Third/Fourth Quarter, 1999

VSE/ESA™ Software Newsletter

News and information for VSE/ESA customers and vendors

G225-4508-19

Trademarks

About This Newsletter

The *VSE/ESA Software Newsletter* is published twice a year as a service to users of VSE systems. On the Internet, the *Newsletter* is available in PDF format (Adobe Acrobat Reader) at:
http://www.ibm.com/s390/vse/vsehtmls/newslett.htm

The information contained in this document is based on the best knowledge of the authors and on current development, has not been submitted to any formal IBM® test, and is distributed on an "As Is" basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent program may be used instead. Any performance data contained in this document was determined in a controlled environment and, therefore, the results which may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country or not yet announced by IBM. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services.

Editor: Joel Hermann, IBM Entwicklung GmbH, Department 3257, Schönaicher Str. 220, D-71032 Böblingen, Germany (jhermann@de.ibm.com)
Subscriptions: See the subscription form at the end of the *Newsletter*, or contact Margarete Büttner (buettnem@de.ibm.com).

Contents

Editor's Notes

A Special "Thank You!"

I am extremely grateful to *each and every* person who contributed an article to this issue of the *VSE/ESA Newsletter*. Writing is just plain hard work; and no one know that better than I.

This issue features a number of valuable articles related to CICS Transaction Server™ for VSE/ESA (CICS® TS). All but one have been contributed by members of the CICS/VSE® Development Team in Hursley. What I especially respect in these folks is their willingness to take the time to do "extra work," as these articles show. It's not that Clare Jackson, Chris Smith, Dave Fitzpatrick, or any of the other team members aren't busy, of course. They have more than enough to do. But look at this issue of the *Newsletter*, past issues, or the VSE/ESA home page. We'd all be "information deprived" without their continued support – without their going the extra mile.

Thanks, authors! Thanks, CICS/VSE team!

Address Corrections / Subscription Changes

If you are subscribed to the *VSE/ESA Software Newsletter* and would like to make address changes or other corrections via e-mail, send a note to Margarete Büttner (**buettnem@de.ibm.com**) with that information.

New Redbooks For CICS Transaction Server For VSE/ESA

Migration To VSE/ESA 2.4 And CICS Transaction Server for VSE/ESA 1.1

This redbook (SG24-5595-00) is intended for VSE customers and IBM technical personnel responsible for planning and migrating to VSE/ESA 2.4 and the CICS Transaction Server for VSE/ESA 1.1.

CICS has significantly changed in this new release. CICS TS 1.1 provides functions to improve application programming, system programming, system management, and data security and availability. CICS internal security has been removed, thus requiring the use of an external security manager if security is needed.

This redbook will help you to understand the impact of migrating to CICS TS 1.1. It will also help you to install and configure the new CICS TS 1.1 by providing detailed guidance and samples for the changes that you have to perform.

You will understand the new features and their potential, the impact of the removal of obsolete functions, and the possibilities of CICS TS 1.1 and CICS for VSE/ESA 2.3 coexistence. You will learn about security and performance issues, and samples are provided for first-level problem determination through the use of dumps or trace tools.

Implementation of VSE/ESA 2.4 and CICS Transaction Server for VSE/ESA 1.1

This redbook (SG24-5624-00) is intended for VSE customers and IBM technical personnel responsible for implementing VSE/ESA Version 2 Release 4 with the CICS Transaction Server for VSE/ESA 1.1 and its new features.

It includes an overview and detailed implementation steps for many of the new functions introduced with this new release of VSE/ESA and CICS TS. You will become familiar with the new system management features and the new application programming and system programming facilities introduced in CICS Transaction Server for VSE/ESA 1.1. Many samples are included which show you how to exploit the new functions.

In addition, you will find a summary of valuable hints to improve your VSE system management.

Related Web Resources

VSE/ESA Home Page

Besides the *VSE/ESA Newsletter*, VSE Development also supports a VSE/ESA home page. The site is not "fancy," but it does have a wide range of information about VSE and VSE-related products from IBM and other vendors. Check out: **http://www.ibm.com/s390/vse/**

VM/ESA® Home Page

VM Development has an excellent home page with a wealth of information. This site is at: **http://www.ibm.com/s390/vm/**

VSE-L Discussion Forum

VSE-L is an Internet community where VSE users gather to share tips, experiences, and support. To subscribe to this forum, which is sponsored by Lehigh University, send an e-mail to: **listserv@lehigh.edu**

In the first line of the message body, enter:
SUBSCRIBE VSE-L *your full name, your company name*

Articles Related To CICS Transaction Server For VSE/ESA

CICS Transaction Server For VSE/ESA Release 1 – Overview

Author

Chris Smith
CICS Technical Strategy and Plans
smithch@uk.ibm.com

Editor's Note

Chris contributed a longer, more in-depth introduction to CICS Transaction Server for VSE/ESA in Issue 17 of the *Newsletter* (Third/Fourth Quarter, 1998). That article is available on the Web at:
http://www.ibm.com/s390/vse/vsehtmls/vsenew17.htm

You also can get a printed copy of Issue 17 by sending a note to Ulrich Ellwart (**ellwart@de.ibm.com**).

Major New Version Of CICS For The VSE/ESA Platform

As previously announced, CICS Transaction Server for VSE/ESA delivers a major new version of CICS, which is based on CICS for MVS/ESA. This is a significant milestone in the continuing evolution of CICS on the VSE/ESA platform, with the new product providing:

- Improved reliability and availability
- Extensive virtual storage constraint relief
- Expanded application programming capabilities
- New e-business and network computing capabilities
- Improved system administration and management
- Closer compatibility with CICS Transaction Server for OS/390®
- A coexistence environment to aid migration from CICS/VSE Version 2.3
- CICS Server, CICS Client and CICS Gateway function in one product
- A new inclusive product package

Expanded Capabilities And A Foundation For Growth

CICS Transaction Server for VSE/ESA protects your investment in CICS and enables you to grow to meet ever increasing business demands. Built on the tried and tested CICS for MVS/ESA™ V4.1 code base and also incorporating function from CICS Transaction Server for OS/390, CICS TS for VSE/ESA provides a robust environment to enable you to grow your applications and workload and to develop new applications to take your business forward.

As well as increased reliability and availability, it provides significant new functions and enhancements, including many that you have asked for – for example, **Storage Protection** to avoid CICS failures caused by application error, extensive **Virtual Storage Constraint Relief** which frees up storage for application use, **Shared**

Data Tables which significantly improves performance for VSAM file access, the **External CICS Interface** (**EXCI**) that allows batch programs to invoke CICS programs online, a **Front-End Programming Interface** (**FEPI**) to enable flexible integration with existing applications, **Resource Definition Online** (**RDO**) **for Files** which improves system management and availability, and many more.

New e-business and network computing capabilities also are available. There is support for the new **CICS Universal Clients** and the new **CICS Transaction Gateway**, which are included as part of the product package. These products allow access to CICS applications from popular desktop environments, Web browsers, and Network Computers. In addition and available on a later schedule is the **CICS Web Interface**, which allows direct access to CICS applications from a Web browser or Network Computer without the need for an intermediate gateway.

Coexistence Environment To Aid Migration From CICS/VSE Version 2 Release 3

As explained in “Migrating To CICS Transaction Server For VSE/ESA Release 1” on page 7, the recommended migration path to the new CICS version is via CICS/VSE V2.3. This is Year 2000 ready and is the only prior CICS release that is supported for coexistence with the new version.

To assist in a smooth migration, CICS/VSE V2.3 is supported for concurrent operation and MRO connectivity with CICS TS for VSE/ESA on the same VSE/ESA system image (the CICS *coexistence* environment). The coexistence environment provides timing flexibility for migration to the new version. For example, old macro-level applications can continue to run on CICS/VSE V2.3 with end users connected to CICS TS for VSE/ESA accessing them via Transaction Routing. Similarly, access to applications on CICS TS for VSE/ESA from BTAM terminals connected to CICS/VSE 2.3 can be achieved via Transaction Routing.

Tools such as the **Application Migration Aid**, the **DFHMSCAN utility**, and the **Security Migration Aid** are also provided to help the migration task. Also, the CICS System Definition File used by RDO can be shared between CICS/VSE V2.3 and CICS Transaction Server for VSE/ESA.

New Inclusive Product Package

The CICS Transaction Server for VSE/ESA product package contains both the new version of CICS and CICS/VSE V2.3, plus several other related products and features:

- **CICS Transaction Server for VSE/ESA Release 1** – the new version of CICS that is based on CICS for MVS/ESA V4.1
- **CICS/VSE Version 2.3** – the prior CICS release which will coexist with CICS TS for VSE/ESA
- **Report Controller Facility** – previously a priced feature of VSE/ESA which is provided in two versions, one to work with CICS/VSE V2.3 and one to work with CICS TS for VSE/ESA
- **CICS Universal Clients Version 3** – an enhanced replacement for the CICS Clients product which runs on all popular desktops to provide client access to CICS

- **CICS Transaction Gateway Version 3** – which provides access to CICS from a Web browser or Network Computer and replaces both the CICS Internet Gateway and the CICS Gateway for Java
- **REXX for CICS** – which enables CICS applications to be written in this popular programming language
- **CICS/DDM** – previously a separately priced product, this provides access to CICS files from distributed systems such as AS/400

There is *one* price for the complete package, and there is *no increase* in price over CICS/VSE V2.3.1[1]. The product license also allows the CICS Transaction Gateway to be used for Web access to applications on CICS/VSE V2.3 immediately.

What's New Since The Original Product Announcement?

- CICS Transaction Server for VSE/ESA Version 1 Release 1 was made generally available at the end of June 1999.
- Early availability of the CICS Web Interface and 3270 bridge is planned for 1Q00.
- REXX for CICS has been added to the product, with planned delivery of 12/99.
- CICS/DDM is now included as part of the product package.
- A facility has been added for the viewing of CSMT messages produced by CICS/VSE V2.3 in the coexistence environment.
- Translated versions of CICS Transaction Server for VSE/ESA messages and RCF panels are provided in Japanese, Simplified Chinese, and German.
- The complete set of entitlement publications for the product is available for download from the CICS Web site.

Additional Information

Additional information on CICS Transaction Server for VSE/ESA, the CICS family, and VSE is available from the following sources:

- IBM Software Announcement Letters

  298-373 (original CICS TS for VSE/ESA announcement letter)
  299-156 (CICS TS for VSE/ESA availability and update letter)

- CICS product Web site

  http://www.ibm.com/software/ts/cics/

- CICS Transaction Server for VSE/ESA Web site

  http://www.ibm.com/software/ts/cics/platforms/cicsvse/vse.html

- VSE/ESA Web site

  http://www.ibm.com/s390/vse/

- ITSO Redbooks Web site

  http://www.redbooks.ibm.com/

- CICS product publications

[1] There may be price variations in some countries due to currency fluctuations.

Note that all CICS Transaction Server for VSE/ESA entitlement publications can be downloaded from the CICS Web site.

- *CICS Transaction Server for VSE/ESA Release Guide*, GC33-1645-00
- *CICS Transaction Server for VSE/ESA Migration Guide*, GC33-1646-00
- *CICS Family Client/Server programming*, SC33-1435-04
- *CICS/VSE V2.3 Server Support for CICS Clients*, SC33-1712
- All CICS, DB2, IMS publications on CD-ROM, Order No. SK2T0730

- CICS ITSO Redbooks
  - *Migration to VSE/ESA 2.4 and CICS TS for VSE/ESA*, SG24-5595
  - *Revealed! CICS Transaction Gateway and CICS Universal Clients*, SG24-5277
  - *Revealed! Architecting Web Access to CICS*, SG24-5466
  - *CICS/VSE in a Networking World*, SG24-2047
  - *New VSE/ESA support for CICS Clients (EPI)*, SG24-2038
  - *The CICS Internet Gateway for VSE/ESA*, SG24-4799

Migrating To CICS Transaction Server For VSE/ESA Release 1

Authors

Dr. Clare K. Jackson
CICS Information Development
clare_jackson@uk.ibm.com

David Fitzpatrick
CICS/VSE Development
david_fitzpatrick@uk.ibm.com

CICS is a complex product used in a variety of ways. This means that the migration from one release to another is not a simple step-by-step process with a single set of instructions. Despite this, migration can still be divided into stages that everyone must consider. These are:

1. Planning your migration
2. Installing CICS TS
3. Tailoring the system
4. Coexisting with a previous release
5. Changing your resource definition
6. Changing for application programming
7. Customizing your system
8. Setting up your security
9. Using monitoring and statistics

Planning Your Migration To CICS TS For VSE/ESA

To simplify the migration of your CICS system to the latest release, it is important to start from a common point. **You must migrate to CICS for VSE/ESA 2.3 (CICS/VSE 2.3) if you are not already there.** (You should already be using CICS/VSE 2.3 to ensure that your system is Year 2000-proof.)

If you are not already doing so, start using Resource Definition Online (RDO) on CICS/VSE 2.3 to familiarize yourself with it. This is important, as some resources in CICS TS can be defined only with RDO.

You need to decide whether to run your installation in a "coexistence" mode, that is, with CICS TS for VSE/ESA (CICS TS) connected to a CICS/VSE 2.3 region (or regions). This the most probable environment and is made possible by Multi Region operation (MRO) extensions that allow communication between two different versions of CICS. However, you can run CICS TS totally isolated if you want.

You need to communicate with your vendors early in this migration process. Although IBM has worked closely with Independent Software Vendors (ISVs) throughout the development of CICS TS – enabling versions of many of the key products that you may use to be supported on the new platform – your vendors may provide other products that they need to change.

You must familiarize yourself with the new functions that CICS TS offers you, the hardware and software requirements, and the effects of the removal of functions such as BTAM terminal support, Macro Level API and CICS internal security.

When planning the migration, don't try to include any of the new functions at this stage. Aim to get the new system as close to your old one as possible as the first step. And most importantly, *plan a staged test, cutover, and implement some form of change management*. Try to avoid simultaneous changes, and have a fallback plan in case of failure.

Installing And Tailoring CICS TS For VSE/ESA

As with previous releases of CICS, CICS TS is an integral part of VSE/ESA. CICS TS only runs on VSE/ESA 2.4 and subsequent releases. It makes extensive use of the new facilities offered by VSE/ESA 2.3, particularly the new OS/390 services. The Interactive Interface and VSE/ICCF are both designed to run only with CICS TS. CICS is restored to the PRD1.BASE library as part of the base installation process for all the environments supplied.

Although you can run both CICS TS and CICS/VSE 2.3 together on VSE/ESA 2.4, you must install CICS/VSE 2.3 (which is provided on the extended base tape) as an optional product using the Interactive Interface *Optional Product* install dialog. It must also reside in a different library from CICS TS and is restored to the PRD2 library. As a result, CICS/VSE 2.3 startup jobs need changes to the LIBDEFs.

Because the Interactive Interface and VSE/ICCF are supported only with CICS TS, a number of functions are no longer available when running CICS/VSE 2.3. One of these – the viewing of the SYSLST message log online – is very important operationally to many customers. Therefore, this specific function will be replaced by a native CICS solution for viewing the message log. It will be delivered initially as a PTF.

Tailoring Your System

The main areas of change are in virtual storage requirements, changed and new datasets, and system initialization. The virtual storage requirements of the predefined environments have changed, as shown in Table 1. To run both CICS TS and CICS/VSE 2.3, you should use environment **B**.

Table 1. Predefined Environments

Environment	**A**	**B**	**C**
Number of address spaces	24	44	12
VSIZE	100MB	250MB	100MB
Max. dynamic partitions	12	32	–
PASIZE	30MB	50MB	30MB
Max. DSPACE size	8MB	20MB	8MB
CICS TS partition size	30MB	50MB (2: F2 and F8)	30MB
CICS/VSE 2.3 partition size	–	30MB (F4)	

The CICS-related datasets are summarized in Table 2 on page 9.

Table 2. CICS-Related Datasets

File Name	Use	Type	Notes
DFHAUXT	Auxiliary Trace	Seq	Unchanged
DFHCSD	CICS System Definitions	KSDS	Now mandatory
DFHDMPA/B	CICS Dumps	Seq	Transaction dumps only
DFHGCD	CICS Global Catalog	KSDS	New
DFHLCD	CICS Local Catalog	KSDS	New
DFHNTRA	Intrapartition Transient Data	ESDS	Unchanged
DFHRSD	Restart Data Set	KSDS	Used only for Emergency Restart
DFHTEMP	Temporary Storage	ESDS	Unchanged
DFHJ01A/B	CICS System Logs	Seq	Unchanged
DFHSTM/N	Monitoring and Statistics	Seq	Obsolete; replaced by DMF datasets

Recovery and restart in CICS TS have been extensively changed, and this is reflected by the introduction of two new Catalog datasets. In previous releases, a part of the Restart Dataset was used for the catalog functions. The Restart Dataset is retained in CICS TS, but it is used only for Emergency Restart purposes.

Because some resources can be defined only online in CICS TS, the **CICS System Definitions Dataset** (**DFHCSD**) is now mandatory. The datasets used for monitoring and statistics collection (DFHSTM/N) are obsoleted and are replaced by DMF datasets, which are defined outside of the CICS partition.

There are many new system initialization parameters available in CICS TS. Until you are up and running, you should let most of these remain as defaults. Many system initialization parameters are obsoleted. For a complete list of changes, you should refer to the *CICS Migration Guide*.

CICS TS now manages eight Dynamic Storage Areas, four above and four below the 16MB line. The overall size of the dynamic storage areas managed by CICS and the maximum number of tasks CICS allows to exist in a CICS system at any one time need to be set using non-default values. **DSALIM** and **EDSALIM** values control the overall size of the DSAs, below and above the 16MB line respectively. **MXT** values control the maximum number of tasks.

Your DSALIM can be calculated by first working out your 24-bit GETVIS requirement for VSAM, DL/I, vendor products and so on, in the same manner as you did in CICS/VSE 2.3 and then adding 500KB to this value. The remaining partition storage below the 16MB line is the upper limit for your DSALIM. To avoid wastage of your 24-bit storage, you should ensure your CICS startup JCL specifies `SIZE=DFHSIP` on the EXEC DFHSIP statement.

The EDSALIM is calculated by allowing for around 3MB of non-DSA 31-bit GETVIS, plus any you need for VSAM and vendor products. The remainder can be managed by CICS.

To improve performance, CICS TS pre-allocates CICS task stack storage in CICS DSA, based on the MXT setting. For each task, CICS allocates about 2KB of 24-bit DSA and 12KB of 31-bit DSA. You should ensure that you do not set too high a value.

Most of the CICS management code has been relocated into the CICS-managed DSA storage above the 16MB line. Thus you generally need to increase the size of your partitions by about 5MB to 6MB compared to CICS/VSE 2.3. In some cases, it may be possible to run CICS in a smaller partition than CICS/VSE 2.3. This is because application programs loaded above the 16MB line are now released when not in use, and the data tables reside in a data space and not in partition storage.

Coexisting Between CICS TS And CICS/VSE 2.3

Multi Region Operation (MRO) extensions allow CICS/VSE 2.3 and CICS TS to communicate with each other. Transaction routing, function-shipping, distributed program link, and distributed transaction processing can all be performed in either direction by defining the necessary connections and sessions between the two systems.

These facilities allow you to stage the migration. For example, you can move all conforming command-level applications and a VTAM® terminal network immediately to CICS TS while maintaining, for instance, a BTAM network and macro-level applications on CICS/VSE 2.3. Then users of the BTAM terminals can access command-level applications running in CICS TS using transaction routing. Macro-level applications on CICS/VSE 2.3 also are accessible from terminals attached via VTAM on CICS TS using transaction routing.

You also can use the coexistence scenario to allow macro-level applications on CICS/VSE 2.3 and command-level applications on CICS TS to access shared data. The shared data must reside on CICS/VSE 2.3, because macro-level file requests cannot be function-shipped. The CICS TS applications can access the data using MRO function shipping.

Of course, these are simple configurations. Examples of more complex scenarios are given in the *CICS Migration Guide*.

Care is required with CICS modules shared in the SVA. Many of the CICS management programs have the *same name* in both systems, even though the code is quite different. When such a naming conflict occurs, the CICS TS version of the module cannot be placed in the SVA, because it is not possible to prevent CICS/VSE 2.3 from using it.

To overcome this problem, we have provided a VSE/POWER source library inclusion member, `DFH$SVEX`. If you add this to the system initialization overrides for a CICS TS job, it ensures that commonly-named modules are never accessed from the SVA by that system.

Changing Your Resource Definitions

There are four main changes to resource definition for CICS TS:

1. The CICS system definition data set (CSD) is now a mandatory CICS resource, as some resources can only be defined via Resource Definition Online (RDO). It is, therefore, a critical resource and should be backed up on a regular basis.

2. RDO support has been introduced for VSAM file control and LSRPOOL resources. LSRPOOL support has been extended to provide definition of separate data and index subpools.
3. Support is introduced for a new RDO resource called **TRANCLASS**. This replaces and extends the CMXT mechanism.
4. Obsolete tables are removed.

CICS System Definition Datasets

You now have to define your transaction, program, profile, mapset and VTAM terminal resources through RDO. The old macros are still supplied with CICS TS, but only to enable you to reassemble your old PPT, PCT, and TCT tables for migration to the CSD using the DFHCSDUP facility. Other tables (DCT, JCT, PLT) must be assembled against the CICS libraries even if you don't change the source. There are some changes to the associated macros you need to be aware of, so refer to the *Migration Guide* for details.

Although the resources described above must be defined using RDO, you should consider using the terminal autoinstall facility and the new program autoinstall facility, which should help you reduce the number of resource definitions required and which in turn will reduce both COLD start times and maintenance costs.

RDO Support For VSAM File Control Resources

CICS TS has introduced RDO support for VSAM and LSRPOOL resource definitions. You can continue to use the DFHFCT macro to define your VSAM resources, but CICS TS is the last release of CICS to support definition of VSAM resources in this way. You are recommended to migrate your VSAM definitions to the CSD.

Support For An RDO Resource TRANCLASS

The CMXT system initialization parameter is obsoleted in CICS TS. We have replaced it with a new RDO resource called **TRANCLASS resource definition**. You can define an unlimited number of transaction classes by name, rather than the ten numbered classes you had available in CICS/VSE 2.3. Each class has an associated limit to the number of tasks in that class allowed to be active at one time and a purge threshold to limit the size of a queue for a particular transaction class.

Obsolete Tables

Because of the restructure of the CICS loader, both the Nucleus Load Table (NLT) and the Application Load Table (ALT) are now obsolete. Similarly, because of the removal of CICS internal security, the Sign On Table (SNT) is removed.

Changes For Application Programming

To assist in the migration of applications to CICS TS, CICS TS allocates all your program working storage, the EIB, the TWA control blocks, and any EXEC CICS SET storage below the 16MB line. This allows you to port applications without the need to change resource definitions.

If your applications are AMODE(31)-capable, you should modify the RDO TRANSACTION resource definition to specify `TASKDATALOC(ANY)` and the RDO PROGRAM definition to specify `DATALOCATION(ANY)` to achieve the maximum amount of VSCR. You can run your command-level applications on CICS/VSE 2.3 alongside CICS TS if you ensure that you compile the program with a supported compiler (Assembler, COBOL for VSE, PL/I for VSE, and C for VSE).

If your application is in Assembler or compiled with any of the LE-enabled compilers, you can port your application without the need to re-compile or link-edit. **You must not:**

- Issue any CICS macros (and this includes undocumented interfaces such as DFHTM, DFHWTO, and DFHSEC).
- Issue any EXEC CICS ADDRESS CSA commands.
- Rely on general registers 12 and 13 addressing the TCA and CSA on entry. They contain an address of a piece of fetch-protected storage, and any use of these addresses results in an abend.

You can still address the EIB but, as in previous releases, direct modification of the EIB is not supported. However, it is recognized that users do modify the EIB for various reasons; and it is probable that if such a modification works on CICS/VSE 2.3, it will work on CICS TS.

Customizing Your System

The internal structure of CICS TS is very different to CICS/VSE 2.3. Consequently, the facilities provided to allow you to customize the product have changed as well.

We have improved the global exits to make them more efficient. We have provided a new Exit Programming Interface – the **XPI** – specifically for use by GLUEs (global user exits).

You can also use more of the command level API/SPI and in more places than you could before. This has been achieved by standardizing the GLUE interface, but it does mean that you have to change your existing exits in the following manner:

- Your exit programs must be fully reentrant (if using the XPI) and be written to 31-bit addressing standards.
- Your global user exits must be altered to conform to the new calling and exit programming interface.
  - You can no longer pass parameters in registers. You must use exit-specific fields.
  - There are some extensions to the DFHUEPAR parameter list.
  - The addresses of the CSA and TCA point to fetch-protected storage areas. If your current exits make use of data obtained via these addresses, you need to review the code and use XPI and/or API/SPI as alternatives.

Using User Replaceable Modules (URMs)

These have been changed, too. They must be 31-bit capable (as must any module they call) and must be written to command-level API standards. The access to control block fields in previous releases is replaced by passing a COMMAREA containing information required by the module. They can be written in any supported language with the exception of two journaling exits (**XJCC** and **XJCO**), which must be Assembler. Although the support for journaling exits is there, you should try to use the new journal archive facility in preference.

Several URMs are now obsolete – **DFHACEE**, **DFHRTY**, **DFHUAKP**, **DFHXSE**, **DFHXSP** – and new ones have been added – **DFHPDADX** (program autoinstall) and **DFHZATDY** (terminal and APPC autoinstall). As in previous releases, VSE/ESA provides replacements for some URMs. Existing NEPs, PEPs, and so on must be reassembled even if they already conform to the new standards.

Using System Generation

You can no longer customize CICS using system generation. It is unnecessary, as CICS TS delivers full function modules which can be customized using the appropriate exit facilities.

Using Task-Related User Exits

These are not significantly changed, but you should be aware that you can issue a resource manager request within an exit which is processing another request for the same or different resource manager. There is also a new link-edit mode option on the EXEC CICS ENABLE command that allows the exit to be invoked in its link-edit AMODE rather than the mode of its caller, which is the default. The CSA and TCA addresses passed in the exit parameter list (as with GLUEs) point to fetch-protected storage.

Performing Initialization

Program list table (PLT) sequencing has changed because of the introduction of a two-phase process for initialization. Your first-stage processing programs must be written in Assembler and link-edited to run in AMODE31. The API you can use is limited to EXEC CICS ENABLE, EXEC CICS HANDLE, and EXEC CICS RETURN. Your phase one program definition is ended by a DFHDELIM statement in the table definition. ALL other PLT programs should be run in phase 2.

Because of the sequencing changes, you should check for PLT programs that issue EXEC CICS START commands with zero delay interval, as it is possible that the started transaction may run before all your CICS resources are available. In such cases, you need to add an `EXEC CICS INQUIRE SYSTEM CICSSTATUS` command to determine if CICS is active before proceeding.

If you enable the storage protect facility in the SIT, all your PLT programs must be defined as CICS key. The system initialization overlay module facility has been removed, so any such modules should be converted to run during the second stage of PLT processing.

Setting Up Your Security

CICS internal security has been removed in CICS TS. All security checking is done by an external security manager (ESM). This means that any applications you have that depend on internal security need changing.

Internally, CICS TS uses the RACROUTE interface to route requests via the new VSE system authorization facility (SAF) to an installed external security manager whenever there is a need to authenticate a request. In addition to the security facilities for signon, transaction attach, and resource security checks that existed in CICS/VSE 2.3, CICS TS also provides:

- Command-level security that protects all system programming (EXEC CICS INQUIRE and SET) from unauthorized use.
- Non-terminal task security, where a task (even if not initiated from a terminal) has an associated user ID against which security checks can be performed.

New options on the EXEC CICS START command, the DCT macro, and the new SIT options allow you to control which user ID is associated with a particular non-terminal task.

- Surrogate user security, which prevents an authorized user starting a task with another user ID to do work that he/she is not authorized to do.

There are several ESMs a customer can choose from, and VSE supplies a basic security manager (BSM) which supports both CICS signon and transaction attach security. If, for instance, you require resource level security, you need to purchase a full-function ESM. CA-Top Secret is available direct from IBM on the VSE/ESA 2.4 extended base tape.

To ease your migration to the BSM and CA-Top Secret, migration utilities are supplied to migrate SNT data to the ESM database. These build on the security migration aid that was introduced as a PTF on CICS/VSE 2.3.

Using Monitoring And Statistics

To provide a fast, system-wide data collection capability, CICS TS introduces the **Data Management Facility** (**DMF**), to which all CICS monitoring and statistics data is directed. To use this, you need to define and initialize DMF data sets and generate the DMF startup table using the **DFHDMFM** macro. A default startup table, **DFHMFSU** is provided to help.

You need to create jobstreams to process the DMF data. CICS TS provides you with a large amount of statistics and monitoring data, and thus a fast means of collection is necessary. DMF runs in a separate VSE/ESA partition and receives data via cross-memory services from CICS TS partitions (and potentially from other subsystems). The data – in SMF110 record format – is saved in a VSE/ESA dataspace (to improve performance) and then is periodically written to VSAM ESDS datasets.

