232

CICS update

March 2005

In this issue

A tcі PUBLICATION

CICS Update

Published by

Xephon Inc
PO Box 550547
Dallas, Texas 75355
USA

Phone: 214-340-5690
Fax: 214-341-7081

Editor

Trevor Eddolls
E-mail: trevore@xephon.com

Publisher

Colin Smith
E-mail: info@xephon.com

Subscriptions and back-issues

A year's subscription to *CICS Update*, comprising twelve monthly issues, costs $270.00 in the USA and Canada; £175.00 in the UK; £181.00 in Europe; £187.00 in Australasia and Japan; and £185.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the December 2000 issue, are available separately to subscribers for $24.00 (£16.00) each including postage.

CICS Update on-line

Code from *CICS Update*, and complete issues in Acrobat PDF format, can be downloaded from our Web site at http://www.xephon.com/cics; you will need to supply a word from the printed issue.

Contributions

When Xephon is given copyright, articles published in *CICS Update* are paid for at the rate of $160 (£100 outside North America) per 1000 words and $80 (£50) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of $32 (£20) per 100 lines. To find out more about contributing an article, without any obligation, please download a copy of our *Notes for Contributors* from www.xephon.com/nfc.

CICS threadsafe

In this article I will discuss the concept of threadsafe and also cover the history behind it.

The Open Transaction Environment (OTE) TCBs in CICS have generated a great deal of interest for sites that are migrating to CICS Transaction Server Version 2. The potential to improve transaction throughput by eliminating TCB switching can be high. However, as some sites have already discovered, achieving this goal is not as simplistic as issuing a DB2 call to cause a TCB switch. Apart from system performance and application integrity, achieving consistent results regardless of system load is a major consideration when implementing threadsafe applications.

Before OTE, all application code ran under the main CICS TCB, called the Quasi-Reentrant (QR) TCB. The CICS dispatcher sub-dispatches the use of the QR TCB between the CICS tasks. Each task voluntarily gives up control when it issues a CICS service, which then issues a CICS dispatcher wait. There is only ever one CICS task active at any one time on the QR TCB.

Programs are said to be quasi-reentrant programs because they take advantage of the behaviour of the CICS dispatcher and the QR TCB. This means that although the same program can be executed by multiple CICS tasks, only one of those CICS tasks is active at any given point in time.

Quasi-reentrant programs always run under the QR TCB and can access shared resources such as the Common Work Area (CWA) or shared storage obtained via the EXEC CICS GETMAIN SHARED instruction. OTE introduces a new class of TCB, called an open TCB, which can be used by applications. An open TCB is characterized by the fact that it is assigned to a CICS task for the life of the CICS task and multiple OTE TCBs may run concurrently in CICS.

No sub-dispatching of any other CICS tasks under the open TCB will take place. An application executing under an open TCB can issue non-CICS API requests, which may involve the TCB becoming blocked. Because only this TCB will be halted, blocking will be allowed. If a block occurs in a QR TCB, the whole of CICS is halted.

Because multiple tasks can potentially access shared resources at the same time when executing under an OTE TCB, applications accessing shared resources such as the CWA must take responsibility to ensure integrity of those resources by implementing an appropriate serialization technique. CICS assumes responsibility for ensuring integrity of the resources it manages. It achieves this by having code changed to run on multiple TCBs or by ensuring that the code will execute on the QR TCB. If you use non-threadsafe CICS commands, they will run on the QR TCB. This will incur a performance hit because of the need to switch TCBs.

The main reasons for migrating applications to be threadsafe are:

1 To improve performance

2 To reduce costs

3 For future positioning.

If you normally experience any of the following, performance improvement may be achieved:

1 CICS QR TCB is CPU constrained.

2 Application tasks wait excessively for the QR TCB.

3 The CICS region generally experiences CPU constraints.

4 Significant numbers of EXEC SQL calls are invoked by each task.

By redesigning as threadsafe, it is possible to significantly reduce the path length of the application tasks. Transactions that will achieve the greatest CPU reduction normally have the following functionality:

1 Significant numbers of EXEC SQL calls invoked per task.

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

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

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

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

If the previous functionality is achieved, then threadsafe will all but eliminate TCB switches for the CICS tasks associated with these calls.

By fully implementing OTE in CICS applications you will be ready for future releases of CICS Transaction Server. IBM intends to move all processing to threadsafe and recommends that any new application be designed to take advantage of this technology. Applications that can be defined as threadsafe now will be able to exploit future CICS enhancements with the minimum amount of migration effort.

It is probably worthwhile mentioning just what a 'threadsafe application' is. A threadsafe program, as defined by IBM, is a program that uses appropriate serialization techniques, such as compare and swap or enqueue, when accessing any shared 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. This means code has to be written to a certain format and standard. It is important that programs should not be defined in CICS as threadsafe until they are truly coded correctly. If it is defined as threadsafe but is not truly threadsafe then you will see unpredictable results throughout the CICS region in which it is executing.

In a CICS open transaction environment threadsafe application program, open API task-related user exits, global user exit programs, and user-replaceable modules cannot rely on quasi-reentrancy because they can run concurrently on an open TCB. In addition, even quasi-reentrant programs are at risk if they access resources that can also be accessed by a user task running concurrently under an open TCB. This forces developers to make sure any programs that will access shared resources must take into account the possibility of simultaneous access from other programs. If programs use the correct serialization techniques when accessing shared resources they are termed threadsafe.

CICS itself will ensure that access occurs in a threadsafe manner for temporary storage queues, transient data queues, and VSAM files – CICS processing automatically ensures access in a threadsafe way. CICS ensures threadsafe access either because the CICS API code has been made threadsafe or CICS ensures it is running on the QR TCB and effectively serializes access.

Other resources, such as shared storage, become the responsibility of the user programs to ensure threadsafe processing. If user programs issue EXEC CICS commands such as ADDRESS CWA, EXTRACT EXIT, or GETMAIN SHARED, they are more than likely not threadsafe. This is because they are allowing access to global storage areas. IBM provides a sample command table named DFHEIDTH, which can be utilized with the DFHEISUP utility to scan load modules for occurrences of these commands.

DFHEIDTH does not test the programs it scans for non-threadsafe CICS commands; it determines whether the application is using CICS commands giving rise to the possibility of being non-threadsafe. There are many different techniques you can use to provide threadsafe processing when accessing a shared resource.

Even though the term ‘threadsafe’ is defined in the context of individual programs, for a user application as a whole to be threadsafe all the application programs accessing shared resources have to obey the threadsafe rules. A program written to threadsafe standards cannot safely update shared resources if another program accessing the same resources does not obey the threadsafe rules.

So how do you achieve this ‘threadsafe’ status in your applications? As mentioned earlier, IBM is recommending that all new CICS application programs be written to threadsafe standards.

The following list tells you where to look and what you need to do to be threadsafe.

1 Ensure all programs are written to current CICS standards, as documented in the *CICS Application Programming Guide*, SC34-6231.

2 Programs should be compiled and link-edited as reentrant, and reside in read-only storage. This is not an absolute requirement for threadsafe programming, but if a program is capable of overwriting itself, then the program itself is effectively shared storage, and access to it should be serialized.

3 Use only published CICS interfaces to external resources.

4 Use of the CICS Common Work Area (CWA) should be avoided where possible.

5 Programs should not create or access shared storage created by the EXEC CICS GETMAIN SHARED command.

6 Try to avoid the use of Global Work Areas (GWAs) in user exits.

7 All programs, user exits, and URMs should use only threadsafe EXEC CICS commands.

8 Ensure all programs that have been written or identified as threadsafe are defined to CICS with the CONCURRENCY(THREADSAFE) attribute.

9 Review the use of function shipping within applications.

Function ships will cause threadsafe EXEC CICS commands to become non-threadsafe.

10 Check with IBM for the latest threadsafe-related APARs.

As always, changing to a new application standard will not be easy and often the light at the end of the tunnel seems a long way off, but there are many benefits to be gained by redeveloping existing applications and ensuring that new applications are developed to the threadsafe standard. Also note that there is quite a lot of recommended maintenance to apply to both CICS and DB2 that relates to threadsafe set-up. IBM can give full information on the various APARs and related PTFs.

John Bradley
Systems Programmer
Meerkat Computer Services (UK)

CICS TS 3.1 highlights and an overview of new enhancements

The world's most popular transaction monitor, CICS, celebrated its 35th anniversary in 2004. CICS has a proven track record of successfully delivering new technology. IBM continues enhancing CICS functionality, leveraging existing investments and skills, while exploiting new technologies, and on 30 November 2004 announced a new release of CICS Transaction Server, Version 3.1, with a general availability date of 25 March 2005.

Most companies today are looking for cost reductions by developing better application processes that are easier to manage, leverage existing investments, and effectively meet their customer demands. This article will give you an overview of the new features and cover major enhancements, to help you with planning and show you how you can preserve your investment in CICS and improve integration, application transformation, and enterprise management. This article will give you an overview of Web services and Service-Oriented Architecture (SOA) exploitation by CICS, and will go over a new mechanism that is provided for inter-program data transfer without CICS size limitation. A follow-up article in next month's issue will cover all other enhancements in CICS TS 3.1 and discuss discontinued functions.

SERVICE-ORIENTED ARCHITECTURE

One of the major enhancements in this release is exploitation of Service-Oriented Architecture (SOA) to take advantage of Web services when integrating with existing legacy applications. CICS is now fully up to spec on Web services standards and can fully participate in SOA.

SOA is now a widely-accepted architectural approach, whereby an application is composed of independent, distributed, and cooperating components called services. The key concept of SOA is that the functionality implemented by a service is exposed via a standard-based interface declaration. The implementation details are hidden from the users of the service; they invoke the service based on the operations exposed in these interfaces. The industry adoption of Web services capabilities into development platforms and tools is making it easier for companies to adopt a service-based development approach. CICS exploitation of SOA can help your company build new on-demand solutions that leverage your current CICS investments.

WEB SERVICES

So, why should the Web services support be important to you?

CICS TS V3.1 enhancements to Web services support will allow your company to publish CICS applications as Web services that can be consumed by J2EE or .NET applications, and will enable CICS applications to consume external Web services. CICS now can be a full participant in the Business-to-Business world and can be a Web services provider as well as a requestor of Web services.

Web services are an emerging technology, widely used for implementing the SOA. They employ a program-to-program communication model built on existing Internet-standard eXtensible Mark-up Language (XML) for the specification of data in a platform, language, hardware delivery device, and software vendor neutral manner. Web services do not specify a particular protocol for communication, and communication layer protocols (like HTTP or JMS) can be used in the message exchange process.

Web services utilize the Web Services Description Language (WSDL) to describe content and usage, the emerging standards of Simple Object Access Protocol (SOAP) as a protocol for sending exchange messages between Web services, and the Universal Description, Discovery and Integration (UDDI) specification to allow Web providers to register their services and Web requestors to locate the appropriate services providers.

Now with CICS exploitation of SOA we can standardize on Web services interfaces throughout the corporation and use SOAP, WSDL, and other specifications as the basis of much of the IT architecture. Making use of this latest technology can give your company an opportunity to standardize, so the entire company can use the same approach or methodology.

SIMPLE OBJECT ACCESS PROTOCOL SUPPORT

The SOAP for CICS feature could be ordered with CICS TS V2.2 and V2.3 but can no longer be ordered with CICS TS V3.1. SOAP and Web services support is included in the base CICS TS 3.1 product. If you are already using the SOAP feature under CICS TS V2, your applications will continue to run under V3. IBM is advising customers to migrate to the Web services support capabilities of CICS TS V3.1, since they have major advantages over the SOAP feature support.

SOAP support under CICS has evolved in the past year and has new SOAP 1.1/1.2 infrastructure for a first-class Web Service end-point, simplification of pipeline and system management, and new tooling support for easier application development.

The SOAP and Web services capabilities are compliant with the W3C standards, but are optimized primarily for the CICS TS environment. This provides major advantages in performance, system management, and problem determination. The ability to standardize on WSDL for describing all services available in the enterprise is a major advantage enabled by embedding SOAP run-time capabilities into CICS TS and comprehensive tooling support in WSED.

