Implementation of Popular Business Solutions with CICS Tools

IBM® CICS® Transaction Server (TS) is the IBM general-purpose transaction processing software for IBM z/OS® meeting the transaction-processing needs of both large and small enterprises. It builds on z/OS and IBM System z® facilities to provide high availability and scalability at a low cost per transaction. It also supports large transaction volumes with a fast and consistent response time.

Many organizations are running applications that are 10, 20, or more years old. These applications still provide the robust levels of service expected from CICS applications, but some might not meet the requirements of today's modern application paradigm. Today, applications allow for easy scalability and provide high availability. They have built-in redundancy, using various communication interfaces ranging from terminals and web browsers to web services and ATOM feeds.

The good news is that these well-proven applications do not have to be scrapped in favor of replacing them with entirely new applications. Instead, by taking advantage of newer features of z/OS and CICS TS, these applications are able to participate in the modern application environment.

IBM CICSPlex® System Manager (CICSPlex SM) provides for flexible management of the CICS infrastructure. You can add or remove additional CICS regions dynamically based on workload changes. With Business Events and System Events, you can emit data from your CICS applications without modifying application code. You can use the emitted event data as input to dashboards that track your business activities. Such tracking allows you to make real-time decisions to improve your business.

So what is the best way for your organization to use these newer CICS features? In today's environment of shrinking budgets and increasing work demands, it is more difficult to change your applications. IBM has tools that make this process simpler and faster, while reducing the risk associated with change.
This IBM Redpaper™ publication presents several scenarios involving a sample general-purpose application to demonstrate various ways to modernize a traditional CICS application using CICS tools. It should be of interest to various parties in your organization, including managers, architects, application developers, system administrators, and performance specialists.

Overview

CICS Transaction Server for z/OS is the premier high-volume transaction processing software available today. CICS uses application programming interfaces (APIs) to communicate directly with resources defined to it, for example, files, databases and system connections. An entire environment has developed around CICS, dedicated to supporting the life cycle of both CICS and its applications. The CICS tools are pivotal to this environment, and this paper describes certain methods used by the CICS tools that support it.

Introduction

This paper illustrates how an application can be converted from a single-region architecture to a multi-region architecture. This change makes possible a continuous 24x7 operation by making it available in several locations. Those locations can be individual CICS address spaces or even z/OS LPARs spread across a sysplex.

This paper also provides a high-level introduction into several of the facilities available in the CICS family of products that help you modernize your CICS environment. The scenarios in this paper involve the use of a general insurance application available with IBM GENAPP CB12 SupportPac. The general insurance application GENAPP is a CICS COBOL application that simulates transactions made by an insurance company to create and manage customer and policy data.

This paper uses a model of GENAPP running a single region with data stored in both Virtual Storage Access Method (VSAM) files and IBM DB2® tables. In the first part of the example modernization strategy, GENAPP is moved from a single-region model into a four-region model using CICSPlex SM. This strategy separates the application, presentation, and data access layers into unique address spaces. The strategy can be expanded to deliver differing interfaces into the application, although these topics are not covered in this paper.

The second part of the modernization strategy is to move the data out of VSAM into DB2. This operation might raise a concern that there is not enough time or budget to rewrite the application code to use SQL. This paper also explores CICS tools that help with modernization of your applications.

The paper consists of five sections, each describing an activity in which CICS tools help improve the modernization process.

▶ Section 1: Understanding the GENAPP application

This section shows how CICS Interdependency Analyzer is used to gain insight into the GENAPP application before moving it into a CICSPlex. CICS Interdependency Analyzer identifies resource dependencies, resource affinities, and key information needed when making this move.
Section 2: Building the cicsplex infrastructure
This section describes the steps required to build a CICSPlex. The section also shows how CICS Deployment Assistant helps provision CICS regions in a CICSPlex environment.

Section 3: Resource definition management
This section shows how CICSPlex SM Business Application Services (BAS) serves as a repository for CICS resource definitions. It assists with the migration of resource definitions from a CICS system definition (CSD) to a BAS repository.

Section 4: Moving VSAM data to DB2
This section shows how CICS VSAM Transparency can be used to migrate data from a VSAM file to DB2 table without changing the application programs to access the DB2 table.

