>
      *************************  TOP OF QUEUE  *******************************
 00001 WRITE Failure resp=14..
 00002 DUPREC RID=137..
 00003 WRITE Failure resp=14..
 00004 DUPREC RID=147..
 00005 WRITE Failure resp=14..
 00006 DUPREC RID=176..
 00007 WRITE Failure resp=14..
 00008 DUPREC RID=180..
*************************  BOTTOM OF QUEUE  *****************************
```

Figure 11-41 TS QUEUE logo showing DUPREC errors

```
RDB3 LAST 10 UPDATES TO REDBOOKF

    ITEM 1    KEY=000000B4 written by AYS
    ITEM 2    KEY=000000B2 written by AYS
    ITEM 3    KEY=000000B0 written by AYS
    ITEM 4    KEY=000000AF written by AYS
    ITEM 5    KEY=000000AE written by AYS
    ITEM 6    KEY=000000AC written by AYS
    ITEM 7    KEY=000000AA written by AYS
    ITEM 8    KEY=000000A8 written by AYS
    ITEM 9    KEY=000000A7 written by AYS
    ITEM 10   KEY=000000A6 written by AYS
```

Figure 11-42 RDB3 output showing missing record keys

Example 11-11 shows us the EXEC CICS COMMANDS called when running program REDBK1 as THREADSAFE program, in the OPENAPI and a storage key of CICS. There are three things to note here:

- EXEC CICS WRITEQ to a shared TSQueue called REDBOOKQ. As seen previously, this command is only executed during error conditions. IA, therefore, indicates that an error message has been written to the queue. See Figure 11-41 on page 398.
- The program starts on an L8 TCB. This is because it is defined as having a storage key of CICS.
- All EXEC CICS FILE commands are made threadsafe by CICS by switching to the QR TCB. Note that this causes two TCB switches in this case.

Example 11-11 IA report for program REDBK1 in threadsafe mode running on the OPENAPI

```
---------+---------+---------+---------+---------+---------+---------+---------+
-- SHOW ME ALL RESOURCES USED BY PROGRAM REDBK1 IN REGION REDBV31
SELECT DISTINCT PROGRAM, FUNCTION, TYPE, OBJECT, TCBMODE
  FROM CIU4_CICS_DATA
  WHERE PROGRAM='REDBK1' AND APPLID='REDBKV31';
---------+---------+---------+---------+---------+---------+---------+---------+
PROGRAM   FUNCTION  TYPE      OBJECT                   TCBMODE
---------+---------+---------+---------+---------+---------+---------+---------+
REDBK1    ADDRESS   CWA       CWA                      L8
REDBK1    ASSIGN    APPLID    REDBKV31                 L8
REDBK1    WRITE     FILE      REDBOOKF                 QR
REDBK1    WRITEQ    TSSHR     REDBOOKQ                 L8
DSNE610I NUMBER OF ROWS DISPLAYED IS 4
```

Threadsafe output with changed program

One solution to enable our sample application to run as threadsafe is to put an ENQ and DEQ around the address CWA and its subsequent increment. This we did (see Figure 11-43) and ran the 75 instances of transaction RDB1 again. This time the results were the same as the non-threadsafe example (that is, there were no DUPREC errors and there were sequential key values in the file).

```
int *CWAptr;                          /* Pointer to CWA                 */
int RidFld;                           /* Rid field for EXEC CICS WRITE  */
                                      /* request                        */
struct CWA_INFO {
  int  Counter;                       /* Just a number                  */
  char Userid[9];                     /* This userid                    */
  char Date[9];                       /* The current Date               */
  char Time[9];                       /* the current Time               */
};
CWA_INFO  *CWA;
        EXEC CICS ENQ RESOURCE(EnqName) LENGTH(sizeof(EnqName))
EXEC CICS ADDRESS CWA(CWAptr) RESP(resp);
CWA = (CWA_INFO *)CWAptr;
---> missing code section to loop for a period of time <---
CWA->Counter++;
RidFld = CWA->Counter;
     EXEC CICS DEQ RESOURCE(EnqName) LENGTH(sizeof(EnqName))
EXEC CICS WRITE FILE(File) RIDFLD(RidFld)
     FROM(FileRec) LENGTH(sizeof(FILEINFO)) RESP(resp);
---> missing code section to write any errors to a TS queue <---
EXEC CICS RETURN ;
```

Figure 11-43 Extract of sample application program REDBK1 with ENQ DEQ

Example 11-12 shows us the EXEC CICS COMMANDS called when running program REDBK1 with the changes described above. We can now see that the ADDRESS CWA command is serialized by using the ENQ/DEQ technique.

> **Note:** The ORDER BY TRANSLATE(HEX(OFFSET),'B','F') clause sorts the query by program offset.

Example 11-12 IA report on the modified REDBK1 program

```
---------+---------+---------+---------+---------+---------+---------+---------+
-- SHOW ME ALL RESOURCES USED BY PROGRAM REDBK1 IN REGION REDBV31
SELECT TRANSID, PROGRAM,
       OFFSET,PROGRAM, FUNCTION, TYPE, OBJECT, TCBMODE
  FROM CIU4_CICS_DATA
  WHERE PROGRAM='REDBK1' AND APPLID='REDBKV31' AND TRANSID='RDB1'
  ORDER BY TRANSLATE(HEX(OFFSET),'B','F');
---------+---------+---------+---------+---------+---------+---------+---------+
TRANSID  PROGRAM   OFFSET    PROGRAM   FUNCTION  TYPE      OBJECT      TCBMODE
---------+---------+---------+---------+---------+---------+---------+---------+
RDB1     REDBK1    000001E8  REDBK1    ASSIGN    APPLID    REDBKV31    L8
RDB1     REDBK1    00000288  REDBK1    ENQ       ENQNAME   REDBKENQ    L8
RDB1     REDBK1    000002B0  REDBK1    ADDRESS   CWA       CWA         L8
RDB1     REDBK1    0000074C  REDBK1    DEQ       ENQNAME   REDBKENQ    L8
RDB1     REDBK1    00000956  REDBK1    WRITE     FILE      REDBOOKF    QR
DSNE610I NUMBER OF ROWS DISPLAYED IS 5
```

Part 3

Appendixes

A

CICS IA installation and customization

This appendix describes steps to take before you can use CICS IA V2.1.

The SMP/E installation process for CICS IA is described in the Program Directory that is distributed with the product.

This appendix describes the simplest way to install and customize the CICS Interdependency Analyzer in a single CICS region.

We do not cover migration steps from an older release. For this, refer to the *CICS Interdependency Analyzer for z/OS User's and Reference Version 2 Release 1*, SC34-6685.

CICS IA requirements

To use CICS IA you need:

- OS/390 Version 2.10 or later.
- CICS Transaction Server for OS/390 or CICS Transaction Server for z/OS. Each CICS region on which the CICS IA Collector is to run must have either Language Environment installed and active or a COBOL runtime environment. A CICS region on which the CICS IA Query interface is to be run must have DB2 installed and active.
- To control CICS IA Collectors on multiple regions from a single CICS terminal, the VSAM files to which CICS saves dependency data and control information must be shared across all the regions. To share these files, you can use either:
  - VSAM record-level sharing (RLS). If you use VSAM RLS, all regions must be in the same MVS parallel sysplex. (A parallel sysplex is a sysplex that uses a coupling facility, which is required to support VSAM RLS.)
  - Function shipping to a file-owning region (FOR). For information about CICS function shipping, see the *CICS Intercommunication Guide*.

CICS-related steps

This section describes the CICS-related steps you must take before you can use CICS IA.

We do not recommend that you edit the sample jobs in the library hlq.SCIUSAMP. CICS IA V2.1 provides a customization program that will copy the hlq.SCIUSAMP data set to a data set of your choice and will configure the jobs to meet your installation requirements. It will customize high-level qualifiers for your IA environment, your DB2 environment, and your CICS environment. It also allows you to set up a JOB card.

CICS IA customization

In our case the IA high-level qualifier is set to CIU and the customized data sets high-level qualifier is set to REDBK23.MIG23T31.

To customize CICS IA we need to run the CIUCNFG1 exec. See Figure A-1.

```
  Menu  List  Mode  Functions  Utilities  Help
 ------------------------------------------------------------------------------
                              ISPF Command Shell
Enter TSO or Workstation commands below:

===> EX 'CIU.SCIUEXEC(CIUCNFG1)' 'CIU ENU'


Place cursor on choice and press enter to Retrieve command

=> EX 'CIU.SCIUEXEC(CIUCNFG1)' 'CIU ENU'
=>
=>
=>
=>
=>
=>
=>
=>
=>

 F1=Help      F3=Exit     F10=Actions  F12=Cancel
```

Figure A-1   IA ISPF customization

Follow the instructions in the exec and configure your jobs. See Figure A-2 through to Figure A-7 on page 411.

```
*************** CICS Interdependency Analyzer for z/OS - V2R1M0 ***************
Command ===> ________________________________________________________________
                                                                   More:     +

                  Welcome to the CICS IA Customization Function

  This function will assist you in customizing your CICS IA sample
  jobs and sample SQL.

  NOTE: Before proceeding with this function please read chapter 2,
        "Preparing to use the CICS IA", in the "User's Guide and
        Reference".
        Also, please consult your DB2 Administrator.

  All customized members will be copied from:

    CIU.SCIUSAMP
    CIU.SCIUSAME
    CIU.SCIUCLIS
    CIU.SCIUSQL
    CIU.SCIUDAT1
    CIU.SCIUDAT2

  The original libraries will not be changed.
```

Figure A-2 IA ISPF customization - entry panel

```
*************** CICS Interdependency Analyzer for z/OS - V2R1M0 ***************
Command ===> _________________________________________________________________
                                                                   More:     +

Press ENTER to proceed, PF3 to exit or PF1 for Help
Press PF8 to Scroll Down, PF7 to Scroll Up

Please update the following:

Output Datasets:

OUTPUT DSN FOR SCIUSAMP   : REDBK23.MIG23T31.SCIUSAMP__________________
OUTPUT DSN FOR SCIUCLIS   : REDBK23.MIG23T31.SCIUCLIS__________________
OUTPUT DSN FOR SCIUSQL    : REDBK23.MIG23T31.SCIUSQL___________________
OUTPUT DSN FOR SCIUDAT1   : REDBK23.MIG23T31.SCIUDAT1__________________
OUTPUT DSN FOR SCIUDAT1   : REDBK23.MIG23T31.SCIUDAT2__________________
OUTPUT DEVICE TYPE        : SYSDA_

CICS IA variables:

IA PRODUCT QUALIFIER      : CIU
IA VSAM FILE QUALIFIER    : REDBK23.MIG23T31__________________
IA TCPIP PORT NUMBER      : 10032
```

Figure A-3 IA ISPF customization - change panel 1

```
*************** CICS Interdependency Analyzer for z/OS - V2R1M0 ***************
Command ===> _________________________________________________________________
                                                                   More:   - +
DB2 variables:

DB2 QUALIFIER             : DSN710___________________________
DB2 RUNLIB.LOAD HLQ       : DSN710___________________________
DB2 PROCLIB DATASET NAME  : DSN710____________________________________
DB2 SUB SYSTEM            : DB2P
DB2 TABLE QUALIFIER       : EYJ_____
DB2 TABLE OWNER           : EYJ_____
DB2 DASD VOLUME           : SYSDA_
DB2 VERSION               : V710
DB2 RACF GROUP            : ________

CICS variables:

CICS VERSION              : V230
CICS QUALIFIER            : CICS.V630.CICS____________________
CICS CSD FILE NAME        : REDBK23.REDBKV23.DFHCSD____________________
CICS CSD LIST NAME        : REDLIST_
RLSACCESS FOR VSAM FILES  : NO_
LSRPOOLID FOR VSAM FILES  : NONE
CICS FILE OWNING REGION   : ____
```

Figure A-4 IA ISPF customization - change panel 2

```
*************** CICS Interdependency Analyzer for z/O                 End of data
Command ===> ____________________________________________________________________
                                                                     More:     -
CICS FILE OWNING REGION  : ____

Migration Variables:

PREVIOUS IA QUALIFIER    : __________________________________
NEW IA QUALIFIER         : __________________________________
OLD DB2 TABLE QUALIFIER  : ________
NEW DB2 TABLE QUALIFIER  : ________

General variables:

ASSEMBLER PROGRAM        : ASMA90
LE QUALIFIER             : CEE_______________________________

JOBCARD HEADER:
//_CIUJOB_ JOB USER=EYJ,NOTIFY=EYJ,___________________________________
//           CLASS=A,MSGCLASS=Y,REGION=0M_______________________________
//*___________________________________________________________________
//*___________________________________________________________________

Press ENTER to proceed, PF3 to exit or PF1 for Help
```

Figure A-5 IA ISPF customization - change panel 3

```
*************** CICS Interdependency Analyzer for z/OS - V2R1M0 ***************
Command ===> ____________________________________________________________________
                                                                     More:     +

   All customized members will be copied :

      from  :  CIU.SCIUSAMP
       and  :  CIU.SCIUSAME
        to  :  REDBK23.MIG23T31.SCIUSAMP

      from  :  CIU.SCIUCLIS
        to  :  REDBK23.MIG23T31.SCIUCLIS

      from  :  CIU.SCIUSQL
        to  :  REDBK23.MIG23T31.SCIUSQL

      from  :  CIU.SCIUDAT1
        to  :  REDBK23.MIG23T31.SCIUDAT1

      from  :  CIU.SCIUDAT2
        to  :  REDBK23.MIG23T31.SCIUDAT2

  WARNING: This takes a few minutes to process.
```

Figure A-6 IA ISPF customization - confirmation panel

```
*************** CICS Interdependency Analyzer for z/OS - V2R1M0 ***************
Command ===> ____________________________________________________________________

 CUSTOMIZATION SUCCESSFUL

 Customized have been stored in the following datasets.

 JCL library      :  REDBK23.MIG23T31.SCIUSAMP

 EXEC library     :  REDBK23.MIG23T31.SCIUCLIS

 SQL library      :  REDBK23.MIG23T31.SCIUSQL

 DAT1 library     :  REDBK23.MIG23T31.SCIUDAT1

 DAT2 library     :  REDBK23.MIG23T31.SCIUDAT2

Please contine with chapter 2 'Preparing to use the CICS IA'.