A CLOSER LOOK AT CICS USING WEB SERVICES

There are three objects defining the execution environment that allow a particular CICS application program to operate Web services. These are the pipeline, the Web service binding file, and the Web service description. These objects are defined to CICS as attributes of the WEBSERVICE resource definition. WEBSERVICE resources can be dynamically installed in your running CICS system.

A pipeline defines the set of message handlers that operate on Web service requests and responses. The WEBSERVICE resource specifies a separate PIPELINE resource, which in turn specifies the pipeline configuration file. If you have been using the SOAP feature for CICS TS V2, you should be aware that the structure of the pipeline in this release of CICS is not the same as that used in the feature.

A Web service binding file contains information that is used at run time to perform the mapping between application data structures and SOAP messages. The Web service binding file is generated by the CICS-supplied tools.

A Web service description is used only when run-time validation of SOAP messages is required. Validation of each message is performed against its schema, which is embedded within the Web service description.

Figure 1 illustrates mapping the SOAP body to the application data structure in the server provider.

When a client invokes a Web service in CICS, an inbound Web service request is made that is associated with a WEBSERVICE resource by the URIMAP resource.

The URIMAP identifies the WEBSERVICE resource that applies to the URI in the inbound message; the WEBSERVICE specifies the processing that is to be performed on the message. URIMAP definitions enable CICS to match the URIs of requests from Web clients, or requests to a remote server, and provide information on how to process the requests.

A WEBSERVICE resource defines aspects of the run-time environment for a CICS application program deployed in a

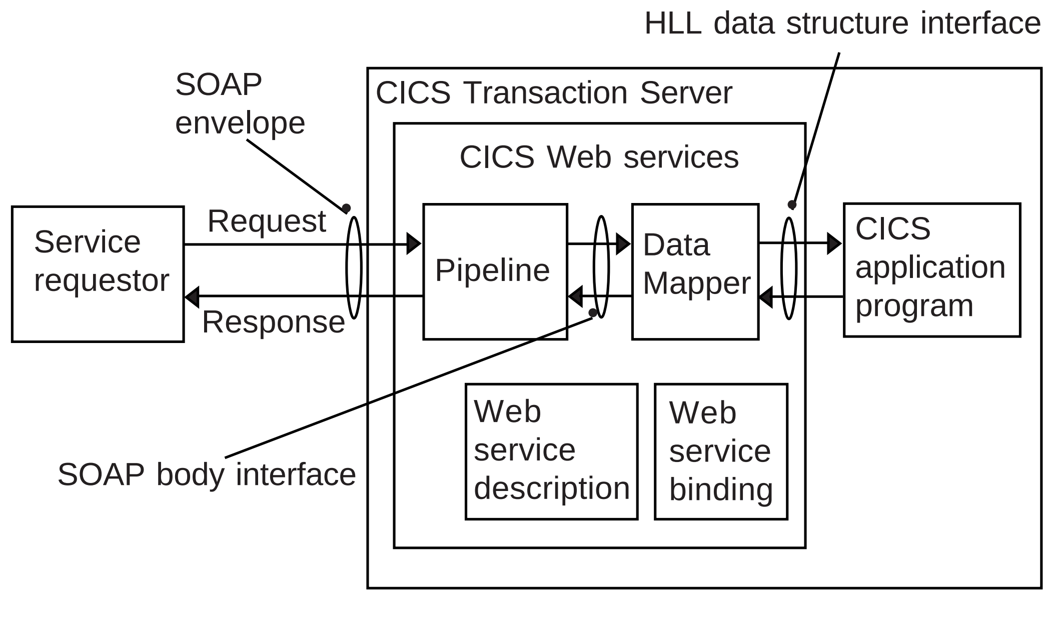


Figure 1: Mapping SOAP body to application data structure

Web services setting, where the mapping between application data structure and SOAP messages has been generated using the CICS Web services assistant.

WEB SERVICES ASSISTANT UTILITY PROGRAMS

The CICS Web services assistant can help you deploy an application with the least amount of programming effort by generating the CICS resources that you need in order to deploy your application. When your application runs, CICS transforms your application data into a SOAP message on output, and transforms the SOAP message back to application data on input.

The CICS Web services assistant can transform the data between a high-level data structure used in an application program, and the contents of the <body> element of a SOAP message. When you write your application program, you do not need to parse or construct the SOAP body; CICS will do this for you. In order to perform the mapping, CICS needs information, at run-time, about the application data structure, and about the format of the SOAP messages.

Web services assistant consists in the following utilities:

- DFHWS2LS, which takes a Web service description as a starting point. It uses the descriptions of the messages and the data types used in those messages to construct high-level language data structures that you can use in your application programs.
- DFHLS2WS, which takes a high-level language data structure as a starting point. It uses the structure to construct a Web services description containing descriptions of messages and the data types used in those messages derived from the language structure.

Both utility programs generate a Web services binding file that CICS uses at run-time to perform the mapping between the application program's data structures and the SOAP messages.

SAMPLE PROGRAMS UPDATED

The following programs have been updated to support the PIPELINE and WEBSERVICE resources:

- DFH0STAT – the sample statistics program.
- DFH$FOR – the DB2 formatting sample program (Assembler).
- DFH$FORP – the DB2 formatting sample program (PL/I).
- DFH0FORC – the DB2 formatting sample program (COBOL).
- DFH$DB2T – the DB2 table definitions for DFH$FORA, DFH$FORP, and DFH0FORC.
- DFH$SQLT – input for the DB2 table load utility.

STATEMENTS OF DIRECTION

IBM announced, as a statement of direction, an intention to add CICS TS 3.1 support to WebSphere Studio Enterprise Developer (WSED) during 2005. This will enable developers with skills in COBOL, PL/I, Java, and Web services to easily reuse, build, and deploy components that integrate into an enterprise-wide SOA. WSED will provide the visual development environment supporting Web services, SOAP for CICS, and aggregation of CICS resources.

IBM views WSED as a strategic development environment, now with added CICS TS V3 support. An optimized CICS data exchange capability and the ability to use a single development tool like WSED will provide enhanced application transformation capabilities and increase developer productivity.

New application development tools will extend WSED to: ‘Enable the composition of CICS application assets to form business service functions that can be exposed as Web services. This will enable an external business process engine, such as WebSphere Business Integration Server Foundation, to externally orchestrate business service functions implemented in CICS. Customers extending the use of their CICS applications can do so in a service-oriented manner, integrating their CICS investments with more parts of the business across an Enterprise Service Bus.

‘It will also include BMS Map editing and Enterprise Generation Language (EGL) generation supporting the conversion of VisualAge Generator Web transactions. In addition it will provide a batch program for use by automated software build procedures, such as JCL, which will input the XML Schema Definition (XSD) or language structure declaration to generate client Web Services Description Language (WSDL) and converters for the CICS Web services implementation. This batch program will be made available early by download for use with CICS TS V3.1’

NO MORE COMMAREA SIZE LIMITATION

Another much anticipated enhancement in CICSTS 3.1 is the allowing of the exchange of data that isn’t limited to a 32KB CICS COMMAREA. A new mechanism is provided for inter-program data transfer without CICS size limitation. This is an alternative to using a communication area (COMMAREA), which has 32KB restriction. This enhanced data transfer requires minimal application changes for exploitation and allows better structuring of application data. With this enhancement CICS can pass large payloads associated with XML-heavy Web services standards.

CICS TS 3.1 introduces containers and channels, which provide flexible and a more structured method of passing data between programs. Any number of containers can be passed between programs. Containers are grouped together in named channels. Standard CICS mechanisms for exchanging data between programs are supported with channels. A channel can be passed using EXEC CICS LINK, XCTL, START, and RETURN commands, and can be used by applications written in any of the programming languages supported by CICS.

There are several advantages to using the channel/container model over the COMMAREAs for exchanging data within CICS programs:

- This implementation removes the need for programs to know the exact size of the data returned and simplifies coding interfaces between programs.
- Programmers now are relieved of storage management concerns, because containers going out of scope are automatically destroyed. It should be noted that although there is no specific limit on the size of data that can be passed, the size of a container is limited by the amount of storage available. There is no limit to the number of containers that can be added to a channel.
- Using channels instead of a COMMAREA can reduce the amount of storage needed per transaction.
- The risk of getting a storage violation is greatly reduced. Programs utilizing containers use any standard Application Programming Interface (API) to get and put to containers, and should not get any storage violations. There is no chance of overriding another container.
- IBM has no plans to enlarge the COMMAREA, so using containers better positions developers to accommodate future application changes that require large amounts of data to be passed.
- The data conversion model used by channel applications is simpler than that used by COMMAREA applications. In channel applications, conversion is controlled by the application programmer using simple API commands. In applications using a COMMAREA, application data conversion is controlled by the systems programmer.

It is easy to migrate your current applications that use COMMAREA to use the channels. IBM recommends replacing a COMMAREA with a channel with a single container. For example, if your current application performs a LINK command, you would replace it as follows:

- Existing code for Program#1:

```
EXEC CICS LINK PROGRAM(myprogram) COMMAREA(structure)
```

Replace with:

```
EXEC CICS PUT CONTAINER(structure-name) CHANNEL(channel-name)
FROM(structure)
EXEC CICS LINK PROGRAM(myprogram) CHANNEL(channel-name)
...
EXEC CICS GET CONTAINER(structure-name) CHANNEL(channel-name)
INTO(structure)
```

Here is another example on using containers to START a transaction with data:

- Existing code for Program#2:

```
EXEC CICS START TRANSID(ABCD) FROM(structure)
```

Replace with:

```
EXEC CICS PUT CONTAINER(structure-name) CHANNEL(channel-name)
FROM(structure)
EXEC CICS START TRANSID(ABCD) CHANNEL(channel-name)
```

- Existing code for Program#3:

```
EXEC CICS RETRIEVE INTO(structure)
```

Replace with:

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

There are two sample C language programs from the IIOP Bank Account (transaction BNKQ) that show how a channel can be passed on EXEC CICS LINK and RETURN commands. DFH$IIBQ is the top-level program, and DFH$IICC is the program it links to. These sample programs use the channel and container model instead of the COMMAREA model.

Please note that if you are passing data on a DPL call between CICS systems connected by either MRO or ISC, both CICS regions should be at the CICS TS 3.1 level; otherwise, additional maintenance is required.

If the remote region is not another CICS TS 3.1 system, to get it to support channels you have to apply one of the APARs listed below, applicable to your remote environment:

- CICS Transaction Server for z/OS, Version 2 Release 3 – APAR PQ92437.
- CICS Transaction Server for z/OS, Version 2 Release 2 – APAR PQ92437.
- CICS Transaction Server for OS/390, Version 1 Release 3 – APAR PQ93048.
- CICS Transaction Sever for VSE/ESA Release 1.1 – APAR PQ83049.

The follow-up article will cover all other enhancements in CICS TS 3.1 and discontinued functions.