Section 5: Performance considerations
This section examines CICS performance concerns surrounding a CICS environment. CICS Performance Analyzer is used to analyze specific performance issues with the GENAPP application.

Understanding the GENAPP application

You can enhance applications by modifying code, exposing sections as web services, or in this case, looking to move an application into a CICSPlex environment. However, you need to understand the impact these types of changes have on your applications.

This section examines the sample GENAPP application to obtain a better understanding of the application topology. We explore what resources make up the application, such as programs, transactions, files, tables, or temporary storage queues (TSQs), and how these resources interact with each other. Having a clear understanding of the application is critical as you look to modernize our CICS environment. Many CICS applications running today are many years old, and chances are, the documentation is out of date, if it even exists at all.

As migration from GENAPP into a CICSPlex proceeds, two things are necessary:

▶ Analysis of resource dependencies
▶ Analysis of resource affinities

IBM CICS Interdependency Analyzer (IA®) is a runtime tool from IBM that captures information from running CICS applications. It helps analyze resource interdependencies and affinities and assess the impact of application changes quickly and efficiently. As CICS IA gathers information from the CICS environment, the information is stored in a relational database. The CICS IA plug-in to the IBM CICS Explorer® provides a rich query interface for analyzing the collected data. CICS IA also provides batch-reporting facilities for analyzing the data.

CICS IA is used to evaluate data for both inter-transaction and transaction-system affinities, information that is essential when implementing dynamic workload balancing. By using information gathered by CICS IA, you can make well-informed decisions on how to split workloads efficiently and move applications across CICS regions accordingly. CICS IA also can generate affinity group information ready-formatted for input into CICSPlex System Manager.
Analyzing resource dependencies

CICS IA assists in understanding, in a controlled manner, the interrelationships between the shared common resources of applications and services. For example, to change the content or structure of a file, you must know which programs use this file because they need to be changed. CICS IA can identify the programs and the transactions that drive the programs.

CICS IA records the interdependencies between resources, such as files, programs, IBM WebSphere® message queues (WMQs), DB2 tables, and transactions. CICS IA records the connections by monitoring programming commands as they run in the CICS environment. CICS IA stores the collected data in a relational database. You can query the CICS IA data with your own SQL or supplied sample SQL. However, the preferred method is to use the powerful query interface of CICS IA using a plug-in to the CICS Explorer. You can use the IA plug-in to perform all of the numerous data analysis activities on the collected data without writing your own SQL.

In the GENAPP application, customer agents use the transaction SSC1 to inquire about and add customer information to or from the customer file. Using the CICS IA Explorer plug-in, you can find all of the resources involved with transaction SSC1.

To see this list of resources, under the Transactions tab in the IA perspective in the CICS Explorer, right-click the SSC1 transaction. Then, select the Uses Resources option from the context menu, as shown in Figure 1.

![Figure 1 Uses Resources query](image-url)
CICS Explorer displays the results under the **Uses** tab in the Resources used window, as shown in Figure 2.

From this display, you can see all the programs, temporary storage queues, maps, files, and other resources involved with transaction SSC1. By clicking any of these resources, CICS IA identifies additional information about that resource. For example, in Figure 2, temporary storage queue GENAWMQC is selected, and to the right, CICS IA shows that program LGICUS01 uses this temporary storage queue.

![Figure 2   Uses Resource results](image)

It is not unusual for transactions and programs to take separate flows based on the data they are given. In our GENAPP example, we know that transaction SSC1 performs Inquiry and Add functions. It is good to know the differences in program flows for a SSC1Inquiry as opposed to a SSC1Add.

With the CICS IA Command Flow feature, you can do exactly that. The Command Flow feature captures all CICS, DB2, IBM IMS™, and IBM WebSphere MQ commands in chronological order. It also captures a wide range of related information, including current and previous task control block (TCB) ID, response and reason codes, time of day, control section (CSECT) offset, and more. With Command Flow, you can see the resources used by a specific instance of a transaction. This information can help you understand the flow and structure of an umbrella transaction such as SSC1.
Figure 3 shows the differences in command flow in SSC1 for an Inquiry and an Add. From the side-by-side command flows, you can see that the inquiry function starts program LGICUS01, while the Add function starts program LGACUS01. Likewise, the inquiry function only reads file KSDSCUST, while the Add function writes to it.