You need to ensure there is sufficient space to accommodate the expected volume of statistics and monitoring data. Up to 36 datasets can be defined. The statistics datasets DFHSTM and DFHSTN are no longer used and need to be removed from any startup job streams. Similarly, the TD queues CSSM and CSSN are also obsolete and should be removed from your DCT.

Additional Information

- Sessions related to CICS TS for VSE/ESA from the 1999 VM and VSE Technical Conference in Orlando are accessible from:

  http:/www.ibm.com/s390/vse/

  These sessions cover topics such as:

  - Migrating to CICS TS for VSE/ESA
  - Overview of the CICS Web Interface and 3270 Bridge
  - Overview of CICS TS Gateway and CICS Universal Clients
  - CICS TS for VSE/ESA: New Application Support
  - CICS TS for VSE/ESA Performance
  - Problem Solving under CICS TS
  - VSE/ESA 2.4 Security

- The CICS TS home page contains down-loadable versions of all the unlicensed manuals:

  http://www.software.ibm.com/ts/cics/

- Information is in IBM ITSO redbooks:

  http://www.redbooks.ibm.com/

  Specific redbooks have information about migration to VSE/ESA 2.4 and CICS TS for VSE/ESA 1.1 (**SG24-5595**) and the implementation of VSE/ESA 2.4 and CICS Transaction Server for VSE/ESA 1.1 (**SG24-5624**).

- These IBM Software Announcement Letters also can be referenced:

  – VSE/ESA 2.4 – **298-372**
  – CICS TS for VSE/ESA Release 1 – **298-373**
  – CA-Top Secret for VSE/ESA – **298-350**

Shared Data Tables – CICS Transaction Server Provides Improved VSAM Read-Only Access

Author

Chris Smith
CICS Technical Strategy and Plans
smithch@uk.ibm.com

Overview

The Data Tables Facility, an extension to CICS File Control, was introduced in CICS/VSE Version 2 to offer improved performance for read-only access to VSAM KSDS files. This support has now been significantly enhanced with the introduction of Shared Data Tables (SDT) in CICS Transaction Server for VSE/ESA.

SDT completely replaces and extends the original data table services that were provided as part of the base product in CICS/VSE Versions 2.2 and 2.3. Under SDT, all files that are defined as data tables can potentially be shared by multiple CICS systems using cross-memory services. No changes are required to the file definitions for existing data tables.

Shared data tables is of particular benefit to read intensive enquiry and search applications and improves performance by:

- Accessing data from memory instead of from DASD
- Accessing a file of data from memory using services integrated within CICS file management instead of using VSAM services and a local shared resource (LSR) pool
- Using VSE/ESA cross-memory services instead of CICS function shipping to share a file of data between two or more CICS systems that are running in different partitions under the same VSE/ESA operating system

The use of cross-memory services is one of the major enhancements to data table services that are included in the SDT facility. This enhancement improves the performance of applications that currently use function shipping and makes file sharing feasible for applications that cannot accept the performance overhead of function shipping.

The other major enhancement is that nearly all read requests are supported for use with data tables. This enhancement extends the use of data tables to applications that include browse requests and read requests that use an imprecise key.

What Are Data Tables?

A CICS file is a representation of a data set on DASD. If you specify that the file is to use data table services, CICS copies the contents of the data set into a VSE/ESA data space when the file is opened and uses that copy whenever possible. This is what is referred to as a *data table*.

Because of the way that records are accessed, data table services can be used only with a VSAM key-sequenced data set (KSDS). The KSDS is called the *source* *data set*. The copy in memory is called the *data table*. The process of copying the records is called *loading the data table*.

The source data set must be a base VSAM KSDS, not an alternate index. However, updates made to the KSDS using an alternate index are reflected in a CICS-maintained data table. The VSAM definition of the KSDS supplies the values for maximum record length and key length.

When the file is read by a CICS application, the record is normally retrieved from the data table. When the file is updated by a CICS application, the effect depends on the type of data table that you have defined for the file.

Types Of Data Tables

There are two types of data tables – CICS-maintained data tables (CMT) and User-maintained data tables (UMT).

The records of a CMT are automatically reflected in the source data set. When you update the file, CICS changes both the source data set and the data table. A CMT is easy to implement. You need to know little about the data table services, you do not need to change your existing application programs, and full recovery support of the file is retained.

The records of a UMT are not automatically reflected in the source data set. When you update the file, CICS changes only the data table. A UMT lets you optimize the benefits of using a data table by allowing you to eliminate activity on the source data set, for update requests as well as read requests. A small number of file operations are not supported for UMTs. Thus, you might need to make minor changes to existing application programs. Also, recovery of the file is supported after a transaction failure, but not a system failure.

The Shared Data Tables Environment

The environment for sharing a data table is the same as for any CICS file. One CICS region, known as a *file-owning region* (FOR), owns the data table. Any other CICS region that uses the data table is known as an *application-owning region (AOR)*. In the FOR, the file is known as a *local file*. In the AOR, the file is known as a *remote file*.

In the context of shared data tables, the FOR is also known as a *server*, and the AOR is known as a *requester*. The same region can be both an FOR for some data tables and an AOR for others.

The data table records are stored in a VSE/ESA data space, whether the data table is to be shared by more than one CICS region or not. A separate data space is used for each VSE partition. It is obtained when the first file that is defined as a data table is opened in the partition. It is used by all CICS data tables that are owned by that partition. And it is retained until the shutdown of CICS in the partition.

The data space storage that is used by the data table is freed when the file is closed in the FOR. This storage is made available for reuse in such a way that the integrity of any AOR that was using the data table is protected.

Figure 1 illustrates how a number of AORs can use cross-memory services to execute reads or browses, using shared data table services in an FOR to access the data table.

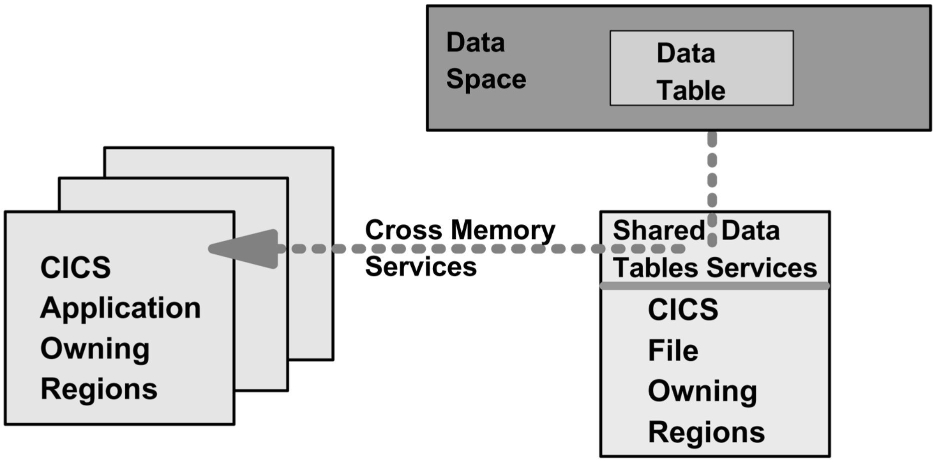


Figure 1. The Shared Data Tables Environment

Global User Exits

Three global user exits are provided to extend the normal processing done by data table services:

1. **XDTRD** provides the ability to select the records that are copied to the data table during loading when the file is opened. For a UMT, it can also be used to modify the records.
2. **XDTAD** provides the ability to select the records that are copied to the data table when new records are added to the file.
3. **XDTLC** can be used to perform processing at the end of the loading operation.

Benefits Of Shared Data Tables

SDT offers many additional benefits over the data table services that were included as part of CICS/VSE V2.2 and V2.3. For example:

- Very large reductions in path length can be achieved for remote accesses, because function shipping is avoided for most read and browse requests.
- When cross-memory services are used, the requests are processed by the AOR, thus freeing the FOR to process other requests. This can increase multiprocessor exploitation.
- Increased security of data is provided because the record information in shared data tables is stored outside the CICS region and is not included in CICS system dumps.
- For CMTs, all forms of non-update, keyed access (including browse requests and imprecise-key read requests) are processed by reference to the data table.
- For UMTs, all forms of non-update, keyed access (including browse requests and imprecise-key read requests) are supported.

- Any number of files referring to the same source data set that are open at the same time can retrieve data from the one CMT.
- An enhancement to the XDTRD user exit allows you to skip over a range of records while loading the data table.

Additional Information

Full details of Shared Data Tables is in the *CICS Shared Data Tables Guide*, SC33-1668. This is delivered with CICS Transaction Server for VSE/ESA and can also be obtained from the CICS Web site at:

http://www.ibm.com/software/ts/cics/

Additional information also is in *Implementation of VSE/ESA 2.4 and CICS Transaction Server for VSE/ESA 1.1*, SG24-5624. This is a redbook which can be obtained from the ITSO Redbooks Web site at:

http://www.redbooks.ibm.com/

The External CICS Interface – New Support For Batch Program Access To CICS Applications

Author

Chris Smith
CICS Technical Strategy and Plans
smithch@uk.ibm.com

Overview

New in CICS Transaction Server for VSE/ESA, the CIC External Interface (EXCI) provides access to CICS applications from a non-CICS environment. The EXCI is an application programming interface that allows a non-CICS program (a client program) running in VSE to call a program (a server program) running under CICS Transaction Server for VSE/ESA. A CICS communications area (COMMAREA) is used for the exchange of information, the CICS server program being invoked as if linked to by another CICS program in the same CICS system. Client programs using the EXCI may be written in Assembler, COBOL, C, or PL/I.

The EXCI allows sessions to be allocated and opened to a CICS system and for distributed program link (DPL) requests and responses to pass over them. These sessions are known as *pipes*, a pipe being a communication path between a sending process and a receiving process. The communication is implemented via the Multi Region Operation (MRO) facility of CICS Interregion Communication (IRC). When EXCI is used, each pipe maps onto one MRO session and carries information from the client program to the CICS system.

A client program that uses EXCI can have multiple sessions for different users (either under the same or separate tasks), all coexisting in the same VSE address space without knowledge of, or interference from, each other. Where a client program attaches another client program, the attached program runs under its own task.

The provision of this new programming interface means that a VSE batch job can:

- Update resources with integrity while they are also being accessed by CICS.
- Take CICS resources offline (and back online) at the start (and end) of the batch job.

For example, CICS files can be opened and closed and CICS transactions can be enabled and disabled – which could eliminate the need for a master terminal operator during system backup and recovery procedures.

Programming Interfaces

The external CICS interface provides two forms of the EXCI program interface that client programs can use to call application programs running in a CICS region – the **EXEC CALL** interface and the **EXCI CICS** interface. Either or both of these could be used in the same client program, but it is unlikely that they would be used together.

The EXEC CALL interface is a macro level interface consisting of six commands which:

- Allocate and open sessions to a CICS system from a non-CICS program.

  These are the Initialize-User, Allocate_Pipe, and Open_Pipe commands.
- Issue DPL (Distributed Program Link) requests to specific CICS application programs over these sessions.

  Here the DPL call command is used.
- Close and deallocate the sessions when DPL requests have completed.

  These are the Close_Pipe and Deallocate_Pipe commands.

The EXEC CICS interface consists of one command only. This is EXEC CICS LINK PROGRAM. It performs all six functions of the EXCI CALL interface in one invocation. That is, a single call will allocate and open a session, issue a request, and then close the session. This command is similar, but not identical, to the DPL command of the CICS command-level application programming interface. Client programs using this form of the interface must be processed by the standard CICS translator.

The EXEC CICS form of the interface is illustrated in Figure 2.

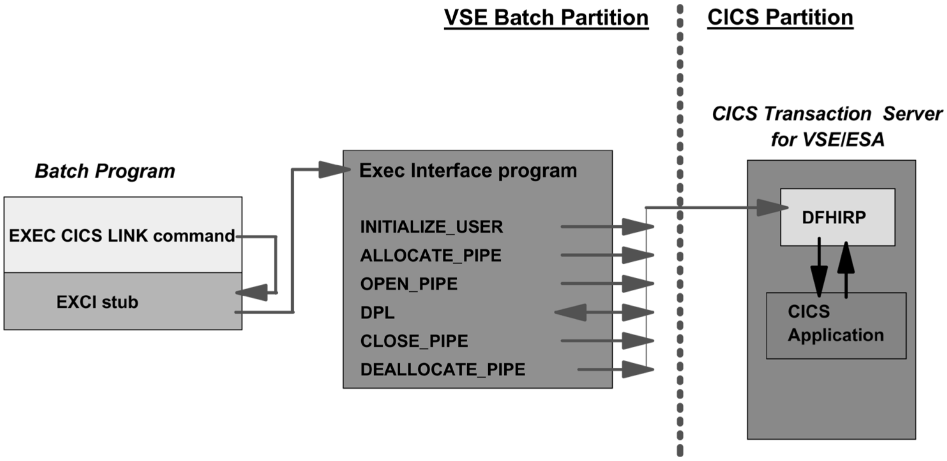


Figure 2. The EXCI EXEC CICS Interface

Choosing Between The Two Forms Of The Programming Interface

Both the CALL interface (all six commands) and the EXEC CICS LINK interface can be used in the same program to perform separate requests. However, as a general rule it is unlikely that this would be done in a production program.

Each form of the external CICS interface has its own particular benefits:

- For low-frequency or single DPL requests, it is recommended that the EXEC CICS LINK command is used. It is easier to code and, therefore, less prone to programming errors. However, each invocation of an EXEC CICS LINK command causes the external CICS interface to perform all the functions of the CALL interface, thus incurring overhead each time.

- For multiple or frequent DPL requests from the same client program, it is recommended that the EXCI CALL interface is used. This is more efficient because the Initialize_User and Allocate_Pipe commands need be performed once only, at or near the beginning of your program, and the Deallocate_Pipe once on completion of all DPL activity. In between these functions, the pipe can be opened and closed as necessary. While the pipe is open, as many DPL calls can be made as required.

Sample server and client programs are provided illustrating how both forms of the EXCI interface can be used.

Using The EXCI

The following steps are a guide to using the external CICS interface:

1. Client programs that use the EXEC CICS LINK form of EXCI must be translated before assembly or compilation, using the translator option EXCI.
2. When any program that uses the EXCI to pass DPL requests to a server region is link-edited, the application program stub DFHXCSTB must be included. It is supplied in the VSE/ESA PRD1.BASE library.
3. EXCI applications should be written to 31-bit addressing standards and specify AMODE(31).
4. The appropriate CICS-supplied copybook for RESPONSE and REASON codes should be included in the source of EXCI client programs.
5. Libraries that contain the CICS-supplied external interface modules needed by the job and by the client program should be included in the LIBDEF statement. The external interface modules, including the sample client program, are in the VSE/ESA PRD1.BASE library.
6. The size of the GETVIS available in the partition must be large enough to take the size of internal trace table specified on the TRACESZE parameter in the DFHXCOPT options table.
7. The CICS-supplied sample programs for EXCI are provided in the VSE/ESA PRD1.BASE library.
8. Any external program that uses the EXCI LINK must specify `OS390` as the last parameter on the // EXEC statement.
9. In all CICS systems that are to be accessed by EXCI client programs, the resource definition groups DFH$EXCI and DFH$FILA must be installed, and Interregion Communication must be opened (either during CICS initialization or by using CEMT SET IRC OPEN).
10. A CICS server program invoked by an external CICS interface request is restricted to the DPL subset of the CICS application programming interface. This DPL subset of the API commands is the same as for a CICS-to-CICS server program and is described in the *CICS Application Programming Guide*, SC33-1657.

Note also that client programs running in a VSE address space can communicate only with CICS server regions running under CICS Transaction Server for VSE/ESA and must use the level of the Interregion Communication program (DFHIRP) supplied with CICS Transaction Server for VSE/ESA Release 1 or later. This is because of the changes to the MRO connection definition to support EXCI.

Additional Information

Full details of the EXCI are in the *CICS External Interface Guide*, SC33-1669. This is delivered with CICS Transaction Server for VSE/ESA and can also be obtained from the CICS Web site at:

http://www.ibm.com/software/ts/cics/

Additional information also is in *Implementation of VSE/ESA 2.4 and CICS Transaction Server for VSE/ESA 1.1*, SG24-5624. This redbook can be obtained from the ITSO Redbooks Web site at:

http://www.redbooks.ibm.com/

E-Business Support With CICS Transaction Server For VSE/ESA

Author

Chris Smith
CICS Technical Strategy and Plans
smithch@uk.ibm.com

Introduction

e-business is all about combining information technology and the Internet to perform commercial operations from the Web – including, of course, transaction processing. Over the last few years, the CICS product has moved forward and added function to support this, initially with gateway access and then with direct access to CICS on MVS™ and OS/390. With the arrival of CICS Transaction Server for VSE/ESA, more function is now being provided for CICS on the VSE/ESA platform.

The main objective of CICS e-business support is to enable end users at Web browsers or Network Computers to access CICS application programs and transactions using standard Internet protocols – in particular the HyperText Transfer Protocol (HTTP) over TCP/IP networks. As shown in Figure 3 on page 25, the two main scenarios for connecting from Web browsers or Network Computers to CICS on the VSE/ESA platform are:

1. Via a Web server running on an intermediate system which connects to CICS – a 3-tier scenario
2. Via a direct connection to CICS without the need for an intermediate – a 2-tier scenario

This article provides an overview of CICS technology for customers on the VSE/ESA platform that enables Web access to CICS applications and transactions. It is delivered via the following functions:

- The CICS Transaction Gateway
- The CICS Web Interface
- The 3270 Bridge

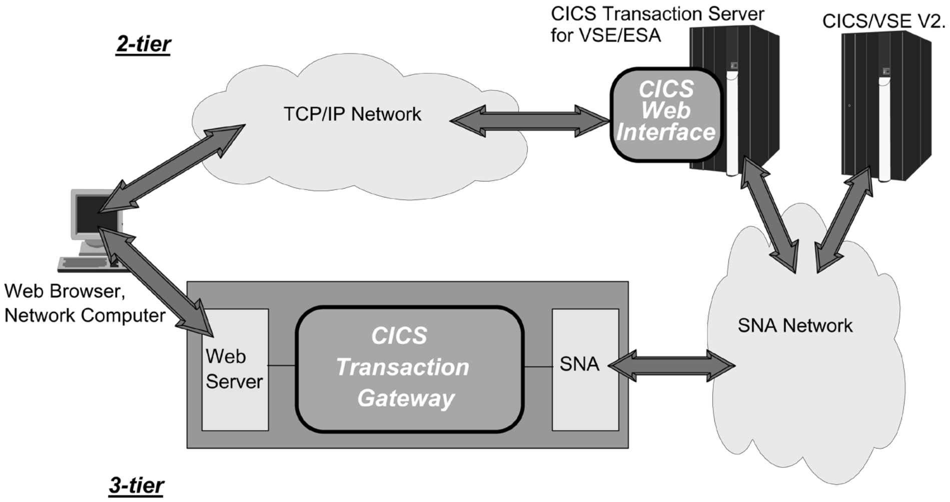


Figure 3. Web Access to CICS on the VSE/ESA Platform

The CICS Transaction Gateway

Overview

The CICS Transaction Gateway provides a comprehensive set of Java-based Web server facilities for accessing CICS application programs and transactions from a Web browser. These include Java classes and Java beans for writing Java servlets (programs that run on the Web server), Java applets (programs that run on a Web browser), and Java applications which can run on any other Java execution platform.

The CICS Transaction Gateway enables access to new and existing CICS applications from modern Java Web-based front-ends in a 3-tier scenario. It runs on the same intermediate system as a Web server and connects via SNA LU6.2 to CICS/VSE V2.3 and CICS Transaction Server for VSE/ESA.

Java Support

The Java support provided can be used in the following main ways:

- **Writing servlets**: The CICS Transaction Gateway supplies Java classes and Java beans which allow servlets to be written to handle browser requests that require services from CICS. Servlets are programs that are loaded by the Web server and run in the Web server environment.

  The provided Java classes and Java beans enable access to existing CICS applications, or new applications, or a combination of both. Included are data conversion and manipulation functions that are needed to invoke CICS transactions with a 3270 or COMMAREA interface and to communicate with a browser.

- **Writing applets**: The same Java classes and Java beans can be used to write an applet to access CICS in the same ways as a servlet. The difference is that an applet executes under a Web browser on the end-user workstation after being downloaded from a Web server. It requires a Java-enabled browser on the workstation, such as Netscape or Internet Explorer.

Applets have some restrictions compared to Java servlets and Java applications. For example, they may not access local resources such as local files.

- **Writing applications**: Again, the same Java facilities are available to access CICS as for servlets and applets. The main difference is that Java applications do not run with a Web server or Web browser. Instead, they run in any other Java execution environment and have none of the limitations that an applet has.

  A Java application can run on the same system as the CICS Transaction Gateway or remotely on another system or workstation.

- **Turnkey access to 3270 transactions**: The CICS Transaction Gateway supplies one pre-written servlet, called the "terminal servlet". This provides turnkey Web browser access to CICS 3270-interface transactions. The servlet translates browser requests to CICS interactions on a one-for-one basis, so that the user sees the browser equivalent of the "green" screens that would appear on a real 3270 terminal. It uses the Java classes and beans described above to convert URL-encoded HTTP Web browser input to 3270-format data inbound and to convert 3270-format transaction output to HTML for return to the Web browser.

- **Tailored access to 3270 transactions**: The CICS Transaction Gateway also provides facilities for customizing (without programming) the "green" screen interface produced by the terminal servlet. For example, it is possible to map specific screens to specific HTML pages, to display specific HTML text, and to change or suppress PF Keys. Variable substitution also can be used with Web server server-side includes to map one Web browser request to several 3270 interactions or to invoke a 3270 transaction and extract data from the returned screen.

Major Components

Figure 4 on page 27 illustrates the overall CICS Transaction Gateway structure and environment. These major components are delivered in the product:

- A **Java gateway application** that resides on a Web server. On non-OS/390 platforms, it communicates with CICS applications through facilities provided by the CICS Universal Client component. On OS/390 systems, it communicates with CICS applications via the CICS External Interface (EXCI).

- A **CICS Universal Client** that provides the basic interfaces and connectivity to host CICS systems, as well as terminal emulation function.

- A **CICS Java class library** that provides an API for the communication between a user-written Java application or applet or servlet and a host CICS system. For example, the JavaGateway class is used to establish communication with the gateway process, the ECIRequest class is used to specify External Call Interface calls, the EPIRequest class is used to specify External Presentaton Interface calls, and the ESIRequest class is used to specify External Security Interface calls.

- A **set of Java EPI beans** for creating Java front-ends to existing CICS 3270-based applications without the need for programming. Java beans are self-contained, reusable software components that can be visually composed into applications, applets, or servlets via the use of visual application builders – for example, the IBM VisualAge® for Java product.

- The **terminal servlet** described previously that allows a Web browser to be used to access existing 3270-based CICS transactions.
- A **set of Common Connector Framework beans**, which are supplied to support IBM's common connector framework strategy. These map to ECI and EPI for CICS.

  The objective of CCF is to provide consistent client application interaction with enterprise resources such as CICS or DB2 from any Java execution environment.
- A **TN3270 server process** which provides support to allow multiple end-users using telnet TN3270 clients to connect to CICS.

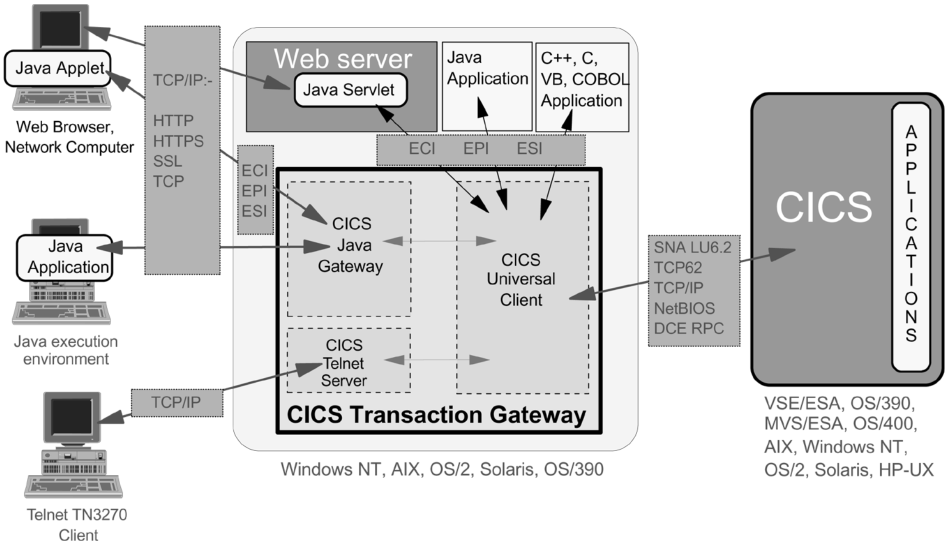


Figure 4. The CICS Transaction Gateway Environment

Application Programming Interfaces

The CICS Transaction Gateway provides three APIs, which are implemented via Java classes:

1. The **External Call Interface** (**ECI**) enables a client (a Java applet, servlet, or application) to call a CICS program and is equivalent to Distributed Program Link. It uses a COMMAREA interface; and to the target CICS application program, it appears as if it is being called by another program in the same host CICS system.

   Multiple CICS programs can be called, either in the same CICS host system or in different systems, within the same invocation of the client.
2. The **External Presentation Interface** (**EPI**) enables a non-CICS client application to emulate a 3270 terminal and drive a CICS 3270-based application, thus allowing existing applications to be enhanced with modern GUI front-ends.

   Essentially the client application simulates a 3270 terminal, and there is no change to the CICS 3270 application. The basic interface is at the 3270 datastream level, but there are higher-level Java EPI classes and Java EPI beans that can be used to shield the programmer from this.

   The same client can be connected to multiple CICS systems or act as multiple 3270s to the same CICS system.

3. The **External Security Interface** (**ESI**) enables a client application to invoke services provided by APPC Password Expiry Management (PEM) – for example, to verify or change passwords. The ESI requires an external security manager on S/390 on which the host CICS system is running. This provides support for an APPC architected sign-on transaction that signs on users to a CICS host system and which can processes requests for a password change.

Note: These same APIs also are available with the CICS Universal Client product for COBOL, C, C++, and Visual Basic for access to CICS from workstations outside of a Web environment. These include client applications running on the same system as the CICS Transaction Gateway.

Connectivity

The CICS Transaction Gateway supports connection to host CICS systems using SNA LU6.2. For Windows NT, Windows 95/98 and OS/2®, there is also support for TCP62 via the IBM Communications Server. This enables transport of SNA LU6.2 data over TCP/IP connections to CICS on MVS or OS/390.

In addition, the CICS Transaction Gateway supports TCP/IP and DCE RPC connection to a host CICS system running on AIX®, HP-UX, Solaris, and Windows NT. This could be used as an intermediate CICS system which can connect to CICS on S/390 using SNA. The NetBIOS protocol is also supported for connection to CICS for OS/2

Note: For connection to CICS/VSE V2.3 or CICS Transaction Server for VSE/ESA, SNA LU6.2 is the only protocol supported.

Network protocols supported for connectivity to Web browsers include TCP/IP Sockets, SSL, HTTP and HTTP-S. Support of Secure Sockets Layer (SSL) enables data encryption for network security.

Supported Platforms

The CICS Transaction Gateway is supported for execution on:

- AIX
- OS/2
- Windows NT
- Windows 98 and Windows 95 (for Development purposes only)
- Solaris
- OS/390

In addition, IBM has issued a Statement of Direction for support of HP-UX.

The terminal servlet requires a Web server or servlet engine that has support equivalent to the Java Servlet Development Kit (JSDK) V1.1 or later, as provided by the IBM WebSphere™ Application Server products.

The CICS host servers supported for use are:

- CICS Transaction Server for OS/390
- CICS for MVS/ESA V4.1
- CICS Transaction Server for VSE/ESA
- CICS/VSE V2.3
- CICS Transaction Server for OS/2
- TXSeries™ (CICS for AIX, Solaris, HP-UX, Windows NT)
- CICS for OS/400®

Licensing And Packaging

The CICS Transaction Gateway is licensed and delivered with:

- CICS Transaction Server for OS/390 Version 1.1, or later
- CICS Transaction Server for VSE/ESA Version 1.1
- TXSeries, Version 4.2, for AIX, Solaris, Windows NT, or HP-UX
- CICS Transaction Server for OS/2 Warp, Version 4.1
- VisualAge for Java Professional or Enterprise Edition Version 2.0 and Version 3.0
- WebSphere Application Server Advanced or Enterprise Edition Version 2.02, or later

The license entitles unlimited use and unlimited right to copy. Note that the license delivered with CICS Transaction Server for VSE/ESA entitles immediate use of the CICS Transaction Gateway with CICS/VSE V2.3.

The CICS Transaction Gateway also may be downloaded from the CICS Web site for use under a license obtained via one of the products listed above.