Elena Nanos
IBM Certified Solution Expert in CICS Web Enablement and MQSeries
Zurich NA (USA)

Threadsafe via autoinstall

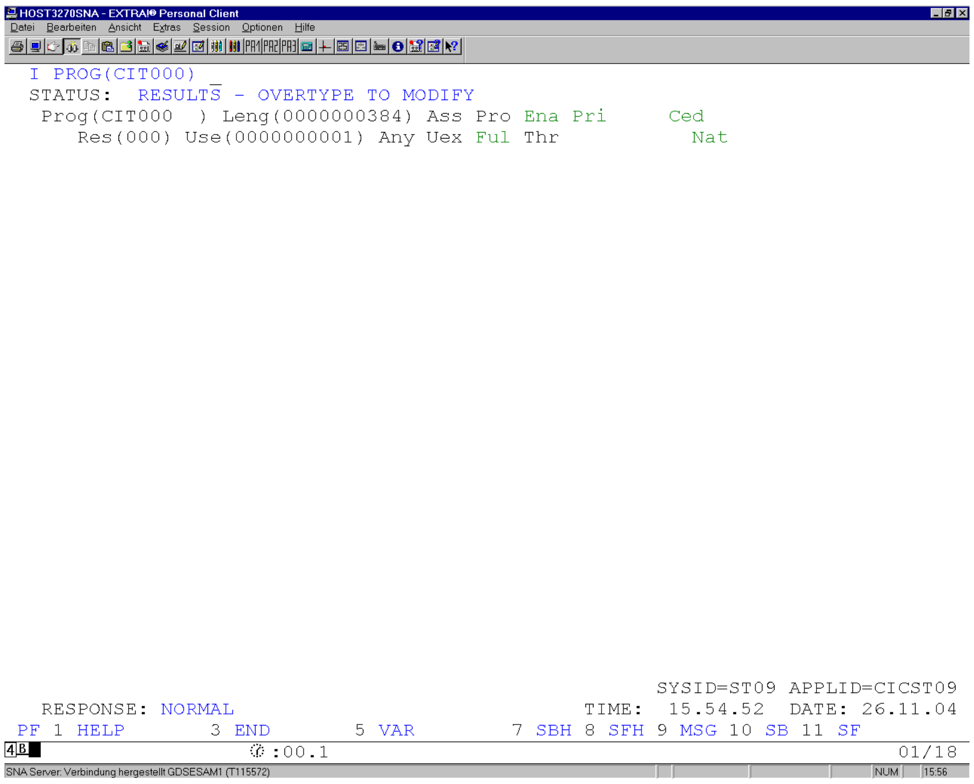
Previous issues of *CICS Update* have discussed how programs can be autoinstalled quickly, safely, and with ease.

In the meantime IBM has made available threadsafe technology, so CPU consumption can be reduced enormously and performance improved drastically.

So long as the program is threadsafe (which is far less complicated than was originally assumed!) it can be defined as such in CICS via autoinstall with the simple but important instruction:

```
MVI    PGAC_CONCURRENCY,PGAC_THREADSAFE
```

The autoinstall program we use is a customized version of


HOST3270SNA - EXTRA!® Personal Client
Datei Bearbeiten Ansicht Extras Session Optionen Hilfe

```
  I PROG(CIT000) _
  STATUS:  RESULTS - OVERTYPE TO MODIFY
   Prog(CIT000  ) Leng(0000000384) Ass Pro Ena Pri      Ced
      Res(000) Use(0000000001) Any Uex Ful Thr            Nat





                                                   SYSID=ST09 APPLID=CICST09
   RESPONSE: NORMAL                          TIME:  15.54.52  DATE: 26.11.04
 PF 1 HELP       3 END       5 VAR        7 SBH 8 SFH 9 MSG 10 SB 11 SF
4B                      :00.1                                        01/18
SNA Server: Verbindung hergestellt GDSESAM1 (T115572)               NUM  15:56
```

Figure 1: Program CIT000 after autoinstall

IBM's DFHPGADX, which you can find in your SDFHSAMP library.

With the CEMT instruction you can verify whether the threadsafe attribute is set or not.

The DFHPG0209 message gives additional information.

Figure 1 shows program CIT000 after autoinstall via NLVPGADX with attribute THREADSAFE.