From these operations, we gain a good understanding of all the components that comprise the application and how they interact. Next, we need to examine our application to determine whether it contains affinities that can affect our ability to route transactions dynamically in a CICSPlex environment.

**Analyzing resource affinities**

CICS applications can use programming techniques that require several transactions to be run in the same region, thus creating an inter-transaction affinity. Likewise, a transaction-system affinity occurs when programming techniques are used that require a specific transaction always to run in one particular region.

The ability to identify transaction affinities is useful in a dynamic routing environment. You need to know of any restrictions that prevent particular transactions from routing to particular application-owning regions (OARs), or conversely, require particular transactions to route to particular OARs.

The affinity-related functions of CICS IA help users of CICS dynamic routing determine whether any of the transactions in their CICS applications use programming techniques requiring them to be run in the same region. Doing so creates an inter-transaction affinity or an affinity in a particular region, thus creating a transaction-system affinity.
The affinity-related functions of CICS IA work in a way similar to the interdependency functions by collecting information about programs and transactions that issue specific commands. However, in this case, the objective is to detect affinities rather than interdependencies. CICS IA stores the collected affinity in a relational database just like dependency and command flow data.

The CICS IA plug-in provides query capabilities to analyze existing affinity groups and to help identify potential affinities, as shown in Figure 4.

![Figure 4 Potential affinities query](image)

CICS IA also contains batch-reporting features, including a batch affinity reporter. Figure 5 shows a section of the CICS IA affinity report that identifies the affinity on temporary storage queue GENACUSTNUM.

![Figure 5 Batch affinity report](image)

CICS IA Affinities Reporter also creates a file of basic transaction affinity groups for input to the CICS IA Builder. The CICS IA Builder runs as a batch job to build affinity transaction group definitions suitable for input to CICSPlex SM. The Builder uses the basic affinity transaction groups built by the Affinity Reporter as input. It then combines those groups and produces a file containing combined affinity transaction groups. CICSPlex SM requires a transaction identifier to be in one transaction group only. The Builder satisfies this requirement by combining groups that contain the same transaction identifier.
Building the cicsplex infrastructure

In this section, we describe the steps that are required to set up CICSPlex SM.

**CICSPlex and cicsplex**: CICSPlex (mixed case) is the name of the IBM product, and cicsplex (lowercase) designates a set of CICS regions in a plex.

**Preliminary setup**

A cicsplex is simply a set of CICS systems that CICSPlex SM manages. You might have a number of cicsplexes in your enterprise, but an individual CICS system can only belong to (be managed by) one CICSPlex.

To set up CICSPlex SM, you need two new CICS systems. In this scenario, you need to create two new address spaces, one for a CMAS and another for a Web User Interface (WUI) server.

A CMAS is a CICS system that manages a cicsplex. It is a dedicated system, so no other workloads run inside it. A WUI server is a CICS system that acts as a web server for the WUI into CICSPlex SM. If you use the CICS Explorer with CICS TS V4, the CICS Explorer uses the WUI to acquire information about your CICS systems and resources.

Because you are operating in a single region (Figure 6), you need to put in place the required address spaces to support CICSPlex SM on the host. Information about setting up the CMAS and WUI server address spaces can be found in the CICS Information Center under the Installation and Scenarios sections.

![Figure 6  Single CICS region architecture](image-url)
After you define the two address spaces, you can use the definitions in GENAPP CB12 SupportPac CB12, in CNTL data set of member CPSMDE2, to define the CICS and cicsplex regions. See Example 1.