Press ENTER to Exit
```

Figure A-7 IA ISPF customization - completion panel

Creating the VSAM files

To create the VSAM files, we need the jobs CIUJCLCC and CIUJCLCA.

> **Sizing considerations:** A customer who uses CICS IA in more than 300 CICS regions had the following experience:
>
> > One important thing we do is a twice-monthly delete/define. This gives us a fresh copy and prevents continuous growth. Several things can cause continuous growth in the repository, such as resources that use a variable in the resource name. One such example would be a program that builds an enqueue name from a record key. Nasty little things like this can exist everywhere and would eventually cause the VSAM repository to balloon out of control. What we look at and keep historically is in the DB2 table. We use the standard reorg/purge utility to maintain the correct amount of data in the DB2 repository. This enables us to discard old or unneeded records, and at the same time provides a backup of that discarded data.

We have to create the files in Table A-1.

Table A-1 CICS IA VSAM files and associated jobs

File	Description	Job
hlq.CIUCNTL	The control record file. A recoverable file used to hold control information.	CIUJCLCC
hlq.CIUINT1	The CICS dependency data file. A non-recoverable file used to record dependencies on CICS resources with names up to 32 bytes long.	CIUJCLCA
hlq.CIUINT2	The DB2 dependency data file. A non-recoverable file used to record dependencies on DB2 resources.	CIUJCLCA
hlq.CIUINT3	The MQ dependency data file. A non-recoverable file used to record dependencies on MQ resources.	CIUJCLCA
hlq.CIUINT4	The IMS dependency data file. A non-recoverable file used to record dependencies on IMS resources.	CIUJCLCA
hlq.CIUINT5	The CICS >32 dependency data file. A non-recoverable file used to record dependencies on CICS resources with names longer than 32 bytes.	CIUJCLCA
hlq.CIUAFF1	The CICS affinity data file. A non-recoverable file used to record affinities with names up to 16 bytes long.	CIUJCLCA
hlq.CIUAFF2	The CICS affinity data file. A non-recoverable file used to record affinities with names up to 32 bytes long.	CIUJCLCA
hlq.CIUAFF3	The CICS affinity data file. A non-recoverable file used to record affinities with names greater than 32 bytes long.	CIUJCLCA

The values in the jobs CIUJCLCA and CIUJCLCC have already been configured and stored in REDBK23.MIG23T31.SCIUSAMP.

Edit, review, and submit CIUJCLCA and CIUJCLCC.

Keep in mind that this sample is only for a single-region installation.

To create dependency data and control record files for multiple regions, do either of the following:

- Use VSAM RLS or function shipping to share the files you created for the first region across all the CICS regions on which you intend to run the collector.
- Edit and run copies of the CIUJCLCC and CIUJCLCA jobs for all of the other CICS regions on which you intend to run the collector.

Defining resources to CICS

The configured jobs provide JCL and statements to define the following resources:

- CICS IA program components
- CICS IA transactions CINT, CINB, and CINQ
- VSAM files CIUCNTL, CIUINT1/2/3/4/5, and CIUAFF1/2/3
- DB2ENTRY) definitions
- CINT transient data queue for messages

To define English language resources to CICS, edit, review, and run job CIUJENCR.

To define all other resources to CICS select the appropriate configured job for your CICS release. Select the appropriated job for your release.

We are installing IA in a CICS TS 2.3 region. Edit, review and run job CIUJ23CR.

Note: To add the group to your RDO list uncomment the last RDO command:

```
*********************************************************************
**   ADD TO CICS STARTUP GROUP LIST IF DESIRED.                    **
**   ENABLE THE ADD COMMANDS BY DELETING THE PRECEDING '*'         **
*********************************************************************
* ADD GROUP(CIU63G21) LIST(REDLIST)
```

Tailoring the CICS startup job

To enable CICS IA to run in your CICS region, do the following:

1. Specify the system initialization parameter DB2CONN=YES. This is necessary to run the CINQ query transaction and the CICS IA client.
2. If you plan to use VSAM RLS to share the dependency data file and the control record file across multiple regions, specify the system initialization parameter RLS=YES.

3. Set the ICVR system initialization parameter to at least 10 seconds (that is, ICVR=10000 (or a larger value)). If you do not do this, the collector or one of your own transactions may end prematurely with an abend code of AICA.
4. If you want to run the CICS IA client, specify the system initialization parameter TCPIP=YES.
5. If you want to run the CICS IA client, review the system initialization parameter EDSALIM. The client can request large amounts of data. We recommend that you set EDSALIM to 250 M or greater.
6. Add the following to the DFHRPL concatenation in the startup job JCL:
   - CIU.SCIULOAD
   - CIU.SCIULODE
7. Add the following DD statement for the CINT transient data message log:
   ```
   //CINT DD SYSOUT=*
   ```
8. In any region on which you intend to collect DB2 data, ensure that the user ID under which CICS runs has permission to access the SYSIBM.SYSPACKSTMT and SYSIBM.SYSSTMT DB2 tables.

If you want to start and stop CICS IA from the PLT, refer to *CICS Interdependency Analyzer for z/OS User's and Reference Version 2 Release 1*, SC34-6685.

Restarting your CICS region

After all changes have been made, you have to restart your CICS region using the modified CICS startup job. Make sure that the new RDO definitions are installed by using the system initialization parameter START=COLD or START=INITIAL. If this is not possible, install the new RDO group after the CICS warm start with the CEDA INSTALL GROUP(CIU*xx*G21) command, with *xx* being the release qualifier of your CICS.

Customize the DB2 environment

This section describes how to set up the DB2 environment for CICS IA. The following steps have to performed:

1. Create the database tables with the job CIUDBCR.
2. Bind the DBRMs using job CIUDBND.
3. Authorize user IDs to access the plans using CIUDBND and SCIUSQL member CIUGRANT.
4. Define the default and IVP applications using the jobs CIUALOAD and CIUANEW.

Al of the above jobs have been customized and can be found in REDBK23.MIG23T31.SCIUSAMP.

Creating the DB2 tables

Edit and review member REDBK23.MIG23T31.SCIUSQL(CIUMAIN).

Edit, review, and submit job CIUDBCR.

Binding and authorizing the IA application

Edit and review member REDBK23.MIG23T31.SCIUSQL(CIUGRANT).

Edit, review, and submit job CIUDBND.

Creating the default and IVP application definitions

Edit, review, and submit the following jobs in sequence:

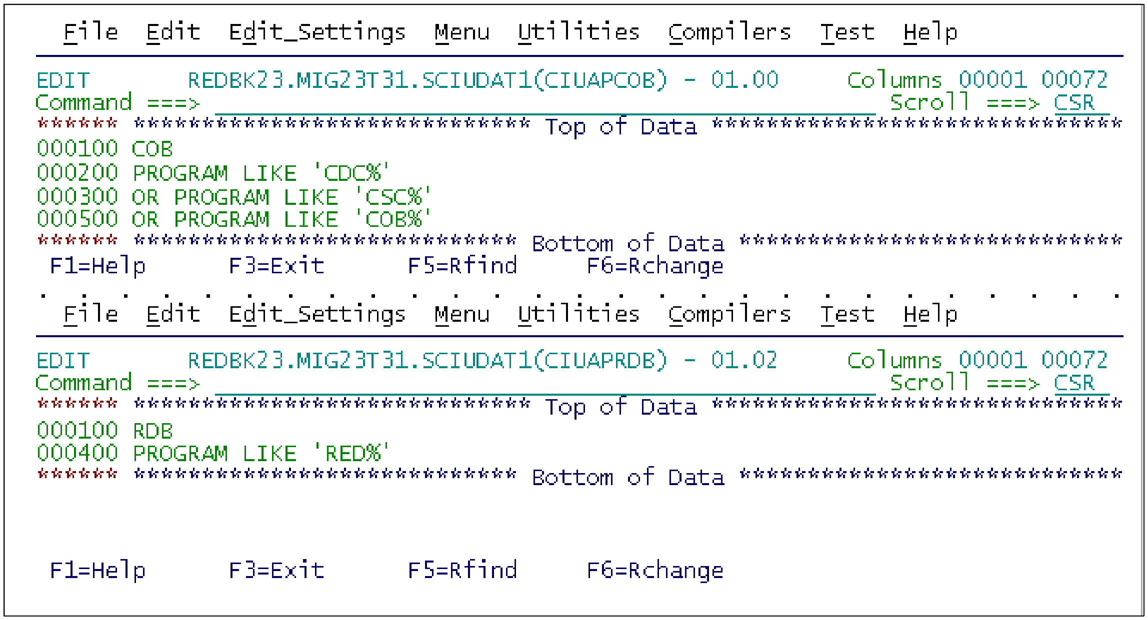
- CIUANEW
- CIUALOAD
- CIUADESC

Creating new applications

For the purpose of this book we create new applications to define the set of programs to be migrated from CICS TS 2.3 to CICS TS 3.1. We create two applications:

- All programs with the prefix RED. The three-character code for this application will be RDB.
- All programs with the prefix COB, CSC, or CDC. The three-character code of this application will be COB.

Create two new members in REDBK23.MIG23T31.SCIUDAT1 called CIUAPRDB and CIUAPCOB. Use CIUAPNEW as a base. Edit these members to represent the two applications mentioned above. See Figure A-8.



```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       REDBK23.MIG23T31.SCIUDAT1(CIUAPCOB) - 01.00     Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ******************************
000100 COB
000200 PROGRAM LIKE 'CDC%'
000300 OR PROGRAM LIKE 'CSC%'
000500 OR PROGRAM LIKE 'COB%'
****** **************************** Bottom of Data ****************************
 F1=Help      F3=Exit      F5=Rfind     F6=Rchange
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       REDBK23.MIG23T31.SCIUDAT1(CIUAPRDB) - 01.02     Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ******************************
000100 RDB
000400 PROGRAM LIKE 'RED%'
****** **************************** Bottom of Data ****************************


 F1=Help      F3=Exit      F5=Rfind     F6=Rchange
```

Figure A-8 IA - Define applications

Edit, customize, and run job CIUANEW to create the new applications. The value that needs editing in this job is _xxx_. We add two new applications. Add steps to create both applications. See Example A-1.

Example: A-1 CIUANEW JCL

```
//INFILE   DD DSN=REDBK23.MIG23T31.SCIUDAT1(&MEMBER),DISP=SHR
//INTEMPL  DD DSN=REDBK23.MIG23T31.SCIUDAT2(CIUATMPL),DISP=SHR
//OUTFILE  DD DSN=REDBK23.MIG23T31.SCIUDAT2(&MEMBER),DISP=SHR
//SYSTSIN  DD DSN=REDBK23.MIG23T31.SCIUCLIS(CIUNEWAP),DISP=SHR
//          PEND
//*
//APPL1    EXEC NEWAPP,MEMBER=CIUAPIVP
//APPL2    EXEC NEWAPP,MEMBER=CIUAPRDB
//APPL3    EXEC NEWAPP,MEMBER=CIUAPCOB
```

Edit, customize, and run job CIUALOAD to load the new data into the DB2 table CIU4_SQL_DATA. The value that requires editing in this job is _xxx_. We add two new applications. Add steps to load SQL data for both applications. See Example A-2.

Example: A-2 CIUALOAD JCL

```
//* CIUAPNON IS ALWAYS REQUIRED
//DSNUPROC.SYSREC DD DISP=SHR,
//     DSN=REDBK23.MIG23T31.SCIUDAT2(CIUAPNON)
//     DD DSN=REDBK23.MIG23T31.SCIUDAT2(CIUAPIVP),DISP=SHR
//     DD DSN=REDBK23.MIG23T31.SCIUDAT2(CIUAPRDB),DISP=SHR
//     DD DSN=REDBK23.MIG23T31.SCIUDAT2(CIUAPCOB),DISP=SHR
//*****************************************************************
//* FOR USER DEFINED APPLICATION DEFINITIONS, THE MEMBERS GENERATED
//* BY THE JOB CIUANEW MUST BE CONCATENATED ABOVE. REMOVE THE
//* ASTERISK AND REPLACE "_xxx_" WITH THE APPLICATION CODE.
```

Create two new members in REDBK23.MIG23T31.SCIUSQL called CIUADRDB and CIUADCOB. Use CIUADNEW as a base. Edit these members to describe the two applications. Figure A-9 on page 418 shows the changes required to describe application RDB.

Edit, customize, and run job CIUADESC to load these descriptions into the DB2 table CIU4_APPLS_DESC. The value that requires editing in this job is _xxx_. We are add two new applications. Add steps to load SQL data for both applications. See Example A-3.

Example: A-3 CIUADESC JCL

```
//SYSIN  DD DSN=REDBK23.MIG23T31.SCIUSQL(CIUAPDEL),DISP=SHR
//       DD DSN=REDBK23.MIG23T31.SCIUSQL(CIUADIVP),DISP=SHR
//***************************************************************
//* FOR USER DEFINED APPLICATION DEFINITIONS, THE MEMBERS CREATED
//* IN DATASET SCIUSQL MUST BE CONCATENATED ABOVE. REMOVE THE
//* '*REM'  AND REPLACE "_xxx_" WITH THE APPLICATION CODE.
//***************************************************************
//       DD DSN=REDBK23.MIG23T31.SCIUSQL(CIUADRDB),DISP=SHR
//       DD DSN=REDBK23.MIG23T31.SCIUSQL(CIUADCOB),DISP=SHR
```

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       REDBK23.MIG23T31.SCIUSQL(CIUADRDB) - 01.00      Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ******************************
000010 --------------------------------------------------------------
000020 -- Template for adding new Application descriptions
000030 --------------------------------------------------------------
000110    insert into   CIU4_APPLS_DESC
000200                  (
000300                  APPLIC_NAME,
000400                  APPLIC_CODE,
000410                  APPLIC_COMMAND
000420                  )
000430                  values
000440                  (
000450                  'Redbook Application 1',
000460                  'RDB',
000470                  'See member SCIUDAT1(CIUAPRDB) for SQL Queuries'
000480                  );
000490    COMMIT;
****** **************************** Bottom of Data ****************************

 F1=Help      F3=Exit      F5=Rfind     F6=Rchange
```

Figure A-9 IA - Application description

Running the installation verification programs

You must perform four steps to verify that the installation was done correctly:

1. The CIUIVPLD job uploads sample interdependency data to the database so you can use the query interface to view it.
2. Create the IVP application. This was done in the previous section.
3. The transaction CIUV verifies that all CICS RDO resources are correctly defined and available.
4. Query the DB2 tables to show which resources are used by the IVP application.

Running the CIUIVPLD job

Edit, review, and run job CIUIVPLD.

Running the CIUV transaction

This installation verification step checks CICS RDO definitions to ensure that all software elements (programs, maps, transactions, files, TD-Queues, and DB2 entries) are correctly defined and available.

In a CICS session clear the screen and run the CIUV transaction. It should end with this message:

```
CIU1002I INSTALLATION VERIFICATION ENDED SUCCESSFULLY
```

The resources used by the IVP application

We will use the CICS IA online query transaction CINQ to show what resources are used by application IVP.