NLVPGADX

```
***********************************************************************
*                                                                     *
* MODULE NAME = NLVPGADX                                              *
*                                                                     *
* DESCRIPTIVE NAME = CTS         Program Autoinstall program exit     *
*                                                                     *
*                                                                     *
*   COPYRIGHT = 5655-Ø18 (C) COPYRIGHT IBM CORP. 1994                 *
*               THIS MODULE IS "RESTRICTED MATERIALS OF IBM"          *
*               LICENSED MATERIALS - PROPERTY OF IBM                  *
*               REFER TO COPYRIGHT INSTRUCTIONS                       *
*               FORM NUMBER G12Ø-2Ø83                                 *
*                                                                     *
* STATUS = CTS 2.3.Ø                                                  *
*                                                                     *
* FUNCTION = Provides user input for the program autoinstall function *
*                                                                     *
*        There are ASM, PL/I, COBOL and C versions of this program.   *
*                                                                     *
*        This program is a sample version in Assembler of the program *
*        autoinstall exit. The program is invoked when a program      *
*        is being autoinstalled on behalf of the user and the         *
*        autoinstall exit name is set to the default, NLVPGADX.       *
*        The exit may be used to specify requirements for the         *
*        program definition.                                          *
*                                                                     *
*        A parameter list is provided as input to the program. The    *
*        parameter list is passed to the program via the COMMAREA.    *
*        The parameter list is defined in DFHPGACD.                   *
*        The parameter list is addressed by the program using the     *
*        normal conventions for a COMMAREA.                           *
*                                                                     *
*        The parameter list specifies the name of the program to be   *
*        autoinstalled and the module type. The user may use the      *
*        parameter list to return information for the program to be   *
*        autoinstalled. The user may also indicate using the          *
*        return_code parameter that the program should not be         *
*        defined.                                                     *
*                                                                     *
* NOTES :                                                             *
*                                                                     *
*    THIS IS A PRODUCT SENSITIVE SAMPLE.                              *
*    REFER TO PRODUCT DOCUMENTATION.                                  *
*                                                                     *
*    DEPENDENCIES = S/39Ø                                             *
*    MODULE TYPE = Executable                                         *
*    PROCESSOR = Assembler                                            *
*    ATTRIBUTES = Read only, Serially Reusable                        *
*                                                                     *
*---------------------------------------------------------------------*
*                                                                     *
* ENTRY POINT = NLVPGADX                                              *
*                                                                     *
*     PURPOSE = All functions                                         *
*                                                                      *
*     LINKAGE =                                                        *
*         This entry point is called by the autoinstall function       *
*         to link to the program autoinstall exit program.             *
*         The parameters are passed to the exit program via the        *
*         COMMAREA. The control block for the parameter list is in     *
*         DFHPGACD.                                                    *
*                                                                      *
*     INPUT =                                                          *
*         The input parameters provide the user with the name          *
*         and module type for the program to be autoinstalled.         *
*         The following input parameters are passed to the program     *
*         via the COMMAREA:                                            *
*         PGAC_PROGRAM        - name of program to be autoinstalled    *
*         PGAC_MODULE_TYPE    - program, mapset or partitionset        *
*                                                                      *
*     OUTPUT =                                                         *
*         The output parameters may be used to specify user            *
*         requirements for the program definition.                     *
*         The following output parameters may be returned to the       *
*         autoinstall function via the COMMAREA:                       *
*         PGAC_MODEL_NAME     - autoinstall model program name         *
*         PGAC_LANGUAGE       - Assembler, COBOL, C370, LE370, PL/I    *
*         PGAC_CEDF_STATUS    - CEDF status, yes or no                 *
*         PGAC_DATA_LOCATION  - data location, below or any            *
*         PGAC_EXECUTION_KEY  - execution key, CICS or user            *
*         PGAC_LOAD_ATTRIBUTE - reload, transient, resident, reuseable*
*         PGAC_USE_LPA_COPY   - use LPA copy, yes or no                *
*         PGAC_EXECUTION_SET  - use DPL subset or full API             *
*         PGAC_REMOTE_SYSID   - remote system ID                       *
*         PGAC_REMOTE_PROGID  - remote program name                    *
*         PGAC_REMOTE_TRANSID - remote transaction ID                  *
*                                                                      *
*     EXIT-NORMAL = Exit is via an EXEC CICS RETURN command.           *
*         The following return codes may be returned via the           *
*         COMMAREA:                                                    *
*         PGAC_RETURN_CODE = PGAC_RETURN_OK                            *
*         PGAC_RETURN_CODE = PGAC_RETURN_DONT_DEFINE_PROGRAM           *
*                                                                      *
*     EXIT-ERROR =                                                     *
*         If the program abends, an error response is returned         *
*         to the autoinstall function. A message is issued by the      *
*         autoinstall function and the autoinstall function  is        *
*         disabled.                                                    *
*                                                                      *
*--------------------------------------------------------------------- *
*                                                                      *
* EXTERNAL REFERENCES =                                                *
*         None.                                                        *
*                                                                      *
*     ROUTINES =                                                      *
*         EXEC CICS RETURN - return to the calling program.           *
*                                                                     *
*     CONTROL BLOCKS =                                                *
*         The PGAC control block, which includes the input and        *
*         output parameters, is in DFHPGACD.                          *
*         See INPUT and OUTPUT description above for a description    *
*         of the parameters.                                          *
*                                                                     *
*-------------------------------------------------------------------- *
*                                                                     *
* DESCRIPTION                                                         *
*         The default program autoinstall exit simply sets the        *
*         return code to OK and returns.                              *
*         The user may customize this program to provide information  *
*         for the autoinstalled definition based on the program       *
*         name and the module type.                                   *
*                                                                     *
*-------------------------------------------------------------------- *
*                                                                     *
* CHANGE ACTIVITY :                                                   *
*                                                                     *
*        $MOD(NLVPGADX) COMP(PROGRAM) PROD(CICS/ESA):                 *
*                                                                     *
*     PN= REASON REL YYMMDD HDXXIII : REMARKS                         *
*    $LØ= 646    41Ø 93Ø222 HDBVDMC : Program Autoinstall             *
*    $P1= M83159 41Ø 93Ø713 HDBVDMC : M83159: DSECTGEN changes        *
*                                                                     *
*   Ø1-Ø1  in 2ØØ3    cr   Define programs as threadsafe.             *
*   nn-nn  tt.mm.jj   ??   ?????????????????????????????????????????? *
*   nn-nn  tt.mm.jj   ??   ?????????????????????????????????????????? *
*   nn-nn  tt.mm.jj   ??   ?????????????????????????????????????????? *
*                                                                     *
***********************************************************************
***********************************************************************
DFHEISTG DSECT  ,
*
*    Insert your own storage definitions here
*
PPRIVATE DS    ØCL2Ø
PSYSID   DS    CL4
PSERVICE DS    CL1
PRESP    DS    F
         ORG   PPRIVATE+2Ø
*
*    Copy the commarea definitions
*
         COPY DFHPGACD                Autoinstall commarea
*
***********************************************************************
```

```
NLVPGADX CSECT
NLVPGADX AMODE 31
NLVPGADX RMODE ANY
         DFHREGS ,
*
*        If there is no commarea, return
*
         OC    EIBCALEN,EIBCALEN
         BZ    RETURNØ
*
*        Address the commarea
*
         L     R2,DFHEICAP
         USING PGAC,R2
*
*        Add user specific code here
*
         CLI   PGAC_MODULE_TYPE,PGAC_TYPE_PARTITIONSET
         BE    RETURNDD                 Accept only programs and maps
*
*       MVI   PGAC_RETURN_INFORMATION,C' '          Clear Output-Area
*       MVC   PGAC_RETURN_INFORMATION+1(L'PGAC_RETURN_INFORMATION-1),P*
*              GAC_RETURN_INFORMATION
*
         MVI   PPRIVATE,X'ØØ'           FORMAT WORKINGSET
         MVC   PPRIVATE+1(L'PPRIVATE-1),PPRIVATE
*
*        Assign the sysid
*
         EXEC CICS ASSIGN SYSID(PSYSID) RESP(PRESP)
         CLC   PRESP,DFHRESP(NORMAL)    ANY ERRORS DETECED ?
         BNE   RETURNDD                 IF YES: DON'T AUTOINSTALL
*
         LA    R7,SERTAB          LOAD SERVICE-TAB
*
SERV1ØØØ DS    ØH
         CLI   Ø(R7),C'*'         END OF TABLE ?
         BE    SERV9ØØØ           YES: IT'S NOT A SERIVCE-CICS
         CLC   PSYSID+2(1),Ø(R7)  ENTRY IN TABLE ?
         BE    SERV19ØØ           YES: IT'S A SERVICE-CICS
         LA    R7,L'SERTAB(R7)    NEXT ENTRY
         B     SERV1ØØØ
*
SERV19ØØ DS    ØH
         MVI   PSERVICE,C'1'
*
SERV9ØØØ DS    ØH
*
*                                       ---- Program ------------
*
```

```
         CLI   PSERVICE,C'1'            is it a service-cics ?
         BNE   PGMØ5ØØ                  no... load aor-table
         LA    R7,PGMTABS               first entry in table
         LA    R1Ø,PGMCNTS              number of programs
         B     PGM1ØØØ
*
PGMØ5ØØ  DS    ØH
         LA    R7,PGMTAB                first entry in table
         LA    R1Ø,PGMCNT               number of programs
*
PGM1ØØØ  DS    ØH
         LA    R8,7                     max.-length -1 (EX!)
         LR    R9,R7                    addr. r9 eq addr. r7
         LA    R9,7(R9)                 last possible character
*
PGM2ØØØ  DS    ØH
         CLI   Ø(R9),C' '               true only gt blank
         BH    PGM3ØØØ                  if gt..compare
         BCTR  R9,Ø                     next column
         BCT   R8,PGM2ØØØ               go on
*
PGM3ØØØ  DS    ØH
         EX    R8,COMPPGM               compare
         BE    PROCESS                  yes... go on
         LA    R7,L'PGMTAB(,R7)         next entry
         BCT   R1Ø,PGM1ØØØ              go on
*
*                                       ---- Maps ---------------
*
         CLI   PSERVICE,C'1'            is it a service-cics ?
         BNE   MAPØ5ØØ                  no... load aor-table
         LA    R7,MAPTABS               first entry in table
         LA    R1Ø,MAPCNTS              number of maps
         B     MAP1ØØØ
*
MAPØ5ØØ  DS    ØH
         LA    R7,MAPTAB                first entry in table
         LA    R1Ø,MAPCNT               number of maps
*
MAP1ØØØ  DS    ØH
         LA    R8,7                     max.-length -1 (EX!)
         LR    R9,R7                    addr. r9 eq addr. r7
         LA    R9,7(R9)                 last possible character
*
MAP2ØØØ  DS    ØH
         CLI   Ø(R9),C' '               true only gt blank
         BH    MAP3ØØØ                  if gt..compare
         BCTR  R9,Ø                     next column
         BCT   R8,MAP2ØØØ               go on
*
```

```
MAP3ØØØ  DS    ØH
         EX    R8,COMPPGM                compare
         BE    MAP                       yes... go on
         LA    R7,L'MAPTAB(,R7)          next entry
         BCT   R1Ø,MAP1ØØØ               go on
         B     RETURNDD                  no map ... goback
*
MAP      DS    ØH
         MVC   PGAC_MODEL_NAME,=CL8'DFHPGAMP'       SET DEFAULT-TYPE
         B     RETURNOK
*
PROCESS  DS    ØH
         MVC   PGAC_MODEL_NAME,=CL8'DFHPGAPG'       SET DEFAULT-TYPE
         CLC   PGAC_PROGRAM(3),=C'ZSS'              NO EDF IF
         BNE   PROCØ5ØØ                             IT IS ZSS
         MVI   PGAC_CEDF_STATUS,PGAC_CEDF_NO        "CEDF = NO"
         B     PROCØ7ØØ
*
PROCØ5ØØ DS    ØH
         MVI   PGAC_CEDF_STATUS,PGAC_CEDF_YES       "CEDF = YES"
*
PROCØ7ØØ DS    ØH
         MVI   PGAC_DATA_LOCATION,PGAC_LOCATION_ANY TASKDATALOC=ANY
         MVI   PGAC_EXECUTION_KEY,PGAC_USER_KEY     "EXECKEY=USER"
         MVI   PGAC_CONCURRENCY,PGAC_THREADSAFE     THREADSAFE
         LA    R7,RESTAB                            LOAD FIRST ENTRY
*
PROC1ØØØ DS    ØH
         CLI   Ø(R7),C'*'         END OF TABLE ?
         BE    RETURNOK           YES: LOAD WITH "RESIDENT=NO"
         CLC   PGAC_PROGRAM,Ø(R7) PROGRAM IN TABLE ?
         BE    PROC19ØØ           YES: LOAD WITH "RESIDENT=YES"
         LA    R7,L'RESTAB(R7)    NEXT ENTRY
         B     PROC1ØØØ
*
COMPPGM  CLC   PGAC_PROGRAM(Ø),Ø(R7)                PROGRAM TRUE?
*
PROC19ØØ DS    ØH                 LOAD PROGRAM RESIDENT
         MVI   PGAC_LOAD_ATTRIBUTE,PGAC_RESIDENT
*
*        Set the return code to OK
*
RETURNOK DS    ØH
         MVI   PGAC_RETURN_CODE,PGAC_RETURN_OK
         B     RETURNØ
*
*        Branch to this label if you elect not to define the program
*
RETURNDD DS    ØH
         MVI   PGAC_RETURN_CODE,PGAC_RETURN_DONT_DEFINE_PROGRAM
```

```
*
RETURNØ  DS    ØH
         EXEC CICS RETURN
         EJECT
*
* PRIVATE DEFINITIONS
*
RESTAB   DS    ØCL8                        TABLE FOR PROGRAMS, WHICH
         DC    CL8'CI119Ø  '               MUST BE LOADED RESIDENT !
         DC    C'*'
*
SERTAB   DS    ØCL1                        SERVICE-CICS-TABLE
*        DC    C'D'                        DOR
         DC    C'T'                        TOR
         DC    C'V'                        FOR
         DC    C'*'
*
PGMTAB   DS    ØCL8                        TABLE FOR PROGRAMS, WHICH
         DC    CL8'CE      '               LE                      "
         DC    CL8'CI      '               NLV                     "
         DC    CL8'CN      '               NLV                     "
         DC    CL8'CSFTT   '               Subroutine for FTP      "
*        DC    CL8'CSO     '               VisualAge               "
         DC    CL8'CT      '               NLV                     "
         DC    CL8'DN      '               K-DIALOG("-NEU")        "
         DC    CL8'DOPDLG  '               DOPE (ITV-GA)           "
         DC    CL8'DZ      '               K-DIALOG("-NEU")        "
         DC    CL8'DØ      '               K-DIALOG("-NEU")        "
         DC    CL8'D1      '               K-DIALOG("-NEU")        "
         DC    CL8'D2      '               K-DIALOG("-NEU")        "
         DC    CL8'D3      '               K-DIALOG("-NEU")        "
         DC    CL8'D4      '               K-DIALOG("-NEU")        "
         DC    CL8'D5      '               K-DIALOG("-NEU")        "
         DC    CL8'D6      '               K-DIALOG("-NEU")        "
         DC    CL8'D7      '               K-DIALOG("-NEU")        "
         DC    CL8'D8      '               K-DIALOG("-NEU")        "
         DC    CL8'D9      '               K-DIALOG("-NEU")        "
         DC    CL8'EDC     '               C                       "
*        DC    CL8'ELA     '               VisualAge               "
*        DC    CL8'EZE     '               VisualAge               "
         DC    CL8'FSN     '               ASF                     "
         DC    CL8'HM      '               KEYFAST                 "
         DC    CL8'IBM     '               PL1                     "
         DC    CL8'IED     '               C                       "
         DC    CL8'IGZ     '               COBOL                   "
         DC    CL8'IIBM    '               PL1                     "
         DC    CL8'IIGZ    '               COBOL                   "
         DC    CL8'KA      '               CSP 4.1                 "
         DC    CL8'KL      '               CSP 4.1                 "
         DC    CL8'KS      '               CSP 4.1                 "
         DC    CL8'KT      '               CSP 4.1              "
         DC    CL8'KU      '               CSP 4.1              "
         DC    CL8'MCP     '               KOSSY                "
         DC    CL8'MS      '               KOSSY                "
         DC    CL8'NKV     '               CSP 4.1              "
         DC    CL8'OC      '               KOSSY                "
         DC    CL8'PKR     '               NKV                  "
         DC    CL8'PSZ     '               NKV                  "
         DC    CL8'PTS     '               TABSYS               "
         DC    CL8'PZB     '               NKV                  "
         DC    CL8'TCP     '               KOSSY                "
         DC    CL8'TSD     '               KOSSY                "
         DC    CL8'ZB      '               CSP 4.1              "
         DC    CL8'ZSS     '               ZSS                  "
PGMTABE  DC    C'*'                        ARE SUPPORTED BY AUTOINST.!
PGMCNT   EQU   (PGMTABE-PGMTAB)/8
*
MAPTAB   DS    ØCL8                        TABLE FOR MAPS,     WHICH
         DC    CL8'CM      '               NLV-MAP'S            "
         DC    CL8'CP      '               KOSSY                "
         DC    CL8'SD      '               KOSSY                "
         DC    CL8'SU      '               KOSSY                "
         DC    CL8'TS      '               TABSYS               "
MAPTABE  DC    C'*'                        ARE SUPPORTED BY AUTOINST.!
MAPCNT   EQU   (MAPTABE-MAPTAB)/8
*
PGMTABS  DS    ØCL8                        TABLE FOR PROGRAMS, WHICH
         DC    CL8'CE      '               LE                   "
         DC    CL8'EDC     '               C                    "
         DC    CL8'IBM     '               PL1                  "
         DC    CL8'IED     '               C                    "
         DC    CL8'IGZ     '               COBOL                "
         DC    CL8'IIBM    '               PL1                  "
         DC    CL8'IIGZ    '               COBOL                "
         DC    CL8'ZSSØØ1  '               ZSS                  "
         DC    CL8'ZSSØØ5  '               ZSS                  "
         DC    CL8'ZSSØØ6  '               ZSS                  "
PGMTABES DC    C'*'                        ARE SUPPORTED BY AUTOINST.!
PGMCNTS  EQU   (PGMTABES-PGMTABS)/8
*
MAPTABS  DS    ØCL8                        TABLE FOR MAPS,     WHICH
         DC    CL8'CMZ     '               ZSS                  "
         DC    CL8'********'               NLV-MAP'S            "
MAPTABES DC    C'*'                        ARE SUPPORTED BY AUTOINST.!
MAPCNTS  EQU   (MAPTABES-MAPTABS)/8
*
         END   NLVPGADX
```

DFHPG0209 MESSAGES IN CICS

```
DFHPGØ2Ø9 26/11/2ØØ4 15:54:46 CICSTØ9 TAFB BØØ1475 TØØØ PPT entry for
CITØØØ has been autoinstalled using model DFHPGAPG.
```

Claus Reis
CICS Systems Programmer
Nuernberger Lebensversicherung AG (Germany)