The CICS Web Interface

Overview

The CICS Web Interface is a new set of services within CICS Transaction Server for VSE/ESA. It enables direct access to CICS applications and transactions from Web browsers or Network Computers in a 2-tier scenario, which does not require an intermediate system running a Web server or gateway. The CICS Web Interface provides support for TCP/IP and the HTTP protocol, new APIs for "Web aware" applications, and the ability to access existing 3270 transactions via the new 3270 Bridge function.

One or more TCP/IP ports are assigned to CICS on which CICS listens for incoming requests and then processes them. The CICS network address and port are specified in a standard Universal Resource Locator (URL) which is sent from a Web browser. Also included in the URL is the name of the CICS application program to process the request.

The CICS Web Interface and 3270 Bridge were first provided on CICS Transaction Server for OS/390. Early availability on CICS Transaction Server for VSE/ESA is planned for the first quarter of 2000 at no extra charge.

Architecture

A CICS application is invoked from a Web browser via a standard URL of the general form:

http://machine.name:port/converter/alias/program?optional-token

where:

- **machine.name** is the IP address or DNS name of the CICS region.
- **port** is the configured CICS Web Interface listening port number.
- **converter** is the name of the program used for Decode and Encode processing. `CICS` or `cics` is specified if a converter is not required.
- **alias** is the transaction id of the alias transaction. `CWBA` is the supplied alias.

- **program** is the name of the application program to be invoked.
- **optional-token** is optional data to be passed with the request.

For example, **http://cicstest.hur.ibm.com/cics/cwba/Webpgm1**

The general flow, illustrated in Figure 5, is as follows:

1. A request from a Web browser comes into CICS via the CICS TCP/IP interface and the CICS sockets listener, which then attaches a transaction for Web attach processing.
2. Web attach processing receives the incoming request, translates the HTTP headers to EBCDIC and links to the Analyser. (See page 31.) The Analyser interprets the request and specifies the CICS resources needed to process it. On return, if requested by the Analyser, the body of the HTTP request is translated to EBCDIC. Web attach processing then starts an alias transaction to handle all further processing of the request and terminates.
3. If requested, the alias first calls a converter for decode functions to allow modification of the request before it is passed to the application in the COMMAREA. The alias then calls the CICS application program, passing the COMMAREA as setup by decode.
4. The CICS application program processes the request and builds an HTTP response either using the new CICS Web-related API commands or directly in the COMMAREA.
5. On return from the application, a converter is called if requested for encode functions that can build or modify the headers and body of the HTTP response. If requested, the alias translates the response to ASCII and then returns the response to the CICS TCP/IP interface (sockets domain) and terminates.
6. A call is then issued to the operating system TCP/IP services to send the complete response back to the Web browser.

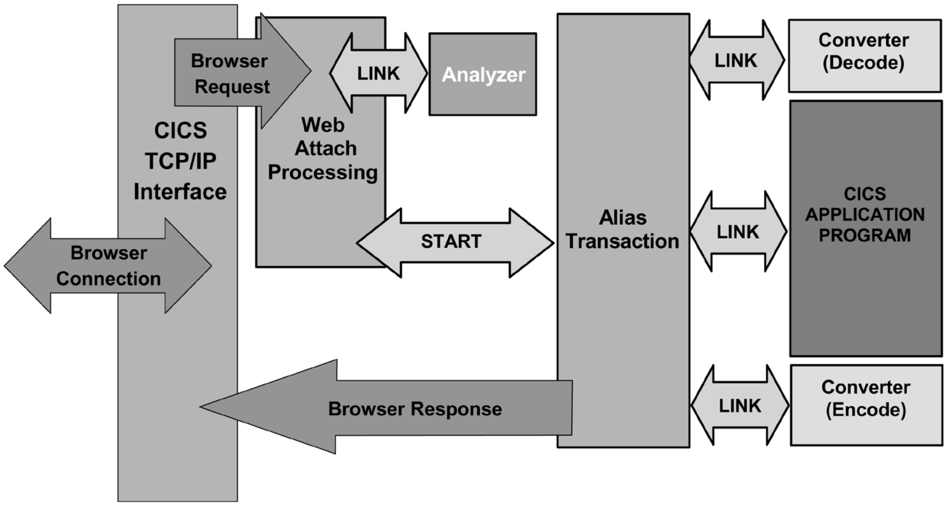


Figure 5. The CICS Web Interface Architecture

The Analyser

The Analyser is a User Replaceable Module which is mandatory, since it defines codepage conversion. It interprets the incoming request and specifies the CICS resources needed to process it. For example, this can be the name of the CICS program, name of a converter, user ID or terminal ID to be associated with the alias transaction, or codepage conversion. There is an Analyser for each TCP/IP port on which CICS listens for requests. The supplied default Analyser:

- Supports the general CICS Web Interface URL format
- Supports an optional token of eight bytes in length
- Provides ISO-8859-01 codepage conversion (standard latin-1 HTML code page)

A replacement Analyser could be provided for a number of reasons – for example, to allow a different URL convention or to support multiple codepages.

The Converter

The Converter is an optional User Replaceable Module which provides decode and encode functions. Decode is invoked before the incoming request is passed to the CICS application program. Encode is invoked after the CICS application program has processed the request.

The main purpose of decode is to provide the COMMAREA in the format expected by the application. If no decode function is specified in the URL, then the entire HTTP request is passed in the COMMAREA to the application program.

The main purpose of encode is to provide a complete HTTP response – that is, the HTTP response headers together with the body of the response. It can create or modify any headers and the body supplied by the application program. If no encode function is specified, then the application program must build the entire HTTP response in the COMMAREA.

New Web-Related APIs

There are two new sets of APIs which are provided for the development of Web-aware applications – **EXEC CICS WEB** and **EXEC CICS DOCUMENT**. It is recommended that these new APIs be used, although it is possible to write COMMAREA-style applications for handling HTTP requests and responses.

The new EXEC CICS WEB API provides an easier way of handling HTTP requests and responses compared to processing the data as one long string in a COMMAREA. The WEB commands are:

EXTRACT – Get information about the incoming HTTP request

READ – Get the value of a specific header

STARTBROWSE, READNEXT, ENDBROWSE – Get header information sequentially

RECEIVE – Get the body of the HTTP request

WRITE – Add a specific output header to the response

SEND – Select a document for delivery as the body of the HTTP response

RETRIEVE – Get the token of the document sent via an earlier WEB SEND command

The new EXEC CICS DOCUMENT API introduces the concept of *documents* – collections of formatted data areas. Its prime use is for constructing HTML pages for sending to a Web browser. (Note, however, that other markup languages could be used.)

Documents can be made up of both text and binary elements and can contain templates and symbols. Symbol substitution is done at execution time. Bookmarks can be used to insert data at specific points in documents, and documents can be imbedded within documents.

Templates which contain static data and symbols can be created offline. Both are new CICS resources defined via RDO with a resource type of `DOCTEMPLATE`. They can reside in several places (for example, CICS files, Transient Data queues, Temporary Storage queues, VSE library members).

The DOCUMENT commands are:

CREATE – Create a new document

INSERT – Insert data into an existing document

SET – Add symbols and associated values to the document symbol table

RETRIEVE – Retrieve a document into application storage

Accessing Existing COMMAREA-Based Applications

To access an existing application that is COMMAREA driven, a Converter could be used to shield the application from HTTP and HTML.

A decode function could be used to reformat the COMMAREA into the form expected by the application. And similarly, an encode function could be used to build the HTTP headers and the HTML body of the response after the application program has processed the request and puts its results in the COMMAREA.

Accessing Existing 3270 Transactions

Existing 3270-based transactions are accessed via use of the new 3270 Bridge function, described in “The 3270 Bridge” on page 33. The supplied Web bridge exit program automatically handles conversion of 3270 datastreams to and from HTML and supports both BMS and 3270 terminal control. No user programming is required, and no changes are required to the 3270 transaction or associated application programs.

Support is provided for customization of the standard HTML output produced, both for BMS and 3270 terminal control applications. For example, it is possible to suppress a PF key or change the name of the text displayed to represent it, to supply a different HTML page background or its color, or to insert additional text to be displayed in the HTML page.

The 3270 application is invoked with a URL that must specify `DFHWBTTA` as the CICS program to be invoked, followed by the 4-character ID of the required 3270 transaction to be executed. For example:

http://testcics/cics/cwba/dfhwbtta/CECI

Managing The CICS Web Interface

To be able to use the CICS Web Interface, CICS must be started with TCPIP=YES. This is a new parameter in the System Initialization Table (SIT). TCPIP=NO is the default.

A new CICS resource type, TCPIPSERVICE, defines the instance and characteristics of a TCP/IP service. Multiple TCPIPSERVICE definitions can be used to enable CICS to listen on many ports and IP addresses at the same time. TCPIPSERVICE is defined online via the CEDA transaction in the same way as other CICS resources.

There also are new CEMT and System Programming Interface (SPI) commands to perform INQUIRE and SET operations on TCP/IP and Web resources.

The 3270 Bridge

The 3270 Bridge is a new function within CICS Transaction Server that provides the ability to run existing 3270-based transactions without a 3270 and without changing the associated applications. It enables invocation of existing 3270 applications in new ways. This includes adding new GUI front-ends or accessing CICS from non-CICS environments – in particular, from the Internet or via MQSeries®.

With this new support, all 3270 terminal I/O requests are intercepted and handled by a Bridge Exit program. The Bridge Exit program is a User Replaceable Module which essentially emulates a 3270 terminal transparently to the 3270 transaction being executed. It intercepts all the 3270 input and output commands issued by the application program and does all necessary data transformation and routing. The 3270 bridge environment is illustrated in Figure 6 on page 34.

Two Bridge Exit programs are supplied. One is a sample in source form which uses Temporary Storage or Transient Data queues for input and output. The other is a specific program for use only with the CICS Web Interface to enable access to existing 3270-based transactions. It is supplied in object code form.

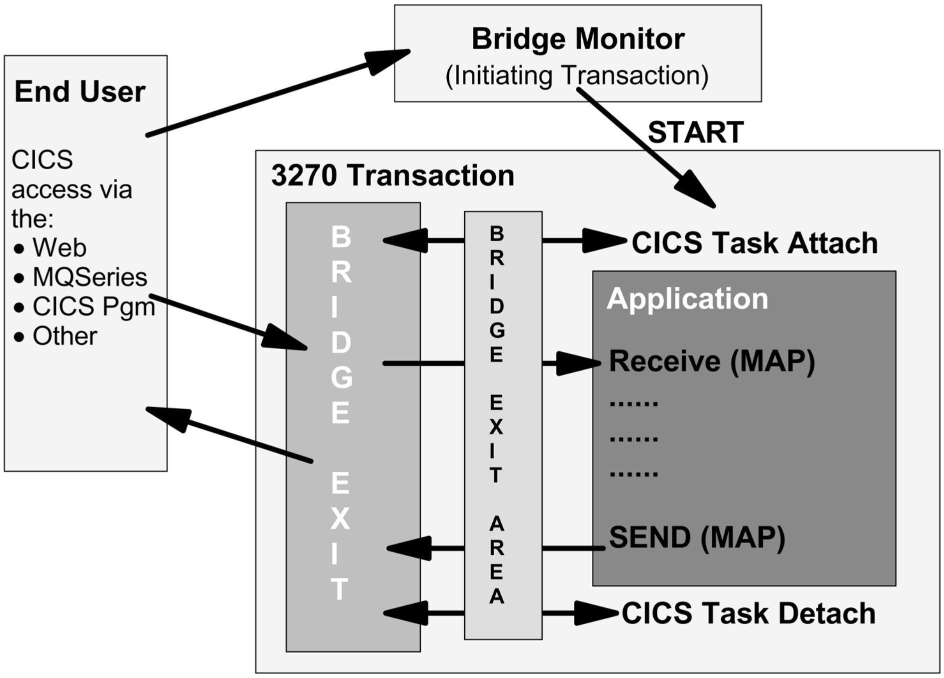


Figure 6. The 3270 Bridge Environment

Additional Information

Further information on the CICS Transaction Gateway, the CICS Web Interface, and the 3270 Bridge is on these Web sites:

- **http://www.ibm.com/software/ts/cics/** (CICS)
- **http://www.redbooks.ibm.com/** (ITSO Redbooks)

Relevant publications available on the CICS Web site include:

- A CICS Web Enablement Selection Guide (white paper)
- The set of CICS Transaction Gateway books
- CICS Transaction Server for OS/390 V1.3 books:
  - *CICS Internet Guide* (includes CICS Web Interface details)
  - *CICS External Interfaces Guide* (includes 3270 Bridge details)

Relevant publications available on the ITSO Redbooks Web site include:

- *Revealed! Architecting Web Access to CICS*
- *Revealed! CICS Transaction Gateway with More CICS Clients Unmasked*
- *CICS Transaction Server for OS/390 V1.3: Web Support and 3270 Bridge*

IBM also provides a services offering, *Planning Services for CICS Web Enablement*. Details are at:

http://www.as.ibm.com/asww/offerings/mww73bE.html

CICS Transaction Server For VSE/ESA – Problem Determination

Author

Charles E. (Chuck) Olsen
VSE/ESA Level 2
colsen@us.ibm.com

Overview

Suddenly, a noise penetrates your dreams. The waves and sand vanish in an instant, and you are back in your darkened bedroom. The phone jangles insistently from the nightstand. Giant green numbers reflect 2:05 off the ceiling. As you reach for the phone with one hand and the light with the other, deep in your heart you know that you will hear:

> *"John, the system is down again!"*

This article won't prevent you from getting these calls. It will, however, show some techniques to quickly establish the extent of the outage, most of which can be done over the phone with the systems operator, while you are deciding whether to make the 45 minute drive to the office. All this without even taking a dump.

This article discusses problem determination techniques for the new CICS Transaction Server shipped in VSE/ESA 2.4. CICS TS, as it is known by its friends, is certainly the most revolutionary change in VSE/ESA online processing in this decade. However, before we discuss specific problem determination for CICS TS, we need to backup and get a running start. Let's start out by discussing problem isolation for the VSE/ESA system in general.

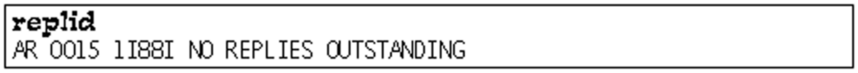
Many users, particularly our online customers, are unable to distinguish between a system outage and a transaction hang. To them, they pressed `ENTER`, and the terminal only glowered at them. To the VSE/ESA system, there are five varieties of problems which will prevent access to system resources. In addition to these, CICS TS adds a few of its own.

1. System disabled wait state (hard wait)
2. System disabled loop
3. System enabled wait state (soft wait)
4. Partition wait
5. Partition loop

Is The System In Disabled Wait State (Hard Wait) Or Disabled Loop?

Even though it may seem trivial, it is important to go through these steps each time when faced with a system outage. What is happening in other partitions can have a significant impact on the availability of our online system. So, even though we want to dive head-first into CICS TS, let's see what is going on with the VSE/ESA system. Let's first establish if the attention routine is responding.

There are two different kind of attention commands, immediate commands (which are processed inline by $IJBAR) and more complex commands (which require calling other modules or even dispatching a separate VSE task). Start off with an immediate command such as `REPLID` (Figure 7 on page 36).



```
replid
AR 0015 1I88I NO REPLIES OUTSTANDING
```

Figure 7. REPLID Command Output

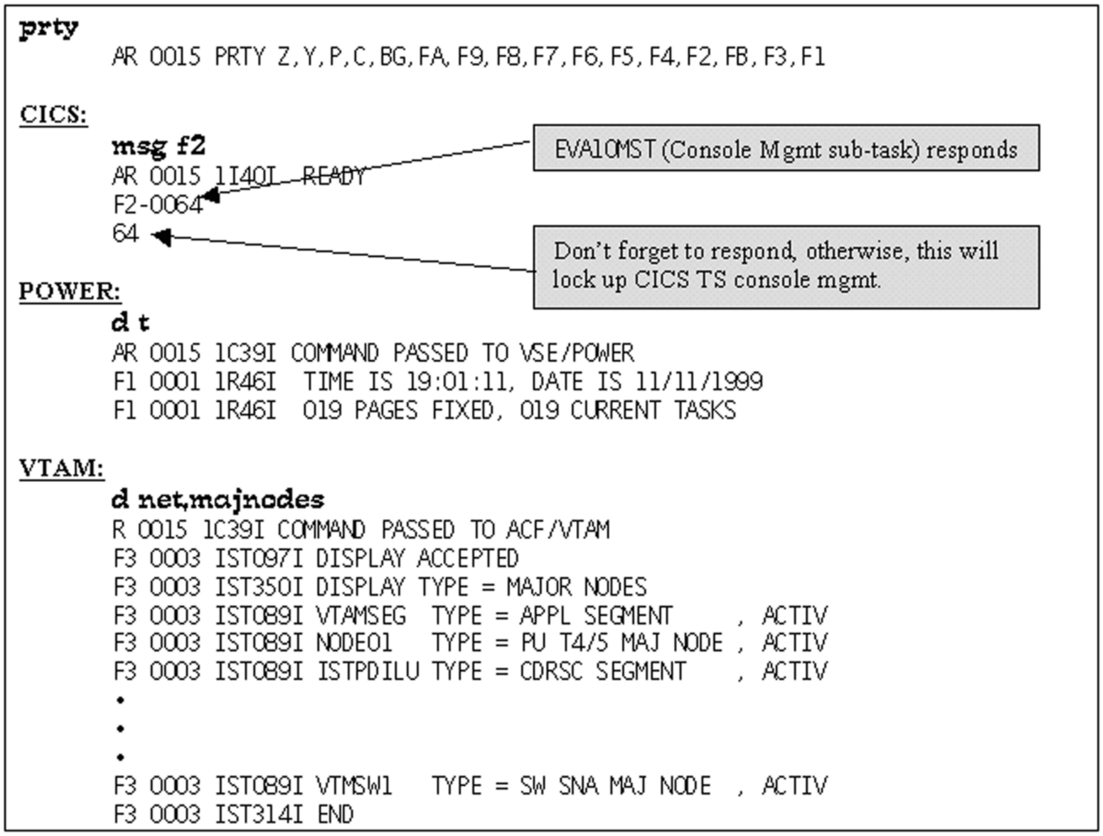
Did we get a reply? No, we have a system problem. However, VSE/ESA system problem determination for hard waits or disabled loops is beyond the scope of this article, so we will assume we did and move on to checking out the higher priority partitions.

Check The Status Of Higher-Priority Partitions

Unless our CICS TS partition is running as the highest priority partition in the system, we need to see what the higher priority partitions are doing. If they are hung, no sense even looking at CICS TS.

Figure 8 shows some specific commands we can use to check the status of higher priority partitions. Did the partitions respond? If not, we should transfer our attention to them. V.s.p.d.i.b.t.s.o.t.a.1.[2]

We will assume the partitions did respond. That being the case, let us check out the CICS TS partition. For the sake of this article, we will assume CICS TS is running in F2.



```
prty
       AR 0015 PRTY Z,Y,P,C,BG,FA,F9,F8,F7,F6,F5,F4,F2,FB,F3,F1

CICS:
       msg f2
       AR 0015 1I40I  READY
       F2-0064
       64

POWER:
       d t
       AR 0015 1C39I COMMAND PASSED TO VSE/POWER
       F1 0001 1R46I  TIME IS 19:01:11, DATE IS 11/11/1999
       F1 0001 1R46I  019 PAGES FIXED, 019 CURRENT TASKS

VTAM:
       d net,majnodes
       R 0015 1C39I COMMAND PASSED TO ACF/VTAM
       F3 0003 IST097I DISPLAY ACCEPTED
       F3 0003 IST350I DISPLAY TYPE = MAJOR NODES
       F3 0003 IST089I VTAMSEG  TYPE = APPL SEGMENT     , ACTIV
       F3 0003 IST089I NODE01   TYPE = PU T4/5 MAJ NODE , ACTIV
       F3 0003 IST089I ISTPDILU TYPE = CDRSC SEGMENT    , ACTIV
       •
       •
       •
       F3 0003 IST089I VTMSW1   TYPE = SW SNA MAJ NODE  , ACTIV
       F3 0003 IST314I END
```

EVA1OMST (Console Mgmt sub-task) responds

Don't forget to respond, otherwise, this will lock up CICS TS console mgmt.

Figure 8. VSE Subsystem-Specific Commands

[2] VSE/ESA system problem determination is beyond the scope of this article.

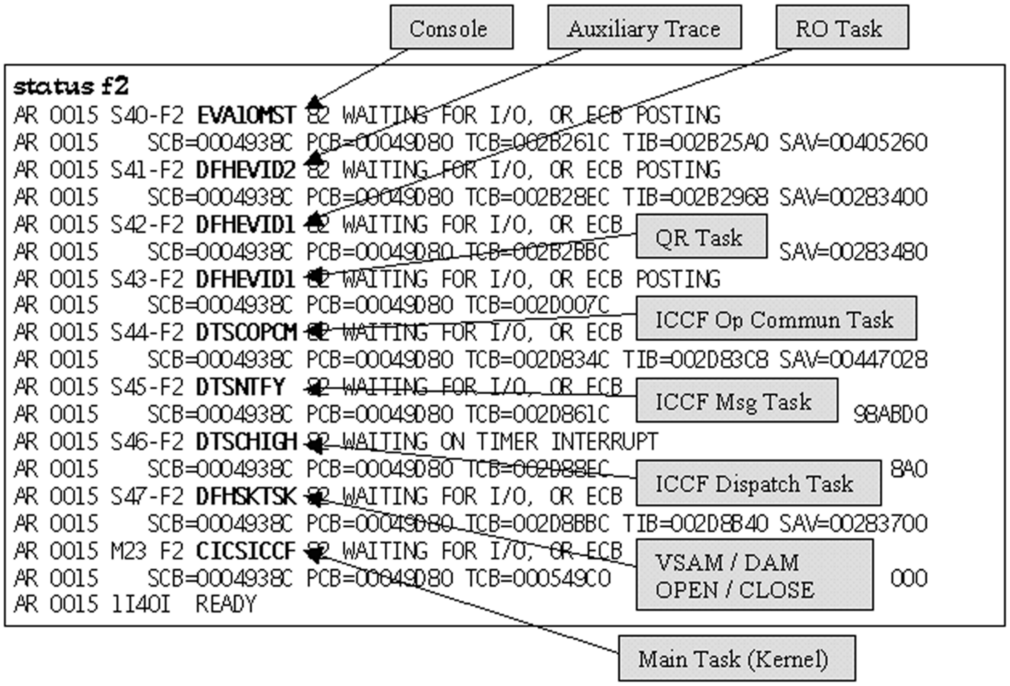
CICS TS Use Of VSE/ESA Subtasks

Under CICS/VSE, user transactions ran in the CICS partition main task. VSE subtasks were primarily used for file control open/close and dump processing.

CICS TS now uses multiple VSE/ESA subtasks. The main task (**Kernel**) is used primarily for system startup and termination. Since PC and AB exits are acquired during system startup, system recovery routines also run under the main task. Three VSE/ESA subtasks are used to run CICS TS user transactions and have the same name (**DFHEVID1**):

- **Quasi-Reentrant** (**QR**) task – This runs "normal" task code.
- **Resource-Owning** (**RO**) task – CICS code will switch to this task when a long VSE/ESA wait will occur that would stop other CICS tasks from being dispatched via the QR subtask (for example, OPEN and CLOSE TD files), CICS-managed files such as dump datasets, load a program, journal archive submit, security SVC interface, etc.
- **Secondary LU Usage** (**SZ**) task – Front End Programming Interface (FEPI)

In the VSE/ESA `STATUS` command display, the order from top is RO, QR, then SZ if FEPI is active. Thus the subtask we are primarily interested in would be the second DFHEVID1 task. Figure 9 has an example of `STATUS` output for the CICS TS partition.



```
status f2
AR 0015 S40-F2 EVA1OMST 82 WAITING FOR I/O, OR ECB POSTING
AR 0015    SCB=0004938C PCB=00049D80 TCB=002B261C TIB=002B25A0 SAV=00405260
AR 0015 S41-F2 DFHEVID2 82 WAITING FOR I/O, OR ECB POSTING
AR 0015    SCB=0004938C PCB=00049D80 TCB=002B28EC TIB=002B2968 SAV=00283400
AR 0015 S42-F2 DFHEVID1 82 WAITING FOR I/O, OR ECB
AR 0015    SCB=0004938C PCB=00049D80 TCB=002B2BBC                SAV=00283480
AR 0015 S43-F2 DFHEVID1 82 WAITING FOR I/O, OR ECB POSTING
AR 0015    SCB=0004938C PCB=00049D80 TCB=002D007C
AR 0015 S44-F2 DTSCOPCM 82 WAITING FOR I/O, OR ECB
AR 0015    SCB=0004938C PCB=00049D80 TCB=002D834C TIB=002D83C8 SAV=00447028
AR 0015 S45-F2 DTSNTFY  82 WAITING FOR I/O, OR ECB
AR 0015    SCB=0004938C PCB=00049D80 TCB=002D861C                    98ABD0
AR 0015 S46-F2 DTSCHIGH 82 WAITING ON TIMER INTERRUPT
AR 0015    SCB=0004938C PCB=00049D80 TCB=002D88EC                       8A0
AR 0015 S47-F2 DFHSKTSK 82 WAITING FOR I/O, OR ECB
AR 0015    SCB=0004938C PCB=00049D80 TCB=002D8BBC TIB=002D8B40 SAV=00283700
AR 0015 M23 F2 CICSICCF 82 WAITING FOR I/O, OR ECB
AR 0015    SCB=0004938C PCB=00049D80 TCB=000549C0                       000
AR 0015 1I40I  READY
```

Figure 9. Status Command Output for CICS TS Partition

CICS TS Lockout Conditions

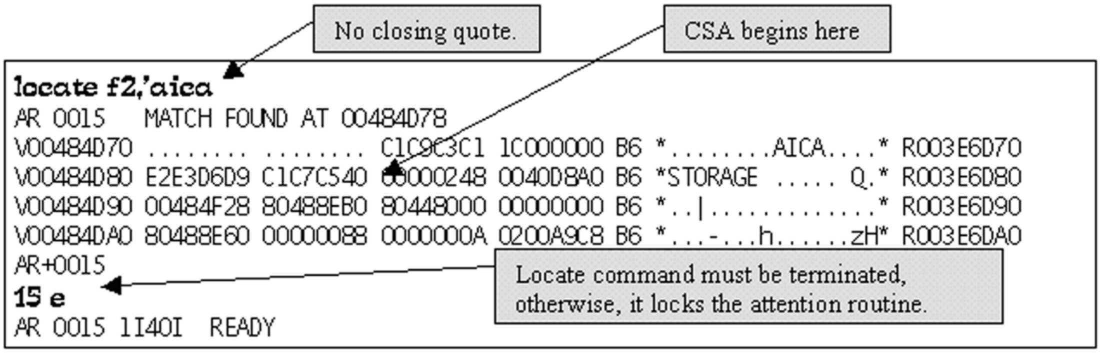
Online users may experience a lock-out condition, even when the rest of the VSE/ESA system is experiencing no problems. With VSE/ESA, a loop or wait in the CICS/VSE main task would prevent any transactions from running. With CICS TS, this is much less likely, since the user transactions run in the DFHEVID1 subtask (usually Quasi-Reentrant task). There are still conditions, however, where the QR task may hang:

1. QR subtask may be in a VSE wait state. This is normally caused by a transaction issuing VSE function calls without going through the normal CICS TS interfaces. An example would be a transaction accessing VSE/VSAM files directly without using CICS TS File Services.
2. QR subtask may be looping. Normally, CICS TS sets a "timer pop" and will interrupt a transaction which has run too long. Therefore, it is imperative that the runaway task control (ICVR timer) not be disabled (CEMT SET SYSTEM RUNAWAY[nnn]).

   There is never more than one QR task in a CICS system, and CEMT runs under the QR task. Thus in the case of a runaway task, the operator will not even be able to invoke CEMT.
3. An individual transaction may be in a quiesce or wait condition.

There is a problem determination trick to see if CICS (either CICS TS or CICS/VSE) is currently servicing transactions. Each time the task dispatcher (DFHKCP on CICS/VSE or Dispatcher Domain on CICS TS) acquires control, it updates the current dispatch time in the Common System Area (CSA) – displacement X'50' in both CICS/VSE and CICS TS.

Locate the CSA by searching for `AICA` starting from the beginning of the CICS TS partition, as shown in Figure 10. Under both CICS/VSE and CICS TS, the CSA immediately follows the eyecatcher: "AICA STORAGE".



No closing quote.