1. In a CICS session clear the screen and run the CINQ transaction.
2. Choose option 1 for CICS resources. See Figure A-10.

```
CIU400        CICS Interdependency Analyzer for z/OS - V2R1M0         2006/06/12
                                   Query Menu                          12:26:41PM

Select one of the following.  Then press Enter.

1    1    Inquire on CICS Resources.
     2    Inquire on DB2 Resources.
     3    Inquire on MQ Resources.
     4    Inquire on IMS Resources.
     5    Additional inquire on DB2 Resources.
     6    Inquire on CICS Affinities.









CICS Sysid:  RB23   CICS Applid:  REDBKV23   TermID:  CP89
CIU7000I 5697-J23 (C) Copyright IBM Corp. 2001,2005
F1=Help      F2=          F3=Exit      F4=          F5=          F6=
F7=          F8=          F9=          F10=         F11=         F12=Exit
```

Figure A-10 CINQ - Query Menu

3. Enter `IVP` for the application and select Y for resource details. See Figure A-11.

```
CIU500        CICS Interdependency Analyzer for z/OS - V2R1M0        2006/06/12
                           CICS Query Menu                            12:27:10PM

      Select the resource type to query:    _

      1. Transactions
      2. Programs
      3. TSQs
      4. TDQs
      5. Maps
      6. Files
      7. Applications
      8. Regions

      OR display all resources in application IVP   detailed Y (Y/N)

      Enter the application's 3 character code or
      ? for a list of applications available.
      WARNING : Option 7 may take a long time.

CICS Sysid:  RB23   CICS Applid:  REDBKV23   TermID:  CP89

F1=Help      F2=          F3=End       F4=Exit      F5=          F6=
F7=          F8=          F9=          F10=         F11=         F12=End
```

Figure A-11   CINQ - CICS Query Menu

Figure A-12 shows the first page of output for the query "Which resources are in IVP application."

```
CIU585       CICS Interdependency Analyzer for z/OS - V2R1M0        2006/06/12
 For your CICS Query                                                12:27:38PM
 WHICH RESOURCES ARE IN IVP Application                         Page   1 of  34

 In  Tran Program  Resource Resource Resource
Regn               Function Type     Name
T304 EQAC EQZ3ACTL RETURN   TRANSID  EQAC
     EQAC EQZ3ACTL RECEIVE  MAP      EQZMA01
     EQAC EQZ3ACTL SEND     MAP      EQZMA01
     EQAC EQZ3ACTL LOAD     PROGRAM  EQZTSCT
     EQAC EQZ3ACTL XCTL     PROGRAM  EQZ3SSUP
     EQAC EQZ3ACTL XCTL     PROGRAM  EQZ3SWCF
     EQAC EQZ3ACTL ASSIGN   APPLID   IYCYZC44
     EQAC EQZ3SSUP START    TRANSID  EQSS
     EQAC EQZ3SSUP RECEIVE  MAP      EQZMB01
     EQAC EQZ3SSUP SEND     MAP      EQZMB01
     EQAC EQZ3SSUP READ     FILE     EQZMSGS
     EQAC EQZ3SSUP LOAD     PROGRAM  EQZTSCT
     EQAC EQZ3SSUP XCTL     PROGRAM  EQZ3ACTL

 CICS Sysid:  RB23   CICS Applid:  REDBKV23   TermID:  CP89

 F1=          F2=          F3=End       F4=Exit      F5=          F6=
 F7= Page Up  F8= Page Down F9=         F10=         F11=         F12=End
```

Figure A-12 CINQ - Output for IVP application query

Configuring the CICS IA client interface

The section describes how to configure the client/server interface in a CICS TS 2.3 environment. Perform the following steps:

1. Install the server in a CICS TS 2.3 region.
2. Install Eclipse on your workstation.
3. Install the CICS IA client.
4. Configure the CICS IA client.
5. Verify that the client works.

Installing the server in a CICS TS 2.3 region

To create the CICS IA BTS repository you must edit and run the customized job REDBK23.MIG23T31.SCIUSAMP(CIUJCLCD).

To define the CICS IA server resources to CICS you must edit and run the customized job REDBK23.MIG23T31.SCIUSAMP(CIUJCLC2).

> **Note:** The TCPIPSERVICE port number will be used later when configuring the client. Make a note of it.

Installing Eclipse on your workstation

In order to run the client you must download Eclipse on to your workstation. Perform the following steps:

1. Download the compressed zip file Eclipse Version 3.1 or later from:

   `http://www.eclipse.org/downloads/`

   See Figure A-13.

2. Extract the files to the base c:\ directory. The extraction process generates a top-level directory called eclipse.

3. Create a start menu link or desktop shortcut to the Eclipse.exe file in c:\eclipse.

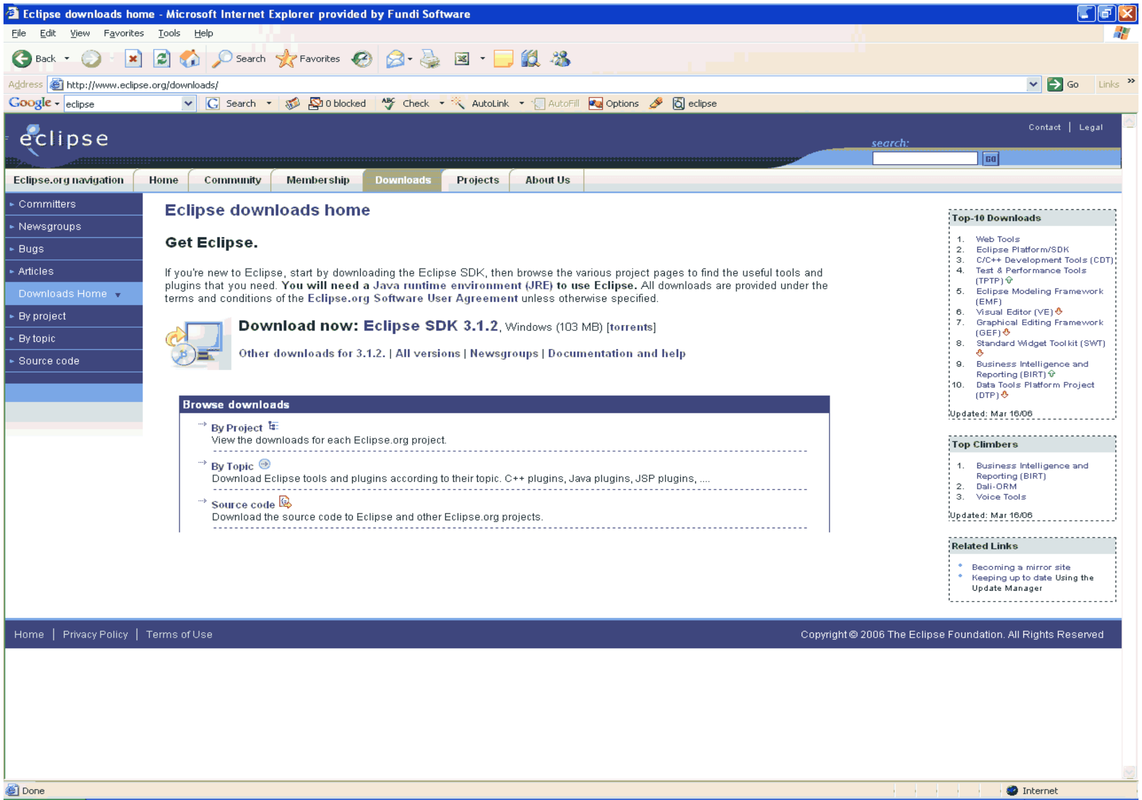


Figure A-13 Eclipse home page

Installing the CICS IA client

Use FTP to download the CICS IA plug-in zip file from the CIU.SCIUJAVE(CIUJCLNT) data set in binary format:

1. Download the compressed zip file, Eclipse Version 3.1 or later, from:

   `http://www.eclipse.org/downloads/`

   See Figure A-14 on page 424.

2. Extract the contents of the CICS IA zip file to the c:\eclipse\plugin directory.
3. Check that the plug-in has been installed successfully.
   a. Open the Eclipse workbench and select **Window** → **Preferences** from the menu.

b. In the Preferences dialog, click **Plug-in Development** → **Target Platform**. A list of all of the plug-ins that have been installed is displayed. See Figure A-14. Check that the com.ibm.cicsia(2.1.0) is in the list and is selected.

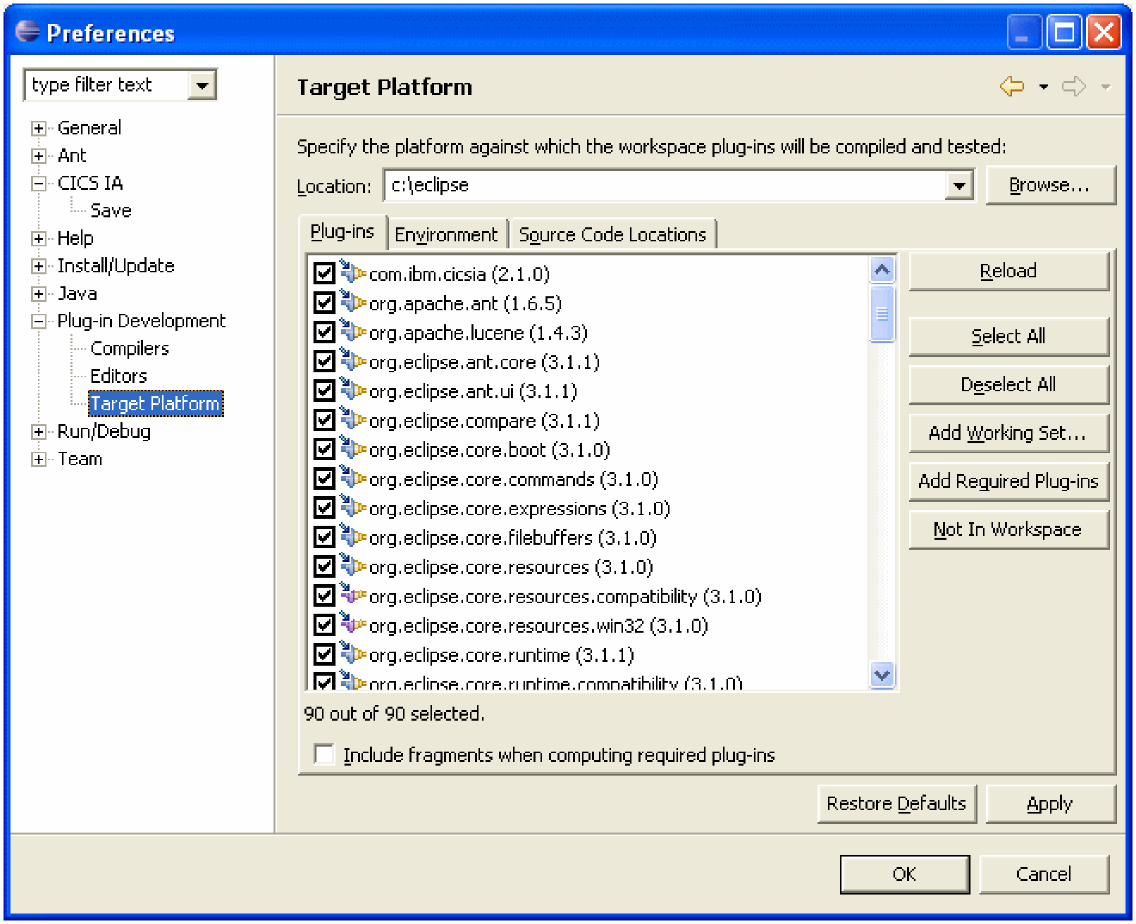


Figure A-14 IA Eclipse plug-in preferences

Configuring the IA client

In order to connect to the CICS IA server we need to tell the client which server and port number to use. In this example the server address is 172.17.69.25 and the port number is 10329.

1. Select **Window** → **Preferences** → **CICS IA** in the CICS IA in the Eclipse workbench. The CICS IA plug-in Preferences dialog appears. See Figure A-15.
2. Enter the CICS IA server target address. In this case it is HTTP://172.17.69.25:10329/CICS/CWBA/CIUWSDSH.
3. Select other preferences as required.

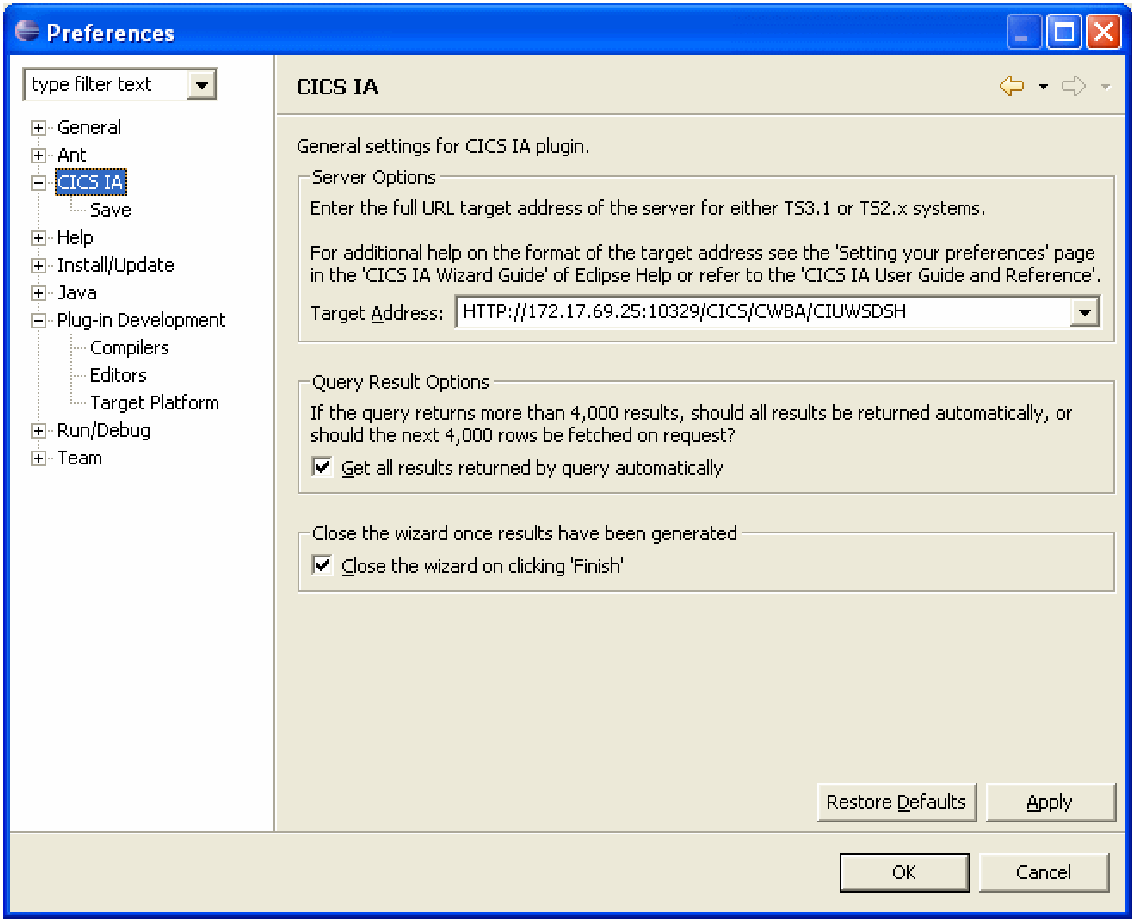


Figure A-15 IA Eclipse Preferences

Verifying the client

We will verify that the client works by asking it to display all resources used by application IVP.