External CICS Interface (EXCI) client interface to DFHEDAP – part 2

This month we conclude the code for the EXCI client interface to DFHEDAP, which is an example of how easy it is to write an EXCI client program that is controlled by CM420.

```
***********************************************************************
*-------------------------------------------------------------------*
*- A 1 Ø Ø _ I N I T I A L I Z E :  Perform Initialization          -*
*-------------------------------------------------------------------*
***********************************************************************
*- R2   DSECT - CM411COM                                             -*
*- R3   BASE CSECT CM411                                             -*
*- R4   ---> DYNSTOR                                                 -*
*- R5   Length of DYNSTOR                                            -*
*- R6   Work Register                                                -*
*- R7   Work Register                                                -*
*- R8   DSECT - CM4Ø1COM                                             -*
*- R9   DSECT - CM4Ø2COM                                             -*
*- R1Ø  DSECT - CM4Ø3COM                                             -*
*- R11  EIBREG DSECT - DFHEIBLK                                      -*
*- R13  DYNREG DSECT - DFHEISTG                                      -*
*- R14  Linkage                                                      -*
***********************************************************************
A1ØØ_INITIALIZE  DS   ØH
*-------------------------------------------------------------------*
*- Clear DYNSTOR.                                                    -*
*-------------------------------------------------------------------*
         LA    R4,DYNSTOR              ADDRESS DYNSTOR
         LA    R5,DYNSTORL             LENGTH OF DYNSTOR
         XR    R6,R6                   FROM ADDRESS NOT REQUIRED
         XR    R7,R7                   SET LENGTH TO Ø
         ICM   R7,8,=C' '              SET PADDING TO BLANKS
         MVCL  R4,R6                   CLEAR STORAGE TO BLANKS
*
         ST    R14,A1ØØSR14            SAVE REGISTER 14
         MVI   DUMP_REQD,FLAG_OFF      INITIALIZE DUMP REQD FLAG
*--------------------------------------------------------------------*
*- Address CICS EIB                                                 -*
*--------------------------------------------------------------------*
EIBADD   EXEC  CICS ADDRESS                                            X
               EIB(EIBREG)
*--------------------------------------------------------------------*
*- Establish addressability and map CM4Ø1COM, CM4Ø2COM and CM4Ø3COM  -*
*--------------------------------------------------------------------*
         LA    R8,CM4Ø1ST              ADDRESS CM4Ø1COM
         USING CM4Ø1COM,R8             MAP CM4Ø1COM
         MVC   CM4Ø1COM_EYECATCH,=C'<<  CM4Ø1COM  >>'  EYECATCHER
         LA    R9,CM4Ø2ST              ADDRESS CM4Ø2COM
         USING CM4Ø2COM,R9             MAP CM4Ø2COM
         MVC   CM4Ø2COM_EYECATCH,=C'<<  CM4Ø2COM  >>'  EYECATCHER
         LA    R1Ø,CM4Ø3ST             ADDRESS CM4Ø3COM
         USING CM4Ø3COM,R1Ø            MAP CM4Ø3COM
         MVC   CM4Ø3COM_EYECATCH,=C'<<  CM4Ø3COM  >>'  EYECATCHER
*--------------------------------------------------------------------*
*- Establish USER, TERMINAL, and TRANSACTION Identifiers            -*
*--------------------------------------------------------------------*
A1ØØASGN EXEC  CICS ASSIGN                                             X
               USERID(USER_ID)                                         X
               RESP(CM4Ø2COM_RESP1)                                    X
               RESP2(CM4Ø2COM_RESP2)
*
         MVC   CM4Ø2COM_FUNC(L'CM4Ø2COM_FUNC),EIBFN   MOVE FUNCTION
         CLC   CM4Ø2COM_RESP1,DFHRESP(NORMAL)    NORMAL COMPLETION?
         BE    A1ØØTERM                YES - OBTAIN TERMINAL-ID
*                                      NO  - PROCESS ERROR
         BAL   R14,Z1ØØ_CICS_DIAGS     PERFORM CICS DIAGNOSTICS
         MVC   CM4Ø1COM_MSGNO,=CL(L'CM4Ø1COM_MSGNO)'CM411Ø1ØE' NUMBER
         MVC   CM4Ø1COM_TEXT(L'CM411Ø1ØE),CM411Ø1ØE            TEXT
         BAL   R14,Z2ØØ_ISSUE_MESSAGE  ISSUE MESSAGE
         B     A1ØØABND                ABEND TRANSACTION
*
A1ØØTERM MVC   TERM_ID(L'TERM_ID),EIBTRMID    MOVE TERMINAL IDENTIFIER
A1ØØTRAN MVC   TRAN_ID(L'TRAN_ID),EIBTRNID    MOVE TRANSACTION ID.
*--------------------------------------------------------------------*
*- Issue start of program messages to TDQ.                          -*
*--------------------------------------------------------------------*
A1ØØMSGS MVC   CM4Ø1COM_MSGNO,=CL(L'CM4Ø1COM_MSGNO)'CM411ØØØI' NUMBER
         MVC   CM4Ø1COM_TEXT(ASMEYEL),ASMEYE                   TEXT
         BAL   R14,Z2ØØ_ISSUE_MESSAGE  ISSUE MESSAGE
*
         MVC   CM4Ø1COM_MSGNO,=CL(L'CM4Ø1COM_MSGNO)'CM411ØØ1I' NUMBER
         MVC   CM4Ø1COM_TEXT(L'CM411ØØ1I),CM411ØØ1I            TEXT
         MVC   CM4Ø1COM_TEXT+12(L'TRAN_ID),TRAN_ID         ADD TRANID
         MVC   CM4Ø1COM_TEXT+22(L'USER_ID),USER_ID         ADD USERID
         MVC   CM4Ø1COM_TEXT+4Ø(L'TERM_ID),TERM_ID         ADD TERMID
         BAL   R14,Z2ØØ_ISSUE_MESSAGE  ISSUE MESSAGE
*----------------------------------------------------------------------*
*- Check that a COMMAREA has been provided, if it isn't present      -*
*- issue an error message and then abend. If a COMMAREA is           -*
*- present then map it using the DSECT CM411COM.                     -*
*----------------------------------------------------------------------*
A1ØØCOMM DS    ØH                      CHECK COMMAREA
         OC    EIBCALEN,EIBCALEN       ANY COMMAREA?
         BNZ   A1ØØACOM                YES - ADDRESS COMMAREA
*                                      NO  - BUILD MESSAGE
         MVC   CM4Ø1COM_MSGNO,=CL(L'CM4Ø1COM_MSGNO)'CM411ØØ3E' NUMBER
         MVC   CM4Ø1COM_TEXT(L'CM411ØØ3E),CM411ØØ3E            TEXT
         BAL   R14,Z2ØØ_ISSUE_MESSAGE  ISSUE ERROR MESSAGE
A1ØØABND MVC   ABNDCODE,=C'A1ØØ'       SET ABEND CODE
         B     Z9ØØ_ABEND_TRAN         ABEND TRANSACTION
A1ØØACOM DS    ØH                      ADDRESS AND MAP CM411 COMMAREA
         L     R2,DFHEICAP             ADDRESS COMMAREA
         USING CM411COM,R2             MAP COMMAREA
         MVC   CM4Ø2COM_EYECATCH,=C'<<  CM411COM  >>'  EYECATCHER
*
         XC    CM411COM_CM411_RC,CM411COM_CM411_RC  INITIALIZE RC
         XC    CM411COM_DIAG_RC,CM411COM_DIAG_RC    INITIALIZE RC
         XC    CM411COM_EXEC_RC,CM411COM_EXEC_RC    INITIALIZE RC
*----------------------------------------------------------------------*
*- Return to caller                                                  -*
*----------------------------------------------------------------------*
A1ØØRET  L     R14,A1ØØSR14            RESTORE REGISTER 14
         BR    R14                     RETURN TO CALLER
         EJECT
************************************************************************
*----------------------------------------------------------------------*
*- A 5 Ø Ø _ L I N K _ D F H E D A P  :  CEDA Processing             -*
*----------------------------------------------------------------------*
************************************************************************
*- R2   DSECT - CM411COM                                             -*
*- R3   BASE CSECT CM411                                             -*
*- R4   Work Register                                                -*
*- R8   DSECT - CM4Ø1COM                                             -*
*- R9   DSECT - CM4Ø2COM                                             -*
*- R1Ø  DSECT - CM4Ø3COM                                             -*
*- R11  EIBREG DSECT - DFHEIBLK                                      -*
*- R12  DSECT - CM411COM_STG_HEADER                                  -*
*- R13  DYNREG DSECT - DFHEISTG                                      -*
*- R14  Linkage                                                      -*
************************************************************************
A5ØØ_LINK_DFHEDAP  DS  ØH
*
```

```
         ST    R14,A5ØØSR14            SAVE REGISTER 14
*----------------------------------------------------------------------*
*- Set required address pointers for link to DFHEDAP                 -*
*----------------------------------------------------------------------*
A5ØØPTRS XR    R4,R4                   CLEAR REGISTER
         LA    R4,CM411COM_COMMAND     ADDRESS COMMAND
         ST    R4,CM411COM_COMMAND_PTR STORE POINTER
         LA    R4,CM411COM_COM_LNG     ADDRESS COMMAND LENGTH
         ST    R4,CM411COM_COM_LNG_PTR STORE POINTER
         LA    R4,CM411COM_FLAG        ADDRESS OUTPUT FLAG
         ST    R4,CM411COM_FLAG_PTR    STORE POINTER
         LA    R4,CM411COM_STORAGE     ADDRESS STORAGE AREA
         ST    R4,CM411COM_STORAGE_PTR STORE POINTER
         LA    R4,CM411COM_STG_LEN     ADDRESS STORAGE LENGTH
         ST    R4,CM411COM_STG_LEN_PTR STORE POINTER
*----------------------------------------------------------------------*
*- LINK to DFHEDAP - CEDA Programmable Interface                      -*
*----------------------------------------------------------------------*
A5ØØLINK EXEC  CICS LINK                                               X
               PROGRAM('DFHEDAP')                                      X
               COMMAREA(CM411COM_POINTERS)                             X
               RESP(CM4Ø2COM_RESP1)                                    X
               RESP2(CM4Ø2COM_RESP2)
*----------------------------------------------------------------------*
*- Check EXEC response codes and produce diagnostics if required.    -*
*----------------------------------------------------------------------*
         MVC   CM4Ø2COM_RESOURCE,EIBRSRCE             MOVE RESOURCE
         MVC   CM4Ø2COM_FUNC(L'CM4Ø2COM_FUNC),EIBFN   MOVE FUNCTION
         CLC   CM4Ø2COM_RESP1,DFHRESP(NORMAL)  NORMAL COMPLETION?
         BE    A5ØØRDD                 YES - CHECK CEDA RC'S
*                                      NO  - PROCESS ERROR
         BAL   R14,Z1ØØ_CICS_DIAGS     PERFORM CICS DIAGNOSTICS
         MVC   CM4Ø1COM_MSGNO,=CL(L'CM4Ø3COM_MSGNO)'CM411Ø14E' NUMBER
         MVC   CM4Ø1COM_TEXT(L'CM411Ø14E),CM411Ø14E            TEXT
         BAL   R14,Z2ØØ_ISSUE_MESSAGE  ISSUE MESSAGE
         MVC   CM411COM_CM411_RC,=F'16'  SET RETURN CODE FOR CM411
*----------------------------------------------------------------------*
*- Check CEDA diagnostics output. If there is an error set the        -*
*- transaction dump required flag in order to produce detailed        -*
*- diagnostics.                                                       -*
*----------------------------------------------------------------------*
A5ØØRDD  LA    R12,CM411COM_STORAGE    --> DFHEDAP DIAGNOSTICS OUTPUT
         USING CM411COM_STG_HEADER,R12     AND MAP IT
         CLC   CM411COM_STG_RC,=H'Ø'     IF RC = ZERO
         BE    A5ØØRDE                     THEN CHECK EXECUTION
         MVC   CM411COM_DIAG_RC+2(2),CM411COM_STG_RC  ELSE SAVE RC
         MVI   DUMP_REQD,FLAG_ON                      AND FLAG DUMP
*----------------------------------------------------------------------*
*- Check CEDA execution output ONLY if the diagnostics code was       -*
*- not greater than 4. If required set transaction dump flag.         -*
```

```
*----------------------------------------------------------------------*
A5ØØRDE  CLC   CM411COM_DIAG_RC,=F'4'     IF DIAG RC > 4
         BH    A5ØØCHDU                     THEN DON'T SET EXEC RC
         AH    R12,CM411COM_STG_LENGTH --> DFHEDAP EXECUTION OUTPUT
         CLC   CM411COM_STG_RC,=H'Ø'      IF RC = ZERO
         BE    A5ØØCHDU                     THEN CHECK IF DUMP REQD
         MVC   CM411COM_EXEC_RC+2(2),CM411COM_STG_RC  ELSE SAVE RC
         MVI   DUMP_REQD,FLAG_ON                      AND FLAG DUMP
*----------------------------------------------------------------------*
*- Check whether a transaction dump is required. If it is produce     -*
*- a CICS transaction dump with the code 'DIAG'. This has no effect   -*
*- on subsequent processing, ie it is not an abend!                   -*
*----------------------------------------------------------------------*
A5ØØCHDU CLI   DUMP_REQD,FLAG_ON        IF DUMP FLAG NOT ON
         BNE   A5ØØRET                    THEN RETURN
*
A5ØØDUMP EXEC  CICS DUMP TRANSACTION                                   X
               DUMPCODE('DIAG')
*----------------------------------------------------------------------*
*- Return to caller                                                   -*
*----------------------------------------------------------------------*
A5ØØRET  L     R14,A5ØØSR14             RESTORE REGISTER 14
         BR    R14                      RETURN TO CALLER
         EJECT
************************************************************************
*----------------------------------------------------------------------*
*- A 9 Ø Ø _ T E R M I N A T I O N  :  Termination Processing         -*
*----------------------------------------------------------------------*
************************************************************************
*- R3   BASE CSECT CM411                                              -*
*- R8   DSECT - CM4Ø1COM                                              -*
*- R11  EIBREG DSECT - DFHEIBLK                                       -*
*- R13  DYNREG DSECT - DFHEISTG                                       -*
*- R14  Linkage                                                       -*
************************************************************************
A9ØØ_TERMINATION  DS  ØH
*
         ST    R14,A9ØØSR14             SAVE REGISTER 14
*----------------------------------------------------------------------*
*- Issue end of program messages to TDQ                               -*
*----------------------------------------------------------------------*
A9ØØTDQ  DS    ØH
         MVC   CM4Ø1COM_MSGNO,=CL(L'CM4Ø1COM_MSGNO)'CM411ØØ2I' NUMBER
         MVC   CM4Ø1COM_TEXT(L'CM411ØØ2I),CM411ØØ2I            TEXT
         MVC   CM4Ø1COM_TEXT+12(L'TRAN_ID),TRAN_ID         ADD TRANID
         MVC   CM4Ø1COM_TEXT+22(L'USER_ID),USER_ID         ADD USERID
         MVC   CM4Ø1COM_TEXT+4Ø(L'TERM_ID),TERM_ID         ADD TERMID
         BAL   R14,Z2ØØ_ISSUE_MESSAGE  ISSUE MESSAGE
*----------------------------------------------------------------------*
*- Return to caller                                                   -*
```

```
*-----------------------------------------------------------------------*
A9ØØRET  L     R14,A9ØØSR14            RESTORE REGISTER 14
         BR    R14                     RETURN TO CALLER
         EJECT
***********************************************************************
*----------------------------------------------------------------------*
*- Z 1 Ø Ø _ C I C S _ D I A G S :  Perform CICS Diagnostics          -*
*----------------------------------------------------------------------*
***********************************************************************
*- R3   BASE CSECT CM411                                               -*
*- R8   DSECT - CM4Ø1COM                                               -*
*- R9   DSECT - CM4Ø2COM                                               -*
*- R11  EIBREG DSECT - DFHEIBLK                                        -*
*- R13  DYNREG DSECT - DFHEISTG                                        -*
*- R14  Linkage                                                        -*
***********************************************************************
Z1ØØ_CICS_DIAGS  DS  ØH
*
         ST    R14,Z1ØØSR14            SAVE REGISTER 14
*-----------------------------------------------------------------------*
*- LINK to CM4Ø2 - Diagnose CICS Error.                                 -*
*-----------------------------------------------------------------------*
Z1ØØLINK EXEC  CICS LINK                                                X
               PROGRAM('CM4Ø2')                                         X
               COMMAREA(CM4Ø2COM)                                       X
               LENGTH(L'CM4Ø2ST)                                        X
               RESP(Z1ØØ_RESP1)                                         X
               RESP2(Z1ØØ_RESP2)
*-----------------------------------------------------------------------*
*- Check EXEC response, if there is an error "try" to issue a          -*
*- message and then abend.                                              -*
*-----------------------------------------------------------------------*
         MVC   Z1ØØ_FUNC(L'Z1ØØ_FUNC),EIBFN    MOVE FUNCTION
         CLC   Z1ØØ_RESP1,DFHRESP(NORMAL)      NORMAL COMPLETION?
         BE    Z1ØØRET                 YES - RETURN TO CALLER
*                                      NO  - ISSUE ERROR MESSAGE
         MVC   CM4Ø1COM_MSGNO,=CL(L'CM4Ø1COM_MSGNO)'CM411Ø12E' NUMBER
         MVC   CM4Ø1COM_TEXT(L'CM411Ø12E),CM411Ø12E            TEXT
         BAL   R14,Z2ØØ_ISSUE_MESSAGE  ISSUE MESSAGE
Z1ØØABND MVC   ABNDCODE,=C'Z1ØØ'       SET ABEND CODE
         B     Z9ØØ_ABEND_TRAN         ABEND TRANSACTION
*-----------------------------------------------------------------------*
*- Return to caller                                                     -*
*-----------------------------------------------------------------------*
Z1ØØRET  L     R14,Z1ØØSR14            RESTORE REGISTER 14
         BR    R14                     RETURN TO CALLER
         EJECT
***********************************************************************
*-----------------------------------------------------------------------*
*- Z 2 Ø Ø _ I S S U E _ M E S S A G E : Write Message to TDQ         -*
```

```
*-----------------------------------------------------------------------*
*************************************************************************
*- R3   BASE CSECT CM411                                               -*
*- R8   DSECT - CM4Ø1COM                                               -*
*- R9   DSECT - CM4Ø2COM                                               -*
*- R1Ø  DSECT - CM4Ø3COM                                               -*
*- R11  EIBREG DSECT - DFHEIBLK                                        -*
*- R13  DYNREG DSECT - DFHEISTG                                        -*
*- R14  Linkage                                                        -*
*************************************************************************
Z2ØØ_ISSUE_MESSAGE  DS  ØH
*
         ST    R14,Z2ØØSR14             SAVE REGISTER 14
*-----------------------------------------------------------------------*
*- LINK to CM4Ø1 - Write Message to TDQ                                -*
*-----------------------------------------------------------------------*
Z2ØØLINK EXEC  CICS LINK                                                X
               PROGRAM('CM4Ø1')                                         X
               COMMAREA(CM4Ø1COM)                                       X
               LENGTH(L'CM4Ø1ST)                                        X
               RESP(CM4Ø2COM_RESP1)                                     X
               RESP2(CM4Ø2COM_RESP2)
*-----------------------------------------------------------------------*
*- Check EXEC response, if there is an error "try" to issue a          -*
*- WTO and then abend.                                                 -*
*-----------------------------------------------------------------------*
         MVC   CM4Ø2COM_FUNC(L'CM4Ø2COM_FUNC),EIBFN   MOVE FUNCTION
         CLC   CM4Ø2COM_RESP1,DFHRESP(NORMAL)  NORMAL COMPLETION?
         BE    Z2ØØRET                  YES - BRANCH BACK
*                                       NO  - PROCESS ERROR
         BAL   R14,Z1ØØ_CICS_DIAGS      PERFORM CICS DIAGNOSTICS
         MVC   CM4Ø3COM_MSGNO,=CL(L'CM4Ø3COM_MSGNO)'CM411Ø11E' NUMBER
         MVC   CM4Ø3COM_TEXT(L'CM411Ø11E),CM411Ø11E            TEXT
         BAL   R14,Z3ØØ_ISSUE_WTO       ISSUE WTO
Z2ØØABND MVC   ABNDCODE,=C'Z2ØØ'        SET ABEND CODE
         B     Z9ØØ_ABEND_TRAN          ABEND TRANSACTION
*-----------------------------------------------------------------------*
*- Return to caller                                                    -*
*-----------------------------------------------------------------------*
Z2ØØRET  L     R14,Z2ØØSR14             RESTORE REGISTER 14
         BR    R14                      RETURN TO CALLER
         EJECT
*************************************************************************
*-----------------------------------------------------------------------*
*- Z 3 Ø Ø _ I S S U E _ W T O :  Write To Operator Message           -*
*-----------------------------------------------------------------------*
*************************************************************************
*- R3   BASE CSECT CM411                                               -*
*- R8   DSECT - CM4Ø1COM                                               -*
*- R9   DSECT - CM4Ø2COM                                               -*
*- R1Ø  DSECT - CM4Ø3COM                                              -*
*- R11  EIBREG DSECT - DFHEIBLK                                       -*
*- R13  DYNREG DSECT - DFHEISTG                                       -*
*- R14  Linkage                                                       -*
***********************************************************************
Z3ØØ_ISSUE_WTO  DS ØH
*
         ST    R14,Z3ØØSR14            SAVE REGISTER 14
*---------------------------------------------------------------------*
*- LINK to CM4Ø3 - Write To Operator                                  -*
*---------------------------------------------------------------------*
Z3ØØLINK EXEC  CICS LINK                                               X
               PROGRAM('CM4Ø3')                                        X
               COMMAREA(CM4Ø3COM)                                      X
               LENGTH(L'CM4Ø3ST)                                       X
               RESP(CM4Ø2COM_RESP1)                                    X
               RESP2(CM4Ø2COM_RESP2)
*---------------------------------------------------------------------*
*- Check EXEC response, if there is an error "try" to issue a        -*
*- message and then abend.                                            -*
*---------------------------------------------------------------------*
         MVC   CM4Ø2COM_FUNC(L'CM4Ø2COM_FUNC),EIBFN   MOVE FUNCTION
         CLC   CM4Ø2COM_RESP1,DFHRESP(NORMAL)  NORMAL COMPLETION?
         BE    Z3ØØRET                 YES - BRANCH BACK
*                                      NO  - PROCESS ERROR
         BAL   R14,Z1ØØ_CICS_DIAGS     PERFORM CICS DIAGNOSTICS
         MVC   CM4Ø1COM_MSGNO,=CL(L'CM4Ø1COM_MSGNO)'CM411Ø13E' NUMBER
         MVC   CM4Ø1COM_TEXT(L'CM411Ø13E),CM411Ø13E            TEXT
         BAL   R14,Z2ØØ_ISSUE_MESSAGE  ISSUE MESSAGE
Z3ØØABND MVC   ABNDCODE,=C'Z3ØØ'       SET ABEND CODE
         B     Z9ØØ_ABEND_TRAN         ABEND TRANSACTION
*---------------------------------------------------------------------*
*- Return to caller                                                   -*
*---------------------------------------------------------------------*
Z3ØØRET  L     R14,Z3ØØSR14            RESTORE REGISTER 14
         BR    R14                     RETURN TO CALLER
         EJECT
***********************************************************************
*---------------------------------------------------------------------*
*- Z 9 Ø Ø _ A B E N D _ T R A N : Abend Transaction                  -*
*---------------------------------------------------------------------*
***********************************************************************
*- R3   BASE CSECT CM411                                              -*
*- R11  EIBREG DSECT - DFHEIBLK                                       -*
*- R13  DYNREG DSECT - DFHEISTG                                       -*
*- R14  Linkage                                                       -*
*********************#*************************************************
Z9ØØ_ABEND_TRAN  DS  ØH
*
Z9ØØABND EXEC  CICS ABEND                                              X
```

```
               ABCODE(ABNDCODE)
*----------------------------------------------------------------------*
*- Return to caller                                                   -*
*----------------------------------------------------------------------*
* As a result of the abend control is, of course, returned to CICS.
*
         EJECT
************************************************************************
*----------------------------------------------------------------------*
*- M O D I F I A B L E   I N S T R U C T I O N S                      -*
*----------------------------------------------------------------------*
************************************************************************
* None !!
         EJECT
************************************************************************
*----------------------------------------------------------------------*
*- C O N S T A N T S                                                  -*
*----------------------------------------------------------------------*
************************************************************************
FLAG_ON  EQU   X'FF'
FLAG_OFF EQU   X'ØØ'
         EJECT
************************************************************************
*----------------------------------------------------------------------*
*- T A B L E S                                                        -*
*----------------------------------------------------------------------*
************************************************************************
* None !!
         EJECT
************************************************************************
*----------------------------------------------------------------------*
*- M E S S A G E S                                                    -*
*----------------------------------------------------------------------*
************************************************************************
CM411ØØ1I  DC  C'Transaction=     User=         Facility=     started.'
CM411ØØ2I  DC  C'Transaction=     User=         Facility=     ended.'
CM411ØØ3E  DC  C'Invalid Invocation - No COMMAREA provided.'
CM411Ø1ØE  DC  C'Error processing ASSIGN command.'
CM411Ø11E  DC  C'LINK to CM4Ø1 failed.'
CM411Ø12E  DC  C'LINK to CM4Ø2 failed.'
CM411Ø13E  DC  C'LINK to CM4Ø3 failed.'
CM411Ø14E  DC  C'LINK to DFHEDAP failed.'
         EJECT
************************************************************************
*----------------------------------------------------------------------*
*- E N D   CM411                                                      -*
*----------------------------------------------------------------------*
************************************************************************
         END   CM411
```

