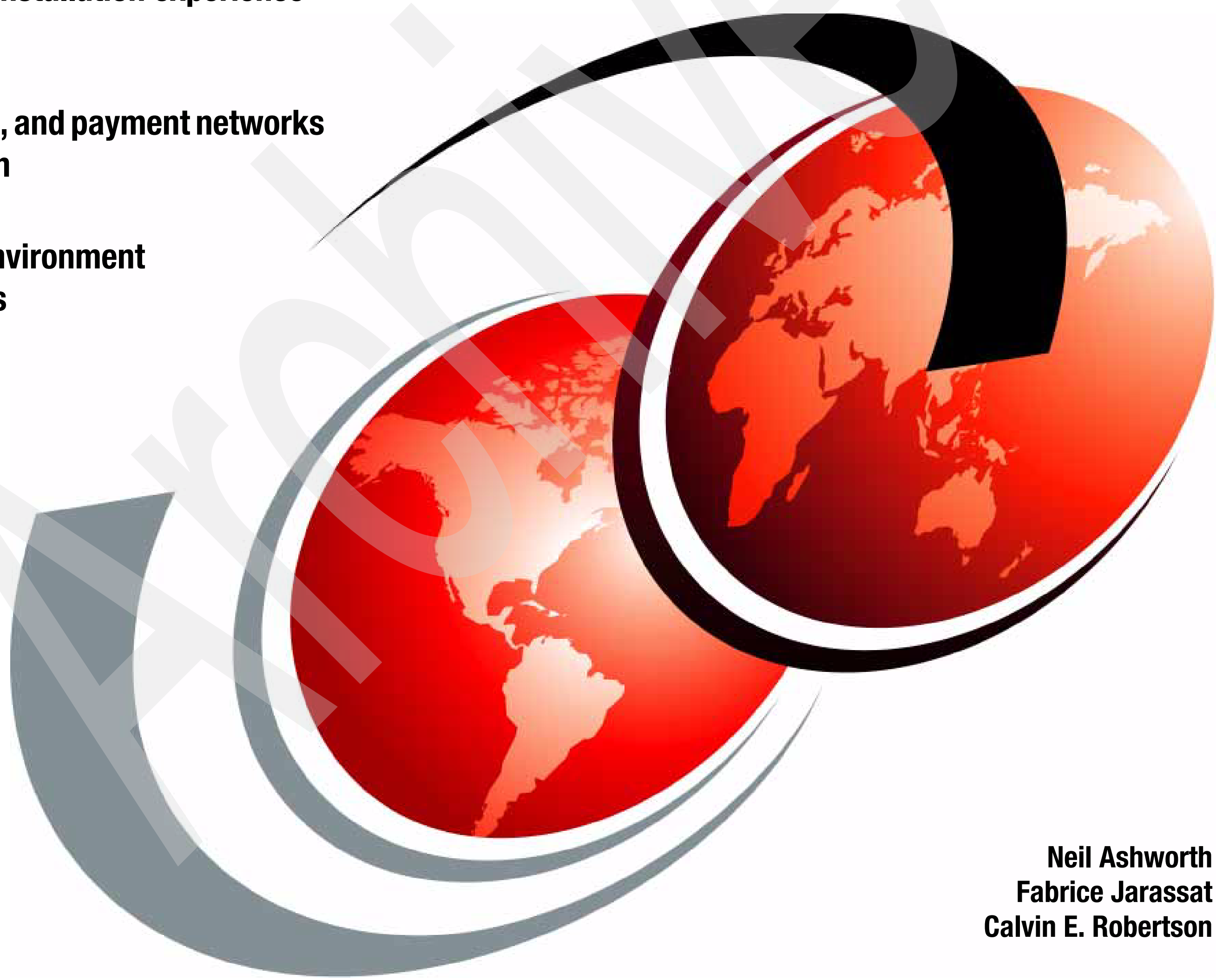
ACI Worldwide's BASE24-eps V6.2: A Supplement to SG24-7268

Practical installation experience

ATM, VISA, and payment networks integration

Sample environment definitions



Neil Ashworth
Fabrice Jarassat
Calvin E. Robertson

Redpaper

ibm.com/redbooks

International Technical Support Organization

ACI Worldwide's BASE24-eps V6.2: A Supplement to SG24-7268

December 2007

REDP-4338-00

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

First Edition (December 2007)
This edition applies to ACI Worldwide’s BASE24-eps version 06.2 product.