1. From within the Explorer View in Eclipse expand **CICS Resources** and select **TRANSACTIONS**. See Figure A-16.

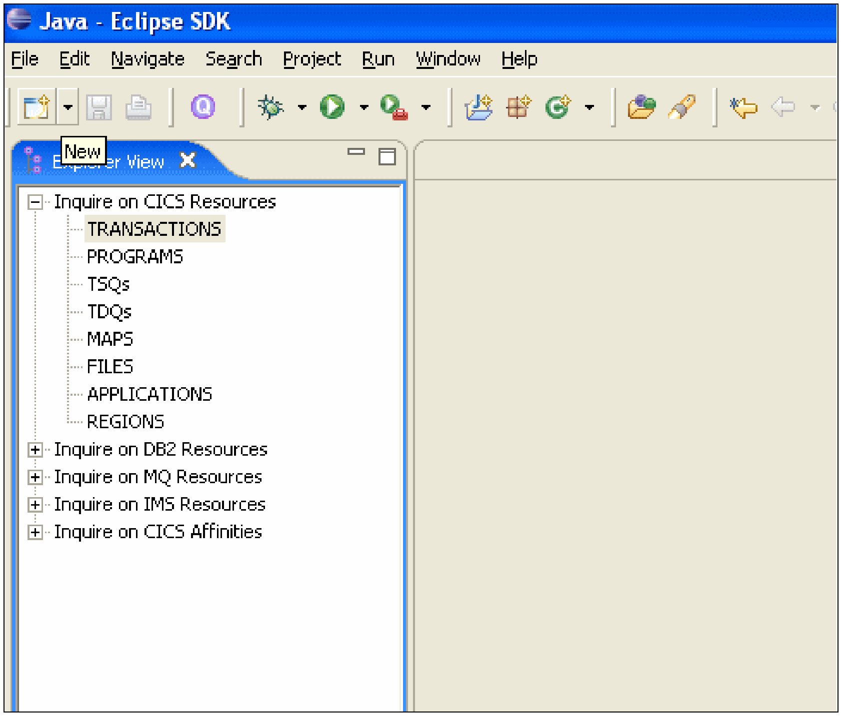


Figure A-16 IA Eclipse - Select CICS transactions

2. Enter `IVP` for the application and select **Next**. See Figure A-17.

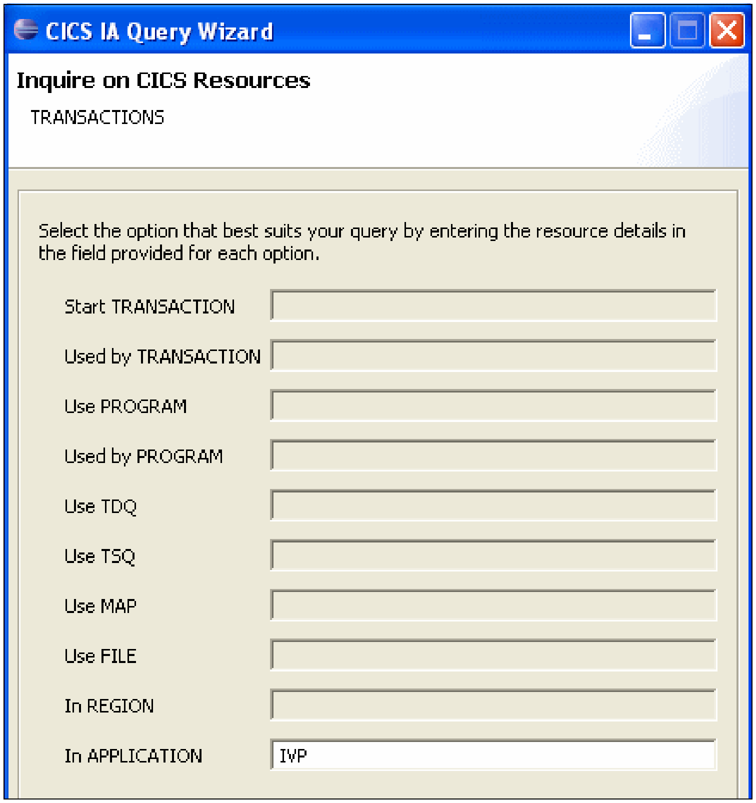


Figure A-17 IA plug-in - Select IVP application

3. Customize the required columns, as in Figure A-18.

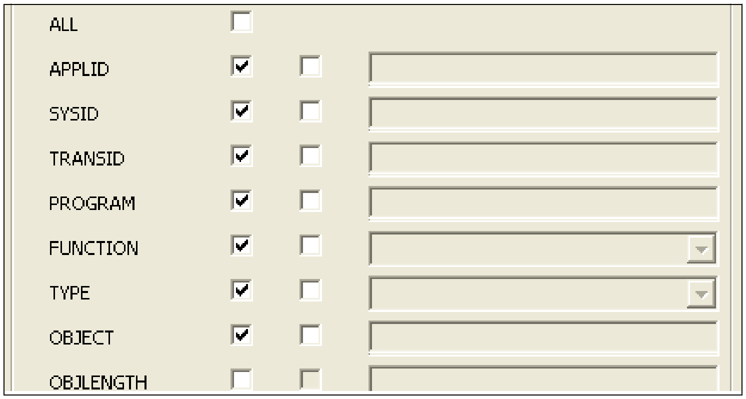
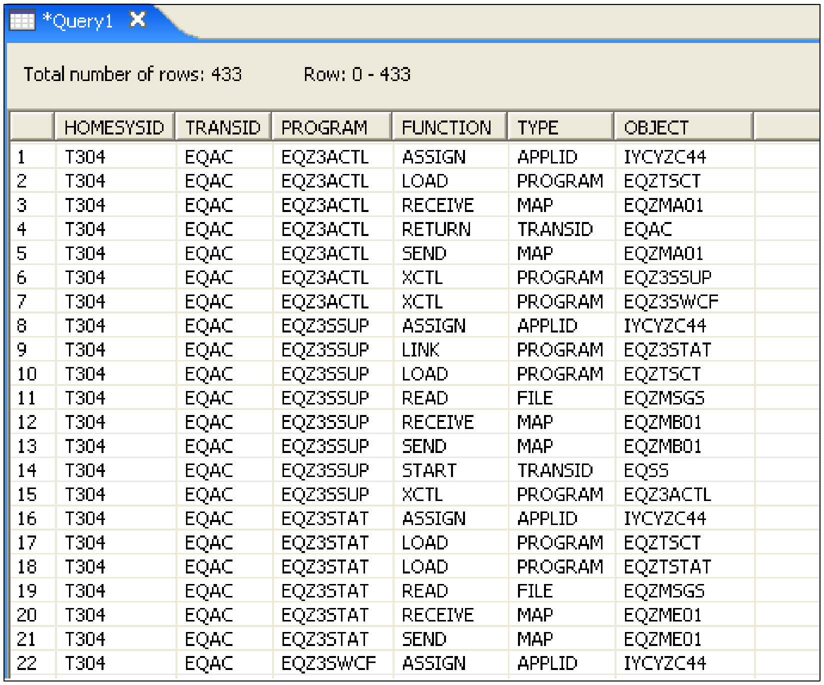


Figure A-18 IA plug-in - Select columns

Figure A-19 shows the first 22 rows of output.



	HOMESYSID	TRANSID	PROGRAM	FUNCTION	TYPE	OBJECT
1	T304	EQAC	EQZ3ACTL	ASSIGN	APPLID	IYCYZC44
2	T304	EQAC	EQZ3ACTL	LOAD	PROGRAM	EQZTSCT
3	T304	EQAC	EQZ3ACTL	RECEIVE	MAP	EQZMA01
4	T304	EQAC	EQZ3ACTL	RETURN	TRANSID	EQAC
5	T304	EQAC	EQZ3ACTL	SEND	MAP	EQZMA01
6	T304	EQAC	EQZ3ACTL	XCTL	PROGRAM	EQZ3SSUP
7	T304	EQAC	EQZ3ACTL	XCTL	PROGRAM	EQZ3SWCF
8	T304	EQAC	EQZ3SSUP	ASSIGN	APPLID	IYCYZC44
9	T304	EQAC	EQZ3SSUP	LINK	PROGRAM	EQZ3STAT
10	T304	EQAC	EQZ3SSUP	LOAD	PROGRAM	EQZTSCT
11	T304	EQAC	EQZ3SSUP	READ	FILE	EQZMSGS
12	T304	EQAC	EQZ3SSUP	RECEIVE	MAP	EQZMB01
13	T304	EQAC	EQZ3SSUP	SEND	MAP	EQZMB01
14	T304	EQAC	EQZ3SSUP	START	TRANSID	EQSS
15	T304	EQAC	EQZ3SSUP	XCTL	PROGRAM	EQZ3ACTL
16	T304	EQAC	EQZ3STAT	ASSIGN	APPLID	IYCYZC44
17	T304	EQAC	EQZ3STAT	LOAD	PROGRAM	EQZTSCT
18	T304	EQAC	EQZ3STAT	LOAD	PROGRAM	EQZTSTAT
19	T304	EQAC	EQZ3STAT	READ	FILE	EQZMSGS
20	T304	EQAC	EQZ3STAT	RECEIVE	MAP	EQZME01
21	T304	EQAC	EQZ3STAT	SEND	MAP	EQZME01
22	T304	EQAC	EQZ3SWCF	ASSIGN	APPLID	IYCYZC44

Figure A-19 IA plug-in - sample output

B

Migrating from CICS TS 1.3 considerations

In this appendix we discuss the new, changed, and removed functionality when migrating from CICS TS 1.3 to CICS TS 3.1. The areas we focus on are:

- Software prerequisites
- General external changes
- RDO
- Application and systems programming interfaces
- Global user exits
- Monitoring and statistics
- Functional changes
- Language Environment
- Obsolete function removal

CICS Transaction Server V3.1 elements

CICS Transaction Server V3.1 Elements

- **CICS 0640**
- **CICSPlex SM 310**
- **CICS Information Center**
- **REXX Development System and Runtime Facility for CICS/ESA**
- **CICS Application Migration Aid Version**
- **CICS Integrator Adapter for z/OS**
- **WebSphere Studio Enterprise Developer V5.1 promotion**
  - 1 unrestricted entitlement, no service entitlement
  - Integrated development environment for CICS and WebSphere
    - COBOL, PL/I, Java for CICS and J2EE applications

Figure B-1 CICS TS 3.1 elements

When you buy CICS Transaction Server V3 R1 you get the following

- CICS Functional level CICS TS 3.1 (Internal level CICS 0640)
- ONC RPC support, CICS Web interface, CICS DB2 attachment facility, CICS/DDM
- CICSPlex SM at functional level CICS TS 3.1

  This is updated to support new levels of function in CICS. CICSPlex SM becomes an exclusive element in CICS TS Release 3. IBM CICSPlex System Manager for MVS/ESA Version 1 Release 3 continues to be available for customers who are not yet ready to migrate to CICS TS (for example, a customer with CICS/ESA Version 4 Release 1 or earlier).
- Application Migration Aid at functional level CICS TS V1 R1

  First available in 1990, this element is still available stand-alone as IBM Customer Information Control System (CICS) program offering, CICS Application Migration Aid, program number 5695-061.
- REXX for CICS at functional level CICS TS V1 R2 (REXX for CICS/ESA V1R1)

This is separately available as REXX for CICS, program number 5655-B54.

- IBM CICS Integrator Adapter for z/OS

  This is the server runtime environment to those adapter services that are modelled, generated, and deployed using the Service Flow Modeller plug-in of the WebSphere Developer for zSeries product.

- One unrestricted entitlement to WebSphere Studio Enterprise Developer V5

Software prerequisites

The prerequisites are:

- z/OS V1.4 or later
  - CICS will not initialize unless the minimum prerequisite level of the operating system is installed.
  - Some components of CICS are installed in PDSE and HFS files:
    - The OMVS address space, UNIX Systems Services, must be active in full-function mode during the install process.
    - The jobs to create the HFS files and directories require superuser authority.
  - LE library SCEERUN must be available to CICS during CICS initialization.
  - z/OS Conversion Services must be enabled.
- IBM SDK for z/OS, Java 2 Technology Edition Version 1.4

  This must be at the 1.4.2 level. PTF UQ90449.

The CICS installation process does not alter if you have data conversion requirements.

However, to get the benefits of z/OS conversion services, if your system requires support for the conversion of UTF-8 or UTF-16 data to EBCDIC, you must enable the z/OS conversion services and install a conversion image that specifies the conversions that you want CICS to perform.

Refer to the instructions in the *z/OS Support for Unicode: Using Conversion Services manual*, SA22-7649, to see the steps needed to set up and configure conversions supported though the operating system services.

CICS TS 3.1 requires the IBM Software Developer Kit for z/OS, Java 2 Technology Edition Version 1.4.2. The 1.4.2 level is available by applying PTF UQ90449.

Optional software minimum levels

For WS-Security support, the IBM XML Toolkit for z/OS V1.7 is required. This is a no-charge product, program number 5655-J51.

The following levels of other products are supported for use with CICS TS for z/OS Version 3.1:

- IMS Database Manager V7 (5655-B01), IMS Database Manager V8 (5655-C56), IMS Database Manager V9 (5655-J38).
- DB2 Universal Database Server for OS/390 V6.1 (5645-DB2). For SQLJ/JDBC support, with PTF for APAR PQ84783 DB2 V6 does not support DB2 Group Attach.
- DB2 Universal Database Server for OS/390 V7.1 (5675-DB2). For SQLJ/JDBC support, with PTFs for APARs PQ84783 and 86525. For DB2 Group Attach, with APARs PQ44614, PQ45691, and PQ45692.
- DB2 Universal Database for z/OS V8.1 (5625-DB2). For SQLJ/JDBC support, with PTFs for APARs PQ84783 and 86525.
- WebSphere MQ for z/OS V5.3 (5655-F10).
- Tivoli Decision Support for OS/390 (5698-ID9) V1.6, with necessary service applied.
- Tivoli Business Systems Manager V3.1.
- CICS Universal Client Version 5.0 or later.
- CICS Transaction Gateway Version 5.0 or later.

Installation process

This release of CICS Transaction Server is installed using the SMP/E RECEIVE, APPLY, and ACCEPT commands. The SMP/E dialogs may be used to accomplish the SMP/E installation steps.

The process is described in the CICS TS 3.1 program directory. It is in line with IBM Corporate Standards, and may be familiar to those who have installed other z/OS products.

The traditional method of installing the CICS Transaction Server, DFHISTAR, is still available.

New SIT parameters for CICS TS 3.1

Systems Initialization Table: New parameters

- **AIBRIDGE**
  - Specifies if the autoinstall URM is to be called for 3270 bridge facilities
- **BRMAXKEEPTIME**
  - Specifies how long idle bridge facilities are to be maintained
- **CLINTCP**
  - Default client code page when DFHCNV CLINTCP=SYSDEF
- **CRLPROFILE (PK04622)**
  - Name of a profile in the RACF LDAPBIND class
- **DEBUGTOOL**
  - Specifies if debugging profiles will be used to select programs
- **EJBROLEPRFX**
  - Prefix to qualify the security role in an EJB deployment descriptor

Figure B-2 SIT new parameters

The default values for these parameters are designed to have minimal impact when you are migrating from an earlier release of CICS.

- AIBRIDGE={AUTO|YES} specifies whether the autoinstall user-repaceable program (URM) is called for bridge facilities (YES) or whether they are defined automatically by CICS (AUTO).
- BRMAXKEEPTIME={86400|timeout} specifies the maximum time in seconds that bridge facilities are kept when unused, with a default value (and upper limit) of 24 hours.
- CLINTCP={437|codepage} specifies the default client code page to be used by the DFHCNV data conversion table, but only if the CLINTCP parameter in the DFHCNV macro is set to SYSDEF.
- CRLPROFILE is the 246-character name of a profile in the RACF LDAPBIND class that contains bind information about an LDAP server that will be used by CICS SSL support to obtain certificate revocation list information.
- DEBUGTOOL={NO|YES} specifies whether debugging profiles will be used to select programs that will run under the control of a debugging tool.
- EJBROLEPRFX=ejbrole-prefix specifies a prefix to qualify the security role defined in an enterprise bean's deployment descriptor.