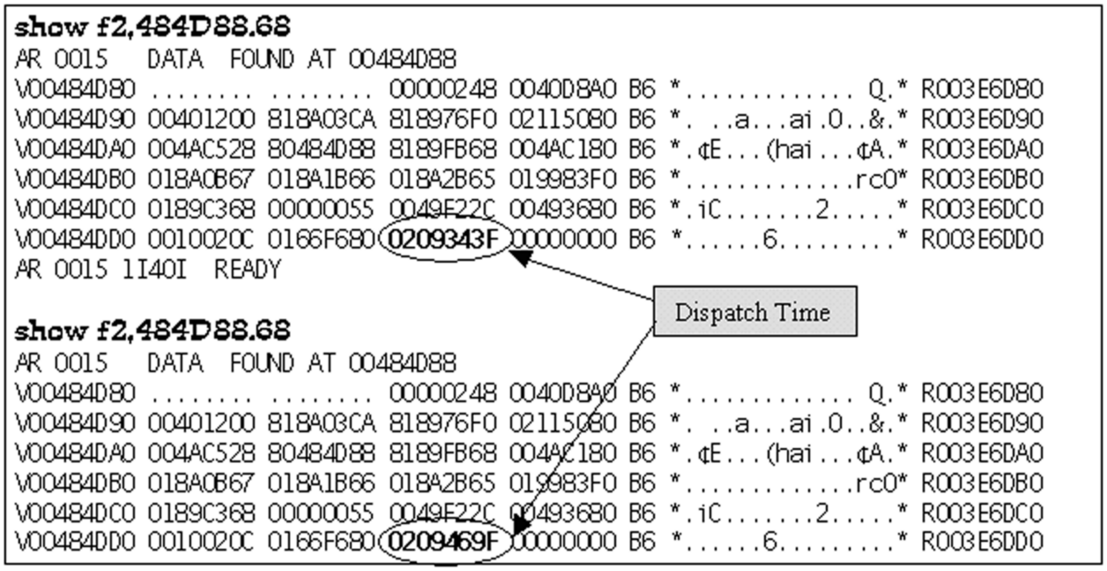
CSA begins here

```
locate f2,'aica
AR 0015   MATCH FOUND AT 00484D78
V00484D70 ........ ........ C1C9C3C1 1C000000 B6 *........AICA....* R003E6D70
V00484D80 E2E3D6D9 C1C7C540 00000248 0040D8A0 B6 *STORAGE ..... Q.* R003E6D80
V00484D90 00484F28 80488EB0 80448000 00000000 B6 *..|.............* R003E6D90
V00484DA0 80488E60 00000088 0000000A 0200A9C8 B6 *...-...h......zH* R003E6DA0
AR+0015
15 e
AR 0015 1I40I  READY
```

Locate command must be terminated, otherwise, it locks the attention routine.

Figure 10. Locate CICS Common System Area

Figure 11 on page 39 shows the location of the current dispatch time at CSA + X'60'. Display the first 60 bytes of the CSA several times, and see if this value changes. If the current dispatch time changes, this indicates that CICS dispatching is working, and the problem is likely due to individual transaction waits. In the example in Figure 11 on page 39, the dispatch time changed from **02:09:34.3** to **02:09:46.9**. (Remember, the call came in at 2:05 am.)



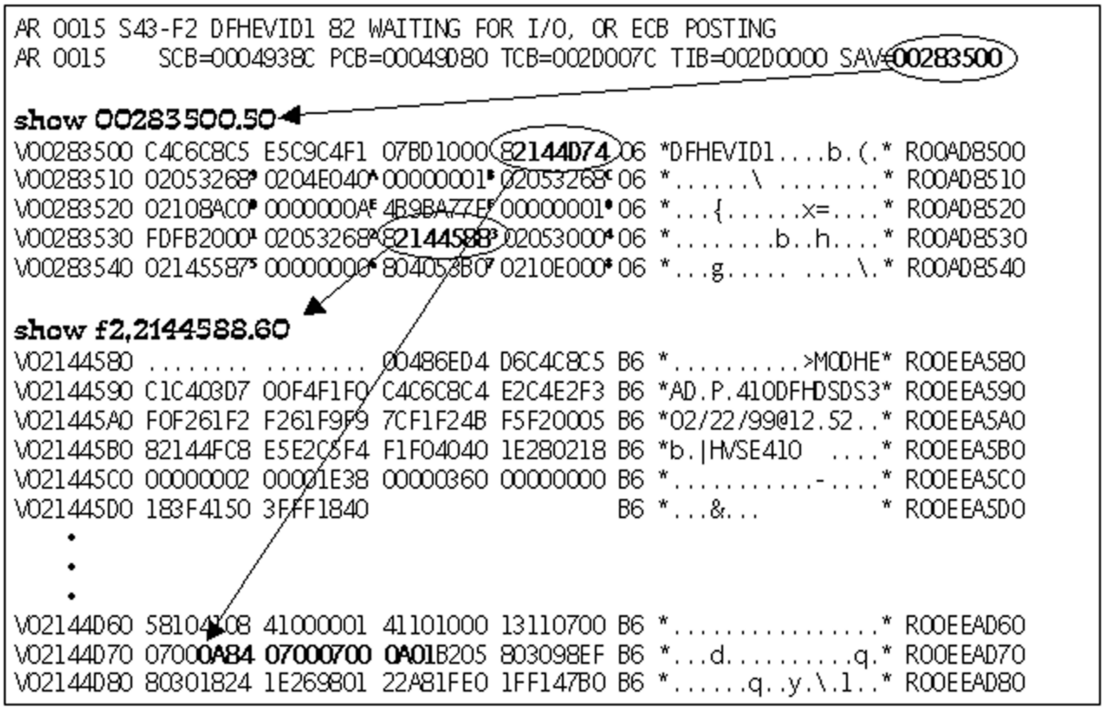
```
show f2,484D88.68
AR 0015   DATA  FOUND AT 00484D88
V00484D80 ........ ........ 00000248 0040D8A0 B6 *............. Q.* R003E6D80
V00484D90 00401200 818A03CA 818976F0 02115080 B6 *.  ..a...ai.0..&.* R003E6D90
V00484DA0 004AC528 80484D88 8189FB68 004AC180 B6 *.¢E...(hai...¢A.* R003E6DA0
V00484DB0 018A0B67 018A1B66 018A2B65 019983F0 B6 *.............rc0* R003E6DB0
V00484DC0 0189C368 00000055 0049F22C 00493680 B6 *.iC.......2.....* R003E6DC0
V00484DD0 0010020C 0166F680 0209343F 00000000 B6 *......6.........* R003E6DD0
AR 0015 1I40I  READY

show f2,484D88.68
AR 0015   DATA  FOUND AT 00484D88
V00484D80 ........ ........ 00000248 0040D8A0 B6 *............. Q.* R003E6D80
V00484D90 00401200 818A03CA 818976F0 02115080 B6 *.  ..a...ai.0..&.* R003E6D90
V00484DA0 004AC528 80484D88 8189FB68 004AC180 B6 *.¢E...(hai...¢A.* R003E6DA0
V00484DB0 018A0B67 018A1B66 018A2B65 019983F0 B6 *.............rc0* R003E6DB0
V00484DC0 0189C368 00000055 0049F22C 00493680 B6 *.iC.......2.....* R003E6DC0
V00484DD0 0010020C 0166F680 0209469F 00000000 B6 *......6.........* R003E6DD0
```

Figure 11. Current Dispatch Time in the CSA

Status Of QR task

Assuming that the CICS TS partition is dispatching properly, let's move on to the user transactions. Since CICS TS transactions run under the Quasi-Reentrant (QR) task, we first need to establish the status of the QR subtask. As shown in Figure 12, from the previous STATUS F2 output, display the QR task save area. Then display the location where the PSW address is pointing.



```
AR 0015 S43-F2 DFHEVID1 82 WAITING FOR I/O, OR ECB POSTING
AR 0015    SCB=0004938C PCB=00049D80 TCB=002D007C TIB=002D0000 SAV=00283500

show 00283500.50
V00283500 C4C6C8C5 E5C9C4F1 07BD1000 82144D74 06 *DFHEVID1....b.(.* R00AD8500
V00283510 02053268 0204E040 00000001 02053268 06 *......\ ........* R00AD8510
V00283520 02108AC0 0000000A 4B9BA77E 00000001 06 *...{......x=....* R00AD8520
V00283530 FDFB2000 02053268 82144588 02053000 06 *........b..h....* R00AD8530
V00283540 02145587 00000000 804053B0 0210E000 06 *...g.....  ....\.* R00AD8540

show f2,2144588.60
V02144580 ........ ........ 00486ED4 D6C4C8C5 B6 *..........>MODHE* R00EEA580
V02144590 C1C403D7 00F4F1F0 C4C6C8C4 E2C4E2F3 B6 *AD.P.410DFHDSDS3* R00EEA590
V021445A0 F0F261F2 F261F9F9 7CF1F24B F5F20005 B6 *02/22/99@12.52..* R00EEA5A0
V021445B0 82144FC8 E5E2C5F4 F1F04040 1E280218 B6 *b.|HVSE410  ....* R00EEA5B0
V021445C0 00000002 00001E38 00000360 00000000 B6 *...........-....* R00EEA5C0
V021445D0 183F4150 3FFF1840                   B6 *...&...         * R00EEA5D0
    •
    •
    •
V02144D60 58104108 41000001 41101000 13110700 B6 *................* R00EEAD60
V02144D70 07000A84 07000700 0A01B205 803098EF B6 *...d.........q.* R00EEAD70
V02144D80 80301824 1E269801 22A81FE0 1FF147B0 B6 *......q..y.\.1..* R00EEAD80
```

Figure 12. Locate Code Currently Executing in QR Subtask

From the status display, we extract the save area pointer (SAV=nnnnnnnn). The format of a system save area (main or subtask) is as follows:

Table 3. Format of a System Save Area

(displ)	(length)	(description)
0	8	Task name In the case of a subtask, this is the phase name called to start the subtask.
8	8	PSW. The second word contains the address of the last instruction executed for this task.
10	40	Registers 9-8.

Normally, if there are no active transactions, the QR subtask points into module `DFHDSDS3`. Register 3 is the initial base register for this module and will be pointing at the eyecatcher.

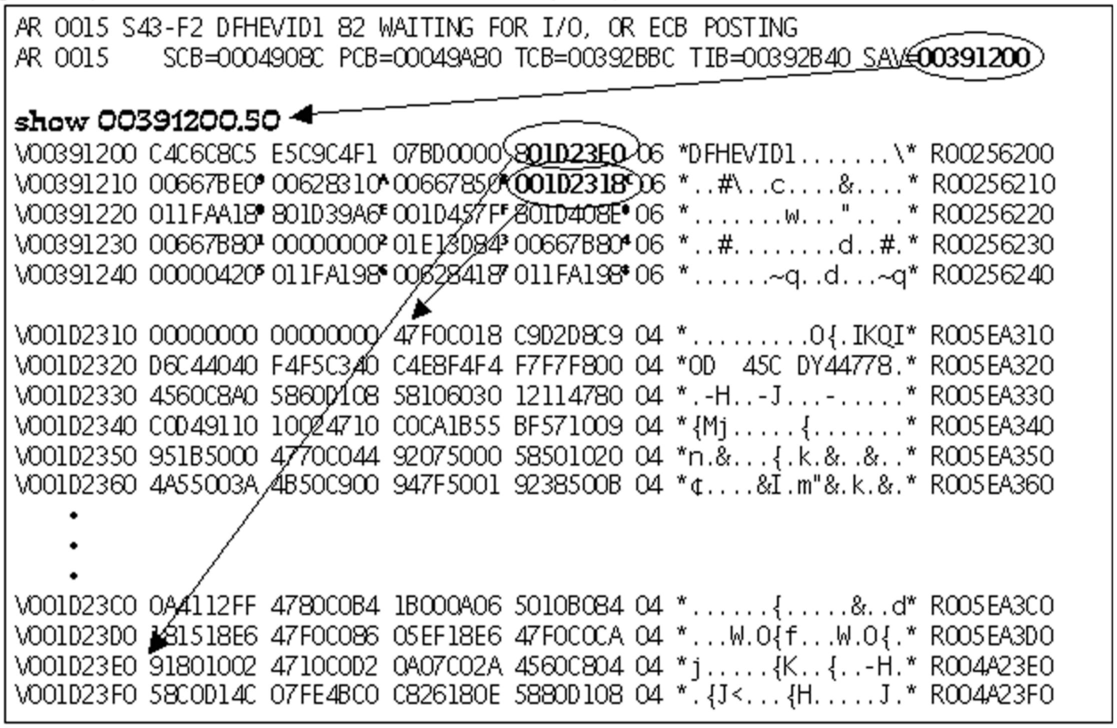
In Figure 12 on page 39, the PSW contains the address `2144D74`, which points to `0A84 07000700 0A01`. (Ignore the leading `8`. This indicates we are running in 31-bit addressing mode, but is not part of the address per se.) `0A84 07000700 0A01` is the expansion of the **SIMSVC** (CICS TS **SIM**ulation **S**uper**V**isor **C**all), instructing the VSE/ESA supervisor to execute the OS/390 SVC 01 (Wait on ECB).

Sometimes, however, the save area PSW for the QR task will be pointing somewhere else. This may cause transaction waits, since the CICS TS dispatcher domain does not receive control to dispatch another CICS transaction.

In Figure 13 on page 41, the task save area for the QR task contains a PSW pointing at X'001D23E0'. This is part of a VSE WAIT macro expansion `91801002 4710C0D2 0A07`.

Under no circumstances should a CICS transaction issue a wait macro, as this prevents other transactions from running. The branch instruction (`4710C0D2`) shows that Register 12 is the base register. Extracting this register from the task save area shows the beginning of the current module at X'001D2318', which is a VSAM module (IKQIOD). The current Task Control Area (TCA) pointer in the CSA + X'4C' tells us which transaction violated the rules. (See Figure 10 on page 38 for assistance in locating the CSA.)

ATCA is assigned to each active task and contains information relating to task execution. TCA + X'1C4' contains the initial transaction ID which was used to initiate this transaction. See Figure 14 on page 41.

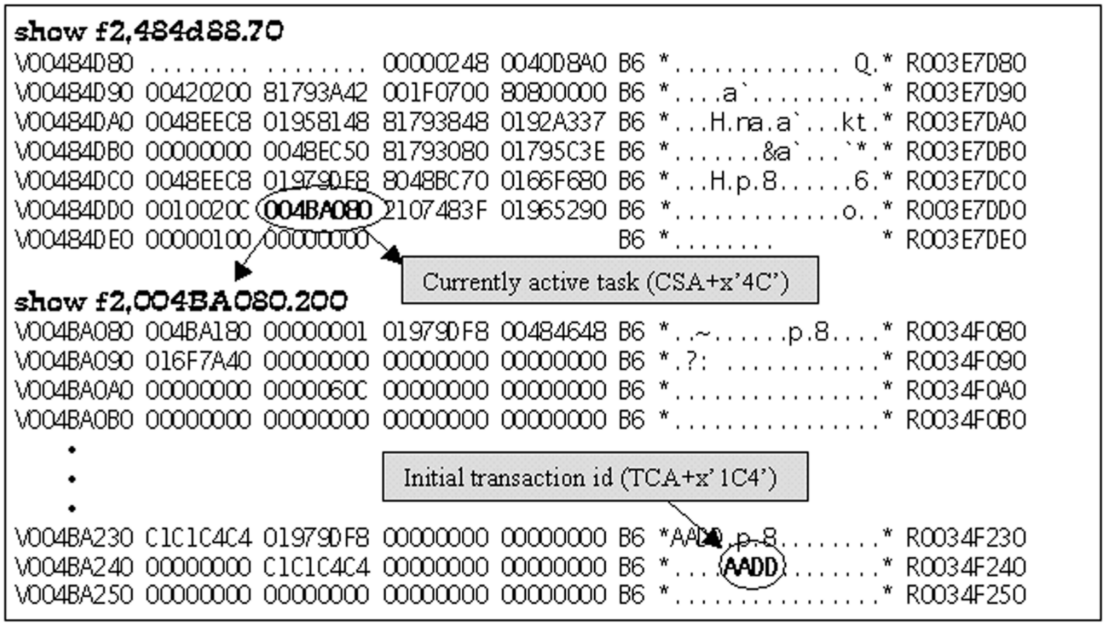


```
AR 0015 S43-F2 DFHEVID1 82 WAITING FOR I/O, OR ECB POSTING
AR 0015    SCB=0004908C PCB=00049A80 TCB=00392BBC TIB=00392B40 SAVE=00391200

show 00391200.50
V00391200 C4C6C8C5 E5C9C4F1 07BD0000 801D23E0 06 *DFHEVID1.......\* R00256200
V00391210 00667BE0 00628310 00667850 001D2318 06 *..#\..c....&....* R00256210
V00391220 011FAA18 801D39A6 001D457F 801D408E 06 *.......w..."..  .* R00256220
V00391230 00667B80 00000000 01E13D84 00667B80 06 *..#........d..#.* R00256230
V00391240 00000420 011FA198 00628418 011FA198 06 *......~q..d...~q* R00256240

V001D2310 00000000 00000000 47F0C018 C9D2D8C9 04 *.........0{.IKQI* R005EA310
V001D2320 D6C44040 F4F5C340 C4E8F4F4 F7F7F800 04 *OD  45C DY44778.* R005EA320
V001D2330 4560C8A0 5860D108 58106030 12114780 04 *.-H..-J...-.....* R005EA330
V001D2340 C0D49110 10024710 C0CA1B55 BF571009 04 *{Mj.....{.......* R005EA340
V001D2350 951B5000 4770C044 92075000 58501020 04 *n.&...{.k.&..&..* R005EA350
V001D2360 4A55003A 4B50C900 947F5001 9238500B 04 *¢....&I.m"&.k.&.* R005EA360
    •
    •
    •
V001D23C0 0A4112FF 4780C0B4 1B000A06 5010B084 04 *......{.....&..d* R005EA3C0
V001D23D0 181518E6 47F0C086 05EF18E6 47F0C0CA 04 *...W.0{f...W.0{.* R005EA3D0
V001D23E0 91801002 4710C0D2 0A07C02A 4560C804 04 *j.....{K..{..-H.* R004A23E0
V001D23F0 58C0D14C 07FE4BC0 C826180E 5880D108 04 *.{J<...{H.....J.* R004A23F0
```

Figure 13. QR Task Lock-Out Situation



```
show f2,484d88.70
V00484D80 ........ ........ 00000248 0040D8A0 B6 *............. Q.* R003E7D80
V00484D90 00420200 81793A42 001F0700 80800000 B6 *....a`..........* R003E7D90
V00484DA0 0048EEC8 01958148 81793848 0192A337 B6 *...H.na.a`...kt.* R003E7DA0
V00484DB0 00000000 0048EC50 81793080 01795C3E B6 *.......&a`...`*.* R003E7DB0
V00484DC0 0048EEC8 019790F8 8048BC70 0166F680 B6 *...H.p.8......6.* R003E7DC0
V00484DD0 0010020C 004BA080 2107483F 01965290 B6 *.............o..* R003E7DD0
V00484DE0 00000100 00000000                   B6 *........        * R003E7DE0

show f2,004BA080.200
V004BA080 004BA180 00000001 019790F8 00484648 B6 *..~......p.8....* R0034F080
V004BA090 016F7A40 00000000 00000000 00000000 B6 *.?: ............* R0034F090
V004BA0A0 00000000 0000060C 00000000 00000000 B6 *................* R0034F0A0
V004BA0B0 00000000 00000000 00000000 00000000 B6 *................* R0034F0B0
    •
    •
    •
V004BA230 C1C1C4C4 019790F8 00000000 00000000 B6 *AADD.p.8........* R0034F230
V004BA240 00000000 C1C1C4C4 00000000 00000000 B6 *....AADD........* R0034F240
V004BA250 00000000 00000000 00000000 00000000 B6 *................* R0034F250
```

Currently active task (CSA+x'4C')
Initial transaction id (TCA+x' 1C4')

Figure 14. Establishing Transaction ID for Currently Active Task

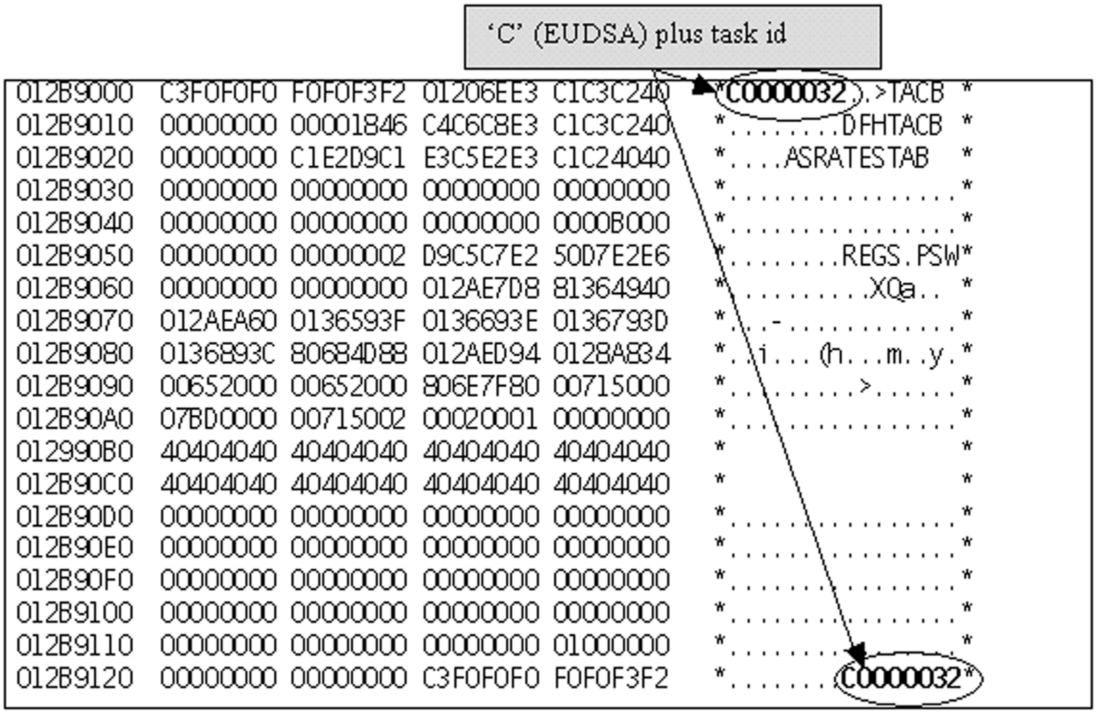
Storage Violation

A storage violation occurs when data is written outside of the storage assigned to the current transaction. Often this overlay over-writes storage belonging to another transaction or CICS system tasks.

To reduce the impact of storage violations, CICS TS isolates user storage in a different partition GETVIS location. Thus storage violations by a user transaction are more likely to impact another user transaction than CICS system areas. In addition, the user can request subsystem storage protection, which isolates CICS storage in a different, inaccessible, storage protect key (`STGPROT=YES` in SIT parameters).

In CICS TS, user storage areas are no longer chained together via Storage Accounting Areas (SAAs). In their place, task data and TIOAs in UDSA (24-bit) and EUDSA (31-bit) subpools are framed with control areas (or "crumple zones") which are checked by CICS during FREEMAIN and end-of-task – `CHKSTRM={CURRENT|NONE}` and `CHKSTSK={ALL|CURRENT|NONE}`. Figure 15 is an example of user storage framed by crumple zones.

Task data is chained at a higher level, so overlaid data no longer destroys the storage chain. If authorized to do so, CICS will attempt to repair storage violations and continue with the task. Otherwise, the task is abended, and its storage is placed out-of-service (not FREEMAINed). This facilitates problem determination. (`STGRCVY={NO|YES}`). In either case, CICS attempts to continue after a storage violation.



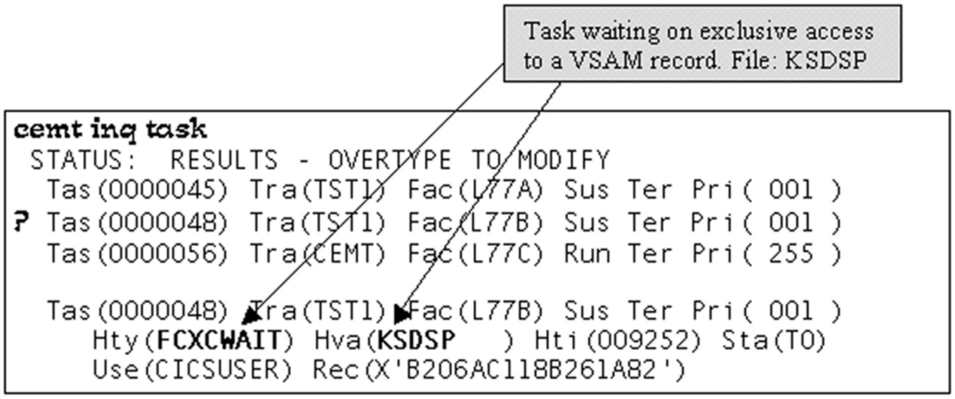
'C' (EUDSA) plus task id

```
012B9000  C3F0F0F0 F0F0F3F2 01206EE3 C1C3C240   *C0000032..>TACB *
012B9010  00000000 00001846 C4C6C8E3 C1C3C240   *........DFHTACB *
012B9020  00000000 C1E2D9C1 E3C5E2E3 C1C24040   *....ASRATESTAB  *
012B9030  00000000 00000000 00000000 00000000   *................*
012B9040  00000000 00000000 00000000 0000B000   *................*
012B9050  00000000 00000002 D9C5C7E2 50D7E2E6   *........REGS.PSW*
012B9060  00000000 00000000 012AE7D8 81364940   *..........XQa.. *
012B9070  012AEA60 0136593F 0136693E 0136793D   *...-............*
012B9080  0136893C 80684D88 012AED94 0128A834   *..i...(h...m..y.*
012B9090  00652000 00652000 806E7F80 00715000   *.........>......*
012B90A0  07BD0000 00715002 00020001 00000000   *................*
012990B0  40404040 40404040 40404040 40404040   *                *
012B90C0  40404040 40404040 40404040 40404040   *                *
012B90D0  00000000 00000000 00000000 00000000   *................*
012B90E0  00000000 00000000 00000000 00000000   *................*
012B90F0  00000000 00000000 00000000 00000000   *................*
012B9100  00000000 00000000 00000000 00000000   *................*
012B9110  00000000 00000000 00000000 01000000   *................*
012B9120  00000000 00000000 C3F0F0F0 F0F0F3F2   *........C0000032*
```

Figure 15. Crumple Zones

File Control Waits

CICS TS has enhanced the output from the CEMT to include a detail option. If a transaction is in a wait state, this detailed output includes a token indicating what resource the transaction is waiting on. In Figure 16, the transaction is waiting on exclusive control access to a VSE/VSAM file.



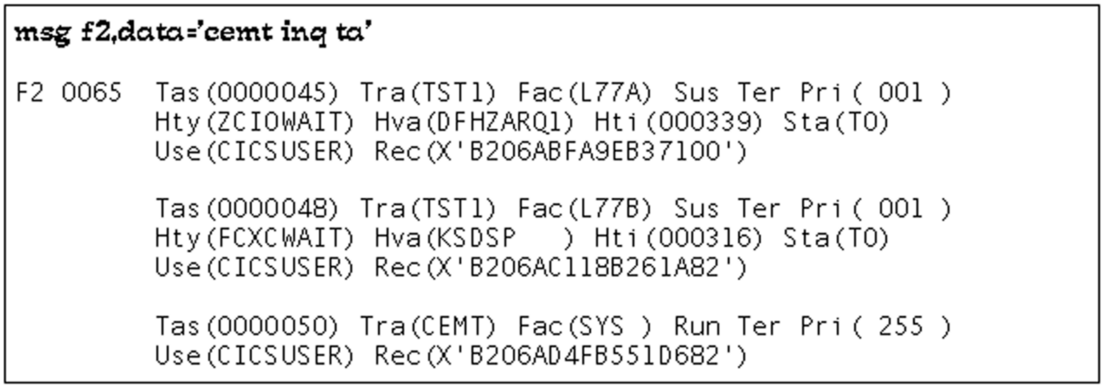
Task waiting on exclusive access to a VSAM record. File: KSDSP

```
cemt inq task
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Tas(0000045) Tra(TST1) Fac(L77A) Sus Ter Pri( 001 )
? Tas(0000048) Tra(TST1) Fac(L77B) Sus Ter Pri( 001 )
  Tas(0000056) Tra(CEMT) Fac(L77C) Run Ter Pri( 255 )

  Tas(0000048) Tra(TST1) Fac(L77B) Sus Ter Pri( 001 )
     Hty(FCXCWAIT) Hva(KSDSP    ) Hti(009252) Sta(TO)
     Use(CICSUSER) Rec(X'B206AC118B261A82')
```

Figure 16. CEMT Inquire Task Output

This level of detail is also available via the new `DATA=` option on the `MSG F2` attention routine command. See Figure 17.



```
msg f2,data='cemt inq ta'

F2 0065  Tas(0000045) Tra(TST1) Fac(L77A) Sus Ter Pri( 001 )
         Hty(ZCIOWAIT) Hva(DFHZARQ1) Hti(000339) Sta(TO)
         Use(CICSUSER) Rec(X'B206ABFA9EB37100')

         Tas(0000048) Tra(TST1) Fac(L77B) Sus Ter Pri( 001 )
         Hty(FCXCWAIT) Hva(KSDSP    ) Hti(000316) Sta(TO)
         Use(CICSUSER) Rec(X'B206AC118B261A82')

         Tas(0000050) Tra(CEMT) Fac(SYS ) Run Ter Pri( 255 )
         Use(CICSUSER) Rec(X'B206AD4FB551D682')
```

Figure 17. Attention Routine Command Access to CICS TS Transaction Status

> *"If your task is waiting on resource type FCXCWAIT, it means that it cannot get exclusive control of a VSAM control interval at the present time. Another task already has shared or exclusive control of the control interval, so your task is suspended pending the release of that control interval. An exclusive control wait on resource type FCXCWAIT occurs within CICS, unlike the similar wait on FCIOWAIT, which occurs within VSAM."*
>
> *CICS TS Problem Determination Guide*, GC33-1663-00

However, the CICS TS only tells us the transaction is waiting on a file. It does tell us which file we are working on, but does not disclose who the current owner of the locked resource is. To access this information, we need another product enhancement added to the VSE/VSAM and CICS TS application interface.