**Example 1 CPSMDE2 member**

```
► CONTEXT REDMAS42
►   DEFINE CICSPLEX GNAPPLEX
►   *
► CONTEXT GNAPPLEX
►   *
► DEFINE REGION REDTOR42 -
►   APPLID REDTOR42 -
►   SYSID EJ04 -
►   CMASID REDMAS42
►   *
► DEFINE REGION REDDEV42 -
►   APPLID REDDEV42 -
►   SYSID EJ07 -
►   CMASID REDMAS42
►   *
► DEFINE REGION REDFOR42 -
►   APPLID REDFOR42 -
►   SYSID EJ06 -
►   CMASID REDMAS42
```

CONTEXT is the name you give to your CMAS when defining a cicsplex. After you establish the cicsplex definition, you then change to that name to CONTEXT to define the CICS regions to that cicsplex. When you put those definitions in place, you can start to expand your single CICS region.
From your single region architecture, you want to develop it into a multi-region cicsplex (and sysplex) enabled architecture, as shown in Figure 7.

To achieve the infrastructure shown in Figure 7, you can use CICS Deployment Assistant (CICS DA) to clone your existing region into the three individual regions. Then change the CSD resource definitions accordingly, but you can do that later when you want to clone the specific region types (TOR, AOR, and DOR). Because you want to migrate certain of your VSAM data to DB2, you can use CICS DA to discover what subsystems you have on your LPAR.
**Using CICS Deployment Assistant for discovery**

You can use CICS DA to obtain more information about the environment you are working with. To do so, follow these steps:

1. Connect to the WUI server. Because you already set up your cicsplex environment, you can connect to it using the CMCI interface. See the CICS Information Center for more information about how to set up the CICS management client interface (CMCI) in a WUI region.

2. In the Create CICS Deployment Assistant Project window, enter a project name in the Project name field, as shown in Figure 8. Browse to the target directory. In our example, project name is WinMVS2F. Click Discover MVS™ Images and then Next, as shown in Figure 8.

![Figure 8 Creating a CICS Deployment Assistant project](image-url)
The DA Projects window opens, showing a new project containing all of the subsystems on your MVS image. Note the entries for the cicsplex and WUI server. You are also shown a number of CICS regions, some of which are managed by CPSM, and others that are not.
Adding CICS regions to the cicsplex

Next, you must add the new CICS regions to the cicsplex:

1. In the DA Projects window, select **CICS Regions - Unmanaged** and ensure that there is a START policy in force for the desired region. You can check for a policy by specifying a command or batch data set for the job control language (JCL) member.

2. Click **Add to CICSPlex**. The Add CICS region to CICSPlex window opens, as shown in Figure 10. Here you add the selected region to the cicsplex.

3. In the Add CICS region to CICSPlex window, you can select an entry to prepare your cicsplex to include multiple types of regions (that is, TORs or AORs).

   You can also create groups. Creating these groups establishes a SYSGRP definition in CICSPlex SM so that you can add your region to the appropriate group for 24x7 availability. To create the groups, you can use CICS DA to clone a region type, such as one of the AORs.
Cloning CICS regions

To clone a region, CICS DA provides a cloning wizard. To use the wizard, take the following steps:

1. Activate the wizard. The Clone a CICS Region window opens. Supply values for the fields in this window. Click **Next**.

![Figure 11 Cloning a CICS region wizard](image-url)
2. The data set selection window opens, as shown in Figure 12. Using the check boxes in the Share column, confirm the required attributes for the new region and click Finish.

![Data set selection window](image)

Figure 12  Data set selection window

3. After you confirm the new data set names, CICS DA will submit a job to create them. You need to customize the new region, such as assigning SYSIN parameters, but much of the effort involved is now automated for you. From here, you can add the new CICS region to the cicsplex.
Resource definition management

In this section, we examine a possible solution for migrating our CICS resource definitions. For this exercise, we use CICS Configuration Manager for z/OS to convert CICS resources into CICSPlex SM Business Application Services (BAS) resources.

Converting CICS resources

Converting these resources allows you to populate your new multi-region environment with a consistent set of definitions that support the new architecture. You can convert resources by following these steps:

1. Using the CICS CM plug-in for the CICS Explorer, open the Groups window (see Figure 13). Select the CSD group that contains your resources. The resulting display tells you the groups that are currently associated with the GENAPP application and the artifacts are needed to support the new application.