Systems Initialization Table: New parameters…

- **IIOPLISTNER**
  - Specifies if this region is an IIOP listener
- **INFOCENTER**
  - Universal Resource Locator for the CICS Information Center
  - Used to provide help information for the web based application debugging profile manager
- **JVMCCPROFILE**
  - JVM profile for the master JVM that initializes the shared class cache
- **JVMCCSIZE**
  - Size of the shared class cache

Figure B-3 New SIT parameters

- IIOPLISTNER={YES|NO} specifies whether the CICS region is to function as an IIOP listener region.
- INFOCENTER {infocenter_url} specifies the Universal Resource Locator (URL) of the root of the CICS Information Center directory structure.
- JVMCCPROFILE {DFHJVMCC|profile} specifies the JVM profile to be used for the master JVM that initializes the shared class cache.
- JVMCCSIZE {24M|number} specifies the size of the shared class cache on an initial or cold start of CICS.

Systems Initialization Table: New parameters…

- **JVMCCSTART**
  - Specifies how the shared class cache is to be started
    - Auto: at CICS initialization
    - Yes: at first JVM request
    - No: by CEMT PERFORM CLASSCACHE START
- **JVMLEVEL0TRACE, JVMLEVEL1TRACE, JVMLEVEL2TRACE**
  - Specifies the default level for JVM level 0. 1 and 2 tracing, corresponds to CICS SJ trace levels 29-31
- **JVMUSERTRACE**
  - Specifies the default level for JVM user tracing, corresponds to CICS SJ trace level 32
- **JVMPROFILEDIR**
  - Specifies the HFS directory that contains the CICS JVM profiles

Figure B-4 New SIT parameters

- JVMCCSTART= {AUTO|YES|NO} determines whether the shared class cache is started during CICS initialization and sets the status of autostart for the shared class cache.
- JVMLEVEL0TRACE, JVMLEVEL1TRACE, JVMLEVEL2TRACE, JVMUSERTRACE {option} specify the default options for the JVM trace levels.
- JVMPROFILEDIR={/usr/lpp/cicsts/cicsts23/JVMProfiles|directory} specifies the name of an HFS directory that contains the JVM profiles for CICS.

Systems Initialization Table: New parameters…

- **KEYRING**
  - Specifies the name of the key king in RACF
- **LOCALCCSID**
  - Default CCSID for local region
- **MAXJVMTCBS**
  - Specifies the maximum number of J8 and J9 TCBs
    - Minimum value now 1
    - Master JVM (JM) does not count towards MAXJVMTCBS
- **MAXSOCKETS**
  - Specifies the number of sockets for the CICS sockets domain
- **MAXSSLTCBS**
  - Maximum number of S8 TCBs for use with SSL

Figure B-5 New SIT parameters

- KEYRING=keyring_name specifies the name of the key ring defined in the security manager's database (for example, as defined by the RACF RACDCERT ADDRING command).
- LOCALCCSID={037|CCSID} specifies the default CCSID for the local region. The CCSID is a value of up to eight characters. If the CCSID value is not specified, the default LOCALCCSID is set to 037.
- MAXJVMTCBS={5|number} specifies the maximum number of open TCBs that CICS can create in the pool of J8-mode and J9-mode TCBs for use by Java programs that run in a JVM (the JVM pool). Within this limit, there are no constraints on how many of the TCBs in the JVM pool are J9 TCBs and how many are J8 TCBs.
- MAXSOCKETS={65535|number} specifies the maximum number of IP sockets that can be managed by the CICS sockets domain. Note that the default value, and any explicit value, is conditional upon the authorization of the CICS region user ID. If the user ID is not defined to UNIX system services as a superuser, the default is restricted to the value specified on the MAXFILEPROC parameter in the BPXPRMxx of SYS1.PARMLIB.
- MAXSSLTCBS={8|number} specifies the maximum number of S8 TCBs that can run in the SSL pool. The default is 8, but you can specify up to 1024 TCBs.

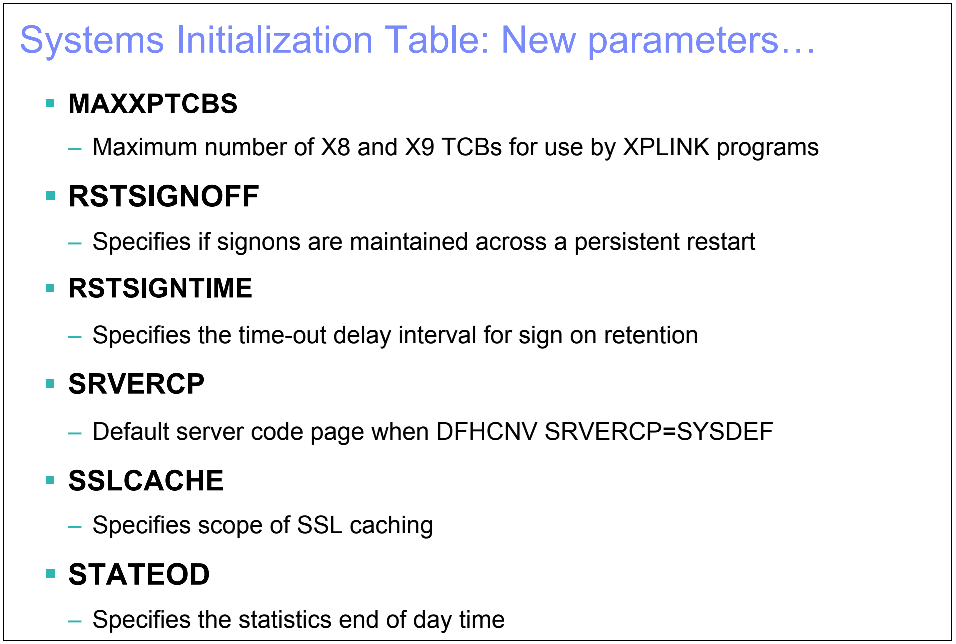


Figure B-6 New SIT parameters

- MAXXPTCBS={5|number} specifies the maximum number, in the range 1 through 999, of open X8 and X9 TCBs that can exist concurrently in the CICS region.
- RSTSIGNOFF={NOFORCE|FORCE} specifies whether all users signed on to a CICS region are allowed to remain signed on following a persistent session restart or an XRF takeover (XRFSOFF is obsolete).
- RSTSIGNTIME={500|hhmmss} specifies the time-out delay interval for sign-on retention during a persistent session restart or an XRF takeover (XRFSTIME is obsolete).
- SRVERCP={037|codepage} specifies the default server code page to be used by the DFHCNV data conversion table, but only if the SRVERCP parameter in the DFHCNV macro is set to SYSDEF.
- SSLCACHE={CICS| SYSPLEX} specifies whether SSL is to use the local or sysplex caching of session IDs.
- STATEOD={0|hhmmss} specifies the end-of-day time for interval statistics collection.
- STATINT={030000|hhmmss} specifies the interval for statistics collection.
- XEJB={YES|NO} specifies whether support of security roles is to be enabled.

Systems initialization table: changed parameters

ENCRYPTION specifies the cipher suites that CICS uses for secure TCP/IP connections. When a secure connection is established between a pair of processes, the most secure cipher suite supported by both is used.

- Use ENCRYPTION=STRONG when you can tolerate the overhead of using high encryption if the other system requires it.
- Use ENCRYPTION=WEAK when you want to use encryption up to 40 bits in length.
- Use ENCRYPTION=MEDIUM when you want to use encryption up to 56 bits in length.

For compatibility with previous releases, ENCRYPTION=NORMAL is accepted as an equivalent to ENCRYPTION=MEDIUM. CICS can use only the cipher suites that are supported by the underlying z/OS or OS/390 operating system.

FORCEQR specifies whether you want CICS to force all CICSAPI user application programs that are specified as threadsafe to run under the CICS QR TCB, as though they were specified as quasi-reentrant programs. This parameter applies to all application programs that are restricted to the current CICS programming interfaces (that is, those that specify API(CICSAPI)) and does not apply to any of the following:

- Java programs that are run in a JVM
- C/C++ programs using XPLINK
- OPENAPI programs

None of these can run on the QR TCB.

- EDSALIM: The default value changed to 30 M.
- LGDFINT: The default value changed to 5 ms.

In earlier releases, MAXOPENTCBS={12|number} applies to all open mode TCBs controlled by the CICS dispatcher domain. The range is changed to 12 through 2000, and it now applies to L8 and L9 mode open TCBs only, which are reserved for use by task-related user exits that are enabled with the OPENAPI option. This includes the CICS DB2 adaptor when CICS connects to DB2 Version 6 or later.

MNFREQ={0|hhmmss}: The minimum time value you can specify is reduced from 15 minutes to one minute, giving a range of 000100–240000 (instead of 001500–240000).

SPCTRxx and STNTRxx: The new domain codes are available for the xx codes in the keyword:

- EJ Enterprise Java domain
- IE ECI over TCP/IP domain
- II IIOP domain
- OT Object transaction services domain
- PI Pipeline domain
- PT Partner management domain
- RZ Request streams domain
- SJ JVM domain

Systems initialization table: obsolete parameters

In this section we review the obsolete parameters.

MAXHPTCBS

Runtime support for Java program objects and hot-pooling (HPJ) has been removed. The system initialization parameter MAXHPTCBS is not required, and is removed. The open TCB mode H8, which was used for hot-pooling Java program objects and was controlled by MAXHPTCBS, no longer exists.

SSLTCBS

This parameter is now obsolete and is only kept for compatibility. If it is specified, it is rejected with a message and MAXSSLTCBS is assumed.

TCAM

This parameter is now obsolete and is only kept for compatibility. If it is specified, it is rejected with a message and TCAM=NO is assumed.

DCT

The destination control table is no longer supported, and all transient data queues must be defined to CICS in the CSD using the TDQUEUE resource type. You can use the old DFHDCT macros for migration purposes only to enable you to migrate your DCT entries to the CSD using the DFHCSDUP MIGRATE command.

KEYFILE

This is replaced by the KEYRING system initialization parameter.

MNEVE

CICS event class monitoring is replaced by support for the MVS workload manager, making MNEVE obsolete.

TCAM

This parameter is now obsolete and is only kept for compatibility. If it is specified, it is rejected with a message and TCAM=NO is assumed.

XRFSOFF

This is replaced by RSTSIGNOFF.

XRFSTIME

This is replaced by RSTSIGNTIME.

CICS-supplied transactions

In this section we discuss CICS-supplied transactions.

Changes to CWXN (Web attach transaction)

There are several changes to the processing carried out by the CICS-supplied transaction CWXN, the Web attach transaction. The most significant of these are:

- If a matching URIMAP definition is found for an HTTP request, CWXN now invokes the analyzer program only if instructed to do so by the URIMAP definition.
- Where the HTTP version of the request is HTTP/1.1, CWXN carries out some of the responsibilities of an HTTP server by performing basic acceptance checks on the request. In response to these checks, CWXN might take action to return a response to the request without involving a user-written application program.
- CWXN pre-processes chunked and pipelined messages received from a Web client so that user-written applications do not have to perform this processing.
- Chunked messages are single messages split up and sent as a series of smaller messages (chunks). CWXN receives and assembles the chunks of the message to create a single HTTP request. CWXN checks that the message is complete before passing it to the user application. The user application can then process the request like any other HTTP request.
- Pipelined messages are multiple messages sent in sequence, where the sender does not wait for a response after each message sent. A server must

respond to these messages in the order in which they are received. To ensure this, CWXN holds pipelined requests and releases them one at a time to the user application. The user application must send a response to the first request before receiving the next request from CWXN.

- Persistent connections are now the default behavior. The connection is only closed if the Web client requests closure, if the timeout period is reached, or if the Web client is an HTTP/1.0 client that does not send a keep-alive header.

New CICS-supplied transactions

CCRL, the certificate revocation lists transaction, is used to create and update the certificate revocation lists (CRLs) that are stored in an LDAP server. You only need to use CCRL if you are implementing SSL in your CICS regions and want each connection checked for a revoked certificate during the SSL handshake.

CPIH is the internal alias transaction for inbound Web Services over http.

CPIR is the internal alias transaction for inbound Web Services using WMQ.

In CICS Transaction Server for z/OS Version 3 Release 1, processing for HTTP requests and processing for non-HTTP requests is kept separate. This ensures that CICS can perform basic acceptance checks on HTTP requests and responses, and that non-HTTP requests are not subjected to these checks. Processing for non-HTTP requests must now be carried out under the user-defined (USER) protocol, which is specified on the TCPIPSERVICE definition for the port that receives the requests.

The new CICS-supplied transaction CWXU, the CICS Web user-defined protocol attach transaction, is the default when the protocol is defined as USER. CWXU executes the CICS program DFHWBXN.

CIRR is the default CICS IIOP request receiver transaction.

CJMJ is the transaction that the CICS master JVM runs under.

CJTR is the CICS Object Transaction Service (OTS) resynchronization transaction.

New CEMT command options

The new options are:

- INQUIRE/SET/DISCARD
  - CORBASERVER
  - DJAR
  - PIPELINE

- URIMAP
  - WEBSERVICE
- DISCARD
  - PIPELINE
  - URIMAP
  - WEBSERVICE
- PERFORM
  - PIPELINE SCAN
  - CORBASERVER SCAN

CEMT supports the standard inquire, set, and discard commands for the new pipeline, urimap, and Web Service resources.

A perform pipeline command initiates a scan of the Web Service binding directory that is specified in the WSBIND attribute of a pipeline definition.

A perform corbaserver scan command initiates a scan of the pickup directory that is specified in the DJARDIR attribute of a corbaserver definition.

Changed CEMT command options

The options are:

- INQUIRE

  Dispatcher, Doctemplate, Program, System, Tcpip, Tcpipservice, Workrequest

- SET

  Dispatcher, Doctemplate, Program, System, Tcpipservice, Workrequest

- PERFORM

  Statistics, Corbaserver, Djar, Pipeline

- INQUIRE SYSTEM
  - CICSTSLEVEL returns 030100
  - RELEASE returns 0640

In terms of obsolete options, inquire/set commands for dispatcher and program have any parameters relating to Java hotpooling and HP TCBs removed.

The dispatcher command now has new parameters relating to XP TCBS for XPLINK. An SSL TCBs.

For programs, a new APIST keyword shows whether the program is defined as OPENAPI or CICSAPI. The existing RUNTIME keyword has a new value of XPLINK.

For DOCTEMPLATES, the HFSFILE keyword returns the full-qualified name of the HFS file where the template resides. TCP/IP commands support the new CRLSERVER and SSLCACHE keywords.

A new MAXDATALEN parameter for TCPIPSERVICE specifies the maximum length of data that may be received by CICS as an HTTP server as a result of upgrading our support to HTTP 1.1.