When VSE/VSAM returns a file access request with exclusive control conflict (RPL Feedback Code X'14'), it also returns detailed information about which string currently has the resource under control. This new message area is chained from the RPL, which is part of the CICS TS VSAM Work Area (VSWA), which is chained off the CICS TS File Control Table Entry (FCTTE). Figure 18 shows the CICS TS control block chaining required to access the VSE/VSAM RPL message area.

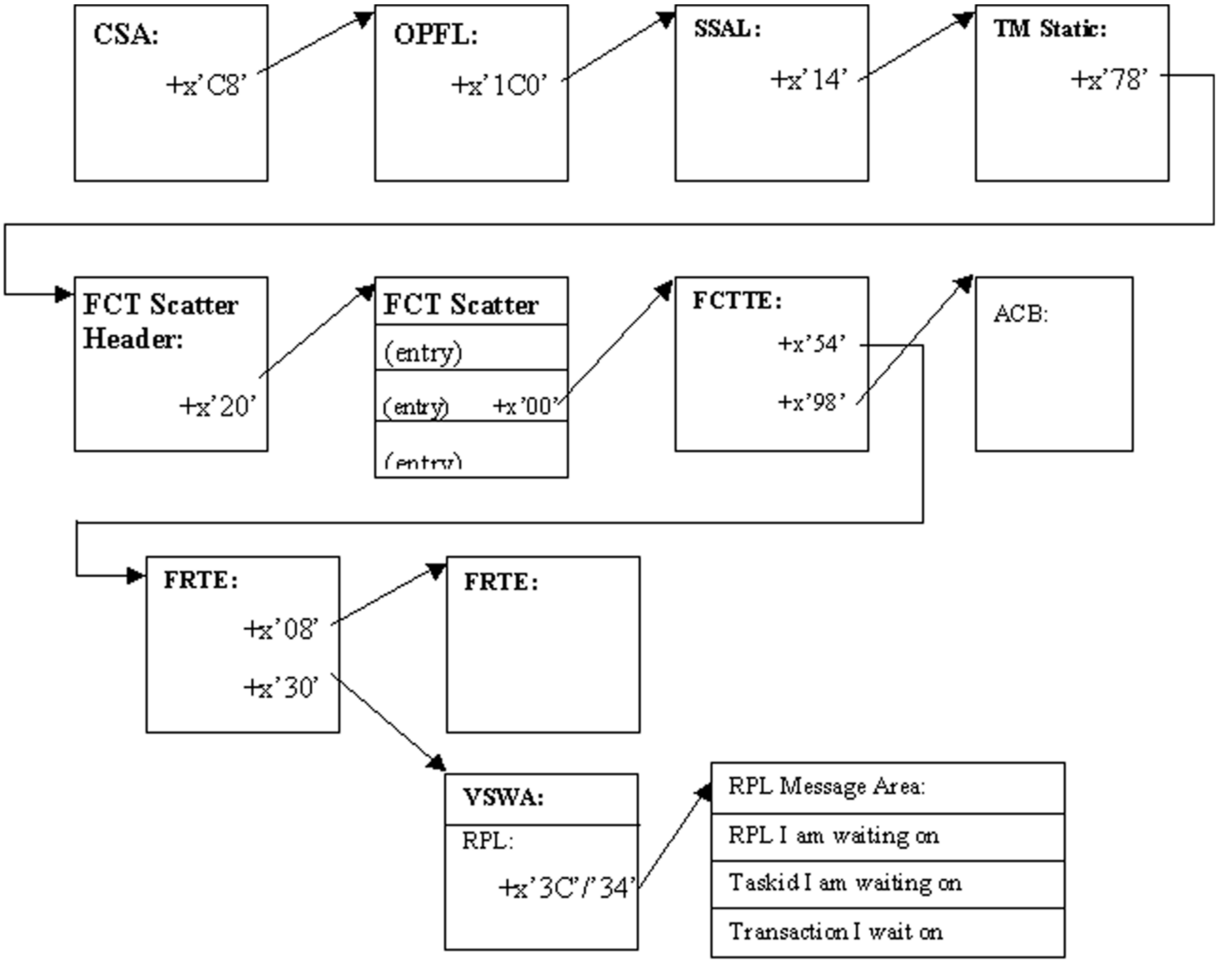


Figure 18. Locating the VSE/VSAM RPL Message Area

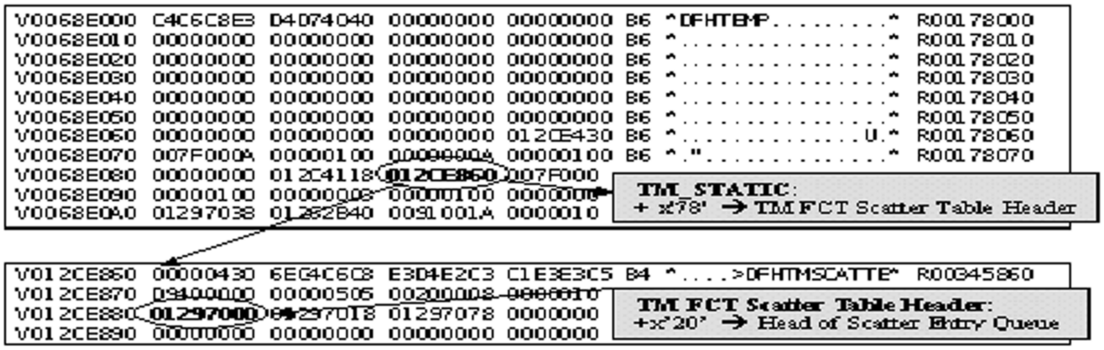
Locating The File Control Table (FCT)

A major change between CICS/VSE and CICS TS has to do with the way tables are managed in CICS TS storage. Originally in CICS/VSE, tables (FCT = File Control, DCT = Destination Control, PPT = Processing Program, etc.) were created by assembling a phase and loaded into CICS/VSE storage during startup. It was impossible to dynamically add a new entry. Adding to the table required a lengthy assembly and re-cycling CICS/VSE. Later, dynamic resource allocation was added via DFHCSDUP and CEDA, and tables entries were allocated and deallocated dynamically within CICS/VSE storage.

CICS TS has added the remainder of the resource tables (including File Control) to dynamic management. CICS TS manages its tables via a Table Manager and scatter tables, which are pointers to pieces of tables, allocated and deallocated dynamically as needed. It is no longer possible to go to the CSA and find a pointer to the FCT lying contiguous in storage.

To find the FCT, we must first locate the Table Manager scatter table for FCT. Using the information from Figure 18 on page 44, follow the CSA + X'C8' to the Optional Feature List (OPFL), then a pointer at OPFL + X'1C0' to the Static Storage Address List. This table also existed in CICS/VSE, but has been substantially changed for CICS TS. SSAL + X'14' points at Table Manager Static Storage (TMSS).

Figure 19 shows an example of the chaining from the TMSS to the Table Manager FCT scatter table. Figure 20 lists all of the tables supported by Table Manager and the displacement of the corresponding pointers. The FCT scatter table pointer is at TMSS + X'78'.



```
V0068E000 C4C6C8E3 D4D74040 00000000 00000000 B6 *DFHTEMP.........* R00178000
V0068E010 00000000 00000000 00000000 00000000 B6 *................* R00178010
V0068E020 00000000 00000000 00000000 00000000 B6 *................* R00178020
V0068E030 00000000 00000000 00000000 00000000 B6 *................* R00178030
V0068E040 00000000 00000000 00000000 00000000 B6 *................* R00178040
V0068E050 00000000 00000000 00000000 00000000 B6 *................* R00178050
V0068E060 00000000 00000000 00000000 012CE430 B6 *..............U.* R00178060
V0068E070 007F000A 00000100 0000000A 00000100 B6 *."..............* R00178070
V0068E080 00000000 012C4118 012CE860 007F000
V0068E090 00000100 00000008 00000100 0000000
V0068E0A0 01297038 01282B40 0091001A 0000010
```

TM_STATIC:
+ x'78' → TM FCT Scatter Table Header

```
V012CE860 00000430 6EC4C6C8 E3D4E2C3 C1E3E3C5 B4 *....>DFHTMSCATTE* R00345860
V012CE870 09400000 00000506 00200008 0000010
V012CE880 01297000 01297018 01297078 0000000
V012CE890 00000000 00000000 00000000 0000000
```

TM FCT Scatter Table Header:
+x'20' → Head of Scatter Entry Queue

Figure 19. Example of Chaining from the Table Manager Static Storage to TM FCT Scatter Table

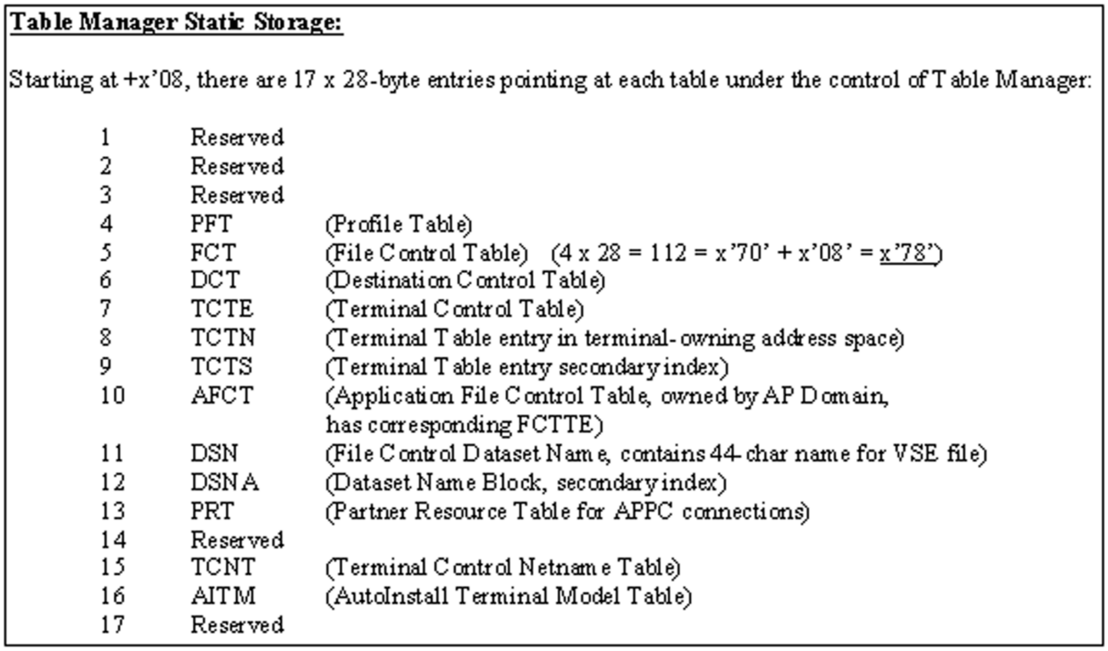


Table Manager Static Storage:

Starting at +x'08, there are 17 x 28-byte entries pointing at each table under the control of Table Manager:

1	Reserved	
2	Reserved	
3	Reserved	
4	PFT	(Profile Table)
5	FCT	(File Control Table) (4 x 28 = 112 = x'70' + x'08' = x'78')
6	DCT	(Destination Control Table)
7	TCTE	(Terminal Control Table)
8	TCTN	(Terminal Table entry in terminal-owning address space)
9	TCTS	(Terminal Table entry secondary index)
10	AFCT	(Application File Control Table, owned by AP Domain, has corresponding FCTTE)
11	DSN	(File Control Dataset Name, contains 44-char name for VSE file)
12	DSNA	(Dataset Name Block, secondary index)
13	PRT	(Partner Resource Table for APPC connections)
14	Reserved	
15	TCNT	(Terminal Control Netname Table)
16	AITM	(AutoInstall Terminal Model Table)
17	Reserved	

Figure 20. List of Tables Supported by the Table Manager and Displacement of Corresponding Pointers

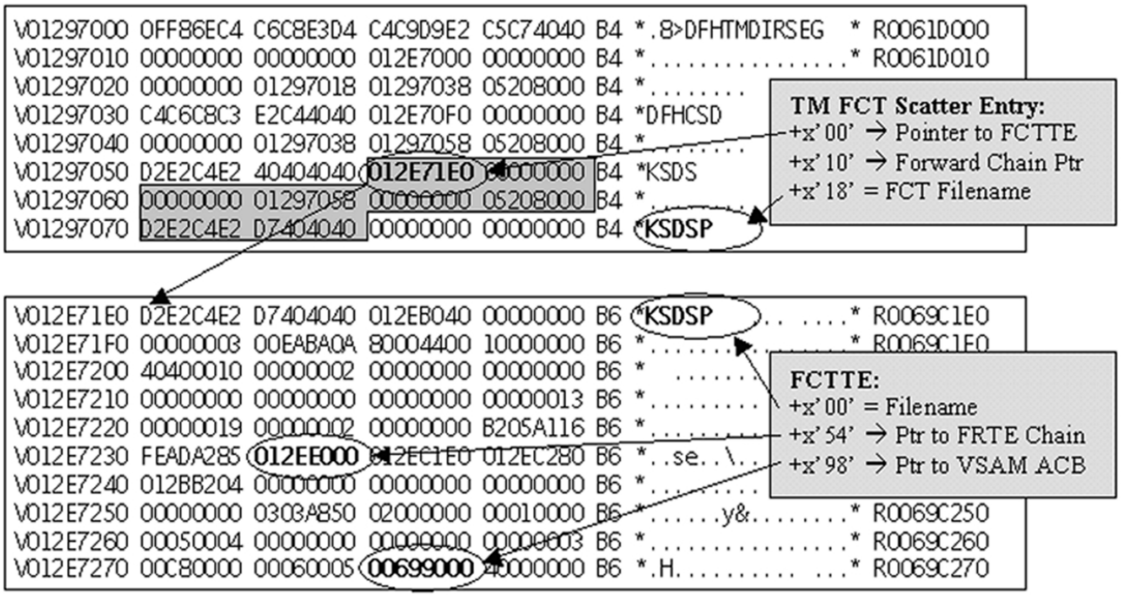
Each scatter table consists of individual entries for each resource in the table. In the case of the FCT scatter table, the entries are X'20' long, with the entry name at the end of the entry. See Figure 21 for an example:

X'00' Pointer to corresponding FCTTE

X'10' Chain pointer to next scatter table entry. The entries are chained together. If the list is very long, we must locate the forward chaining pointer to the next set of table entries.

X'18' Name of the resource (filename for FCTTE)

The File Control Table Entry (FCTTE) has been substantially changed between CICS/VSE and CICS TS. The most noticeable change is the movement of the ACB or DTF from the FCTTE + X'90' to File Control stack storage. In addition, active file control requests (get-for-update, browse) were tracked under CICS/VSE via Deferred Work Elements (DWEs). Under CICS TS, they are chained in new control blocks called File Request Thread Entries (FRTEs).



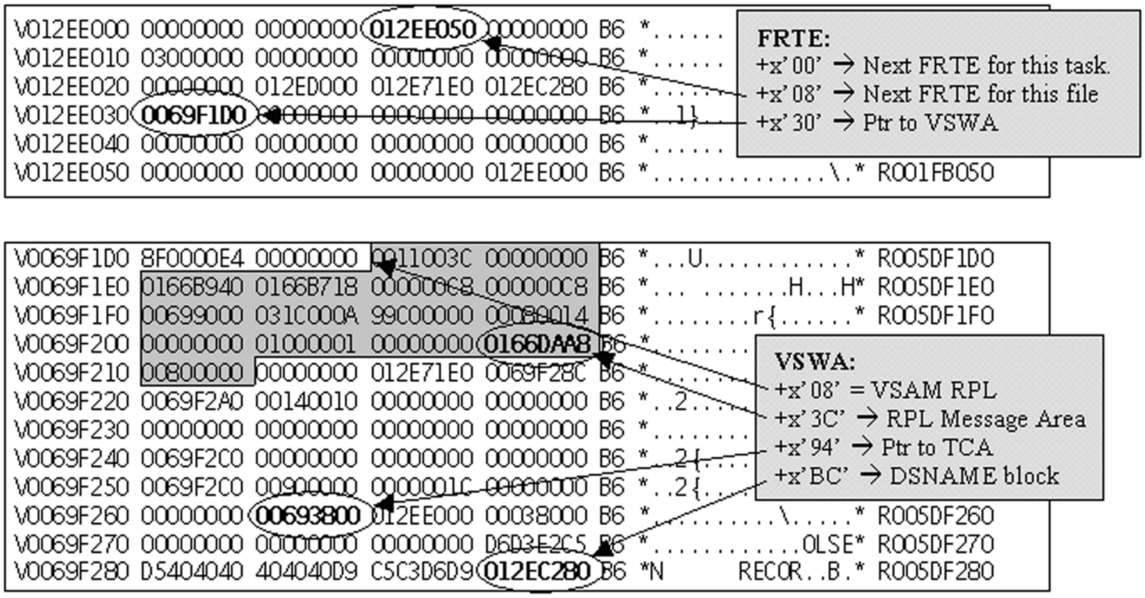
```
V01297000 0FF86EC4 C6C8E3D4 C4C9D9E2 C5C74040 B4 *.8>DFHTMDIRSEG   * R0061D000
V01297010 00000000 00000000 012E7000 00000000 B4 *................* R0061D010
V01297020 00000000 01297018 01297038 05208000 B4 *........
V01297030 C4C6C8C3 E2C44040 012E70F0 00000000 B4 *DFHCSD
V01297040 00000000 01297038 01297058 05208000 B4 *........
V01297050 D2E2C4E2 40404040 012E71E0 00000000 B4 *KSDS
V01297060 00000000 01297058 00000000 05208000 B4 *........
V01297070 D2E2C4E2 D7404040 00000000 00000000 B4 *KSDSP

V012E71E0 D2E2C4E2 D7404040 012EB040 00000000 B6 *KSDSP   ..  ....* R0069C1E0
V012E71F0 00000003 00EABA0A 80004400 10000000 B6 *...............* R0069C1F0
V012E7200 40400010 00000002 00000000 00000000 B6 *   .......
V012E7210 00000000 00000000 00000000 00000013 B6 *.........
V012E7220 00000019 00000002 00000000 B205A116 B6 *.........
V012E7230 FEADA285 012EE000 012EC1E0 012EC280 B6 *..se..\..
V012E7240 012BB204 00000000 00000000 00000000 B6 *.........
V012E7250 00000000 0303A850 02000000 00010000 B6 *......y&........* R0069C250
V012E7260 00050004 00000000 00000000 00000003 B6 *................* R0069C260
V012E7270 00C80000 00060005 00699000 40000000 B6 *.H.........  ...* R0069C270
```

TM FCT Scatter Entry:
+x' 00' → Pointer to FCTTE
+x' 10' → Forward Chain Ptr
+x' 18' = FCT Filename
FCTTE:
+x' 00' = Filename
+x' 54' → Ptr to FRTE Chain
+x' 98' → Ptr to VSAM ACB

Figure 21. FCT Scatter Table to FCTTE Chaining

A quick and convenient way to ascertain what requests are outstanding for a specific file is to examine the FRTE chain. See Figure 22 on page 47 for an example.



```
V012EE000 00000000 00000000 012EE050 00000000 B6 *......
V012EE010 03000000 00000000 00000000 00000000 B6 *......
V012EE020 00000000 012ED000 012E71E0 012EC280 B6 *......
V012EE030 0069F1D0 00000000 00000000 00000000 B6 *...1}...
V012EE040 00000000 00000000 00000000 00000000 B6 *......
V012EE050 00000000 00000000 00000000 012EE000 B6 *..............\.* R001FB050
```

FRTE:
+x'00' → Next FRTE for this task.
+x'08' → Next FRTE for this file
+x'30' → Ptr to VSWA

```
V0069F1D0 8F0000E4 00000000 0011003C 00000000 B6 *...U...........* R005DF1D0
V0069F1E0 0166B940 0166B718 000000C8 000000C8 B6 *... .......H...H* R005DF1E0
V0069F1F0 00699000 031C000A 99C00000 00B80014 B6 *........r{......* R005DF1F0
V0069F200 00000000 01000001 00000000 0166DAA8 B6 *.......
V0069F210 00800000 00000000 012E71E0 0069F28C B6 *.......
V0069F220 0069F2A0 00140010 00000000 00000000 B6 *..2....
V0069F230 00000000 00000000 00000000 00000000 B6 *.......
V0069F240 0069F2C0 00000000 00000000 00000000 B6 *..2{....
V0069F250 0069F2C0 00900000 0000001C 00000000 B6 *..2{...
V0069F260 00000000 00693800 012EE000 00038000 B6 *..........\.....* R005DF260
V0069F270 00000000 00000000 00000000 D6D3E2C5 B6 *............OLSE* R005DF270
V0069F280 D5404040 404040D9 C5C3D6D9 012EC280 B6 *N     RECOR..B.* R005DF280
```

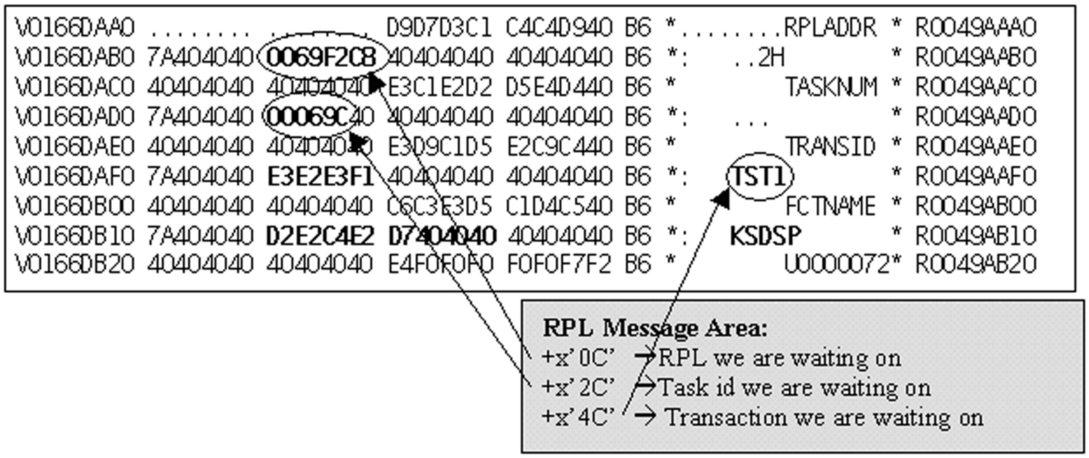
Figure 22. FRTE and VSWA Chaining

An FRTE exists for each active file request. They are chained by task (+ X'00') and by resource (+ X'08'). If the request is for a VSE/VSAM file, FRTE + X'30' points to a VSAM Work Area (VSWA). The main component of the VSWA is the VSE/VSAM Request Parameter List (RPL), which contains a pointer to the ACB (+ X'18'), the current string number (+ X'1C'), last file request (+ X'1D'), and the feedback code for the last file request (+ X'25'). For additional detail on layout and content of RPL fields, see *VSE/VSAM Logic*, LY33-9130.

Also within the VSWA is a pointer to the current task's TCA (+ X'94') and a pointer to the CICS TS DSNAME block (+ X'BC'). The DSNAME block is of particular interest, since it contains the full 44-character name of the file.

Most important for our search here, the VSWA + X'3C' contains a pointer to the RPL Message Area. This control block was added for CICS TS to allow more precise re-starting of VSE/VSAM requests which are waiting on a resource.

When VSE/VSAM returns a file management request because of resource contention, it also returns an RPL message area containing detailed information as to which CICS transaction currently owns this resource. Figure 23 on page 48 is an example of an RPL message area. Each field is preceded by readable text describing the contents.



```
V0166DAA0 ........ ........ D9D7D3C1 C4C4D940 B6 *........RPLADDR * R0049AAA0
V0166DAB0 7A404040 0069F2C8 40404040 40404040 B6 *:    ..2H        * R0049AAB0
V0166DAC0 40404040 40404040 E3C1E2D2 D5E4D440 B6 *         TASKNUM * R0049AAC0
V0166DAD0 7A404040 00069C40 40404040 40404040 B6 *:    ...         * R0049AAD0
V0166DAE0 40404040 40404040 E3D9C1D5 E2C9C440 B6 *         TRANSID * R0049AAE0
V0166DAF0 7A404040 E3E2E3F1 40404040 40404040 B6 *:   TST1         * R0049AAF0
V0166DB00 40404040 40404040 C6C3E3D5 C1D4C540 B6 *         FCTNAME * R0049AB00
V0166DB10 7A404040 D2E2C4E2 D7404040 40404040 B6 *:   KSDSP        * R0049AB10
V0166DB20 40404040 40404040 E4F0F0F0 F0F0F7F2 B6 *         U0000072* R0049AB20
```

RPL Message Area:
+x'0C' → RPL we are waiting on
+x'2C' → Task id we are waiting on
+x'4C' → Transaction we are waiting on

Figure 23. RPL Message Area

Summary

The new CICS TS CEMT INQUIRE TASK command (Figure 16 on page 43) tells us that a task may be waiting on a VSE/VSAM resource, but does not stipulate who the current owner of the resource is.

In order to perform specific recovery procedures without re-cycling the entire CICS partition, it is necessary to locate the RPL Message Area, which tells us which transaction is holding our request. In order to do this, we must be familiar with the CICS TS Table Management tables, locate the File Control Table (FCT) scatter table, and the specific File Control Table Entry (FCTTE) on which we are waiting. From there, we follow the File Thread Request queue until we locate the FRTE which applies to this task. The FRTE points at the VSAM Work Area (VSWA), which contains the RPL, and a pointer to the RPL Message Area.

Most of the demonstrations in this article can be performed via the console using the `LOCATE` and `SHOW` commands. Along with the enhancements to CICS TS, product usability has been significantly improved. To a considerable extent, problem determination can be performed online without the necessity of taking a dump and re-cycling the online partitions.

Other Technical Articles

New MQSeries For VSE/ESA Redbook

Author

Bret Dixon
MQSeries for VSE/ESA Development
IBM GSA APC Perth, Australia
bret@au1.ibm.com

Overview

The new MQSeries for VSE/ESA redbook – SG24-5647 – does not supersede a previous version. *It is the first redbook specifically for MQSeries for VSE/ESA.*

The redbook was written by myself and Gerard Martinelli of IBM France, who is a systems engineer, specializing in VSE/ESA and VM/ESA. Both Gerard and I have experience with development of the MQSeries product.

The redbook takes a "how to" approach in regard to most aspects of the product, including installation, migration, distributed queueing, performance, security, problem determination and other topics. An effort was made **not** to duplicate material already available in other manuals, and specifically the *System Management Guide*. This article reproduces some significant sections directly from the redbook. All bulleted points in this article are covered in greater detail in the red piece which is available on the Web. (See “Additional Information” on page 56.)

Installation And Migration

The first chapter deals with installation and migration. We cover installation for both VSE/ESA 2.3 and 2.4. Migration is specifically for MQSeries 1.4 to 2.1. Once again, we take a "how to" approach and step through each process.

Many problems experienced by MQSeries for VSE/ESA customers are related to installation or incorrect installation issues. Consequently, the redbook provides sample JCL and guidelines for VSAM, CICS and LE definitions to help you get installation right.

Since MQSeries for VSE/ESA is essentially a CICS subsystem, the installation of MQSeries involves modifications to your CICS environment including:

- Language Environment® (LE) definitions
- Systems Network Architecture (SNA) definitions
- Journal Control Table (JCT) definitions
- File Control Table (FCT) definitions
- Destination Control Table (DCT) definitions
- MQSeries definitions
- CICS system startup JCL

Regarding these, the redbook explains that Language Environment for VSE (LE/VSE) provides common services and language-specific routines in a single run-time environment for applications written in LE/VSE-conforming versions of C, COBOL, and PL/I high level languages, and the High Level Assembler language.