CM411A01 – CM411COM

```
***********************************************************************
*                C A R L   W A D E   M C B U R N I E                  *
*                  - I T   C O N S U L T A N T -                      *
*                                                                     *
*                             www.cwmit.com                           *
***********************************************************************
*                                                                     *
* MODULE NAME = CM411AØ1                                              *
* MODULE TYPE = DSECT CM411COM                                        *
* DESCRIPTION = Communications Area for CM411 (Assembler)             *
*               Programmable Interface to CICS CEDA                   *
*                                                                     *
*               NB This member actually contains two DSECTs:          *
*                                                                     *
*               CM411COM maps the storage required for the call       *
*               to CM411.                                             *
*                                                                     *
*               CM411COM_STG_HEADER re-maps the output area           *
*               returned by CM411 within CM411COM. This area          *
*               is only used by CM411 if the CM411COM_FLAG is         *
*               set to X'ØØ'. If it is present the output             *
*               contains one or two concatenated, structured          *
*               fields. The output from a single request comprises    *
*               one field for the translation stage and one or        *
*               none for the execution stage. The variable            *
*               length data following the header contains:           *
*                                                                     *
*                - translation: diagnostic messages                   *
*                - execution: output intended for the CEDA            *
*                  screen, including messages. Each line              *
*                  begins with a new line (NL) character and          *
*                  consists of blanks and uppercase alphanumeric      *
*                  characters.                                        *
***********************************************************************
         EJECT
***********************************************************************
*                                                                     *
* CHANGE HISTORY:                                                     *
* --------------                                                      *
***********************************************************************
         EJECT
***********************************************************************
* C M 4 1 1    -     Communications Area                              *
***********************************************************************
CM411COM               DSECT
CM411COM_ALIGN         DS    ØD             ALIGNMENT
CM411COM_EYECATCH      DS    CL16           EYECATCHER
CM411COM_CM411_RC      DS    F              CM411 RETURN CODE
```

```
CM411COM_DIAG_RC       DS     F              CEDA DIAGNOSTICS RC
CM411COM_EXEC_RC       DS     F              CEDA EXECUTION RC
*
CM411COM_POINTERS      DS     ØF             POINTERS FOR CEDA INTERFACE
CM411COM_COMMAND_PTR   DS     A              --> SOURCE COMMAND
CM411COM_COM_LNG_PTR   DS     A              --> COMMAND LENGTH FIELD
CM411COM_FLAG_PTR      DS     A              --> OUTPUT FLAG
CM411COM_STORAGE_PTR   DS     A              --> STORAGE AREA
CM411COM_STG_LEN_PTR   DS     A              --> STORAGE AREA LENGTH
CM411COM_POINTERS_LEN  EQU    *-CM411COM_POINTERS
*
CM411COM_COM_LNG       DS     H              LENGTH OF CEDA COMMAND
CM411COM_COMMAND       DS     CL1Ø22         CEDA SOURCE COMMAND
CM411COM_FLAG          DS     X              OUTPUT FLAG
CM411COM_FLAG_TERM     EQU    X'8Ø'          OUTPUT TO TERMINAL
CM411COM_FLAG_STOR     EQU    X'ØØ'          OUTPUT TO STORAGE AREA
CM411COM_STG_LEN       DS     H              LENGTH OF STORAGE AREA
CM411COM_STORAGE       DS     CL1Ø24         OUTPUT STORAGE AREA
*
CM411COM_LENGTH        EQU    *-CM411COM     LENGTH OF CM411COM
*
CM411COM_STG_HEADER    DSECT                 MAPS CM411COM_STORAGE
CM411COM_STG_LENGTH    DS     H              INCLUSIVE LENGTH OF OUTPUT
CM411COM_STG_NUM       DS     H              NUMBER OF MESSAGES OUTPUT
CM411COM_STG_RC        DS     H              HIGHEST MESSAGE RETURN CODE
CM411COM_STG_RC_ØØ     EQU    X'ØØ'          OK      - EXECUTION
CM411COM_STG_RC_Ø4     EQU    X'Ø4'          WARNING - EXECUTION
CM411COM_STG_RC_Ø8     EQU    X'Ø8'          ERROR   - NO EXECUTION
CM411COM_STG_RC_12     EQU    X'ØC'          SEVERE  - NO EXECUTION
CM411COM_STG_HDR_L     EQU    *-CM411COM_STG_HEADER
*
CM411COM_STG_DATA      DS     CL(L'CM411COM_STORAGE-CM411COM_STG_HDR_L)
*--------------------------------------------------------------------*
*- E N D   CM411AØ1                                                 -*
*--------------------------------------------------------------------*
         EJECT
```