2. If you select (right-click) one of the CSD groups and then select Show all resources, you can see what programs make up the application (see Figure 14). This list is the basis for part of a later migration package in CICS CM.
3. Store these resources in the CICSPlex SM data repository so that they are ready for deployment in your new regions. With CICS CM, storing resources is a simple operation because you only have to copy them into the relevant configuration. This functionality is currently not available in the CICS CM plug-in, so you must copy the sources through the Interactive System Productivity Facility (ISPF) interface.

By selecting all the resources in one of the groups (see Figure 15), you can then copy them into the target group in our CICSPlex SM data repository. This transfer then creates the resources in a BAS definition format and makes them available for deployment across the cicsplex.

![Program definition in a BAS Group](image)

*Figure 15  Program definition in a BAS Group*
Migrating CICS resource definitions

Now that you have your definitions in a BAS Group, you must move these definitions from one environment to another, for instance, from development to test. This move requires you to identify all the artifacts that constitute the application and populate a Change Package in CICS CM, as follows:

1. Open the ISPF interface to CICS CM, as shown in Figure 16:

   ![Figure 16 Change package selection](image)

   This display shows all the resources that are available in the CSD groups that you installed for the application. For the purposes of this document, assume that this list is comprehensive and shows all the resources required. All you need to do to select a resource is to place the action character P next to each entry. You can achieve this selection quickly by issuing the P command from the command line.
2. After the resources are ready to be packaged, you can also define their additional migration rule sets and transform rules, as shown in Figure 17.

![Figure 17 Migration scheme details](image)

This migration scheme takes resources from the CSD file referenced by the REDDVA42 CICS region. It then moves them across to the CSD file referenced by the REDDVB42 CICS region, as shown in Figure 18. We also specify a transformation rule here:

![Figure 18 Transformation rule details](image)

3. In this transformation rule, when you encounter a file resource with the name of KSDSCUST, you change the data set name from DEVELOP to PRODUCT. Changing the name correctly identify it after migration.
Moving VSAM data to DB2

The GENAPP application uses Virtual Storage Access Method (VSAM) for a number of key data stores. Users have requested particular query capabilities against the KSDSCUST data, but because the data is in VSAM, they cannot perform the particular queries they want. Another solution is to move the KSDSCUST data to DB2. However, due to time and budget limitations, it is not feasible to rewrite all of the programs that access the KSDSCUST VSAM file.

Instead, CICS VSAM Transparency (VT) is used for the DB2 migration. This way, the existing application programs that access KSDSCUST do not have to be rewritten. Using CICS VT greatly reduces the amount of time and risk associated with the DB2 migration. Because there are no application programming changes, testing coding changes is not necessary. You only must test that the migrated data is in correct format and that our application programs can successfully access the DB2 data.

This section describes how CICS tools help speed the migration of VSAM data to DB2 in the following process:
1. Determining where the KSDSCUST file is used
2. Converting KSDSCUST file from VSAM to DB2
3. Testing the conversion

Determining where KSDSCUST file is used

In a previous section, we introduced CICS Interdependency as a tool for affinity analysis to help prepare for dynamic transaction routing. One of the main values of CICS IA is that it helps you understand your CICS applications. Because CICS IA captures the relationship between resources, you can use this tool to learn of the programs in our CICS environment that use KSDSCUST.

Using the CICS IA plug-in to the CICS Explorer, we can create a query to find all the programs that use the KSDSCUST file by transaction ID, as shown. See Figure 19.

![Image of the query interface](image_url)  
**Figure 19** KSDSCUST file usage query
Running the queries provides you with the list of programs and the associated transaction that are using the KSDSCUST file. Notice that the query also shows how each program is accessing the file (READ, STARTBR, REWRITE, and so on).

For the batch component of the application, scan the library containing all of our execution JCL to identify the batch programs using the KSDSCUST File.

You have now identified all of the programs (batch and CICS), batch jobs, and CICS transactions that use the KSDSCUST file. We use this information obtained from CICS IA to drive our CICS testing.

Converting KSDSCUST file from VSAM to DB2

By using CICS IA, we identified all of the locations in the CICS portion of GENAPP that use the KSDSCUST file. However, as stated before, there are not enough time and resources to rewrite all the programs that access the KSDSCUST file to use SQL. Instead, we use CICS VSAM Transparency to convert the KSDSCUST file from VSAM to DB2. Using CICS VT saves the time and effort of rewriting and testing application programs. We only have to test the data migration and data access.