Statistics now support the new Pipeline and WebService resources.

The CICS level number in CICS TS 3.1 is 0640. This number is returned in the RELEASE parameter of the INQUIRE SYSTEM command. The 0640 number also appears in other forms such as 6.4.0 in output from offline utilities such as statistics and dump formatters to identify the level of utility being used, and as the suffix in module names such as DFHPD640.

- CETR

  New activate trace options

  - BR, DP, EJ, IE, II, OT, PI, PT, RZ, SJ
  - Java tracing options

- CEDA changes

  - Multi-line fields

    - All the line entries must be available to edit the fields
    - For example, SHELF, HOST

  - Mixed Case fields

- New CEOT commands

  Temporarily alter the uppercase translation for your terminal

    - UCTRAN, NOUCTRAN, TRANIDONLY

The CETR transaction is enhanced to enable you to set special tracing for the following new components:

BR Bridge domain, DP Debug Tool Interface domain, EJ Enterprise Java domain, IE ECI over TCP/IP domain, II IIOP domain, OT Object transactions services domain, PI Pipeline Manager domain, PT Partner management domain, RZ Request streams domain, SJ CICS JVM domain.

Controlling tracing for JVMs CETR has new option screens to display and update trace settings for JVMs. Press PF6 on the main screen to access the JVM trace options screens. (Although the JVM trace options are part of the SJ component, they are controlled using the JVM trace options screens, rather than the component trace options screen.)There are new options added to the CEOT transaction that allow you to alter the uppercase translation status (UCTRAN) for your own terminal, for the current session only. The new keywords are NOUCTRAN, UCTRAN, or TRANIDONLY. These new options enable to switch between the uppercase translation options as required. For example, you might need to switch off uppercase translation temporarily while you use CEDA to define some resource definitions that require mixed-case attribute values.

CEDA is often used in circumstances where the CICS system, or the particular terminals, are defined so that all input is folded (or translated) to upper case. Web support and Enterprise bean support introduced some resource definition parameters which must be entered in lower case or mixed case, because their values must match those in other systems, where the use of lower case fields is commonplace.

To enable you to input lower case and mixed case values, you were advised in CICS TS V2 to use CEOT to set NOUCTRAN before entering such input. CEDA now knows about these fields. It will observe the setting of UCTRAN, but if upper case translation is in effect CEDA will not alter the fields listed below. They are the ones where input may need to be kept in the case that you entered.

There are new options added to the CEOT transaction that allow you to alter the uppercase translation status (UCTRAN) for your own terminal, for the current session only. The new keywords are NOUCTRAN, UCTRAN, or TRANIDONLY. These new options enable to switch between the uppercase translation options as required. For example, you might need to switch off uppercase translation temporarily while you use CEDA to define some resource definitions that require mixed-case attribute values.

Resource definition

We discuss the resource definition in this section.

CICS System Definition (CSD)

Run the DFHCSDUP utility program, specifying the UPGRADE command, to upgrade the CICS-supplied definitions in your CSD to the latest CICS TS level. You can create a new CSD using the DFHCSDUP INITIALIZE command.

- Define new CSD.
- REPRO existing CSD to new data set.
- Run DFHCSDUP UPGRADE.
  - Use the DFHCSDUP SCAN command to check for user changes.
  - Review the CEE group.
- Share the CSD.

– CICS TS 3.1 CSD can be shared with prior releases. There is no requirement for a DFHCOMPx group to share with CICS TS 2.3

Upgrading other IBM-supplied resource definitions

If you have resource definitions in your CSD that support other IBM products, you may need to upgrade these also. For example, if your Language Environment resource definitions are not at the z/OS Version 1 Release 4 level, we recommend that you delete and replace the CSD group containing these.

You can find the Language Environment resource definitions in the SCEESAMP library in member CEECCSD.

Obsolete IBM-supplied resource groups

In this section we discuss obsolete IBM-supplied resource groups.

DFH$JAVA

IBM-supplied sample application program group DFH$JAVA is removed. This group contained the resource definitions needed for the sample applications for Java support using VisualAge for Java, Enterprise Edition for OS/390. The same sample applications are defined for use with a JVM by the DFH$JVM group.

DFHAUGRP

IBM-supplied group DFHAUGRP is removed. This group contained the resource definitions for the CICS transaction affinities utility.

DFH$AFFY

IBM-supplied sample group DFH$AFFY is removed. This group contained sample resource definitions for the CICS transaction affinities utility that you could modify to suit your requirements.

Removing obsolete definition groups from start-up group lists

Obsolete definition groups have been removed from the CICS-supplied default start-up group list.

Removing obsolete definition groups from start-up group lists

Obsolete definition groups have been removed from the CICS-supplied default start-up group list, DFHLIST.

If you use customized startup group lists, you must remove any obsolete definition groups from them.

Changes to resource definition

Group attach is a DB2 facility that allows CICS to connect to any one member of a data sharing group of DB2 subsystems, rather than to a specific DB2 subsystem. The group attach facility chooses any member of the group that is active on the local MVS image for the connection to CICS (members that are active on other MVS images are not eligible for selection).

If you use the new DB2GROUPID attribute of the DB2CONN definition to specify the ID for the group of DB2 subsystems, instead of using the DB2ID attribute to specify the ID of an individual DB2 subsystem, you will activate the group attach facility. This means that you can use a common DB2CONN definition, specifying a group ID, across multiple cloned AORs, and CICS will connect to any active member of that data sharing group.

Group attach raises considerations regarding the resolution of units of work (UOWs) that are in doubt. Consider where CICS is connected to DB2, which is member 1 of the data sharing group, and the connection is lost, leaving member 1 holding in-doubt units of work. If CICS reconnects to member 1, the in-doubt units of work can be resolved. However, if group attach is requested, CICS could connect to member 2, in which case the in-doubt units of work held by member 1 cannot be resolved.

To solve this problem, CICS maintains a history of the last DB2 data sharing group member to which it was connected, which is cataloged and maintained across warm, emergency, and cold starts (but not initial starts). During connection to DB2, the CICS DB2 attachment facility checks this history to see if any outstanding UOW information is being held for the last DB2 data sharing group member to which it was connected. If no outstanding UOW information is being held, group attach operates normally and chooses any active member of the data sharing group for the connection. If outstanding UOW information is being held, the next action depends on the setting you have chosen for the new RESYNCMEMBER attribute of the DB2CONN definition. If the RESYNCMEMBER attribute is set to YES, indicating that you require resynchronization with the last recorded DB2 data sharing group member, CICS ignores the group attach facility and waits until it can reconnect to that DB2 data sharing group member, to resolve the in-doubt units of work. If the RESYNCMEMBER attribute is set to NO, perhaps because you want to reconnect as fast as possible, CICS makes one attempt to reconnect to the last recorded DB2 data sharing group member. If this attempt is successful, the in-doubt units of work can be resolved. If it is unsuccessful, CICS uses group attach to connect to any active member of the DB2 data sharing group, and the warning message DFHDB2064 is issued, stating that there may be unresolved in-doubt units of work with the last recorded member.

CORBASERVER CIPHERS Keyword added

This specifies a string of up to 56 hexadecimal digits that is interpreted as a list of up to 28 2-digit cipher suite codes. The attribute value is automatically populated with the list of acceptable codes, depending on what level of encryption has been specified by the ENCRYPTION system initialization parameter. For ENCRYPTION=WEAK, the default value is 03060102. For ENCRYPTION=MEDIUM, the default value is 0903060102. For ENCRYPTION=STRONG, the default value is 0504352F0A0903060102.

DOCTEMPLATE HFSFILE attribute

This allows the template to reside on an HFS file.

PROGRAM API attribute

This specifies what application programming interfaces the program will use.

CICSAPI means that the program uses CICS application programming interfaces only. CICS determines whether the program runs on the quasi-reentrant (QR) TCB or on another TCB. This depends upon the value of the CONCURRENCY attribute in the PROGRAM resource definition. If the program is defined as threadsafe it may run on whichever TCB, in use by CICS at the time, is determined as suitable.

OPENAPI means that the program is not restricted to the CICS application program interfaces. CICS executes the program on its own L8 or L9 mode open TCB dependent upon the value of the EXECKEY attribute in the PROGRAM resource definition. If, while executing a CICS command, CICS requires a switch to the QR TCB, it returns to the open TCB before handing control back to the application program. OPENAPI requires the program to be coded to threadsafe standards and defined with CONCURRENCY(THREADSAFE).

TCPIPSERVICE

With TCPIPSERVICE:

- CIPHERS added
- MAXDATALEN added
- PROTOCOL new USER value
- SOCKETCLOSE (changed socketclose(0) recommendation)

REQUESTMODEL

With REQUESTMODEL:

- 1.3 and 3.1 REQUESTMODEL definitions are incompatible.

  Must be defined in different groups

- TYPE={GENERIC|CORBA|EJB}
  - GENERIC: BEANNAME, INTFACETYPE, MODULE, INTERFACE, OPERATION
  - CORBA:MODULE, OPERATION, INTERFACE
  - EJB: BEANNAME, INTFACETYPE, OPERATION

TCPIPSERVICE - CIPHERS keyword added

This specifies a string of up to 56 hexadecimal digits that is interpreted as a list of up to 28 2-digit cipher suite codes. The attribute value is automatically populated with the list of acceptable codes, depending on what level of encryption has been specified by the ENCRYPTION system initialization parameter. For ENCRYPTION=WEAK, the default value is 03060102. For ENCRYPTION=MEDIUM, the default value is 0903060102. For ENCRYPTION=STRONG, the default value is 0504352F0A0903060102.

TCPIPSERVICE - MAXDATALEN keyword added

This defines the maximum length of data that can be received by CICS as an HTTP server, on the HTTP protocol or the USER protocol. The default value is 32 K. The minimum is 3 K, and the maximum is 524288 K. To increase security for CICS Web support, specify this option on every TCPIPSERVICE definition for the HTTP protocol. It helps to guard against denial of service attacks involving the transmission of large amounts of data.

TCPIPSERVICE - USER value added to protocol keyword

Processing for all non-HTTP requests must now be carried out under the USER protocol. No parsing is carried out for messages received on the USER protocol, and requests that have been divided up for transmission across the network are not automatically assembled. This is the same behavior as when handling non-HTTP messages in earlier CICS releases.

TCPIPSERVICE - Change of recommendation for SOCKETCLOSE(0)

In previous releases the recommendation was that if you are using the TCPIPSERVICE for CICS Web Support and are processing only standard HTTP requests, SOCKETCLOSE(0) should be specified to avoid unnecessary CWXN transactions remaining in the system.

However, in CICS TS 3.1, the socket can remain open without involving a CWXN transaction taking up a max task slot. Also, with the upgrade to HTTP 1.1 the recommendation is that if you are using a TCPIPSERVICE for CICS Web Support with the HTTP protocol, SOCKETCLOSE(0) should not be specified. A zero setting for SOCKETCLOSE means that CICS closes the connection immediately after receiving data from the Web client, unless further data is waiting. This means that persistent connections cannot be maintained.

New definitions

In this section we discuss new definitions.

PIPELINE definition

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. The information about the processing nodes is supplied indirectly: the PIPELINE specifies the name of an HFS file that contains an XML description of the nodes and their configuration. An inbound Web Service request (that is, a request by which a client invokes a Web Service in CICS) is associated with a PIPELINE resource by the URIMAP resource.

UIRMAP definition

URIMAP definitions are resource definitions that match the URIs of HTTP or Web Service requests and provide information about how to process the requests. URIMAP definitions are used to provide three different Web-related facilities in CICS:

- Requests from a Web client to CICS as an HTTP server
- Requests to a server from CICS as an HTTP client
- Web Service requests

WEBSERVICE definition

A WEBSERVICE resource defines aspects of the runtime environment for a CICS application program deployed in a Web Services setting, where the mapping between application data structure and SOAP messages has been generated using the CICS Web Services assistant. Although CICS provides the usual resource definition mechanisms for WEBSERVICE resources, they are typically installed dynamically, using the output produced by the assistant. The aspects of the runtime environment that are defined by the WEBSERVICE resource are:

- A pipeline
- A Web Service binding file
- A Web Service description

See *Implementing CICS*, SG24-7206.

- CORBASERVER

  Defines an execution environment for enterprise beans

- DJAR

  Defines an instance of a deployed JAR file, which contains enterprise beans

A CORBASERVER defines an execution environment for enterprise beans and stateless CORBA objects.

The attributes include information that is used to construct Generic Factory Interoperable Object References used by clients that invoke stateless CORBA objects. Information that is used when making outbound method requests on objects in remote EJB or CORBA servers.

A DJAR defines an instance of a deployed JAR file, which contains enterprise beans. The definition identifies a particular instance of a deployed JAR file (in the sense that it is valid to have multiple versions of the same deployed JAR file deployed in different CorbaServers in the same region). The DJAR definition also associates the JAR file instance with its execution environment, the CorbaServer.

A deployed JAR file is an ejb-jar file containing enterprise beans, on which code generation has been performed and that has been stored on the hierarchical file system (HFS) used by z/OS. When the DJAR definition is installed, CICS copies the deployed JAR file (specified by HFSFILE) into a subdirectory of the HFS shelf directory of the specified CORBASERVER.

Application Programming Interface

In this section we discuss the Application Programming Interface.

EXEC CICS

In CICS TS 1.3 and earlier, CICS recognizes the sign-on immediately, and establishes the specified user's security and operating attributes for the terminal. The transaction (and any associated task-related user exits, function shipping, or distributed transaction processing) may have invoked other resource managers (for example, IMS, DB2, or VSAM). It is unpredictable whether these other RMs recognize the sign-on before the transaction terminates, and thus you can only be sure that the new user attributes apply for all resource managers invoked by subsequent transactions at the terminal. Hence, since CICS TS V2, the behavior of EXEC CICS ISGNON and SIGNOFF changed in that a SIGNON and SIGNOFF command does not affect the current transaction issuing the command.

- SIGNON/SIGNOFF
  - Since CICS TS V2 operation is terminal related only
  - Executing transaction security and user ID set at task attach time

    XSNEX Global User Exit (migration aid retained for compatibility)
- VERIFY PASSWORD

  CICS now enforces the revoked status of a user ID or a user's group connection.
- RESP2 values
  - File Control

    RESP2 values always returned for local and remote files
  - Program Control

    New RESP2 values for Java errors
- Mapset Generation (DFHMSD)
  - Will add CSECT, AMODE 31, and RMODE ANY statements, only if the MAPSET does not include a CSECT statement

If you have applications that cannot tolerate the change in the SIGNON and SIGNOFF process, CICS provides a global user exit point (XSNEX) and sample global user exit program that will enable CICS to handle EXEC CICS SIGNON and SIGNOFF as in CICS TS 1.3 and earlier releases. Note that XSNEX is a migration aid only, and you should consider removing all application dependency on the old behavior. CICS TS 3.1 continues to ship this migration aid.

When the command EXEC CICS VERIFY PASSWORD is issued, CICS now enforces the revoked status of a user ID or a user's group connection. For example, if a user has tried to log on too many times, the ID is revoked and the user cannot access the system or resources.

High Performance Java (HPJ) Programs

Non-IIOP applications must be converted to JVM programs.