Carl Wade McBurnie
IT Consultant (Germany)

CICS transaction segregation and region creation

INTRODUCTION

This article will define the CICS transactions supplied by IBM, their functions and security impact, and will provide a template for adequate segregation and control. It will give both a structure for classifying IBM-supplied transactions into logical groups and the JCL necessary for creating the specific CICS General Resource profiles within your own CICS regions. It will also look at providing increased security for the CICS commands through proper segregation within the CCICSCMD/ VCICSCMD General Resource profiles.

RELEASES/VERSIONS

For the purposes of this article, the following IBM versions and releases will be used as examples:

- z/OS Version 1.4
- RACF Version 1.4
- CICS/TS Version 2.2.

NAMING CONVENTIONS

Before we get into CICS profile creation, I want to spend some time specifying the naming conventions to be used. Your organization probably has its own naming conventions – at least, I hope it does! If it doesn't, consider the structure demonstrated here as a template for your convenience. If you already have a naming convention in place, make sure you

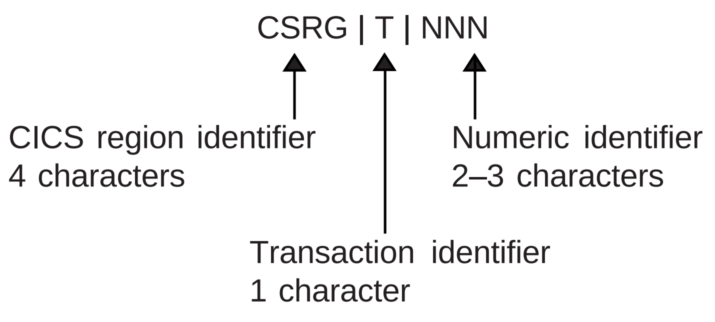


Figure 1: General Resource profile naming convention

stick to it. Otherwise, you've got some major headaches ahead when trying to maintain the CICS region or track down problems.

The primary naming conventions here are for the groups and the GCICSTRN/TCICSTRN General Resource profiles. They're quite similar in structure, but we'll look at them as separate entities.

General Resource profile naming convention

The General Resource profile naming convention is shown in Figure 1.

CSRG: CICS region identifier

The CICS region identifier consists of four alphanumeric characters, used to uniquely define the CICS region to which the transaction profile is related.

Examples:

- DLL1Tnnn – CICS development region, DCICLLL1
- PAP6Tnnn – CICS production region, PCICAPY6
- THM3Tnnn – CICS testing region, TCICHOM3.

As you can see, this allows you immediately to identify a region based on whether it is production, development, or test. This can be expanded as needed for your environment (B for business acceptance testing, U for unit testing, Q for quality control testing, etc).

T: transaction profile type

The transaction profile type consists of a single alpha character. It is used to define the purpose of the transaction profile.

Transaction profile types categorize the profile in relation to transactions defined within the profile. For example, CICS-supplied, system/utility, and application transactions will have separate transaction profile types.

Examples:

- xxxxAnnn – application transactions
- xxxxSnnn – system/utility transactions
- xxxxCnnn – IBM-supplied CICS transactions
- xxxxMnnn – CICS commands (VCICSCMD).

NNN: numeric identifier

The numeric identifier consists of up to three numeric characters. It is used to attach a numeric value to the transaction profile.

The numeric identifier is simply a value that differentiates profiles having the same CICS region identifier and transaction profile type.

Examples:

- xxxxA001 – application transactions profile 1
- xxxxA089 – application transactions profile 89
- xxxxC022 – CICS transactions – Category 2 – Group 2
- xxxxS001 – system/utility transaction profile 001
- xxxxM005 – CICS command profile 005.

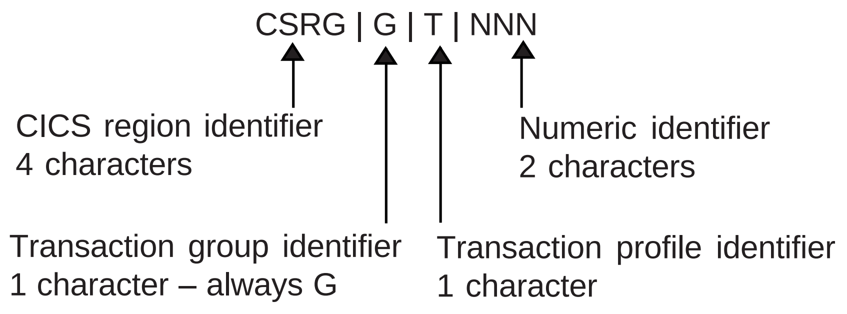

CSRG | G | T | NNN
CICS region identifier 4 characters
Numeric identifier 2 characters
Transaction group identifier 1 character – always G
Transaction profile identifier 1 character

Figure 2: Group naming convention

Group naming convention

The Group naming convention is illustrated in Figure 2.

As you can see, the structure is quite similar to the General Resource profile naming convention, with the exception of the G in column 5 to denote a Group. Group names should relate to the General Resource profile names, for ease of use. For example, xxxxA091 would be connected to group xxxxGA91, xxxxC02E would be connected to group xxxxGC2E, etc.

Now, that may be fine for CICS regions where you have only 99 separate application transaction profiles (xxxxA001-xxxxA099). What do you do if you've got xxxxA101? The truncation of the leading zero won't work, because you'd be relating both xxxxA001 and xxxxA101 to group xxxxGA01. Well, there's a simple solution. If you've really got that many profiles for your application transactions (although heaven only knows why anybody would) change the A to a B for the next group of 99 profiles.

You should always reserve 00 as your owner group for the subgroups 01–99. It's better to segregate these than to just have everything lumped together in a single region group. You'll see what I mean when we get to the Group Structure tree later in the article.

TRANSACTION PROFILES – THE BASICS

According to the *CICS RACF Security Guide*, IBM supplies 123 CICS transactions within CICS Transaction Server V2.2. These transactions serve a variety of program- and operation-specific functions to ensure the continuous processing of your CICS systems. IBM recommends that these transactions be segregated in the following manner:

- 01 – non-terminal transactions
- 02 – specialist and limited access transactions
- 03 – CICS users' transactions.

It is simple, straightforward, and to the point. When you look at the *CICS RACF Security Guide*, you find that IBM devotes only 11 pages to IBM-supplied transaction segregation, and a further seven pages for CICS command descriptions (with no real segregation suggestions for the commands themselves). Beyond that, their guidance is pretty sparse. This is a bit disappointing, of course, but I believe IBM's rationale is that each shop should design its own transaction/command segregation based on its particular needs. While laudable, more guidance would, hopefully, provide more standardized and more secure CICS profile definition across the board.

TRANSACTION SEGREGATION – CATEGORIES 1 AND 3

IBM's divisions for Category 1 (non-terminal) and Category 3 (all CICS terminal user) transactions are quite straightforward. Use the structure as recommended by IBM and you won't have any problems. They really don't need further segregation, so don't bother tinkering around with this.

TRANSACTION SEGREGATION – CATEGORY 2

These are the IBM-supplied transactions that are user-initiated. Some of them have rather a lot of power, such as CEMT or CECI. IBM segregation for Category 2 is defined in 11 subcategories, which are:

- SYSADM
- DEVELOPER
- INQUIRE
- OPERATOR
- DBCTL
- INTERCOM
- ALLUSER
- WEBUSER

- RPCUSER
- IIOPUSER
- AFFINITIES.