For flexibility reasons, it is possible to change run time characteristics by using options tables. Because batch and CICS environments are very different, run time options are also different. LE/VSE comes with IBM-supplied run-time values that you should find in sublibrary PRD2.SCEEBASE. These include:

```
CEEDOPT.A  (for batch environments)
CEECOPT.A  (for CICS)
CEEUOPT.A  (for user specific applications)
```

Sample jobs to generate these options tables are also provided in:

```
CEEWDOPT.Z (for batch environments)
CEEWCOPT.Z (for CICS)
CEEWUOPT.Z (for user specific applications)
```

We recommend you do not change the default values of CEEWCOPT provided by LE/VSE unless requested by IBM maintenance people to fix APARs. However if for any reasons you have to do so, try to avoid changing the following keyword values:

```
ABTERMENC=((ABEND),OVR)
ALL31=((ON),OVR)
ANYHEAP=((4K,4K,ANYWHERE,FREE),OVR)
BELOWHEAP=((4K,4K,FREE),OVR)
HEAP=((4K,4K,ANYWHERE,KEEP,4K,4K),OVR)
LIBSTACK=((4K,4K,FREE),OVR)
STACK=((4K,4K,ANYWHERE,KEEP),OVR)
STORAGE=((NONE,NONE,NONE,0K),OVR)
TERMTHDACT=((MSG),OVR)
```

Defining LE/VSE programs in the CICS System Definition (CSD) file is a required installation step. MQSeries needs the phases specified in the following members defined in the CICS CSD:

```
CEECCSD.Z  (LE/VSE Base)
IGZCCSD.Z  (for COBOL)
EDCCCSD.Z  (for C)
```

You can find these members in the PRD2.SCEEBASE or PRD2.SCEECICS sublibraries. The definitions can be added by submitting JCL that executes the DFHCSDUP program.

Regarding SNA definitions, the redbook explains that since version 2.1, you have a choice to interconnect MQSeries queue managers using either TCP/IP or SNA LU 6.2 (or both). You specify which protocol to use when you define MQSeries channels. With TCP/IP, you do not need any additional definitions for CICS. For LU6.2, you must specify Sessions and Connections through the CEDA transaction.

A connection is the definition of a remote system that your CICS system communicates with using SNA protocols. When you define a CONNECTION with CEDA, you give enough information to identify the system and specify its basic attributes. You put details about sessions you will use to communicate with the system in SESSION definitions. CICS uses the CONNECTION name to identify the other system when the definition has been installed. The redbook provides examples of both types of definitions and shows distributed examples using these definitions.

The redbook also explains that message queues are logical entities. Physically they reside in KSDS VSAM clusters. A unique VSAM cluster may host several queues (not recommended for a production system). Once again, MQSeries is just a normal CICS subsystem. Therefore, all general considerations for CICS files apply also for MQSeries. That is, you have to:

1. Specify the VSAM clusters in the DFHFCT
2. Specify the number of index and data buffers you want to use
3. Decide whether or not they will be part of a LSRPOOL and specify the pool number.

A typical entry looks like the following:

```
MYQUEUE DFHFCT TYPE=DATASET,DATASET=MYQUEUE,               *
               ACCMETH=VSAM,                                *
               SERVREQ=(READ,UPDATE,ADD,BROWSE,DELETE),     *
               LOG=YES,                                     *
               RSL=PUBLIC,                                  *
               BUFND=16,BUFNI=16,STRNO=16,                  *
               RECFORM=(VARIABLE,BLOCKED),                  *
               LSRPOOL=5
```

As you can see, the number of strings is set to 16. Depending on your own environment, this value may be too large or too small. We recommend you start with 16, then check the CICS statistics at the end of a busy day to see whether there were "Wait on String" conditions and the maximum number of strings used. This information helps you tune your MQSeries message queues. Also notice that we have specified LOG=YES. This is because, in a production environment, we cannot imagine running CICS without Recovery/Restart capabilities. Indeed, a DFHJCT must be defined. For CICS TS, file definitions exist in the CSD and may be created with the CEDA transaction.

To make sure no message is lost, it is strongly recommended you use CICS journal logging. If you do not have a CICS system journal, then you should define one. You should check with your CICS system programmer before creating or changing your JCT phase.

A parameter of the DFHJCT macro is SUFFIX, which is a 2-character value. To make this table operational, specify the 2-character suffix as a SIT parameter, JCT=XX, either by recompiling your DFHSIT, or with override parameters in your CICS startup JCL.

When specifying LOG=YES in DFHFCT entries, you are asking for an automatic logging to the system journal. It is used in case of emergency restart, to restore file records of in-flight transactions at abend time to the value they had when the transaction abended. Automatic logging should not be confused with automatic journaling exploited by user applications. If you do not have a DFHJCT entry for the system log (also called system journal) and if you specified LOG=YES for DFHFCT entries, then you receive an AFCL abend condition when MQSeries starts.

Transient Data entries need to be defined in the CICS Destination Control Table (DCT). Make sure that the book IESZDCT is included in your DFHDCT source code, because it specifies the required entries for LE/VSE (CESO and CESE). MQSeries for VSE/ESA has one transient data queue – MQER, an intra-partition destination – for logging MQSeries error messages.

With regard to the CICS startup JCL, the redbook also explains that there is a trap that you must avoid in your CICS startup job if you wish to use TCP/IP communications with MQSeries channels. The TCP/IP sublibrary (normally PRD1.BASE) *must* be specified before PRD2.SCEEBASE. This is because LE/VSE is provided with a dummy phase ($EDCTCPV) needed for TCP/IP. The real phase comes with the TCP/IP product. If you do not specify the TCP/IP sublibrary first, then no TCP/IP applications work under CICS. Do not forget to verify that this phase is correctly specified in the CSD.

Distributed Queueing

In Chapter 5 of the redbook, we look at distributed queuing. More specifically, we look at the objects involved in distributed queueing, how they interact, when they are used, and how they work. In Chapter 6, we cover specific examples of distributed queuing across similar and dissimilar architectures.

Our explanation of distributed queueing covers:

- Remote queues
- Transmission queues
- Channels
- Message Channel Agents
- TCP/IP Listener program
- TCP/IP Server program
- Communication protocols
- Clients

We also provide an example of an application (APP1) putting messages to a remote queue, intended to be read by a second application (APP2) running on a remote platform. The remote queue definition is used to identify a transmission queue as well as the target queue manager and target queue. This information is appended to messages written to the transmission queue. The transmission queue specifies a trigger to be fired when messages arrive. For distributed queueing, this is always program MQPSEND, the sender Message Channel Agent (MCA).

The sender MCA uses the channel definition identified in the transmission queue definition and attempts to establish a connection with the remote host identified in the channel definition. This connection may use LU 6.2 or TCP/IP. Establishing a connection causes the receiver MCA to be started on the remote host. The receiver MCA also uses a channel definition in establishing the connection. The channel name must be the same for both the sender and the receiver.

Once a connection is established, the sender MCA begins to read messages from the transmission queue and send them to the receiver MCA. This process continues until the sender MCA detects that no more messages are available to be sent. It then terminates.

The receiver MCA writes the received messages to the local queue definition identified in header information appended to the messages. For this example, the local queue definition is a local queue with a trigger to start application program APP2. The application program then begins to read messages from the local queue using

MQGET calls. This example shows distributed queueing and data sharing across platforms, which is transparent to the application programs.

Chapter 6 of the redbook includes numerous examples of distributed queueing and details all relevant MQ, SNA and CICS definitions. We also explain the use of standard test programs to verify the configuration once it is defined. The examples include:

- VSE 2.4 to VSE 2.3 using LU 6.2
- VSE 2.4 to OS/390 using TCP/IP
- OS/390 to VSE 2.3 using TCP/IP
- OS2/WARP to VSE 2.3 using LU 6.2
- VSE 2.3 to OS2/WARP using LU 6.2
- Windows/NT to VSE 2.3 using TCP/IP
- Windows 95 Client to VSE 2.3 using TCP/IP
- VM/CMS Client to VSE 2.3 using TCP/IP

These examples are intended to act as prototypes for customer configurations, such that customers can duplicate the definitions to exploit distributed queueing with a minimum of effort.

Performance

Performance for MQSeries for VSE/ESA is not addressed in the *System Management Guide*. Consequently, it is covered in the redbook as it is a subject of considerable importance to many customers. In Chapter 7, we explain that there are many factors which influence the performance of MQ applications within the VSE environment. The MQ application can be viewed as part of a stack of software components which each having a bearing on the overall performance. For example, the:

- Application
- MQ subsystem
- CICS subsystem
- File system (VSAM)
- Network (LU6.2 or TCP/IP)
- Operating system (VSE/ESA)

The redbook covers in detail areas of each of these which are immediately relevant to MQSeries for VSE/ESA. More specifically, we focus on MQSeries, CICS, VSAM and Network performance. For example, regarding MQSeries performance, we explain which system parameters can affect performance and provide guidelines for reasonable settings. We also raise considerations for general application design.

Regarding CICS performance, we suggest that in most cases all advice about tuning a CICS environment also applies to MQSeries for VSE/ESA. You should especially look at performance reports (CICS statistics or reports created by vendor software) in order to discover and eliminate potential bottlenecks. We also explain frequency and conditions under which MQSeries transaction and programs execute so that you can take precautions to eliminate unnecessary processing.

For VSAM, we make recommendations regarding maximum and average record sizes and explain how these can affect performance. We also explain that for MQ VSAM clusters you should not define free space for CIs and CAs, for two reasons:

1. There is no initial loading.
2. Unless you have multiple queues in a VSAM cluster, CI or CA splits will never occur since keys are dynamically created based on an ascending sequence.

Once again, we do not recommend multiple queues per VSAM cluster. All considerations for tuning VSAM clusters equally apply to MQSeries files, especially for the number of buffers for data and indexes. When your MQSeries system is operational, and has reached the number of daily messages you would normally expect, check your CICS file statistics. You should check:

1. The number of WAITs on buffers.
2. The hit read ratio.
3. The number of WAITs on strings. If there are any, increase the value in the DFHFCT for the relevant file.
4. The number of Data EXCP versus Index EXCP. If the number of index EXCP is too high, increase the number of index buffers.
5. Try to use LSR pools. However, this is acceptable only if your CPU is not saturated and paging activity is low.
6. Spread your files to different LSR pools (the response time should be better and the CPU activity should be lower).
7. Do not use the same CI sizes for indexes and data.
8. Define very active queues on separate volumes.

Regarding network performance, we explain that tuning MQSeries from a network point of view, is the same as tuning any other CICS application. This is also true for applications using VTAM, which is relevant to MQSeries APPC (LU6.2) connections. Since the use of TCP/IP is relatively new to MQSeries for VSE/ESA, we do focus on specific TCP/IP parameters that may affect the performance of MQSeries.

Security

Like performance, MQSeries security is not discussed in the *System Management Guide*. Consequently, Chapter 10 of the redbook is dedicated to this subject.

We explain that when you write your own applications to connect to MQSeries, you use the MQSeries MQI and the CICS API. The MQSeries MQI is a set of linkable objects which also use the standard CICS API. Therefore, as far as security is concerned, there is no consideration which could apply to normal applications and not to MQSeries application.

When you use CICS security, users need to identify themselves to CICS. This is achieved either by the logon transaction CESN (or the old CSSN) or from the Interactive User Interface (IUI) logon panel. At this point, CICS internally creates a control block called *SNTTE*, which is associated with a terminal. This control block contains information essentially retrieved from the Signon table (or from the VSE/ESA control file if logged via the IUI). If terminal operators do not log on, they may only run transactions which are not protected.

Transaction security is enabled via transaction security keys. When you define a transaction either with RDO or by assembling a Program Control Table (PCT), you can assign a security characteristic (class) to this transaction. It is called a *Transaction Key* and is represented by a number in range from 1 to 64 for the TRANSEC parameter. If no value is specified for TRANSEC, 1 is used by default.

To be able to use this transaction, a user must be given authorization for the TRANSEC value. This is done via the parameter SCTYKEY of the user SNT entry, or if you are using the IUI, by the *Maintain User Profiles* dialog. Up to 64 security keys are possible for the same user. If users do not log on explicitly, they get a default profile, with SCTYKEY=1.

Resource Security Level (RSL) is a second level of CICS security which extends the security concept to resources such as files, transient data destinations, application programs, and temporary storage. However, using RSL requires a very good understanding of applications, otherwise programs may receive a NOTAUTH abend condition code when they are trying to access resources.

Once you have considered logon, transaction and program level security, you should also consider network access. MQSeries for VSE/ESA 2.1 does not provide any facility to prevent people from being connected. This means that such security should be processed at the TCP/IP level (firewall, exits, and so forth) or at the application level. That is, for the moment, unless you entirely trust your security system, it may be wise to use MQSeries for VSE/ESA only in an intranet environment.

Problem Determination

Problem determination is covered in the *System Management Guide* and is an essential starting point for problem resolution. The redbook provides a supplementary chapter written by Simon Davitt, who is currently responsible for MQSeries for VSE/ESA support at Hursley. Chapter 11 covers:

- Installation pitfalls
- Application program problems
- Common problems
- Known problems

MQSeries for VSE/ESA runs as a subsystem within a CICS/VSE or CICS TS environment. Because of this, it relies on a software stack for its communications and application support functions. For example, the COBOL, C and PL/I language support is performed by LE/VSE; the intersystem communications support is performed by VTAM or TCP/IP; and the file system support is performed by CICS and VSAM. Because of the dependence by MQSeries for VSE/ESA on the software stack, it is extremely important that the stack is at the correct release and service level.

To run successfully, MQSeries requires several CICS groups to be installed. MQSeries itself is defined in a CICS group called *MQM*. This contains the required definitions for the MQSeries programs, transactions, mapsets and so on. In addition to this, there are several others which are required:

1. V2.1 of MQSeries for VSE/ESA uses the EXEC CICS WRITE OPERATOR command to put messages to the VSE console. This requires that the CICS DFHEXEC group is installed.
2. LE, COBOL and C runtime support. The CICS group name required for this is *CEE*, but the definitions are split between several files, one file for each component.

CSI customers with TCP/IP should note that from September 1999, CSI ships the LE socket interface module, $EDCTCPV. This is the interface which MQ uses for all its TCP/IP calls. The module is shipped in their default library PRD2.TCPIP. This library must be placed in the LIBDEF statement before the PRD2.SCEEBASE library. Prior to September 1999, the $EDCTCPV phase is available in PRD1.BASE.

Chapter 11 also examines several methods for debugging MQ application programs. Methods covered include:

- LE source language debugger
- CICS Execution Diagnostic Facility (CEDF)
- CICS auxiliary trace
- The MQSeries MQI monitor

Once a system has been installed and the initial problems encountered during development and testing have been resolved, most customers run their production systems without problems. Even so, problems can occur on a system which has been stable for some time. If problems begin to occur in a system that has previously been running correctly, then you should look for possible changes which may have caused the problem. To this end, you should consider the following:

- Changes or increase in workload or work patterns
- New hardware installed
- New or updated software installed

Additional Information

A "red piece" for SG26-5647 is available via:

http://www.redbooks.ibm.com/

Whenever you encounter a problem which you have fully researched but cannot resolve, it is worth seeking assistance from other sources. Possible sources include IBM support centers, Web sites, user groups, closed APARs, and IBM documentation.

What The TOD And TIME And ZONE Is All About

Author

Axel Pieper
VSE Development
IBM Böblingen
pieper@de.ibm.com

Native VSE Users

When I talk about the processor clock, I refer to it as the TOD (Time Of Day clock). This TOD (normally[3]) always contains the GMT/UTC and it normally keeps running once it has been set. I don't know of any reason why someone would ever need to change this TOD on a production system (unless the clock is really broke). In "Interesting Stuff About The TOD" on page 59, I have added some details about the TOD and what it is about.

The (initial) setting of this clock is accomplished by the
`SET DATE=mm/dd/yyyy,CLOCK=hh/mm/ss,ZONE=zone/hh/mm`
command during the first IPL. The DATE, CLOCK and ZONE are assumed to be the LOCAL values at the time the command is issued.

Based on these values, IPL will CALCULATE the corresponding GMT-time and STORE THIS GMT time into the TOD where it will be incremented every few micro-, or even nano-seconds (for most processors even if they are powered OFF). The ZONE operand, however, is saved within the SYSCOM, will thus be lost on the next subsequent IPL, and is required by all timer-related services (mainly GETIME). This is the reason why you NEED TO HAVE a SET ZONE or SET DEFZONE statement(s) in your IPL ASIPROC. Otherwise, VSE would be unable to calculate the proper LOCAL time.

```
SET ZONEDEF,ZONE=WEST/05/00,EST
SET ZONEDEF,ZONE=WEST/04/00,EDT
```

These above two statements (you can, of course, have more) are telling VSE that you have actually two zones that you work in and that is a zone FOUR/FIVE hours BEHIND (WEST) of GMT. In addition, each statement provides a 3-charcter string (EDT,EST which can freely be chosen) and which VSE is going to associate to the specified ZONE. IPL, when reading these statements, does not yet know which one to apply. For that purpose we need:

```
SET ZONEBDY,DATE=04/05/1998,CLOCK=02/00/00,EST
SET ZONEBDY,DATE=10/25/1998,CLOCK=02/00/00,EDT
SET ZONEBDY,DATE=04/04/1999,CLOCK=02/00/00,EDT
SET ZONEBDY,DATE=10/31/1999,CLOCK=02/00/00,EST
SET ZONEBDY,DATE=04/02/2000,CLOCK=02/00/00,EDT
SET ZONEBDY,DATE=10/29/2000,CLOCK=02/00/00,EST
SET ZONEBDY,DATE=04/01/2001,CLOCK=02/00/00,EDT
SET ZONEBDY,DATE=10/28/2001,CLOCK=02/00/00,EST
SET ZONEBDY,DATE=04/07/2002,CLOCK=02/00/00,EDT
```

[3] When I say normally, then I mean that is the way it should be all over the world, but it assumes that anyone (except GMT zone folks) did provide the proper ZONE in the initial (IPL) SET DATE ... command).

```
SET ZONEBDY,DATE=10/27/2002,CLOCK=02/00/00,EST
SET ZONEBDY,DATE=04/06/2003,CLOCK=02/00/00,EDT
SET ZONEBDY,DATE=10/26/2003,CLOCK=02/00/00,EST
SET ZONEBDY,DATE=04/04/2004,CLOCK=02/00/00,EDT
SET ZONEBDY,DATE=10/31/2004,CLOCK=02/00/00,EST
SET ZONEBDY,DATE=04/03/2005,CLOCK=02/00/00,EDT
SET ZONEBDY,DATE=10/30/2005,CLOCK=02/00/00,EST
SET ZONEBDY,DATE=04/02/2006,CLOCK=02/00/00,EDT
SET ZONEBDY,DATE=10/29/2006,CLOCK=02/00/00,EST
SET ZONEBDY,DATE=04/01/2007,CLOCK=02/00/00,EDT
SET ZONEBDY,DATE=10/28/2007,CLOCK=02/00/00,EST
SET ZONEBDY,DATE=04/06/2008,CLOCK=02/00/00,EDT
SET ZONEBDY,DATE=10/26/2008,CLOCK=02/00/00,EST
SET ZONEBDY,DATE=04/05/2009,CLOCK=02/00/00,EDT
SET ZONEBDY,DATE=10/25/2009,CLOCK=02/00/00,EST
SET ZONEBDY,DATE=04/04/2010,CLOCK=02/00/00,EDT
SET ZONEBDY,DATE=10/31/2010,CLOCK=02/00/00,EST
```

These statements now, based on the current LOCAL time, will tell IPL which ZONE to apply. If the current LOCAL time is EQUAL or GREATER than the local time as specified in the ZONEBDY statement, but LOWER than the ZONEBDY next in sequence, then that associated ZONE-ID (EST/EST) of the current ZONEBDY statement will be used to identify the ZONE that needs to be set (into SYSCOM). Remember, the TOD remains UNCHANGED in this case.

Assuming you have the above statements incorporated into your IPL ASIPROC, then you don't have to be concerned for the next 10 years. Just make sure that you IPL your VSE-system always after a ZONEBDY date and time has just been passed. Then you are OK.

Now some words to the TIME command. The TIME command allows you to either change the DATE and CLOCK (mainly intended for test purposes only) and/or the ZONE on the fly. Changing the ZONE could eventually impact subsystems which use the GETIME macro and subsequently use these values in journal or recovery actions. Such subsystems, if known to you, should be stopped before the ZONE is changed and then brought up again after the ZONE change. Changing the ZONE could also impact Job-accounting data.

When the TIME command is used with DATE and/or the CLOCK operand, then the processor clock (TOD) will be altered right away (unless the clock is secured). If the clock is secured, the operator must enable the clock setting otherwise the update will be rejected. Setting the clock can only be recommended if all subsystems are down and no job is currently running, or on a test-system.

Some words now to why you shouldn't change the TOD. This is simply because there are quite a few programs which use the TOD to set TIMESTAMPS here and there. These programs could get into big trouble if all of a sudden a timestamp is lower than a previous one (which would be the case in Autumn when the time is set back 1 hour). If you don't change the TOD, then the timestamp would always and at any time be correct and would represent the micro- or nano-seconds that have passed since 1.1.1900 in Greenwich. (Right now this is: 3,132,244,833 seconds – quite some time.)

Some Words If You Run Under VM

As you probably know, VM does also provide the facilities to specify Zones, as well as Zone-Boundaries similar to VSE. (We actually copied those definitions.) These changes in VM will become effective as stated in the appropriate VM-manuals (sorry, I don't have one), but it is certainly similar to what VSE does. VM will also just update its Zone-field, but certainly not any TOD, be it the REAL or a VIRTUAL one. With REAL TOD I mean the real processor clock; and the VIRTUAL TOD is the one which is being SIMULATED by VM, but only if you have specified TODENABLE in your virtual guest directory entry.

VM, UNDER NO CIRCUMSTANCES, will allow any virtual guest to update the REAL TOD. Any attempt to do so (by means of the Set-Clock instruction) will simply be IGNORED and has no effect at all. This is different for the VIRTUAL TOD, (established when you have the TODENABLE in your DIR entry). This VIRTUAL TOD clock can be set to any time without any impact to the REAL VM TOD. The VIRTUAL TOD clock will be initialized with the REAL TOD value, but keep in mind it is a copy. (What VM does, in case the virtual guest changes its clock setting, is to simply maintain the difference to the REAL VM TOD (you may consider it a correction factor and nothing else)).

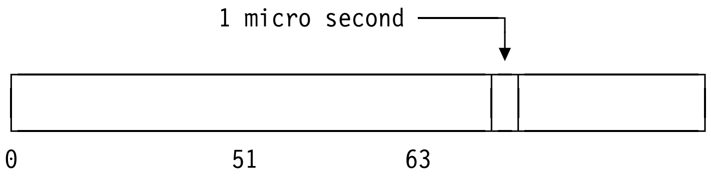
The REAL VM TOD clock is also set to GMT/UTC time and VM, similar to VSE, will require the ZONE-operand to be specified properly. This is all very similar, not to say identical, to what I said for native VSE. VM does also save this ZONE value and it can actually also be retrieved (via a special DIAG command) by any virtual guest. And this is exactly what VSE does (assuming you are on VSE 2.3.x level). IPL will retrieve this ZONE value from VM and will use this ZONE setting as long as it is not overruled by ANY ZONE statement in the IPL ASIPROC. VSE zone definitions will always supersede the VM retrieved ZONE definition.

The VM ZONE setting will ONLY be retrieved during IPL processing. This means that a ZONE change in VM will NOT BE REFLECTED to the VSE systems (generally such a TIME ZONE change interrupt is available in VM, but VSE doesn't have it enabled) as long as the VSE system isn't IPLed. This simply guarantees that VSE will keep running with its initial TOD and ZONE settings although VM could now be working with a different ZONE, but still the same REAL TOD. But as soon as you now IPL the VSE system, you will get the new VM zone as long as it isn't overruled by a VSE ZONE statement.

Unless there is some special need, I would recommend to NOT USE any ZONE statements in VSE if you are running under VM. VSE will be smart enough to recognize the correct VM settings and this way you only need to maintain the VM specifications which you need to do anyway.

Interesting Stuff About The TOD

For all those that want to know it, the TOD clock (processor clock) is a binary counter with the format shown in the following illustration. The bits are numbered from 0 to 63, corresponding to the bit positions of a 64-bit unsigned binary integer.



In the basic form, the TOD is incremented by adding a one in bit position 51 every microsecond. The resolution factor could differ for the different processors, but the effect is the same as if a one would be added to bit position 51 every micro second. This TOD always assumes having been started on the 01/01/1900 (00.00 AM). So, based on GMT it was/is/will be:

```
01/01/1900  0:00:00 AM  =  X'0000 0000 0000 0000'
12/31/1999 11:59:59 PM  =  X'B361 183F xxxx x000'
01/01/2000 00:00:00 PM  =  X'B361 1840 xxxx x000'
```

As you see, the TOD will not be impressed by the millennium change at all.

BUT, WHAT IS GOING TO HAPPEN ON THE 09/17/2042 at 11:53:46 PM GMT

when this TOD clock will wrap around from X'FFFF FFFF FFFF FFFF' to X'0000 0000 0000 0000'?? The situation, as of today, there won't be any notification to the operating system(s) and thus to the programs. IT IS A REAL FALLBACK into the year 1900. All those programs which have used a SETIME, they will have to WAIT approximately another 142 years until they get posted. You see this date is another challenging issue for the IT industry. I am a grey-haired VSE-veteran-fan, so I don't bother too much, but all the VSE-kid-fans among you may want to start asking questions now and then, to get this issue addressed. For, without any question, ...

VSE will certainly be alive at that date.

News For Language Environment For VSE/ESA (LE/VSE)

Authors

Wolfgang Bosch
Heiko Schnell
VSE Service and Development
IBM Böblingen
wbosch@de.ibm.com
hschnell@de.ibm.com

Mixed Language Applications Under LE/VSE (Involving Assembler)

When creating or maintaining mixed language applications in an LE/VSE environment, various supported techniques are available. This summary is intended to help you to select the right implementation. It primarily focuses on how to best exploit:

- LE/VSE-conforming Assembler programs (which means the Assembler routines are prepared with **CEEENRTY**/**CEETERM** macros)
- LE/VSE preinitialization service (**CEEPIPI**)
- C-specific macros (**EDCPRLG** and **EDCEPIL**)

The following sections and Figure 24 on page 62 show recommended techniques. Examples and further details not covered in this article are available on LE/VSE home page:

http://www.s390.ibm.com/le_vse/samples.htm

Key Scenarios

- A LE/VSE-conforming Assembler (main or sub) routine calls an High Level Language (HLL) routine such as COBOL/VSE, PLI/VSE and C/VSE. This is a typical situation where the use of CEEENTRY/TERM macros is required.
- A non-LE/VSE-conforming Assembler driver calls HLL routines. Here CEEPIPI pre-initialization is the appropriate coding method.
- A C/VSE program calls an Assembler routine bridging to another C/VSE subroutine. This Assembler routine is a good candidate for EDPRLG/EPIL macros usage.

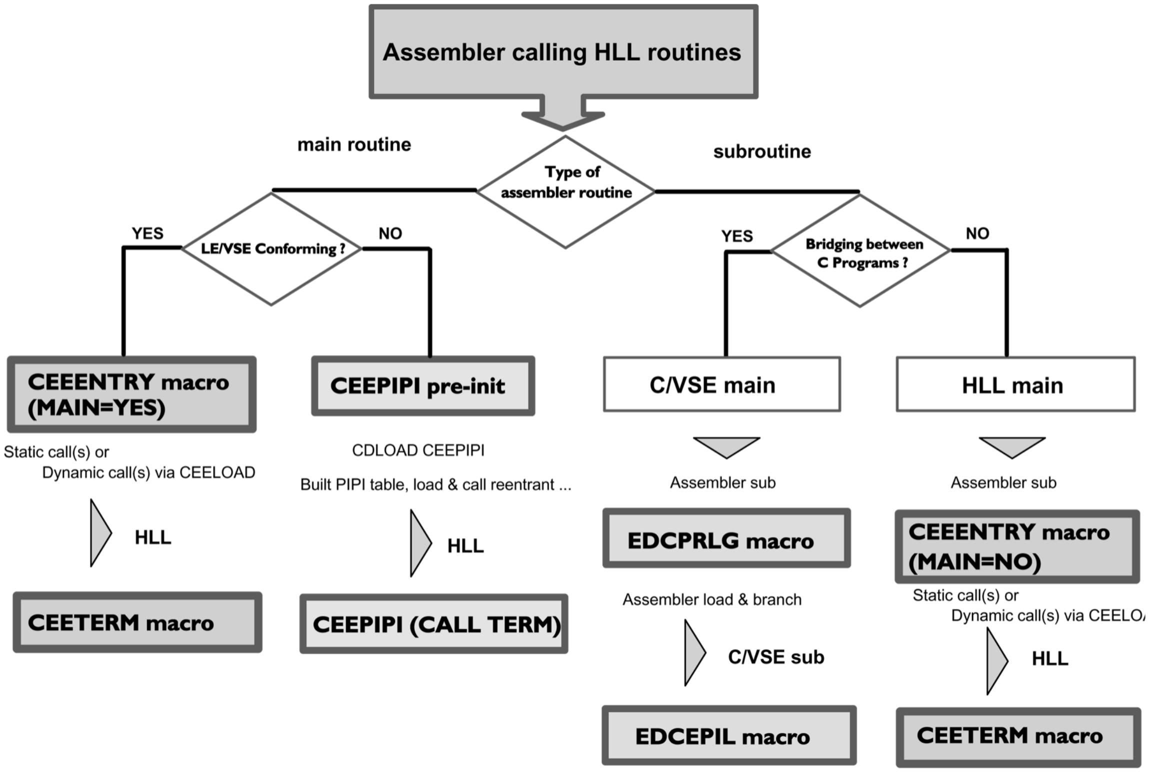


Figure 24. Assembler Calling High Level Language (HLL) Routines

Hints And Tips

- LE/VSE-conforming Assembler macros (for example, CEEENTRY/TERM) are restricted to Assembler batch routines calling HLL routines. These macros preserve the register and calling conventions for the communication between LE/VSE and HLL applications. The macros are not supported under CICS/VSE and CICS TS. Here EXEC CICS LINK and EXEC CICS XCTL commands should be used instead.