Run-time support for Java program objects and for hot-pooling (HPJ) is withdrawn in CICS TS 3.1. Any Java programs that you had processed using the VisualAge for Java, Enterprise Edition for OS/390 bytecode binder (hpj) to run as Java program objects in CICS must be migrated to run in a Java Virtual Machine (JVM).

C/C++ programs

These can now use XPLINK capability.

CICS provides support for C and C++ programs compiled with the XPLINK option by using the multiple TCB feature in the CICS Open Transaction Environment (OTE) technology. X8 and X9 mode TCBs are defined to support XPLink tasks in the CICS key and the USER key, respectively. Each instance of an XPLink program uses one X8 or X9 TCB.

- Activated via XPLINK compiler option.
- New CICS-supplied procedures for Translate, compile, and linkedit.
- Programs run on X8 or X9 TCBs using MVS LE services.
- Programs must be threadsafe to use XPLINK and defined as threadsafe.

To use XPLink, your C or C++ application code must be reentrant and threadsafe. The same code instance can be executing on more than one MVS TCB and, without threadsafe mechanisms to protect shared resources, the execution behavior of application code is unpredictable. This cannot be too strongly emphasized.

CICS provides procedures DFHYITFL for C programs and DFHYITGL for C++ Programs wanting to use XPLINK.

Systems Programming Interface

A new SPI command, EXTRACT STATISTICS, handles statistics for URIMAP, PIPELINE, and WEBSERVICE resources. Use the EXTRACT STATISTICS command to retrieve the current statistics for a single resource, or global statistics for a class of resources. The EXTRACT STATISTICS command performs a function equivalent to COLLECT STATISTICS for the URIMAP, PIPELINE, and WEBSERVICE resources. To collect statistics for other resources use the existing COLLECT STATISTICS command. The syntax of the EXTRACT STATISTICS differs from that of COLLECT STATISTICS.

EXEC CICS EXTRACT STATISTICS:

- PIPELINE
- URIMAP

- WEBSERVICE

All CICS SPI commands are restricted in the number of distinct options they can support. As new resources have been added to CICS over time, the limit has been reached for the COLLECT STATISTICS command, and it is not possible to accommodate the new URIMAP, PIPELINE, and WEBSERVICE resources on the existing command.

The EXTRACT STATISTICS command uses the RESTYPE option, with a CVDA, to specify a CICS resource. As a result, there is no limit on the number of resources that the command can potentially support, although in this release, only the three new resources are supported.

Global user exits

We *highly* recommended that *all* global user exits be analyzed to ensure that they are threadsafe and that their program definitions changed to specify CONCURRENCY(THREADSAFE). XRMIIN, XRMIOUT, XEIIN and XEIOUT are the most important OTE considerations. Changes to the standard parameter list (DFHUEPAR) are UEPGIND, task indicator field; L9, X8, X9, and SP; H8 no longer available.

All user programs defined by a program resource definition have a concurrency attribute, which can be either QUASIRENT or THREADSAFE. By default, global user programs are defined as quasi-reentrant, which means that they are given control on the CICS QR TCB. If the task under which the global user exit is invoked is executing on an open TCB, and the exit program is defined as quasi-reentrant, CICS switches back to the QR TCB for the execution of the exit program.

To avoid unnecessary TCB switching, we strongly recommend that you make sure that your global user programs conform to threadsafe programming standards. When you are satisfied that your exit programs are threadsafe, ensure that they are defined as CONCURRENCY(THREADSAFE). This is particularly important for exits that are invoked by tasks that are using the CICS DB2 interface and running under an L8 TCB.

Exit parameter UEPGIND passed to global user exits includes reference to the mode of the TCB the exit is running on. With the new types of open TCB introduced, exits can now run on these new types of TCB if they are threadsafe and defined to CICS as such.

New global user exits

There are two new global user exits for CICS as an HTTP client: XWBOPEN in the WEB OPEN command and XWBSNDO in the WEB SEND command. (Note that XWBSNDO only applies when the WEB SEND command is used for CICS as an HTTP client, and not for CICS as an HTTP server.)

- XWBOPEN

  This is called during WEB OPEN, before the session is established, and can be used to bar access to a whole host.

- XWBSNDO

  This is called during WEB SEND or WEB CONVERSE, and enables systems administrators to specify a security policy for HTTP requests by CICS.

XWBOPEN

This enables systems administrators to specify proxy servers that should be used for HTTP requests by CICS as an HTTP client, and to apply a security policy to the host name specified for those requests. XWBOPEN is called during processing of an EXEC CICS WEB OPEN command, which is used by an application program to open a connection with a server.

XWBSNDO

This enables systems administrators to specify a security policy for HTTP requests by CICS as an HTTP client. XWBSNDO is called during processing of an EXEC CICS WEB SEND or EXEC CICS WEB CONVERSE command. The host name and path information are passed to the exit, and a security policy can be applied to either or both of these components.

- XICERES

  Enables the user to determine the availability of resources in a remote region for dynamically routed starts

- XPCERES

  Enables the user to determine the availability of resources in a remote region for dynamic distributed program links

XICERES

This is invoked by the interval control program, before CICS processes a non-terminal-related EXEC CICS START request that has been dynamically routed to this region.

XPCERES

This is invoked by the EXEC interface program on the target region before CICS processes either of the following kinds of dynamically routed link request:

- A distributed program link (DPL) call
- A Link3270 bridge request

XFAINTU

Facility Initialization and Tidy Up is called just after a new bridge facility has been built. This may be at the end of a task (when zero keep time is specified) or when a keep time expires before the facility is re-used.

XFCREQ

This exit allows you to intercept a file control application programming interface (API) request before any action has been taken on it by file control. The XFCREQC exit allows you to intercept a file control API request after file control has completed its processing.

- Using XFCREQ, you can analyze the request to determine its type, the keywords specified, and their values. Modify values specified by the request before the command is executed. Set return codes.
- Using XFCREQC, you can analyze the request to determine its type, the keywords specified, and their values. Set return codes for the EIB.

XFCFRIN

This exit is invoked on entry to the main file control request gate, FCFR. It allows you to monitor file control requests and allow them to continue, to be processed by CICS file control, to intercept file control requests, and to bypass CICS file control processing altogether. Redirect the request to a remote region.

XFCFROUT

This exit is invoked after completion of a file control request. It is invoked in both the following cases: after CICS file control has completed its processing, either normally or with an error.

Changed global user exits

Global user exit programs cannot access containers created by application programs. They can, however, create their own channels and pass them to programs that they call.

- Parameter list changes
  - Existence bits with channel name passed to exits

  - XICEREQ, XICEREQC
  - XPCREQ, XPCEREQC
  - XPCTA, XPCFTCH, XPCHAIR, XPCABND

– Exits may not access contents of channels

- XPlink programs
  – XPCTA does not allow a resume address.
  – New flag PCUE_NO_RESUME, in PCUE_CONTROL_BITS.
- XPCFTCH does not allow a modified entry address.
  – New flag PCUE_NO_MODIFY, in PCUE_CONTROL_BITS
  – Alternative is CEEBXITA

When the exit XPCTA is invoked from a C or C++ program that was compiled with the XPLINK option, a flag is set indicating that a resume address, if specified by the exit, will be ignored. This is because XPLINK runs with MVS LE, which has it own recovery procedures that percolate to CICS. By the time CICS recovery gets control, the program environment has gone. When the exit XPCFTCH is invoked from a C or C++ program that was compiled with the XPLINK option, a flag is set indicating that any modified entry point address, if specified by the exit, will be ignored. It is not supported because XPLINK uses MVS LE with CEEPIPI preinitialised interface and PIPI will reject the signature of any assembler program.

Removed global user exits

In this section we discuss removed global user exits.

XTCTIN Terminal control program

This exit was invoked on TCAM input events. It is no longer called because CICS TS 3.1 does not support the TCAM/ACB interface, and it only supports the TCAM/DCB interface indirectly.

XTCTOUT Terminal control program

This exit was invoked on TCAM output events. It is no longer called because CICS TS 3.1 does not support the TCAM/ACB interface, and it only supports the TCAM/DCB interface indirectly.

User-replaceable modules

The user-replaceable programs DFHAPH8O and DFHJHPAT are removed.

- Removed URMs
  - DFHAPH8O (HPJ hotpooling)
  - DFHJHPAT (HPJ)
- New URMs

  DFHAPXPO (XPLINK)
- Changed URMs

  User-replaceable programs cannot access containers created by application code.
- DFHCNV

  Added SYSDEF operand to TYPE=INITIAL

DFHAPH8O was provided to allow you to alter the default Language Environment runtime options for the Language Environment enclave where a Java program object was to be run.

DFHJHPAT was optional and could be used for your own purposes, such as tracing. It was called before a Java program object was invoked.

The new User-replaceable module DFHAPXPO allows you to alter the default Language Environment runtime options for the Language Environment enclave where an XPLINK program is to run.

There are changes to the dynamic routing copybook to support the new channel/container constructs. There is a new field, DYRCHANL, which contains the name of channel associated with the request. There are changes to an existing field, DYRACMAA. If the user application employs a communications area (COMMAREA), this field will contain the 31-bit address of the application's COMMAREA. If the user application employs a channel and has created, within the channel, a container named DFHROUTE, this field will contain the 31-bit address of the DFHROUTE container. If the user application has no COMMAREA and no DFHROUTE container, this field will contain null characters.

The new operand SYSDEF has been added to the TYPE=INITIAL and TYPE=ENTRY macro parameters CLINTCP and SRVERCP. These macros define the user-replaceable data conversion table DFHCNV. The DFHCNV TYPE=INITIAL macro defines the beginning of the conversion table. It gives a list of valid code pages. The DFHCNV TYPE=ENTRY macro specifies a name and type to uniquely identify a data resource. There must be one for each resource for which conversion is required.

Monitoring and statistics

Some performance data fields are added to performance class data records. The result of all these additions is that record length of performance class data records has increased significantly, with the maximum record length now up to 1836 bytes per record.

- Performance class data

  Record size increases to 1836 bytes. Reduce using INCLUDE and EXCLUDE option on the MCT.

- Changes to statistics record

  New and changed DSECTs:

  - DFHCHNL - Container usage
  - DFHPROG - Program statistics
  - DFHSOCK - TCP/IP statistics
  - DFHTASK - Task statistics
  - DFHWEBB - Web support statistics

SMF data sets can quickly fill with unwanted data.You can reduce the amount of data written to SMF by using a monitoring control table (MCT) to selectively include or exclude specified fields.

There are changes to CICS statistics records. These are usually because of new domains, or they are a result of enhancements to CICS. As a result, a number of statistics DSECTs have new or changed fields. The changed DSECTs are:

- DFHDSGDS - Dispatcher global statistics
- DFHPIPDS - Pipeline resource statistics
- DFHPIWDS - Webservice resource statistics
- DFHWBSDS - Urimap global statistics
- DFHMNTDS - Transaction performance monitoring resource statistics
- DFHWBRDS - Urimap resource statistics
- DFHSORDS - TCP/IP resource statistics
- XPLINK - CPU time included in CICS 110 records
  - X8 CPU - Dispatch and delay times
  - X9 CPU - Dispatch and delay times
- OPENAPI
  - New L9 CPU - Dispatch and delay times, as well as existing L8 times
- SP and S8 CPU time
  - SP time included in miscellaneous

  - S8 CPU - Dispatch and dispatch delay times

The CICS 110 record includes new fields to record the CPU time consumed on X8 and X9 TCBs used by XPLINK programs. These contribute to the overall CPU total for the transaction.

Similarly, for openapi programs, L9 TCBs contribute to the overall time, as well as the existing L8 TCBs. For SSL a new SP TCB and S8 TCBS CPU time is captured.

- DB2 CPU time will be included in CICS 110 records.

  DB2 class 1 time will be included in the CICS CPU time. May increase due to now accounting for CREATE thread.

- L8 CPU time will be greater or equal to DB2 class 1 time.
  - May also contain thread create or termination time
  - If application is threadsafe:
    - Will contain CPU time spent in application
    - QR CPU time will decrease

- DB2WAIT field will be zero.
  - Represents elapsed time spent waiting for a DB2 request to complete
  - With OTE there is no CICS dispatcher wait for a subtask

- Can be large difference between DB2 class 1 and class 2 CPU times.
  - CICS RMI code and threadsafe application code
  - CICS tracing

When CICS is connected to DB2 Version 6 or later, and is exploiting the open transaction environment, the CICS DB2 attachment facility uses CICS-managed open TCBs rather than CICS DB2 subtask TCBs. This means the CICS monitoring facility can measure activity that was previously only reported in the DB2 accounting record (the SMF type 101 record). For example, CICS can now measure the processor time consumed on the DB2 thread and the processor time consumed in DB2 (the CLASS 1 and CLASS 2 CPU time). When CICS is using L8 open TCBs, the CPU time reported for these TCBs by the CICS monitoring facility includes the DB2 CLASS 1 processor time.

When CICS is connected to DB2 Version 6 or later, do not add together the processor time from the CICS records (SMF type 110 records) and the DB2 accounting records (SMF type 101 records) when calculating the total processor time for a single transaction, because the DB2 processor time would then be included twice. The total processor time for a single transaction is recorded in the USRCPUT field in the CICS records (performance class data field 008 from group DFHTASK). This field includes all processor time used by the transaction when it was executing on any TCB managed by the CICS dispatcher.

CICS SOAP feature

If you use the SOAP for CICS feature, you can continue to do so. The feature continues to be fully supported in CICS TS 3.1 independently of Web Services in CICS.

The SOAP for CICS feature can interoperate with the support for Web Services in CICS TS for z/OS Version 3.1. The feature can be the service requester or the service provider. It is not orderable with CICS TS 3.1. The existing Version 2 feature may be used with 3.1:

- Intent is to aid migration
- Not intended as a substitute for Web Services

CICSPlex Systems Manager

CICSPlex migrations similar to previous releases, CAS, CMAS, and MAS agent code must all be at the 3.1 level.

The WUI Server and its connected CMAS must be at the 3.1 level. Migrate contents of the WUI Server repository:

1. At the prior CICS level, export view set and menu definitions.
2. Create a new WUI server repository for 3.1.
3. Start the 3.1 WUI server.
4. Import the new starter set definitions.
5. Review the new view formats with your changes.
   - Import previous release view set and menu definitions.
   - Specify SKIP on Duplicate Names field of COVC panel.

Important: Maintenance point CMAS must be upgraded first.

You must migrate your CICSPlex SM CMAS to CICS TS Version 3.1 at the same time as you migrate the CICS system on which it runs. This is because CICS Transaction Server for z/OS Version 2 Release 3 a CICSPlex SM CMAS will run only in a CICS system at the same release level.

Both the Web User Interface server and the CMAS that it connects to must be at the highest level of CICSPlex SM within the CICSplex. This means that both must be at the same level as the maintenance point CMAS.

Before you migrate a Web User Interface server, you must migrate the CMAS that it connects to. You must migrate the Web User Interface server before you migrate any other MASs. If the CMAS that the Web User Interface server connects to is not the maintenance point CMAS, you must migrate the maintenance point CMAS at the same time.

As the CICS system that acts as your Web User Interface server is a local MAS, all the considerations that apply to a local MAS also apply to a Web User Interface server.