What is CICS VSAM Transparency?

CICS VT moves KSDS and RRDS data sets to DB2 without having to rewrite your application programs. You preserve your investment in existing VSAM-based applications but also open the data to new uses using regular DB2 SQL calls. Using CICS VT results in the following benefits:

- Movement of critical VSAM data to DB2 with no program changes
- CICS VT functionality for both CICS and batch
- Integration of your data with new and existing DB2 applications
- Access to DB2 data 24x7
- Supported business intelligence and data analytics
- Fast route to DB2
- Automated VSAM-to-DB2 mapping
- Staged migration of individual VSAM files
CICS VT works by intercepting runtime calls to VSAM files from your application programs. For batch programs, this interception is performed using an MVS subsystem. In CICS, the interception is handled by a global user exit (GLUE). The call interception is known as the \textit{CICS VT run time component}, as depicted in Figure 20.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure20.png}
\caption{CICS VT run time}
\end{figure}

At run time, CICS VT uses two driver modules for each migrated VSAM data set:

- Data set Information Module: Defines the relationship between the VSAM record structure and the DB2 table it relates to
- Data Set Driver Module (DDM): contains the necessary SQL to access the DB2 table.

These modules are generated by standard CICS VT ISPF dialog functions. These functions are part of the \textit{CICS VT mapping component}.

The third component of CICS VT is the \textit{CICS VT data migration component}. This component is involved in all aspects of the initial conversion of your VSAM files to DB2 tables.
Moving KDSCUST file to DB2 using CICS VT

The KDSCUST VSAM file has a record length of 82 bytes, as represented by copybook LGCUST. See Figure 21.

```
01 LG-CUSTOMER-INFO.
   05 LG-CUST-NUMBER   PIC 9(10).
   05 LG-FIRST-NAME    PIC X(10).
   05 LG-LAST-NAME     PIC X(20).
   05 LG-DOB           PIC X(10).
   05 LG-HOUSE-NAME    PIC X(20).
   05 LG-HOUSE-NUM     PIC X(4).
   05 LG-POSTCODE      PIC X(4).
```

Figure 21   KDSCUST copybook
Using the auto-mapping feature of CICS VT, we can create a DB2 table with the structure shown in Figure 22.

```
CREATE TABLESPACE KSDSCUS
    IN CVTDB
    USING STOGROUP CVTSG
    PRIQTY 65862
    SECQTY 2
    SEGSIZE 4
    BUFFERPOOL BF0
;
COMMIT;

CREATE TABLE DMUSERS.KSDSCUS
    ( 
    LG_CUST_NUMBER CHAR(10) NOT NULL,
    LG_FIRST_NAME CHAR(10) NOT NULL,
    LG_LAST_NAME CHAR(20) NOT NULL,
    LG_MID  CHAR(10) NOT NULL,
    LG_HOUSE_NAME CHAR(20) NOT NULL,
    LG_HOUSE_NUM CHAR(4) NOT NULL,
    LG_POSTCODE CHAR(8) NOT NULL,
    PRIMARY KEY ( 
    LG_CUST_NUMBER
    )
    )
    IN CVTDB.KSDSCUS;
COMMIT;

CREATE UNIQUE INDEX DMUSERS.IX_KSDSCUS
    ON DMUSERS.KSDSCUS
    ( 
    LG_CUST_NUMBER ASC
    )
    USING STOGROUP CVTSG
    PRIQTY 12723
    CLUSTER
    BUFFERPOOL BF0
;
COMMIT;
```

**Figure 22** KSDSCUST table definition

You can see that the KSDSCUST file structure is simple. It does not contain `occurs` clauses or `redefines`. We also did not convert the date of birth field into a standard DB2 date column. For complex data structures where data manipulation is needed, CICS VT provides exit facilities for field level and record level re-engineering.