  APAR/PTF (**PQ26738**/**UQ33950**) provides additional flexibility for programmers using CEEENTRY. New capabilities are added to specify the residency mode and allow for multiple base registers (parameters RMODE= and BASE=.

- CEEPIPI is a batch interface which initializes and terminates an LE/VSE environment for specific HLL routines. Its basic use is to allow a non-LE/VSE-conforming Assembler program to call HLL routines.

- If your requirement is to stay in an LE/VSE C run-time environment, the Assembler macros EDCPRLG and EDCEPIL set up the communication conventions for calling subsequent C/VSE subroutines. This bridging Assembler routine has to be called by a C/VSE main program. If communication to other HLLs is required, the CEEENTRY/TERM macros should be used. Be aware that CEExxxx and EDCxxxx macros are mutually exclusive and cannot be coded in the same Assembler program.

Further considerations about mixed language applications are covered in *LE/VSE Writing InterLanguage Communications Applications*, SC33-6686-00 (Chapter 6), *LE/VSE Programming Guide*, SC33-6684-01 (Chapter 26,27), and *LE/VSE C Programming Guide*, SC33-6688-01 (Chapter 15).

Customizing The LE/CICS Abnormal Termination Exit

You may have experienced LE/VSE abend code 4039, which is issued by the default "CICS abnormal termination exit" CEECDATX. This situation either arises if the LE/VSE run-time option TERMTHDACT(UADUMP) is set for the CICS environments or if unhandled conditions such as ASRA, ASRB occur. The resulting problem might be that the default exit suppresses some regular CICS transaction dump symptoms.

Scenario

An operator has purged a CICS transaction "USER" usually resulting in the abend code ATCH. The CICS transaction dump analyzing utility DFHDUP produces a VSE/POWER list queue entry named "USERATCH" ("transaction name" + "abend code"). With the default "CICS abnormal termination exit," the queue name is "USER4039," thereby hiding the original error condition!

How To Cope With 4039 Events

- Do not use LE/VSE run-time option TERMTHDACT(UADUMP) under CICS. Since CEECDATX already drives the request for a transaction dump via "EXEC CICS DUMP TRANSACTION(4039)," any other setting – primarily TERMTHDACT(DUMP) – is recommended.
- Preventively use PCDUMP=YES in the CICS System Initialization Table (SIT), which forces CICS to produce an IDUMP (partition dump).
- If the phenomenon still persists, it is useful to disable the default exit handling and enable the null exit CEEBNATX. This avoids extra LE/VSE dumps and reestablishes the CICS type "Axxx" abend codes. For further diagnosis help, refer to the common CICS problem determination procedures (such as CICS auxiliary trace, etc.).

How To Set Up The Null Exit

- Use CEEBNATX.A from PRD2.SCEEBASE and assemble/link (for example, via "OPTION 8" COMPILE in ICCF Primary Library) a corresponding PHASE representing the null exit. Do NOT run the CICS translator against this piece of code (only use // EXEC ASMA90 ...) because it might cause an CICS abend dependent on the translator options used!
- Make sure that you have installed the PTF for PQ23380 or that you have added a PPT entry for Assembler exit program CEEBNATX by yourself.
- After successful assembly, the user module should be added to the abnormal termination table:
  - Take skeleton CEEWCEXT from ICCF library 62
  - Include copybook CEECXTAN.A.
  - Change line "CEEXART TERMXIT=CEECDATX" to "CEEXART TERMXIT=CEEBNATX".
  - Run CEEWCEXT (assemble and link job).
  - Shutdown CICS, recycle CICS cold.

Also refer to *LE/VSE 1.4 Installation and Customization Manual* (SC33-6682-00, page 68 or SC33-6682-01, page 70) for further consideration.

Year 2000 PTFs For LE/VSE

This is a summary from "YR2000VSE" PSP (subset "LE"). These PTFs are also included in all "VSEESAxxx" specific PSPs (related subset "IBMLANG/1DS" for LE/VSE 1.4).

Table 4. Year 2000 PTFs for LE/VSE

APAR	PTF	Available	Summary
PQ32076	UQ36679	99/11/02	Century fix for Assembler-C macro EDCPRLG
PQ24951	UQ30080	99/05/14	Incorrect CEEPPA macro (hard code century 19)
PQ23866	UQ27896	99/03/18	PL/I MLE & msg IBM0506S ONCODE=2106
PQ23746	UQ27072	99/03/02	PL/I MLE to recognize YR2000 as leap year
PQ15750	UQ18792	98/06/23	Pre-req APAR for PL/I MLE function
PQ15749	UQ18913	98/06/23	Pre-req APAR for PL/I MLE function
PQ15114	UQ18665	98/06/23	Supporting code for COBOL MLE function
PQ15112	UQ18664	98/06/23	Supporting code for COBOL MLE function
PQ15106	UQ18836	98/06/23	Supporting code for COBOL MLE function
PQ04468	UQ05847	97/06/24	Allow to handle new DFSORT date fields
PQ03669	UQ06203	97/06/19	Make LE/VSE C Run-Time YR2000 ready

Useful Run-Time Options

You may find these tips valuable in your environment:

- Use LE/VSE run-time option **TERMTHDACT(UADUMP)** for diagnosis in a batch environment. It generates a VSE partition dump and a corresponding LE/VSE dump.
- The default run-time option **HEAP(..,ANYWHERE,..)** is important because of the correlation to COBOL compile options DATA(24/31), RMODE and RENT. Be aware that there is a 64KB limit for CICS GETMAIN storage requests below the 16MB line. Avoid HEAP(BELOW), since it ignores the COBOL DATA(xx) compile setting for location of dynamic data areas (such as working storage)!
- Setting **TRAP(ON)** ensures LE/VSE's integrity when handling abnormal conditions. TRAP(OFF) should only be activated on request of IBM service personal.

Additional Information

LE/VSE documentation links and pointers include:

- **http://www.s390.ibm.com/le_vse/** (LE/VSE home page)
- *LE/VSE Enhancements Guide*, SC33-6778-00
- *LE/VSE Debugging Guide and Run-Time Messages*, SC33-6681-**01**
- *LE/VSE Installation & Customization*, SC33-6682-**01**
- *LE/VSE Programming Guide*, SC33-6684-**01**
- *LE/VSE Programming Reference*, SC33-6685-**01**
- *LE/VSE C Run-Time Programming Guide*, SC33-6688-**01**

REXX/VSE Sockets Application Program Interface

Author

Ursula Braun-Krahl
REXX/VSE Development
braunu@de.ibm.com

Overview

With year-end 1999, REXX/VSE provides a REXX Socket API within PTF **UQ37224** for VSE/ESA 2.3 and **UQ37225** for VSE/ESA 2.4 (APAR **PQ31258**). It allows you to write socket applications in REXX/VSE for the TCP/IP environment. The REXX Socket Interface – a new function called "SOCKET" – has the same syntax and semantic as the SOCKET function available for REXX/VM and REXX in OS/390.

```
>>--SOCKET(subfunction------------------)----><
                       |              |
                       |   ---<---    |
                       |  |       |   |
                        ----,arg-- ---
```

subfunction name of the socket call

arg parameters for the socket call

A character string is returned from the SOCKET call that contains several values separated by blanks, so the string can be parsed using REXX. The first value in the string is usually the return code. If the return code is zero, the values following the return code are returned by the socket subfunction. If the return code is not zero, the second value is the name of an error, and the rest of the string the corresponding error message.

Here are some examples:

```
SOCKET('GetHostId')   ==> '0 9.4.3.2'
SOCKET('Recv',socket) ==> '1102 EWOULDBLOCK Problem on non-blocking socket'
```

The REXX/VSE SOCKET implementation uses the LE/VSE support to access the TCP/IP socket interface. Thus it maps the socket calls from the C programming language to REXX/VSE.

Tasks You Can Perform Using REXX Sockets

You can use REXX sockets to perform the tasks described in this section.

Processing Socket Sets

A socket set is a number of preallocated sockets available to a single application.

Table 5. REXX Socket Functions for Processing Socket Sets

Function	Purpose
Initialize	Defines a socket set
Terminate	Closes all the sockets in a socket set and releases the socket set
SocketSet	Gets the name of the active socket set
SocketSetList	Gets the names of all the available socket sets
SocketSetStatus	Gets the status of a socket set

Creating, Connecting, Changing, And Closing Sockets

A socket is an endpoint for communication that can be named and addressed in a network.

Table 6. REXX Socket Functions for Creating, Connecting, Changing, and Closing Sockets

Function	Purpose
Socket	Creates a socket in the active socket set
Bind	Assigns a unique local name (network address) to a socket
Listen	Converts an active stream socket to a passive socket
Connect	Establishes a connection between two stream sockets
Accept	Accepts a connection from a client to a server
Shutdown	Shuts down a duplex connection
Close	Shuts down a socket

Exchanging Data

You can send and receive data on connected stream sockets and on datagram sockets.

Table 7. REXX Socket Functions for Exchanging Data

Function	Purpose
Read	Reads data on a connected socket
Write	Writes data on a connected socket
Recv	Receives data on a connected socket
Send	Sends data on a connected socket
RecvFrom	Receives data on a socket and gets the sender's address
SendTo	Sends data on a socket, and optionally specifies a destination address

Resolving Names And Other Identifiers

You can get information such as name, address, client identification, and host name. You can also resolve an Internet Protocol address (IP address) to a symbolic name of a symbolic name to an IP address.

Table 8. REXX Socket Functions for Resolving Names and Other Identifiers

Function	Purpose
GetHostId	Gets the IP address for the host processor
GetHostName	Gets the name of the host processor
GetSockName	Gets the local name to which a socket was bound
GetHostByAddr	Gets the host name for an IP address
GetHostByName	Gets the IP address for a host name
Resolve	Resolves the host name through a name server

Managing Configurations, Options, And Modes

You can obtain the version number of the REXX Sockets function package, get socket options, set socket options, and set the mode of operation.

Table 9. REXX Socket Functions for Managing Configurations, Options and Modes

Function	Purpose
Version	Gets the version and date of the REXX Sockets function package
Select	Monitors activity on selected sockets
GetSockOpt	Gets the status of options for a socket
SetSockOpt	Sets options for a socket
Fcntl	Sets or queries the mode of a socket
Ioctl	Controls the operating characteristics of a socket

Additional Information

This article contains only a small announcement of the new REXX/VSE Socket function. More information is on the VSE/ESA home page:

http://www.s390.ibm.com/vse/

A more detailed article and coding samples are planned for the next issue of the *VSE/ESA Newsletter*.

If you have any questions or comments about this article or about REXX/VSE, please feel free to contact me using my e-mail address.

LPAR Communications Using ESCON Channels

Author

Aubrey Hayes
ISB Company
Aubrey@ISBCo.com

The following definitions allow communications to occur between two VSE/ESA systems running on an IBM Multiprise® server in LPAR mode with ESCON channels. This example has two LPARs – **VSE_A** and **VSE_B**. A terminal defined in one LPAR has access to CICS in either LPAR via the VTAM logon screen.

IOCP

```
CH4C CHPID PATH=4C,TYPE=CTC,PART=(A)
     CNTLUNIT CUNUMBR=4CA,PATH=(4C),UNIT=SCTC,UNITADD0((00,1))
     IODEVICE ADDRESS=700,1),CUNUMBR=4CA,UNIT=SCTC

CH4D CHPID PATH=4D,TYPE=CNC,PART=(B)
     CNTLUNIT CUNUMBR=4DA,PATH=(4D),UNIT=SCTC,UNITADD0((00,1))
     IODEVICE ADDRESS=700,1),CUNUMBR=4DA,UNIT=SCTC
```

VSE/ESA

```
ADD 700,CTCA,EML (in IPLPROC for both VSE_A and VSE_B)
```

VTAM

Normally there should be no need to modify your start list. Review the following parameters, however. Changes to your system may be necessary.

```
CATALOG ATCSTR00.B (for VSE_A)
...
SSCPID=10,HOSTSA=10,SSCPNAME=SSCP10,HOSTPU=NODE10,
...

CATALOG ATCSTR00.B (for VSE_B)
...
SSCPID=11,HOSTSA=11,SSCPNAME=SSCP11,HOSTPU=NODE11,
...
```

Modify the configuration list in order to have VTAM load and activate the lists and major nodes. On the other hand, if it is preferred to occasionally start PNET connection, catalog the books and activate them as needed by issuing the respective `V NET,ACT,ID=xxxxxx` commands.

```
CATALOG ATCCON00.B
...
VSEAPPL,VSECDRM,VSECDRS,VSEPATH,VSECTCA
...
```

The VTAM application major node defines the VSE/POWER PNET application to VTAM (page 69).

```
CATALOG VSEAPPL.B (for VSE_A)
VSEAPPL   VBUILD TYPE=APPL
PNETA   APPL AUTH=(PASS,ACQ),VPACING=3,MODETAB=VTMLOGTB,DLOGMOD=PNET

CATALOG VSEAPPL.B (for VSE_B)
VSEAPPL   VBUILD TYPE=APPL
PNETB   APPL AUTH=(PASS,ACQ),VPACING=3,MODETAB=VTMLOGTB,DLOGMOD=PNET
```

The next use of the VBUILD statement defines the cross-domain resource manager major node. It must be followed by one or more cross-domain resource manager minor nodes. In the following example, the first domain is the local VSE system, and the second domain is the remote VSE system.

Note: In POWER PNET, the number of buffers to be allocated to every transmitter and receiver communicating with a node is defined in the MAXBUF of the PNODE macro. The default value of this operand is three receivers and three transmitter buffers. **Note that the pacing value (VPACING) and the MAXBUF operand of the PNODE macro must agree. Therefore, specify VPACING=3.**

```
CATALOG VSECDRM.B (for VSE_A)
VSECDRM   VBUILD TYPE=CDRM
VSEA      CDRM  SUBAREA=10,CDRDYN=YES,ISTATUS=ACTIVE,VPACING=3
VSEB      CDRM  SUBAREA=11,ISTATUS=ACTIVE,CDRSC=OPT,VPACING=3

CATALOG VSECDRM.B (for VSE_B)
VSECDRM   VBUILD TYPE=CDRM
VSEB      CDRM  SUBAREA=11,CDRDYN=YES,ISTATUS=ACTIVE,VPACING=3
VSEA      CDRM  SUBAREA=10,ISTATUS=ACTIVE,CDRSC=OPT,VPACING=3
```

Now the VBUILD statement is used to define the cross-domain resource major node. It must be followed by one or more cross-domain resource manager minor nodes. They define those cross-domain resources with which applications or logical units within this domain can have a session.

```
CATALOG VSECDRS.B (for VSE_A)
VSECDRS   VBUILD TYPE=CDRSC
PNETB     CDRSC CDRM=VSEB
CICSB     CDRSC CDRM=VSEB

CATALOG VSECDRS.B (for VSE_B)
VSECDRS   VBUILD TYPE=CDRSC
PNETA     CDRSC CDRM=VSEA
CICSA     CDRSC CDRM=VSEA

CATALOG VSECTCA.B (for both VSE_A and VSE_B)
VSECTCA   VBUILD TYPE=CA
CTCGROUP  GROUP LNT=CTCA,REPLYIO=3.0,ISTATUS=ACTIVE
CTCLINE   LINE  ADDRESS=700,DELAY=0.1,MIH=YES,MAXBFRU=8
CTCPU     PU    PUTYPE=4,TGN=3

CATALOG VSEPATH.B (for VSE_A)
LPARB    PATH  DESTSA=11,ER0=(11,3),VR0=0

CATALOG VSEPATH.B (for VSE_B)
LPARA    PATH  DESTSA=10,ER0=(10,3),VR0=0
```

VSE/POWER

The only specification required in the POWER macro is to include PNET=phase, where phase is the name of the phase under which the Network Definition Table has been cataloged.

The NDT establishes the relationship between POWER and VTAM. The NDT is automatically loaded during PNET initialization.

The following jobs will generate the two NTDs for the LPARS.

```
// JOB NDT GENERATON
// LIBDEF PHASE,CATALOG=PRD.CONFIG
// OPTION CATAL
// EXEC ASSEMBLY
         TITLE 'POWER-PNET LPAR_A'
         PRINT NOGEN
NDTA     PNODE NODE=NODE10,APPLID=PNETA,LOCAL=YES
         PNODE NODE=NODE11,APPLID=PNETB,AUTH=NET
         END
/*
// EXEC LNKEDT
/&

// JOB NDT GENERATON
// LIBDEF PHASE,CATALOG=PRD.CONFIG
// OPTION CATAL
// EXEC ASSEMBLY
         TITLE 'POWER-PNET LPAR_B'
         PRINT NOGEN
NDTB     PNODE NODE=NODE11,APPLID=PNETB,LOCAL=YES
         PNODE NODE=NODE10,APPLID=PNETA,AUTH=NET
         END
/*
// EXEC LNKEDT
/&
```

The logmode table is VTMLOGTB, which is supplied by VSE/ESA. The logmode table entry is PNET, which is part of the VTMLOGTB table. For reference, it is included below.

Please note that the values for PACING are all set to 3, to comply with the default MAXBUF operand of the PNODE macro.

```
VTMLOGTB MODETAB
IBMPOWER MODEENT LOGMODE=PNET,PSNDPAC=3,SRCVPAC=3,SSNDPAC=3,
                FMPACFC=X'03',TSPROFC=X'03',
                PRIPROT=X'72',SECPROT=X'72'
         MODEEND
         END
```

CICSVR/VSE – IBM's Enhanced VSAM Dataset Recovery Program

Author

Holly Yamamoto-Smith
CICSVR/VSE and DFSORT/VSE Team Leader
IBM San Jose
hollyyam@us.ibm.com

Do you have recovery procedures set up just in case you lose your CICS VSAM data? If you don't, you're running a BIG risk...and an unnecessary one at that! In November 1999, IBM announced and made available in December 1999, a new release of CICS VSAM Recovery for VSE (CICSVR/VSE V1R2) which will assist you when you need to recover lost or damaged CICS data.

What Is CICSVR?

CICS VSAM Recovery (CICSVR) is an IBM utility which can forward recover and backout CICS transactions that are stored in VSAM datasets. CICSVR can use the information in the CICS system log or user journals to reapply your CICS transactions to the user restored backup copy or to remove partially completed transactions when a CICS backout has failed. CICSVR can be used with CICS TS for VSE or with CICS/VSE V2R3.

Why Do You Need CICSVR?

Without a forward recovery or backout utility, lost or damaged datasets must be recovered manually. If each update is reentered manually, errors can increase, which can threaten the integrity of your VSAM data. If you have high transaction volumes between backups, you want to avoid long interruptions due to failed or lost datasets. CICSVR minimizes the time and effort involved in the recovery process and eliminates the errors that often occur with manual recoveries.

CICSVR Features

- The CICSVR archive utility gives you the ability to automatically create one or more copies of your CICS journals and produce a summary report on all the VSAM files on that journal. The information in the archive summary report will assist you in creating a CICSVR forward recovery job and/or CICSVR backout job.
- The CICSVR forward recovery function helps you to recreate one or more VSAM files from a user restored backup copy of the file.
- The CICSVR backout function is used when an online CICS TS backout has failed. CICSVR backout removes partially completed transactions and is only supported with CICS TS.

The Big Picture

Before CICSVR can be used to recover your VSAM datasets, you must complete some initial preparation.

- You must request dataset recovery in the CICS FCT.

  For CICSVR forward recovery, CICS must produce a forward recovery journal.

For backout, CICS must produce a system log for backout. If CICS backout fails, then CICSVR backout can be used to complete the dataset backout. (Remember, backout is only supported with CICS TS.)

- You must set up CICS to automatically run CICSVR archive to produce a report and create a copy of the journal every time the journal becomes full.

  CICS TS provides an automatic journal archiving feature to ensure that datasets are archived promptly and are not overwritten until archived. CICS/VSE V2 does not provide an automatic journal archiving feature so CICSVR will provide this ability instead. Support for automatic archiving for CICS/VSE V2 will be available as a separate CICSVR PTF.

- You must set up a plan to regularly backup your VSAM datasets.

  CICSVR forward recovery uses the restored VSAM sphere as the starting point for the forward recovery. (A VSAM sphere is a VSAM base cluster that contains the data and optionally, one or more AIX [alternate index] files containing record keys of data records in the base cluster.)

Once you've done the initial preparation, it's really easy to use CICSVR to recover your VSAM data. The following two scenarios illustrate how CICSVR can help you.

- Scenario 1 – Disaster strikes and you need to recreate your VSAM data. Since you have already done the necessary preparation, you can easily recreate your VSAM data using the information in the CICSVR archive report to construct a CICSVR forward recovery job.

  – First, review the CICSVR archive reports to find out which journals contain the after-images of the VSAM sphere you need to recover and construct the CICSVR forward recovery job.

  – Second, restore the backup copy of the VSAM dataset that needs to be recreated.

  – Third, run CICSVR forward recovery to apply the after-images from the forward recovery journals to the restored VSAM sphere. When the forward recovery is complete, CICSVR has recreated the dataset.

- Scenario 2 – CICS online backout fails. If CICS encounters a problem and cannot complete an online backout, you can use the CICSVR backout function if you are using CICS TS.

  – First, review the CICSVR archive reports to find out which system logs contain the before-images of the VSAM sphere you need to backout.

  – Second, run CICSVR backout to apply the before-images from the system log to backout the changes to the VSAM sphere. When the CICSVR backout is complete, CICSVR has removed the partially completed transaction.

Additional Information

For a complete description of the announcement, please see the IBM Software Announcement Letter **299-366**.

Sample VSE And VM Customer "Success" Stories

This section has three short articles that appeared in the December 1999 issue of the *S/390 Bulletin*. I wanted to make sure you see them, because they illustrate the wide range of solutions available to VM and VSE customers.

If you are a customer who would like to showcase your company in a similar article, please contact me (**jhermann@de.ibm.com**). I will help you create the article, which will reach other IBM customers, Business Partners, and IBM employees in over 80 countries.

If you are a third-party vendor or Business Partner, this invitation extends, of course, to you also!

Harrassowitz "Webs It" With VSE/ESA And Java

Authors

Friedemann A. Weigel
Director and Manager
Harrassowitz Booksellers and
Subscription Agents
fweigel@harrassowitz.de

Hans Joachim Ebert
S/390 Technical Support VM/VSE
IBM Munich
eberthj@de.ibm.com

HARRASSOWITZ
Booksellers & Subscription Agents

Harrassowitz Booksellers & Subscription Agents of Wiesbaden, Germany, have valued their VSE system for years. One basic reason is because their market is specialized, and providing better service than your competitor is necessary for success. Recently, Harrassowitz put the first stage of a VSE-based Java application into production on the Web – an application that means better customer communication and better customer service.

Founded On Quality Of Service

Harrassowitz, a bookseller and subscription agent company with customers around the world, was founded in 1872 in Leipzig by Otto Harrassowitz. Now located in Wiesbaden (near Frankfurt), Harrassowitz specializes in providing scientific works to libraries of academic and research institutes and universities. The company's customers include:

- Harvard (over 115 years),
- Yale,
- The University of Chicago,
- Oxford,
- Cambridge, and
- Many other distinguished universities and libraries.

Today, about 600,000 books are registered at Harrassowitz; and 30,000 are added every year. In addition, Harrassowitz has approximately 50,000 periodicals stored in its databases.

About 200 employees currently work in Wiesbaden, and Harrassowitz has branches in the United States and other countries. The company's total revenue is approximately 125 million euro.

Like all other companies that serve customers around the world, Harrassowitz deals with a number of complex issues every day. This is especially true when it comes to presenting detailed, specific descriptions of its offerings and handling the high number of transactions that customers trigger.

In line with Otto Harrassowitz's motto, "uncompromising quality of service," the company is committed to providing its customers and suppliers with the information they request as quickly and efficiently as possible. That means, of course, exploiting the latest IT technologies available.

The "Revolution" Continues

In the early 1970s, Harrassowitz made an important decision and installed its first DOS system on an S/360 processor. Over the years, it has kept its IT support system up-to-date. On the software side, VM/ESA 2.3, VSE/ESA 2.3, and CICS/VSE 2.3 are currently installed, together with VSAM, DB2, COBOL/VSE, and TCP/IP. As in many open environments, AIX and Windows NT devices are connected to the VM and VSE mainframe. In addition, Lotus Notes with its enormous connectivity potential has become an integral part of the total infrastructure.

On the hardware side, VM and VSE run on an IBM 9672-RA4, which has about 250GB of data on RAMAC II DASD (complex publication descriptions). The main system also has 130 Network Stations and 70 3270-based terminals attached to it.

Recently, Harrassowitz made a major step in its IT and customer support structure by investing in a Java - VSE/ESA solution for its Web site. Friedemann Weigel, director and IT manager, sees continuity with actions like this to the past:

"Five hundred years ago, Johannes Gutenberg started a revolution in Western civilization by introducing removable print letters. The Internet and its related IT technologies are our newest tools to further that innovation and contribute to the revolution."

Communication Is Everything!

Like many businesses supported by S/390, Harrassowitz is experiencing fundamental changes in the ways it communicates and interacts with its customers and publishers. Web browsers are, of course, a leading, new communication interface. Use of e-mail and other means of electronic data interchange also is rapidly expanding.

For service providers like Harrassowitz, efficient communication is vital to survival – nothing more and nothing less. Thus the company had a basic business need to expand its use of VSE and its data processing system to take advantage of the Internet.

Beginning in 1998, Friedemann Weigel and his team took several steps to exploit VSE capabilities to present Harrassowitz on the Internet. At the beginning, this involved:

- Presenting marketing information via VSE/ESA's support for a Web server,
- Offering customers a TN3270 path for searching the VSE host's publication database or for placing orders.

The initial solution provided fast global access to simple and powerful applications and high-quality data. But it did not yet offer a full graphical user interface, which is an important issue for the international library community.

CICS Gateway For Java – The Key

Exploiting the CICS Gateway for Java was the next step in linking customers to information at Harrassowitz. It was selected as the way to offer a VSE-based application with a graphical user interface that could be accessed by any Web browser. The design for this application specified that it had to run in real-time and perform well.

The CICS Gateway for Java is part of a three-tier solution and is located on the middle tier, on a Web server. Currently, the Gateway can be downloaded at no cost from the Internet. In the near future, it will be provided through the CICS Transaction Server for VSE/ESA.

In Harrassowitz's implementation, an HTML document calls a Java applet. The applet calls the CICS Gateway for Java, which makes a connection to CICS/VSE and triggers a CICS transaction. Data is exchanged via a normal CICS Communication Area.

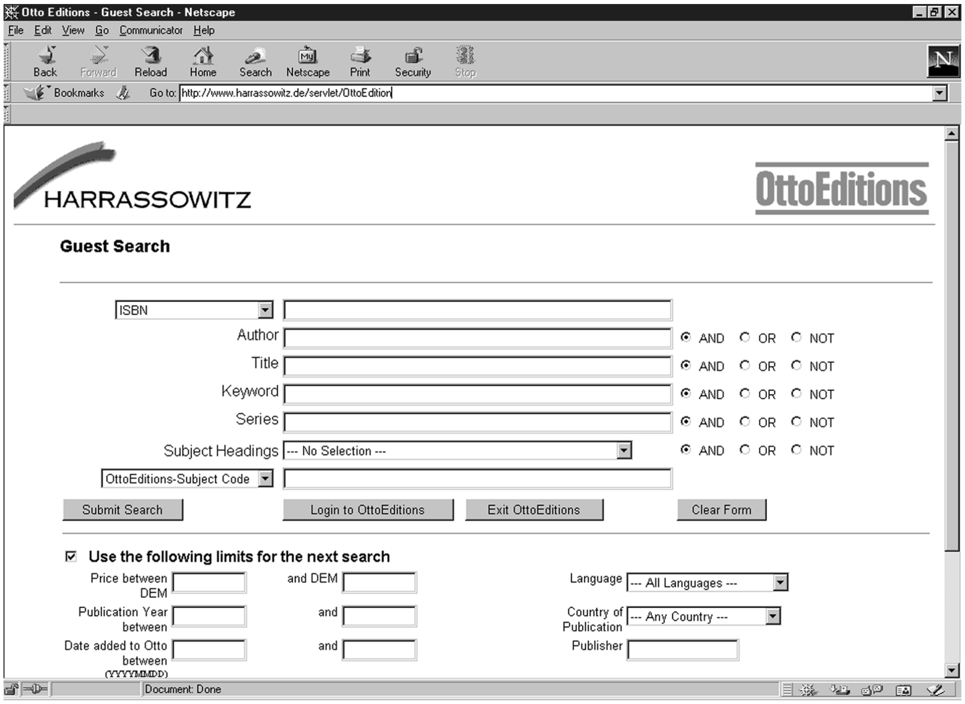
The CICS transaction on the VSE host is not aware of being called from the Internet. It only sees a non-3270 application using the COMMAREA technique for external communication.