Language Environment

Runtime support for OS/VS COBOL programs is withdrawn. OS/VS COBOL programs, which had runtime support in CICS Transaction Server for z/OS Version 2, cannot run under CICS TS 3.1.

OS/VS COBOL programs must be upgraded to Language Environment conforming COBOL, and recompiled against a level of COBOL compiler supported by CICS. Enterprise COBOL for z/OS and OS/390 Version 3 is the recommended compiler.

You can now produce assembler MAIN programs that are Language Environment conforming. Until now, the only way to use Language Environment conforming assembler programs within CICS was to use a call from a COBOL, PLI, or C Language Environment conforming program and linkedit the assembler program with the high-level language (HLL) program. This made the assembler program a Language Environment subroutine. It had to have MAIN=NO on CEEENTRY. The user had to specify NOPROLOG and NOEPILOG and then code the CEEENTRY and CEETERM calls separately. A CICS PROGRAM resource could not be defined as both ASM and LE370.

CICS now supports the coding of Language Environment conforming assembler MAIN programs. A new translator option LEASM causes the Language Environment function to be used to set up the program's environment. Such programs must be linkedited with stub DFHELII rather than DFHEAI.

This support also enables use of the Debugger for Assembler programs.

- CICS TS 2.3
  - CICS interfaces for the VS COBOL II, OS PL/I, and C/370 runtimes removed
  - Need runtime libraries distributed with LE to execute current load modules
- CICS TS 3.1
  - CICS interfaces for the OS/VS COBOL are removed.
  - CICS will terminate any OS/VS COBOL program with an ALIK abend.
  - Support for LE conforming Assembler main programs added.

To enable Language Environment support to be installed correctly by CICS:

1. Specify enough storage for the ERDSA to run CICS and the Language Environment together. They need a minimum of 3500 KB. To this minimum, add an amount of storage sufficient for your own requirements.
2. Ensure that the CICS-Language Environment interface module CEECCICS and the Language Environment modules CEEPIPI and CEECTCB are installed in an APF-authorized library defined in the STEPLIB concatenation in the CICS startup JCL. You can do this by including the Language Environment SCEERUN library in an APF-authorized library in the STEPLIB concatenation of your CICS startup job (for example, in the CICSTS31.CICS.SDFHAUTH library) or in an APF-authorized library in the MVS LNKLSTnn concatenation.
3. Ensure that the program resource definitions for the Language Environment language interface modules have been added to the CICS CSD. These definitions are in the CEE group. The CEE group is added automatically to the CSD and to the grouplist DFHLIST during CICS installation, as part of the DFHCOMDS job. The definitions are also supplied as DEFINE statements in the CEECCSD member of the SCEESAMP library. You can add the CEE group to any CICS startup group list named in the GRPLIST system initialization parameter.
4. Define the Language Environment transient data destinations, CESE, and CESO (DD names CEEMSG and CEEOUT). The CICS-supplied resource definition group, in the CSD, DFHDCTG, contains entries for CESE and CESO. For information about the attributes needed for Language Environment transient data destinations, see the *IBM Language Environment for MVS & VM Programming Guide*, SC26-4818.
5. Define the Language Environment runtime libraries on the CICS STEPLIB and DFHRPL DD statements as follows: Add the SCEERUN library, which contains CEECCICS and CEECTCB, and the SCEERUN2 library, which contains support that is required for the IBM Java Virtual Machine (JVM) and also support for other programming languages, to STEPLIB or to a library in the MVS LNKLSTnn concatenation. Both the libraries, SCEERUN and SCEERUN2, must be APF-authorized. Add the SCEECICS, SCEERUN2, and SCEERUN libraries to DFHRPL, with SCEECICS and SCEERUN2 concatenated before SCEERUN.

Language Environment…

- **Review ERDSA specification for CICS LE requirements**
  - Minimum of 3500K
- **Review RDO definitions for LE programs**
  - LE language interface modules
    - Definitions are in the CEE group
      - Automatically created if DFHCOMDS is run during CICS installation
      - Supplied in CEECCSD member of the SCEESAMP library
- **Define the LE transient data destinations, CESE, and CESO**
  - DD names CEEMSG and CEEOUT
  - RDO group DFHDCTG, contains entries for CESE and CESO
- **Define the LE runtime libraries on the CICS STEPLIB and DFHRPL DD statements as follows:**
  - Add the SCEERUN2 before SCEERUN to STEPLIB or to LNKLST concatenation
  - Add the SCEECICS, SCEERUN2 and SCEERUN libraries to DFHRPL
    - SCEECICS concatenated before SCEERUN2 before SCEERUN
      - XPLINK requires the addition of SCEERUN2 to DFHRPL

Figure B-7 LE support

Java

CICS Transaction Server for z/OS Version 3 Release 1 supports the JVM created by the IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.2 at or later, which features the persistent reusable JVM technology.

CICS Transaction Server for z/OS Version 2 Release 2 supported the JVM created by the IBM Developer Kit for OS/390 Java 2 Technology Edition Version 1.3.1s, which also featured the persistent reusable JVM technology. Java programs that ran under CICS Transaction Server for z/OS Version 2 Release 2 can also run under CICS Transaction Server for z/OS Version 3 Release 1.

The library SDFJAUTH is now required for Java support. SDFJAUTH is the partitioned data set extended (PDSE) version of SDFHAUTH, and it contains some of the components of the SJ domain. A separate library is needed because these components are now built using Extra Performance Linkage (XPLink). As for the SDFHAUTH library, the SDFJAUTH library must be APF-authorized by adding it to the list of APF-authorized libraries in an appropriate member in SYS1.PARMLIB, and a STEPLIB DD statement must be provided for it in your startup job stream.

JVM profiles, which contain the JVM initialization options, are now kept as HFS files, rather than as members of a partitioned data set (PDS). The DFHJVM DD card in the CICS startup JCL, which referred to the PDS for the JVM profiles, is no longer required and should be removed. You can use several different JVM profiles in the same CICS region, and each is stored as a separate HFS file. The name of each JVM profile (that is, the name of the HFS file) must still be eight characters or less, so that it can be used in the program definition. Use the JVMPROFILE attribute of a PROGRAM resource definition to name the JVM profile that is used to construct the JVM that runs the program.

CICS-defined programs now have their own JVM profile, DFHJVMCD, to make them independent of any changes you make to the default JVM profile DFHJVMPR. DFHJVMCD is used by the default request processor program DFJIIRP, which is used by the CICS-supplied CIRP request processor transaction, and by DFJIIRQ, the CICS-key equivalent of DFJIIRP. DFHJVMCD has an associated JVM properties file, dfjjvmcd.props. You need to make changes to DFHJVMCD and dfjjvmcd.props to ensure that the settings in them are suitable for your installation (including the configuration for your JNDI nameserver).

For language migration issues see:

- http://java.sun.com/j2se/1.4/compatibility.html
- http://java.sun.com/products/jdk/1.3/compatibility.html#incompatibilities1.3

The EXECKEY parameter on the PROGRAM resource definition is no longer ignored for Java programs. In CICS Transaction Server for OS/390 Version 1 Release 3 and CICS Transaction Server for z/OS Version 2 Release 2, CICS made all Java programs execute in the CICS key, but they now execute as specified by the EXECKEY parameter. The default for this parameter is EXECKEY(USER), which means that the program runs in a JVM that executes in the user key. (A new type of open TCB, the J9 TCB, is used for these JVMs.) As running applications in the user key extend CICS storage protection, it could be beneficial to let most of your Java programs run in a JVM in the user key.

Before setting up the shared class cache, you must check the options for semaphores that you have set in the BPXPRMxx members of SYS1.PARMLIB. The master JVM that initializes the shared class cache uses a single semaphore ID, and requests a set of 32 semaphores, so you must:

- Ensure that the MNIDS value is enough for the maximum number of semaphore IDs that are in use at one time, including the shared class cache. Depending on the frequency with which you expect to reload the shared class cache, you might want to allow two or possibly three semaphore IDs for the shared class cache. One semaphore ID would be used by the master JVM that controls the active shared class cache, and the remainder would be used by a master JVM that controls a shared class cache that is being phased out, or by a new master JVM that controls a shared class cache that is being loaded. It is unlikely that you would need more than two semaphore IDs for the shared class cache, except in a CICS region that is being heavily used for development and testing.

- Ensure that the MNSEMS value is enough for the maximum number of semaphores that the master JVM requests in a semaphore set. The value must be 32 or greater. If you need to change the MNIDS value, you can do this by using the IPCSEMNSEMS parameter (that is, in the BPXPRMxx members of SYS1.PARMLIB).

Open transaction environment

Applications wanting to use XPLINK or OPENAPI support must be coded to threadsafe standards. Applications must worry about concurrent access to their resources such as shared storage. Unless an application requires overwriting itself (in which case it has to provide serialization of such code — a type of shared storage), then ensure that applications are read-only. The CICS read-only DSA can be used to ensure this.

CICS provides a load module scanner utility with a sample table called DFHEIDTH that looks for applications that issue EXEC CICS ADDRESS CWA, EXEC CICS GETMAIN SHARED, or EXEC CICS EXTRACT EXIT. All these commands give access to shared storage and hence have the potential for the application logic not being threadsafe if the storage is not subsequently updated in a threadsafe way. Applications can use ENQUEUE and DEQUEUE to serialize updates to shared storage. In assembler applications compare and swap instructions can be used.

- OPENAPI and C/C++ XPLINK applications have to be THREADSAFE. CICS will ensure threadsafe access to its managed resources: VSAM files, TS, TD, DLI databases, and DB2 tables. Applications have to ensure threadsafe access to their resources: shared storage (for example, CWA, GETMAIN SHARED).
- Ensure that applications are read-only.
  - Put them in the CICS read-only DSA (linkedit with RENT).
  - Set the SIT option RENTPGM=PROTECT.
- Serialize access to shared resources.
  - CWA or shared storage: Use the load module scanner to look for use of global storage.
  - Use services such as EXEC CICS ENQUEUE and DEQUEUE.

Function removal

Support for the CICS Connector for CICS TS, introduced in CICS TS for z/OS Version 2.1, is withdrawn.

A CICS connector is a software component that allows a Java client application to invoke a CICS application. CICS TS for z/OS Version 2.3 introduced a new CICS connector, the CCI Connector for CICS TS, that performs a similar role to the CICS Connector for CICS TS (that is, it enables a Java program or enterprise bean running on CICS Transaction Server for z/OS to link to a CICS server program). However, while the old CICS Connector for CICS TS implemented the IBM-proprietary Common Connector Framework (CCF) interface, the new CCI Connector for CICS TS implements the industry-standard Common Client Interface (CCI) defined by the J2EE Connector Architecture Specification Version 1.0.

The ECI Base Classes (ECIREQUEST, which were introduced for compatibility with the CICS Transaction Gateway) are not included in CICS TS 3.1. The recommended replacement is the COMMON CLIENT INTERFACE CONNECTOR FOR CICS TS (CCI Connector for CICS TS), introduced in CICS TS V2.3, when it was announced that ECIREQUEST would be removed.

CICS TS 3.1 does not include the detector and reporter components previously provided as part of the CICS Transaction Affinities utility. These components are now incorporated in IBM CICS Interdependency Analyzer for z/OS V1.3, announced in August 2004, which has the capability of analyzing both interdependencies and affinities. The load library scanner component of the CICS Transaction Affinities utility remains in CICS TS 3.1 and can produce reports on application programs that have potential affinities.

Support for defining terminals using the 1-byte console ID is withdrawn. The CONSOLE attribute on the TERMINAL resource definition is obsolete, but is supported to provide compatibility with earlier releases of CICS. You can define terminals using the CONSNAME(name) attribute on the TERMINAL resource definition.

If you have a network of terminals connected by the ACB interface of TCAM to a back-level CICS TOR, you will not be able (as you were in previous CICS releases) to route transactions from them to a CICS TS for z/OS Version 3.1 AOR. You must migrate your connections to use TCAM/DCB or (preferably) ACF/VTAM, or route to a previous version of CICS. (All terminals that support TCAM/ACB also support ACF/VTAM.)

If you have a network of terminals connected by the DCB interface of TCAM to, for example, a CICS TS 2.3 TOR, you will not be able to migrate the TOR to

CICS TS for z/OS Version 3.1. To do so, you must migrate your connections to use ACF/VTAM.

If you have a network of terminals connected by the DCB interface of TCAM to a back-level CICS TOR, you will (as in previous CICS releases) be able to route transactions from them to a CICS TS for z/OS Version 3.1 AOR. However, we recommend that you migrate your connections to use ACF/VTAM.

If you have a network of BTAM terminals connected to a back-level CICS terminal-owning region (TOR), you will not be able (as you were in previous CICS releases) to route transactions from them to a CICS TS for z/OS Version 3.1 application-owning region (AOR). You must either upgrade your terminals or route to a previous version of CICS.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 470. Note that some of the documents referenced here may be available in softcopy only.

- *Patterns: Service-Oriented Architecture and Web Services*, SG24-6303
- *Threadsafe Considerations for CICS*, SG24-6351
- *Using the Web User Interface in CICSPlex SM*, SG24-6793
- *CICS Transaction Server V3R1 Channels and Containers Revealed*, SG24-7227
- *Implementing CICS*, SG24-7206
- *Application Development for CICS Web Services*, SG24-7126

Other publications

These publications are also relevant as further information sources:

- *IBM Language Environment for MVS & VM Programming Guide*, SC26-4818
- *CICS TS 3.1 Migration guide*, SC34-6458
- *CICS TS 3.1 Installation guide*, SC34-6425
- *CICS TS 3.1 Web Service Guide*, SC34-6458

Online resources

These Web sites and URLs are also relevant as further information sources:

- XML description

  `http://www.w3.org/XML/`

- IBM developerWorks

  http://www.ibm.com/developerworks/views/webservices/standards.jsp
- CP12: CICS Performance Analyzer for z/OS Historical Database Reporting

  http://www.ibm.com/support/docview.wss?uid=swg24011321

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

E

F

G

H

I

X

Z

IBM

Redbooks

Migration Considerations for CICS Using CICS CM, CICS PA, and CICS IA

Migration Considerations for CICS Using CICS CM, CICS PA, and CICS IA

Use CICS CM to copy and transform CICS resource definitions

Use step-by-step migration to CICS TS 3.1

Use CICS IA to identify migration issues

This IBM Redbook focuses on CICS Migration to CICS TS 3.1, showing you how the CICS Tools (CICS Configuration Manager, CICS Interdependency Analyzer, and CICS Performance Analyzer) can help you with your migration.

Part 1, “Introduce CICS TS 3.1 and the CICS Tools” on page 1, gives an overview of the new functionality available in CICS TS 3.1 and an overview of the CICS Tools individually.

Part 2, “Migration” on page 111, looks at migration, discussing migration considerations and CICS TS 3.1 exploitation. It also looks at three migration scenarios:

- Migrating CICS TS 2.3 CSD to CICS TS 3.1 CSD
- Migrating CICS TS 2.3 CSD to CICSPlex SM TS 3.1 BAS
- Migrating an Application to CICS Web Services in CICS TS 3.1

SG24-7294-00 ISBN 073849688X