After CICS VT builds the DB2 table, we then use CICS VT and DB2 utilities to move the VSAM data to the new DB2 table. The data on the VSAM file is no longer needed.
Testing the conversion

To activate the runtime component of CICS VT in CICS, the appropriate definitions for the CICS VT runtime modules, along with DB2 definitions, are added to CICS. CICS VT is also activated for the KSDSCUST table. For batch, the DD statements for the KSDSCUST file are modified to point to the VT batch subsystem.

When CICS transactions or batch jobs access KSDSCUST, CICS VT intercepts the VSAM request and issues an SQL call to access the new customer DB2 table.

In the VSAM to DB2 data conversion, where programs are rewritten, the testing process typically includes these tasks:

- Testing data format
- Testing program code changes
- Testing data access

Because we do not have to change any application programs, we only need to test data formatting and data access.

**CICS VT:** For more details about CICS VSAM Transparency, see the CICS TS Information Center under CICS VSAM Transparency.

Performance considerations

In this section, we examine how the IBM CICS Performance Analyzer helps with performance management.

CICS Performance Analyzer overview

Performance management disciplines are key components of successful IT service delivery processes. Good performance tools turn data into information and provide your team with the understanding they need to assess the impact of change. Good performance also helps you spot trends that might lead to poor CICS performance. You then can take rapid action to minimize any downtime or performance degradation, if problems do occur. Performance tools also contribute to optimizing IT resource usage and help you meet today's service level agreements while supporting capacity-planning exercises to satisfy future demands.

IBM CICS Performance Analyzer (CICS PA) for z/OS is an offline performance analysis tool. It combines ease of use, flexibility, and the level of detail required to meet the performance challenges of businesses today. CICS PA is designed to accurately detail how your enterprise uses CICS resources and provides detailed reports and views of all aspects of CICS system and application performance while helping you collect and manage historical performance data.

CICS systems programmers and performance specialists can tailor these reports to access the critical data they need, quickly. Along with application developers, they can all benefit from the intuitive selection, tabular, and graphical views when using the CICS PA plug-in for the CICS Explorer. CICS PA helps you gain the insight you need to manage CICS systems effectively and enhance their function and efficiency.
CICS PA analysis programs use performance and accounting data written to IBM z/OS System Management Facilities (SMF) data sets to generate reports. CICS PA delivers over 180 standard reports providing analysis ranging from various transaction performance reports, to reporting for trend analysis and capacity planning. Using the reporting interface of CICS PA, you can easily customize and generate performance reports and extracts without the use of any specialized programming languages or having in-depth knowledge of the SMF data layout.

Resolving GENAPP performance problems

Customer agents see a significant slowdown with the GENAPP application when adding new customers to the system and performing customer inquiries. The following steps allow you to investigate this issue:

1. Under the Transactions tab of the CICS Explorer, right-click the SSC1 transaction (customer add and inquiry), and a menu opens. In this menu, select Performance history → Response time because customer agents have noticed response time issues. See Figure 23.

![Figure 23 Integration with CICS PA from CICS Explorer SM Perspective](image)
A response time bar chart generated from CICS PA data, is displayed. In this example, the chart shows that the response time is averaging more than 3 seconds for the selected intervals. For two intervals, the response time peaks above 30 seconds. Our service level agreement for this transaction is one second or less, so obviously something is drastically wrong with this transaction.

![Response time chart](image)

**Figure 24  Response time chart**

2. Right-click one of the time intervals, and select **Detail Breakdown** to create a transaction detail chart.

![Detail breakdown](image)

**Figure 25  Detail breakdown**

In this example, we see that transactions are using nearly 100% of the response time in suspend status (36.521965 response time, 36.521009 suspend time).
3. To get an understanding of what is causing the high suspend time, run a CICS PA Wait Analysis report using the ISPF interface of CICS PA, as shown in Figure 26.

![Figure 26  Wait Analysis report](image)

Transaction SSC1 is the traditional transaction using basic mapping support (BMS) screens to communicate with customer agents. In the example, it appears that we have two issues with transaction SSC1. The report shows that over 58% of suspend time is attributed to terminal input and over 41% of suspend time is attributed to first dispatch wait time.