Overall Costs And Benefits

To develop the required application, 60% of Harrassowitz's effort was related to host programming, 30% was for Java, and 10% involved presentation. The physical connection and Java part was implemented with help of Infologica GmbH, an IBM Business Partner located in Stuttgart.

The solution's test period ended this Autumn, and its first phase went into production in November. The total implementation effort was less than one-third of a "person year."

From Harrassowitz's point of view, the benefits to the Internet solution far outweigh any possible risks. Besides using the Web and new Web-based development tools, it takes advantage of existing hardware, software, staff experience to get the job done.



Otto Editions - Guest Search - Netscape

Go to: http://www.harrassowitz.de/servlet/OttoEdition

Guest Search

ISBN				
Author		● AND	○ OR	○ NOT
Title		● AND	○ OR	○ NOT
Keyword		● AND	○ OR	○ NOT
Series		● AND	○ OR	○ NOT
Subject Headings	--- No Selection ---	● AND	○ OR	○ NOT
OttoEditions-Subject Code				

☑ Use the following limits for the next search

Price between DEM		and DEM		Language	--- All Languages ---
Publication Year between		and		Country of Publication	--- Any Country ---
Date added to Otto between (YYYYMMDD)		and		Publisher	

Document: Done

Figure 25. OttoEditions, a New Web Interface Search and Order System for Monographs and Music Scores

Fundamental Value

Harrassowitz's new solution greatly eases customer communication and service, especially for its market in the USA. Presentation online is not purely 3270 or lifted or enhanced. It's a real browser view, without any compromise. Processing and data retrieval, however, take place on the host environment and provide the Internet application with high performance and availability. On the way to e-business, Harrassowitz has gone beyond a "simple" state, and further developments are on the way.

Based on the way German companies are now relating to e-business and acknowledging its value, Friedemann Weigel concludes, *"We simply have to exploit the Web, and we're doing that with VSE/ESA. Today's VSE/ESA offers a great variety of connectors to the Web, and in doing so, it has again shown its fundamental value to Harrassowitz and our IT environment."*

Additional Information

For more information about Harrassowitz, please visit:

http://www.harrassowitz.de/

The home pages for VSE/ESA and CICS are:

http://www.ibm.com/s390/vse/
http://www.ibm.com/software/ts/cics/

Infologica GmbH has its home page (in German and English) at:

http://www.infologica.com/

J&C Migrations Helps Danville Metal Quickly Move From RPG To COBOL

Author

Judy Tymon
J&C Migrations, Pty
jmtconsult@mediaone.net

J&C Migrations is a consulting and service company that provides IT solutions to S/390 customers throughout the world. Its primary service offering, RPG II to COBOL translation service, has helped these customers maintain their investment in legacy applications and reduce the cost of routine maintenance. This article highlights a customer who recently used the JCMRPG translation service to move a legacy RPG application to COBOL running under VM, as part of Y2K preparations.

What To Do With "Legacy" Applications?

Danville Metal Stamping Co. Inc. located in Danville, Illinois, is a manufacturer of metal components for the aerospace and gas turbine industries. The company employs over 450 people and serves companies including Pratt & Whitney, General Electric, AlliedSignal, and Rolls-Royce/Allison. Like many companies, Danville Metal found itself having to make a decision about legacy applications, especially those written in RPG II. Many of these applications are crucial to the business, and Danville Metal depends on them to run smoothly and to operate efficiently on a daily basis. Most of the programs were developed as far back as 1975, however; and the company was finding it difficult to maintain these applications alongside newer ones written in COBOL and CSP.

At the beginning of 1998, Danville Metal decided to make the transition from RPG II to COBOL for several reasons, many of which had to do with the need to make their system Y2K compliant. Another factor in the decision was IBM's announcement that it would no longer support Interactive File Sharing (IFS) on VM.

The company decided to migrate their RPG II code to DB2, COBOL for MVS and VM, and the Display Management System. A reason for this decision was that translating from RPG II to COBOL would allow for replacement of RPG application interface calls to the database. Screen panel calls also could use the Display Management System. About 18,000 lines in 90 programs were targeted for conversion. They included General Ledger, Payroll and Tax Withholding, and Sales and Manufacturing Cost Systems. These applications are running on a P/390.

A Nightmare

Before hearing about the JCMRPG translation service from J&C Migrations, Danville Metal had decided to work with a consultant, who converted the most critical applications. Unfortunately, the generated COBOL program logic was not functionally equivalent to the RPG II, and the differences were such that Danville Metal's IT staff could not recognize their original logic or get the programs to work in the target environment. The financial cost of this failed attempt at conversion was not great; but since programs did not work at all, this money was wasted. Of course, the more critical loss was in precious time and effort.

The first attempt at migrating their code was – as Kay Kepling, the IT manager, puts it – *"a nightmare."* The code was essentially indecipherable and support from the vendor was almost non-existent. With the year-end deadline looming, Danville Metal knew that they needed to act quickly and that the new vendor would have to provide complete support.

The Solution

Danville Metal first learned of J&C Migrations from their IBM representative in July, 1999. Since their first attempt at conversion was unsuccessful – and very frustrating – they were anxious to try another solution that would not only work correctly but would help them meet their year-end deadline.

Kay Kepling first called J&C Migrations on July 7th. By July 29, after a successful Proof of Concept demonstration with two of Danville Metal's programs, she received management go-ahead to use the service to translate the remaining applications. From the time that Danville Metal sent their RPG code to J&C Migrations to the time that they received the translated code, only 2 days elapsed. The code produced was both clear and functionally equivalent. All that Danville's IT staff had to do was to take the resultant COBOL, upgrade the database and screen interaction, and test it to complete the process.

Using their RPG II to COBOL cross compiler, J&C Migrations was able to convert the original RPG II code to functionally equivalent COBOL and – using the COBOL compiler – to verify the new code would compile cleanly. Because Danville Metal's IT staff was able to see the original RPG II code as comments in the COBOL code, it was very easy to correlate the COBOL with the original logic.

Although the majority of J&C Migrations' customers have far larger applications, all customers receive the same quality support and service. *"Support was always available,"* states Kay Kepling. Regarding J & C Migrations, her only request was that *"more IBM customers learn about their service."*

Additional Information

The home page for Danville Metal Stamping Co., Inc. is:

http://www.danvillemetal.com/

To learn more about the services offered by J & C Migrations, access:

http://www.jcmigrations.com/

Speed Is Of The Essence For Sanacorp!

Authors

Erwin Zattler
Sanacorp AG
e.zattler@sanacorp.de

Hans Joachim Ebert
S/390 Technical Support VM/VSE
IBM Munich
eberthj@de.ibm.com

With headquarters in Planegg, Germany (near Munich), Sanacorp AG requires one thing from its data processing support environment – speed. That's because the company is responsible for delivering pharmaceutical items – including prescription medicines – to over 8,000 pharmacies, every single day of the year. Missing a delivery or being late is simply not an option. To ensure that it continues to serve its customers as efficiently and as quickly as possible, Sanacorp recently installed ten new S/390 Multiprise 2000® servers with VSE/ESA. And those systems are already making a positive difference.

550,000 Items Per Day

For over 75 years, Sanacorp Pharmahandel AG has done exactly what the German for "Pharmahandel" means. Its main business is distributing medicines and pharmaceutical items. Today, Sanacorp has 14 branches throughout Germany and serves about 8,200 pharmacies. It has 2,000 employees and an annual revenue of approximately two billion euro.

Sanacorp contacts its customers three times per day to take orders. About 95% of all calls are made via special terminals, and 5% are via phone. In response, Sanacorp also makes three scheduled deliveries per day. On average, it handles 25,000 orders and 550,000 total items each and every day of the year. Sanacorp has about 80,000 in-stock items and about 200,000 listed items.

VSE At The Heart Of The Business

Sanacorp has long exploited IBM solutions, starting in the early 1970s with S/360 processors. From those beginnings, its IT environment has successively migrated to IBM 4300, 9370, and 9221 systems. By 1994, 12 systems served the company's 14 branches.

Sanacorp's warehousing system is mainly based on VSE/ESA and is very complex, handling everything from order entry to delivery logistics. This complexity comes from the need for:

- **Highest availability** – Sanacorp has two strong ordering peaks per day when the system needs to be 100% available. One is at noon, and one is in the afternoon. Each lasts about 90 minutes.
- **Optimal system performance** – This includes support for high-speed data entry.

- **"Horizontal cooperative processing"** – All branches are connected via 2MB network links. Thus if a requested item is not on stock at one branch, an automatic lookup in connected branches takes place.
- **"Vertical cooperative processing"** – Order delivery is highly automated (through robots, for example) and controlled by processors with UNIX or AIX. This "UNIX world" receives the data it needs to operate from VSE/ESA.

Searching For The Optimal System

In 1991, Sanacorp AG felt a strong uncertainty about its IT future, since rumors were circulating that VSE was "dead." As a precaution, it decided to begin implementing a basic client/server solution. Some applications thus were moved to AIX systems. The scare proved to be false, however. IBM continued to make new releases of VSE/ESA regularly available, and Sanacorp chose to keep the main part of the warehouse system on S/390 and VSE/ESA.

The IT manager for Sanacorp AG, Mr. Erwin Zattler, comments, *"Over the years, we've gradually evolved to a client/server support structure. S/390 and VSE/ESA have continued to play the lead role of the server because of this system's reliability, availability, performance, and cost effectiveness."*

"Part two" of searching for the optimal system began early in 1998, as Sanacorp analyzed how to get ready for the millennium. The IBM 9221 and IBM 9345 disks were four years old, and their release of VSE/ESA was not Y2K ready. Some different migration models were discussed, including:

- A one-to-one migration from IBM 9221 to IBM Multiprise 2000 servers,
- A consolidation to four, three, two or one server.

By mid-1998, the decision was made to migrate from the twelve IBM 9221 processors to ten IBM Multiprise 2000 servers, models 203 to 207. Which model went into a branch depended on the volume of that branch's order income.

Rollout of the Multiprise servers started late in 1998 and was finished by July of this year. Only one system was installed later than originally planned.

Sanacorp's main challenge, of course, was to continue operating normally in all branches while the switch to the new systems was being implemented. And besides new servers and moving to a Y2K-enabled level of VSE/ESA 2.3, the IT team also changed the company's support environment by:

- Migrating from DOS PLI to PL/I VSE running with LE/VSE,
- Exploiting TCP/IP for VSE/ESA for file transfer, and
- Adapting all applications for the millennium and testing them.

In addition, Sanacorp also opted to use another network provider for the new environment.

Thus the total project was a giant step forward for Sanacorp. Each system was reviewed both for its functionality and from a performance point of view. After all, *speed* is of the essence to Sanacorp; and the IT team wanted from the onset to get the most out of the new Multiprise servers with their increased processor horsepower, greater storage capacity, and internal disks with read/write cache.

The project was a big challenge for the IT staff, but it was completed and within budget. Today, one of the Multiprise 207 models is installed at company headquarters in Planegg. It has two VSE/ESA 2.3 systems running as VM/ESA 2.3 guests. One guest serves as a normal branch system; the other runs headquarter applications.

Branch systems are clones of the VSE/ESA master system, which is maintained in Planegg. All of these servers run VSE/ESA natively. Updates are synchronized by the IT team in Planegg.

Numerous Benefits

Of course, Sanacorp did not take the risk of migrating to new systems or make a sizeable investment in new hardware and software "just for the fun of it." The careful analysis that Sanacorp and its IBM support team did before starting the project indicated that the company would enjoy numerous benefits upon completion. And the results have proved that to be true. Today at Sanacorp:

1. All IT systems are Y2K compliant.
2. The average internal response time for online applications is about 50% better than before. For mission-critical transactions, it is in the range of 0,1 to 0,25 seconds, depending on processor speed.
3. Batch jobs are running much faster. This is based on internal disk cache and processor speed. Some jobs are running 300% faster than before.
4. The granularity of the Multiprise 2000 server opens the possibility to move smoothly an upgrade, in case the workload increases unexpectedly.
5. VSE/ESA 2.3 and CICS/VSE 2.3 are the base for an upgrade to VSE/ESA 2.4 and CICS Transaction Server for VSE/ESA 1.1, if needed.

Sanacorp AG is prepared for its IT future. Mr. Erwin Zattler remarks, *"For over 75 years, our business has been based on the newest technology. If we didn't stay ahead of our competitors, we'd be out of business in a short period of time. It's just that simple."*

Right now the IT staff at Sanacorp is "relaxing" a little bit by completing its final Y2K tests. Once that is finished, there are numerous ideas to consider for next year. These include putting graphical user interfaces on 3270 applications and implementing a new warehouse system to provide even better customer service. In addition, e-business and VSE/ESA will be investigated, since VSE/ESA also has proved to be a good server in this area.

Always ahead – this is what Sanacorp intends to be in the future, as it has been in the past. And staying ahead means speed, both in the customer service you provide and in your IT environment. Sanacorp AG is not slowing down one bit. In fact, it just geared up for even greater speed!

Additional Information

For more information about Sanacorp AG, you can visit the company's home page (in German) at:

http://www.sanacorp.de/

More Articles And News

Education, Technical Conference, And Teleconference Updates

Author

Julie Liesenfelt
VM/ESA & VSE/ESA S/390 Systems Center
juliev@us.ibm.com

Recent Course Offerings For VSE/ESA

Recently, IBM Learning Services announced a new partnership with *Intelli*Ware Systems, Inc. and is offering five new classes that focus on VSE/ESA, TCP/IP for VSE/ESA, and CICS Transaction Server for VSE/ESA. All of these courses will be hosted by IBM Learning Services and delivered by *Intelli*Ware Systems, Inc. in the United States.

For enrollments, schedules, or more information:

- Call **1-800-IBM-TEACH** (426-8322)
- Visit **http://www.ibm.com/services/learning/us/**

The descriptions, schedules are listed below for:

TCP/IP for VSE/ESA Under the Covers
VSE/ESA Under the Covers
CICS Transaction Server for VSE/ESA Migration
CICS TS for VSE/ESA Tuning & Problem Determination
CICS TS for VSE/ESA Under the Covers

TCP/IP For VSE/ESA Under The Covers (A3750)

Native TCP/IP support is now available for VSE; and with this course, the VSE professional can get the most out of this important new technology. You will learn the fundamentals of TCP/IP and the Internet, and then, specifically the implementation in the VSE environment. Through a series of lectures and labs, you will gain valuable experience working with TCP/IP functions and applications. Hands-on workshops with skilled instructors means you will learn the skills to ensure success in your networking environment.

Who Should Take This Course: Knowledgeable VSE system programmers and system planners responsible for, or directly involved in, integrating their VSE system into a TCP/IP network using TCP/IP for VSE.

What You Are Taught

- Explain the fundamental concepts and terminology of TCP/IP
- Install and configure the TCP/IP for VSE product
- Implement and use TCP/IP application functions such as TELNET, FTP, LPR/LPD and NFS
- Create web serving applications using TCP/IP programming interfaces
- Diagnose and tune a TCP/IP for VSE system

Topics Include

- TCP/IP and Internet fundamentals
- Installation, configuration, and operation
- Using TCP/IP clients
- Sockets, HTML and CGI programming and use
- TCP/IP security
- Performance tuning
- Problem determination
- Telnet, FTP, LPR/LPD and NFS configuration and use
- Web serving implementation
- Online labs

Prerequisites: Experience installing, tailoring, and maintaining VSE and CICS systems. General knowledge in the use of personal computers and Windows operating system.

Schedule at print time: 4.5 days, 2/28/00, Houston, Texas

VSE/ESA Under The Covers (A3713)

This course offers a comprehensive training workshop focused on building the skills VSE systems professionals need to manage their VSE/ESA environment. You will come away from this workshop with the detailed technical understanding and experience critical to maintaining, supporting and tuning your VSE/ESA system. The workshop features hands-on lab exercises that reinforce the wealth of knowledge you will gain from the lecture portions of the class.

Who Should Take This Course

VSE system programmers and system planners requiring a more in-depth understanding of VSE and its subsystems so that they can more rapidly respond to problems that affect system performance and availability.

What You Are Taught

- Identify the architecture of key VSE components
- Diagnose and resolve VSE system and subsystem problems
- Use VSE debugging facilities
- Improve the overall performance of VSE systems
- Program 31-bit applications
- Implement data spaces and virtual disks

Topics Include

- S/390 architecture
- VSE/ESA supervisor and internals
- 31-bit programming
- Data spaces and virtual disks
- VSE/ESA debugging tools
- VSE/ESA performance and tuning
- VSE/POWER structure and functions
- VSE/VTAM and CICS/VSE problem determination
- VSE/ESA enhancements
- VSE/ESA control block flow
- VSE/ESA I/O handling
- VSE/ESA error recovery

- VSE/ESA storage management
- VSE/ESA dispatcher/task selection
- Online Labs

Prerequisites: Six months' experience installing, tailoring, and maintaining VSE systems.

Schedule at print time: 4 days, 2/25/00, Phoenix, Arizona

CICS Transaction Server For VSE/ESA Migration (A3731)

This course prepares you for migration to the new CICS Transaction Server (CICS TS) for VSE/ESA Version 1.1. Through a series of lectures and labs, this course identifies the changes in CICS TS that impact the migration from CICS/VSE and CICS/VS systems. It highlights the associated changes in the areas of CICS TS installation, operation, resource definition, security, customization and programming interfaces. This course provides information that you can use to develop a customized migration plan to ease your transition to CICS TS.

Who Should Take This Course: Knowledgeable CICS/VSE or CICS/VS system programmers and system planners responsible for, or directly involved in, migrating their current systems to CICS TS for VSE/ESA Version 1.1.

What You Are Taught

- Migrate a CICS/VSE or CICS/VS system to CICS TS for VSE/ESA 1.1
- Identify and implement changes to key functional areas that affect:
  - Installation and operations
  - System and resource definition
  - Security
  - Application, system, and exit programming interfaces
  - Statistics and monitoring
  - Set up coexistence environments for CICS/VSE

Topics Include

- Introduction to CICS Transaction Server for VSE/ESA
- Migration planning
- Installation of VSE/ESA 2.4 and CICS TS
- Related product considerations
- Resource definition
- Security implementation
- Application programming compatibility
- Customization changes
- Migration tools
- Monitoring and statistics
- Traces and dumps
- CICS/VSE coexistence scenarios
- Vendor product considerations

Prerequisites: Six months' experience installing, tailoring, and maintaining a CICS/VSE or CICS/VS 1.7 system. Introductory knowledge of CICS TS for VSE/ESA can be obtained by reading the *CICS General Information manual* and *CICS Release Guide*.

Schedule at print time: 3 days, 3/16/00, Phoenix, Arizona

CICS Transaction Server For VSE/ESA Tuning & Problem Determination (A3732)

This course helps you understand how to do performance tuning and problem determination in a CICS Transaction Server (CICS TS) for VSE/ESA Version 1.1 environment. Through a series of lectures and labs, this course identifies the changes in CICS TS that affect performance tuning and problem determination. Learn how various parameters, design concepts, CICS facilities operating systems facilities, and network facilities affect the performance of CICS TS systems in an installation. Learn how changes to CICS supplied transactions, traces, and dumps affect debugging of system and application-related problems.

Who Should Take This Course: CICS/VSE or CICS/VS system programmers and system planners responsible for, or directly involved in, performance tuning and debugging problems of CICS TS systems on VSE/ESA.

What You Are Taught

- Identify performance bottlenecks and parameters that affect performance
- Tune the system to maximize performance
- Create procedures for collecting statistics and monitoring data
- Describe the processing flow of a transaction under CICS
- List common information sources for transaction debugging such as messages, CICS manuals, and language-specific information
- Use debugging tools provided with CICS: Supplied Transactions, Traces, and Dumps
- Use appropriate information and tools to classify and resolve transaction abends, loops, and wait conditions
- Identify and recognize symptoms of storage-related problems
- Describe a methodology to reduce problem resolution time

Topics Include

- Overview of CICS TS
- Performance tuning methodology
- Changes in CICS TS affecting performance
- Task management and storage management changes
- Performance tuning parameters
- Implementing statistics and monitoring
- Using statistics for tuning
- Changes in CICS TS problem determination facilities
- Trace and dump management
- Debugging CICS TS transaction problems
- Debugging CICS TS system problems
- Performance measurement and tuning
- Debugging a CICS TS transaction dump
- Using the CICS TS analyzed dump
- Online labs

Prerequisites: Six months experience installing, tailoring, and maintaining a CICS/VSE or CICS/VS 1.7 system. Ability to code application programs using COBOL in a CICS environment. Knowledge of CICS supplied transactions such as CEMT and CEDF.

Schedule at print time: 3 days, 3/15/00, Washington, DC

CICS Transaction Server For VSE/ESA Under The Covers (A3730)

This course offers a comprehensive training workshop focused on building the skills VSE and CICS systems professionals need to manage their CICS Transaction Server for VSE/ESA environment. You will come away from this workshop with the detailed technical understanding and experience critical to maintaining, supporting and tuning your CICS TS system. This workshop features hands-on lab exercises that reinforce the wealth of knowledge you will gain from the lecture portions of the class.

Who Should Take This Course: VSE and CICS system programmers and system planners who need a more in-depth understanding of the basic structure of CICS TS so that they can more rapidly respond to problems that affect system performance and availability.

What You Are Taught

- Identify the key structural components of CICS TS
- Diagnose and resolve CICS TS system problems
- Use CICS TS debugging facilities
- Improve the overall performance of CICS TS systems
- Implement Shared Data Tables
- Learn new enhancements in CICS TS

Topics Include

- CICS TS enhancements
- CICS TS structure
- CICS TS domain structure
- Shared data tables implementation
- CICS TS performance and tuning
- Task management
- CICS TS storage management
- Subsystem storage protection
- Using CICS TS debugging tools
- Dump and trace management
- CICS TS system dump analysis
- Online labs

Prerequisites: Extensive experience installing, tailoring, and maintaining a CICS/VSE or CICS/VS 1.7 system.

Technical Conference In Orlando, Florida

For 2000, the VM/ESA and VSE/ESA Technical Conference in the United States will be back in Orlando, Florida. We will return to the Renaissance Orlando Resort on Sea Harbor Drive.

This 3 1/2 day conference will be held on May 31st - June 3rd. We will start Wednesday morning and run through Saturday noon. This is a different format due to the Memorial Holiday on Monday the 29th. You can plan to arrive early for a long Memorial Day weekend or stay over after the conference ends for additional classes. A special hotel rate of $129 (space is limited) is available for conference attendees and also valid for those who come early or stay longer in Orlando.

You can save $100 by registering by April 14 (registration fee of $1075). After April 14, the registration fee is $1175.

The conference is intended to help you:

- Discover how VM/ESA and VSE/ESA fit into IBM's e-business Application Framework
- See and hear the recipes of success from VM and VSE users in the areas of e-business solutions, connecting to other platforms, guest support, server consolidations, systems management and more
- Explore every aspect of VM/ESA V2R4 and VSE/ESA V2R4 through over 80 elective sessions covering the basics to advanced topics on e-business, processors and technology, performance, business intelligence, problem solving, application development, integration and future directions

To register, or for more information, you can visit our Web site at:

http://www.ibm.com/services/learning/conf/vmvse/

or call **800-IBM-TEACH** (426-8322).

We look forward to seeing you again in Orlando!

IBM VM/ESA And VSE/ESA Customer Conference Calls

This is the current list of upcoming VSE and VM Calls:

01/20/00	**VM**	Providing a Parallel Sysplex Test Environment Using VM/ESA
02/10/00	**VSE**	CICS TS for VSE/ESA Performance Hints and Tips
03/15/00	**VM & VSE**	Enterprise Storage Management with Tivoli Storage Manager
04/19/00	**VSE**	VSE/ESA V2.4 User Experiences
05/17/00	**VM & VSE**	Establishing a Data Warehouse Environment using DB2 for VSE and VM

Please note that this schedule and topics are subject to change, so please check the web site for the very latest information on this call series:

http://www.ibm.com/s390/events/

If you missed a 1999 call, you can listen to the audio replay over the Internet. Just go to the web site shown above, and look under "previously held conference calls" to see the IBM Enterprise Connection Presentation Library.

The use of this service requires that you check-in before entering the library. Thus you will have to register beforehand. We will post previously held calls up to 3 to 4 months.

Previously Held 1999 VSE and VM Calls:

11/17/99	**VSE**	Overview of the CICS Web Interface and 3270 Bridge
11/16/99	**VM**	Getting More From e-business with VM/ESA
10/27/99	**VM & VSE**	Data Management and Business Intelligence with VM/ESA and VSE/ESA
10/20/99	**VSE**	A Progress Report on VSE/ESA
09/22/99	**VM**	VM/ESA Release Update and Customer Experiences
09/15/99	**VSE**	Migrating to CICS Transaction Server for VSE/ESA
08/26/99	**VSE**	VSE/ESA V2.4 Security

Thanks for listening!

Presenting The Global Solutions Directory From PartnerWorld For Developers

Author

James Kingman
Solution Developer Marketing
jkingman@us.ibm.com

Finally, there's a single source for thousands of business solutions. The Global Solutions Directory is a comprehensive online catalog of applications, tools and services created by some of the industry's leading developers using IBM technologies. You can search this Web site free of charge – anytime, day or night – to find the ideal solutions for your business.

With over 36,000 entries, you're bound to find what you re looking for. Each listing gives you product descriptions, company information and even direct links to company Web sites and e-mail addresses. Free-form text searches help you evaluate and compare solutions. And you can search on an incredibly wide variety of parameters – industry, technology, language, solution type, country of availability and many other criteria.

Why wait another minute to find the solution you need? Just log on to the URL shown below, select a language, and start searching the Global Solutions Directory. It's a quick click from there to the perfect solution for your business.

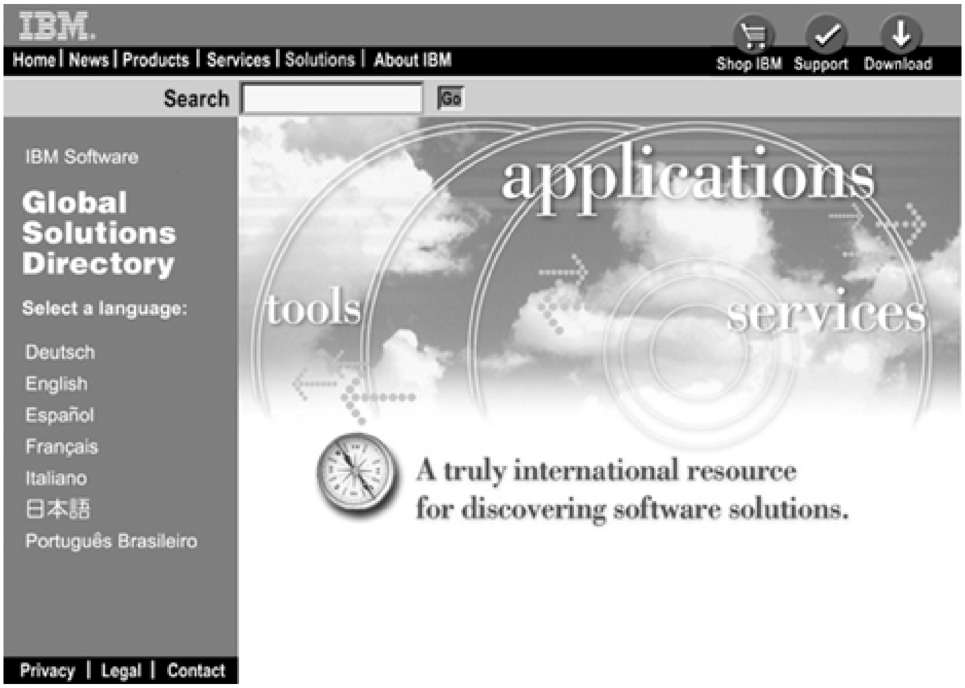

IBM.
Home | News | Products | Services | Solutions | About IBM
Shop IBM
Support
Download
Search
Go
IBM Software
Global Solutions Directory
Select a language:
Deutsch
English
Español
Français
Italiano
日本語
Português Brasileiro
applications
tools
services
A truly international resource for discovering software solutions.
Privacy | Legal | Contact

Figure 26. Global Solutions Directory – http://www.ibm.com/software/solutions/isv/. It's easier to find a needle in a haystack when the haystack comes with a multicriteria, searchable database!

If you have any questions, call **1 800 627-8363** in the U.S. & Canada; **+ 1 770 835-9902** worldwide.

Note: The listing of products in the Global Software Directory does not constitute an express or implied recommendation or endorsement by IBM of any particular product, service, company or technology, but is intended as an informational service.