My only real problem with the IBM segregation is that it doesn’t go quite far enough, as far as I’m concerned. For example, CEMT is embedded in the SYSADM subcategory. Some developers use the CEMT inquire function and don’t require the other items in IBM’s interpretation of this particular transaction grouping. Since security is designed to provide the minimum-required level of access, this subcategory needs further refinement. Therefore, I’ve redefined the subcategory list from 11 to 16 (021 to 02F) to provide greater security. When used with the VCICSCMD segregation (discussed later) it will provide a much finer level of granularity and accountability to your transaction/command combinations:

- 021 – system administration
- 022 – development
- 023 – inquiry only
- 024 – operator transactions
- 025 – intercom transactions
- 026 – RPCUSER transactions
- 027 – AFFINITIES transactions
- 028 – WEBUSER transaction
- 029 – ALLUSER transactions – good morning
- 02A – ALLUSER transaction – CRTX
- 02B – IIOPUSER transaction
- 02C – DBCTL transactions
- 02D – CECI transaction – limited access
- 02E – CIND transaction – limited access

- 02F – CEMT transaction – limited access.

You'll note that I've actually segregated the CRTX transaction from the other two ALLUSER items. I feel that, despite this being recommended by IBM for all users, there are some functions within CRTX that need a bit more overall control. The last three categories above pull the most sensitive transactions – CECI, CIND and CEMT – away from their IBM-recommended grouping for greater security and tracking. Now, you can stick with IBM's transaction structure if you wish. However, if you're at all paranoid about the functionality of these transactions (and every good security technician should be at least a little paranoid) you'll find the isolation as above to be a better choice.

TRANSACTION SEGREGATION – CATEGORY 9

One thing that bothered me during the research for this article was that the transactions mentioned in the *CICS RACF Security Guide* seemed a bit skimpy. After all, we're talking about a transaction server subsystem that comprises millions of lines of code and a flexibility that allows it to function for airlines, banks, manufacturers, butchers, bakers, candlestick makers, etc. A mere 123 IBM-supplied transactions didn't seem quite enough to run such an all-encompassing transaction processing system.

So I started investigating. Granted, I actually enjoy reading IBM manuals (and doctors are still searching for a cure) so the research was not an odious chore. Well, not at first, anyway. I checked around and found that IBM has two other manuals with transactions not mentioned in the *CICS RACF Security Guide*. Quite a few transactions, it turns out. The *CICS Supplied Transactions* and *CICS Resource Definition Guide* were full of obscure and arcane transactions. No doubt this was partly because of past CICS releases (oh, the thrill of backwards compatibility), but the arrangement and location of the transactions were most confusing. It took a while, but I was able to cross-match the transactions to the manuals where they appear.

A full list of the 259 transactions and in which manual each is described can be downloaded from the Xephon Web site at www.xephon.com/extras/trancomp.txt.

Of the 259, the *CICS RACF Security Guide* had 123, the *CICS Resource Definition Guide* 199, and the *CICS Supplied Transactions* 174.

As you can see when you download the file, some of the transactions are common to all thee manuals, whereas some are mentioned in only one book, or two. All in all, it was most worrying to find these transactions listed and described so far apart. I've already advised IBM of the potential discrepancies, and as of this writing the IBM Hursley Laboratories in the UK are diligently researching this issue. Hopefully we'll see some updates to their documentation in the newest release of CICS/TS V3.1.

Now, you're probably asking yourself, why do these transactions matter? If I don't define them, they won't execute anyway, right? True, but only if you've got full transaction security defined within both RACF and CICS, and only if your shop doesn't require some of the more obscure transactions. Being a belt and braces kind of guy (that's belt and suspenders to our American readers – I didn't want the British audience to giggle too much at the image of me in garter belts), I decided that it would be best to define the transactions and then block access to all but the CICS region itself. Hence, you'll find a Category 9 defined in the transaction segregation lists and the accompanying JCL.

By the way, you'll note that there are actually four subcategories for these transactions:

- 091 – defined in *CICS Resource Definition Guide* and *CICS Supplied Transactions*
- 092 – defined in *CICS Resource Definition Guide*
- 093 – defined in *CICS Supplied Transactions*
- 094 – defined in *CICS Front-End Programming Interface (FEPI).*

Although it wasn't in the comparison chart, a separate group was created for CICS Front-End Programming Interface (FEPI) transactions. Since FEPI is pretty much becoming 'standard equipment' in CICS, I've included it for your convenience. If you don't use FEPI, just drop that segment of the JCL.

Now, while these all have UACC(NONE), the CICS region itself still has access to the transactions. You can remove that as well, should you so desire, but if you've got any legacy systems that still use these digital dinosaurs it's best to let the region retain access.

For a detailed list of all the transactions and their individual General Resource profile groupings, please see Appendix A (in next issue). The JCL template to actually create these profiles can be found in Appendix B.

CICS COMMAND SECURITY

Everybody seems to focus on the CICS transactions in security set-ups, and pages can be written on the GCICSTRN/TCICSTRN General Resource profiles. Sadly, though, very little attention is paid to the ugly sister of the RACF GenRes settings, the CCICSCMD/VCICSCMD CICS command security profiles. Even IBM gives this subject short shrift in the *CICS RACF Security Guide*. I find this a bit disappointing, and somewhat disturbing, considering the immense power that some of these commands can wield.

So, since there was no real guidance on this issue, I decided to create my own command segregation. In general, it breaks down in the following manner:

- M001 – inquiry only commands
- M002 – DB2 commands
- M003 – terminal/monitor/TCPIP commands
- M004 – reserved for future use
- M005 – general use commands (segregated by RACF

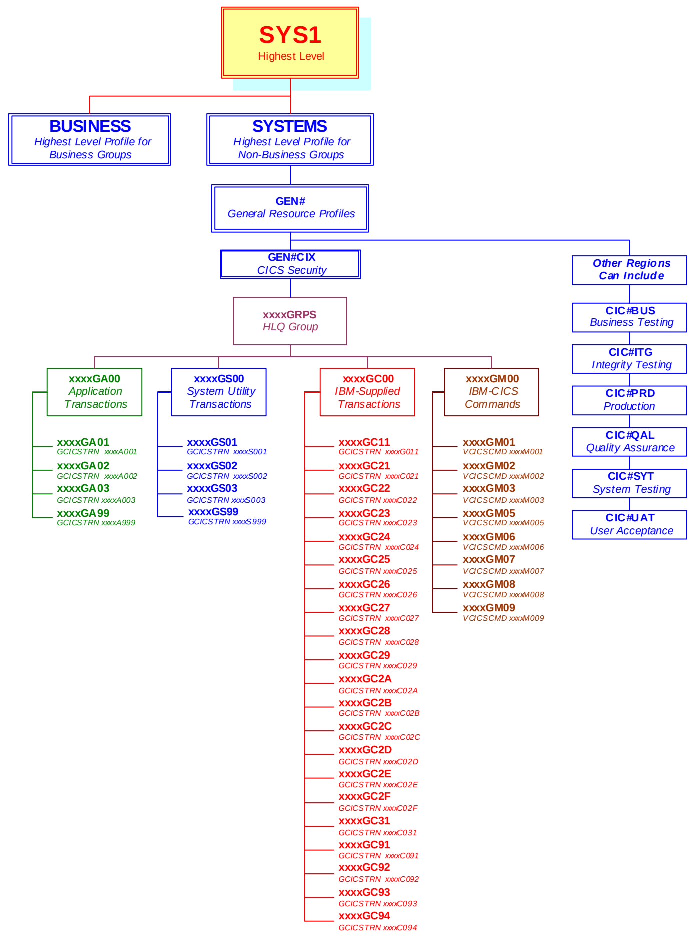


Figure 3: Group tree example

access level READ/UPDATE/ALTER)

- M006 – EXEC CICS-level commands
- M007 – CEMT-level commands
- M008 – high-end technical support commands only
- M009 – SECURITY command only.

The full description of these commands can be found in Appendix C, and the JCL to create the General Resource profiles is listed in Appendix D. (See next issue.)

GROUP STRUCTURES

I’m a big believer in creating a group tree in a structured tiered format. I don’t like group trees that look like low-growing shrubbery. I’m also not too thrilled with groups that make you dizzy from their excessive verticality. If any of you have ever read the four-part series that I did on *RACF Restructuring*, you’ll know that I am heavily into the Christmas Tree format for RACF group alignment (published in *RACF Update*). From top (SYS1) to bottom (xxxxGC94) this group model covers seven levels. Whether you have one CICS region or a hundred, you’re covering only seven levels, and the spread is consistent along the height and width of the group tree. An example is shown in Figure 3.

Now, why should you bother creating all of these profiles? Just because it makes a pretty tree? My big bugaboo about CICS regions, dataset profiles, etc is that some RACF administrators provide permission to these resources by user ID instead of by GROUP! This, to me, is a security accident waiting to happen. It, in effect, bypasses the controls provided by the group structure. I strongly recommend that to avoid this you build a group structure for each CICS region, and then connect the users to only those groups they need to obtain the transactions required for their job function. This goes for all the transaction groups – application, system/utility, IBM-supplied transactions, and CICS commands. The one exception to the group-only rule is to explicitly add the CICS region name (started task) to each General Resource profile. Even though each profile is owned by the CICS region, I still add the region name to the PERMIT list (belt and braces again).

Editor’s note: this article will be continued in next month’s issue.

Doc Farmer
Independent Security Consultant (USA)

BEA Systems has upgraded its Tuxedo mainframe adapters to provide a broader infrastructure for migrating legacy systems to open systems, or for building new applications that require mainframe connectivity.

The company said its Tuxedo Mainframe Adapters, designed to provide end-to-end transactional support between BEA Tuxedo and mainframe applications, can now support applications built on IMS 8.1 and CICS 2.3. According to the company, the adapters can link IMS or CICS applications on the mainframe with open standards-based applications built on Tuxedo. These adapters can also play a role in supporting a migration strategy to open systems, which may require the mainframe applications to co-exist with BEA Tuxedo during the transition period.

For further information contact:
BEA Systems, 2315 North First Street, San Jose, CA 95131, USA.
Tel: (408) 570 8000.
URL: www.bea.com/framework.jsp?CNT=pr01400.htm&FP=/content/news_events/press_releases/2004.

***

MacKinney Systems has announced Release 1.6 of CICS/Hotprint, which replaces and extends the functions of the standard IBM screen-print facility.

The product allows the target printer to be any CICS-defined VTAM printer, allows online update for the target printer per user, maintains the original ‘screen look’ by not suppressing blank lines, and adds a header to screen prints (optional per user) to assist in print distribution.

The product interfaces with MacKinney Systems’ eSendIT to allow screen prints to be e-mailed or faxed. It interfaces with CICS/SPOOLER to allow screen prints to be queued in CICS/Spooler for online display and optional printing on CICS or system printers. It now has added support for MacKinney’s CICS/SPY so it can print spy session screens to a specified CICS printer.

For further information contact:
MacKinney Systems, 4411 E State Hwy D, Suite F, Springfield, MO 65809, USA.
Tel: (417) 882 8012.
URL: www.mackinney.com/products/cics/cics_hotprint.htm.

***

More details of the CICS Transaction Server for z/OS V3.1 announcement are emerging.

CICS TS V3.1 enables customers to extend or integrate current CICS applications to an SOA while supporting the SSL security protocol. The new platform also supports IBM’s WebSphere Studio Enterprise Developer, a toolset for transforming legacy applications. The platform also enables simplified administration through the extension of the CICSPlex System Manager Web User Interface, improved workload throughput, and enhanced C/C++ program performance.

CICS Transaction Gateway, which provides access to CICS from multiple environments, enables data managers to use CICS applications in J2EE and Web services solutions hosted on application servers such as IBM’s WebSphere.

For further information contact your local IBM representative.
URL: www-306.ibm.com/software/htp/cics/tserver/v31.