Application programming standards dictate that all terminal-based CICS transactions run pseudo-conversationally. However, the high rate of wait for terminal input suggests that transaction SSC1 is not pseudo-conversationational. If we look back at the CICS Interdependency Analyzer command flow shown in Figure 3 on page 6, we can see program LGTESTC1 is not doing a return after the initial send map. LGTESTC1 is not pseudo-conversationenal and needs to be corrected.

Transaction SSC1 runs in a special transaction class (DFHTCL09). When SSC1 runs too slow, it causes backups on the transaction class, which reflects high first dispatch time waits.

We notice the issue from statistic alerts from CICS PA. Figure 27 shows the CICS PA Statistic Alert view from the CICS Explorer. Notice that we have multiple alerts for Maximum active transactions in class reached.

![Figure 27  Statistics alerts](image)

The alert for Maximum active transaction in class reached has a critical threshold limit of 10. We see two occurrences of the alert where the actual maximum is 15 and 21.
CICS PA provides many sample alerts, making it easy to implement alert reporting. Remember that CICS PA is a historical performance tool and many times is used with real-time monitors. If you have alerting active in your real-time monitor, you might want to set CICS PA alerts to a lower threshold. This way, you can catch a performance trend before it becomes a real problem and before your real-time monitor issues an alert. Ideally, the CICS PA statistic alert report is empty, indicating all is well in the system and entries are displayed only when issues arise.

CICS PA Transaction Profiling reports, such as shown in Figure 28, compare new transaction activity compared to baseline data. Transaction Profiling reports were used to measure the impact on response time for transaction SSC1 when the application development team made program LGTESTC1 run as pseudo-conversational.

As you can see from the Transaction Profiling report, response time for transaction SSC1 improves by 91.68 percent. The average suspend time drops by 91.72 percent.

CICS PA Transaction Profiling and several other reports, such as the alert features of CICS PA, are all useful for measuring performance as we move the GENAPP application to a CICSPlex environment.

### The team who wrote this paper

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

**Chris Rayns** is an IT Specialist and Project Leader at the ITSO Poughkeepsie Center in New York. Chris writes extensively on all areas of CICS TS and CICS TG. Before joining the ITSO, he worked in IBM Global Services in the United Kingdom as a CICS IT Specialist.

**Eric Higgins** is a Certified Technical Software Specialist for CICS Tools in the U.S and has over 25 years experience with CICS. This experience ranges from applications development to systems programming, and most recently, CICS Tools. He has been with IBM for seven years, with the last five years working as part of the US CICS tools technical pre-sales team. Eric routinely works with prospective clients to help them find ways to improve their CICS environment and become more efficient with the use of CICS tools.

**Peter Siddell** is a CICS Tools Technical Specialist based in the IBM Hursley laboratory in the UK. He has over 30 years of experience in IT with over 20 years spent on CICS. His areas of expertise include CICS systems programming and technical support for z/OS products. He delivers education on CICS and CICSPlex SM and presents regularly at worldwide conferences on the CICS Tools Portfolio.
Thanks to the following people for their contributions to this project:

Stewart Smith
IBM Hursley

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at: 
ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- Find us on Facebook:
  http://www.facebook.com/IBMRedbooks
- Follow us on Twitter:
  http://twitter.com/ibmredbooks
- Look for us on LinkedIn:
  http://www.linkedin.com/groups?home=&gid=2130806
- Explore new IBM Redbooks® publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
- Stay current on recent Redbooks publications with RSS Feeds:
  http://www.redbooks.ibm.com/rss.html
Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.
This document REDP-4824-00 was created or updated on January 25, 2012.

Send us your comments in one of the following ways:
  ▶ Use the online Contact us review Redbooks form found at:
    ibm.com/redbooks
  ▶ Send your comments in an email to:
    redbooks@us.ibm.com
  ▶ Mail your comments to:
    IBM Corporation, International Technical Support Organization
    Dept. HYTD Mail Station P099
    2455 South Road
    Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (© or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

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

CICS Explorer®  IBM®  Redbooks (logo) ™
CICSplex®  IMS™  System z®
CICS®  MVS™  WebSphere®
DB2®  Redbooks®  z/OS®
IA®  Redpaper™

The following terms are trademarks of other companies:

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