Contents

Notices

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

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

- CICS®
- CICSPlex®
- DB2®
- eServer™
- GDPS®
- HyperSwap™
- IBM®
- OMEGAMON®
- Parallel Sysplex®
- Rational®
- Redbooks®
- Redbooks (logo) ®
- System i™
- System p™
- System x™
- System z™
- Tivoli®
- TotalStorage®
- WebSphere®
- z/OS®

The following terms are trademarks of other companies:

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

Paragon, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

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

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

Preface

In this paper, we document our configuration findings from an implementation scenario that we performed from the IBM® Redbooks® publication, *A Guide to Using ACI Worldwide's BASE24-es on z/OS*, SG24-7268. We based this Redpaper on a project that was undertaken by the Product and Solutions Support Center (PSSC) in IBM Montpellier, France.

One of the considerations is that, since the publication date of the Redbooks publication, the product name changed from BASE24-es to BASE24-eps (enterprise payments system).

The Smart Bank project, hosted in Montpellier, implemented BASE24-eps on z/OS®, and then integrated it with both simulated and real ATM devices. A simulated VISA network authorized the payments against an independent host-core system from another independent software vendor (ISV). As a result of this work, we covered some new ground in configuring BASE24-eps on z/OS, which is what we cover in this Redpaper.

The team that wrote this paper

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

Neil Ashworth is a Certified IT Architect working in the Infrastructure solutions department at the PSSC in Montpellier, France. He has been with IBM since 2001. He has over 15 years of experience in the financial services sector, working on international banking client projects. He specializes in integration technologies and techniques around the IBM System z™ platform.

Fabrice Jarassat is an IBM Customer Information Control System (CICS®) and IBM CICSPlex® IT Specialist within the Infrastructure solutions department at the PSSC in Montpellier, France. He joined IBM in 2000. Before joining IBM, he worked at Mag-Info, which is a part of the French Galeries Lafayette group, responsible for managing the transaction system based on CICS. He has a degree in computing from Ecole Superieure Informatique Professionnel, Paris. His areas of expertise include CICS, IBM CICSPlex, CICS e-business technologies for z/OS, and CICS Tools. He is a Subject Matter Expert (SME) in CICS Web enablement.

Calvin E. Robertson is a Lead Engineer and Project Technical Lead with ACI Worldwide Ltd for the BASE24-eps Projects Group. He brings 30 years of IBM experience in software design, product development and project management, with an emphasis on ATM, Interchange, and banking applications. He has been with ACI Worldwide for 17 years (which includes 10 years at SDM International, which was acquired by ACI in 1999) and works with Domestic/United States and international customers worldwide. Calvin has a degree in Computer Programming, from the Electronic Computer Programming Institute (ECPI) in North Carolina, USA.

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

Alex Louwe-Kooijmans
International Technical Support Organization, Poughkeepsie Center

Claus Koefoed
IBM Denmark

Special thanks to the team of specialists from IBM Products and Solution Support Center (PSSC), Montpellier, France, to ACI Worldwide Ltd and the IBM International Technical Support Organization (ITSO) for the cooperation in producing this Redpaper. The PSSC is the largest support center in Europe for IBM eServer™, including IBM System z, IBM System p™, IBM System i™, IBM System x™, and IBM TotalStorage®. The Smart Bank showcase referred to in this Redpaper as the implementation environment for the BASE24-eps version 06.2 application from ACI Worldwide Ltd, is physically implemented on systems based in the PSSC, Montpellier, France.

Special thanks also to Wincor Nixdorf for providing the physical ATM equipment in Montpellier and to Paragon® for providing the injector tool.

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an e-mail 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

1

Smart Bank architecture with BASE24-eps

In this section, we describe:

- How BASE24-eps fits into the Smart Bank architecture
- The role of BASE24-eps
- How BASE24-eps interfaces with other resources

1.1 Solution architecture

Figure 1-1 illustrates the physical architecture of the configuration that was built in Montpellier for the Smart Bank demonstration. ACI Worldwide's BASE24-eps application coexists with other ISVs on z/OS LPARs for high availability and resiliency.

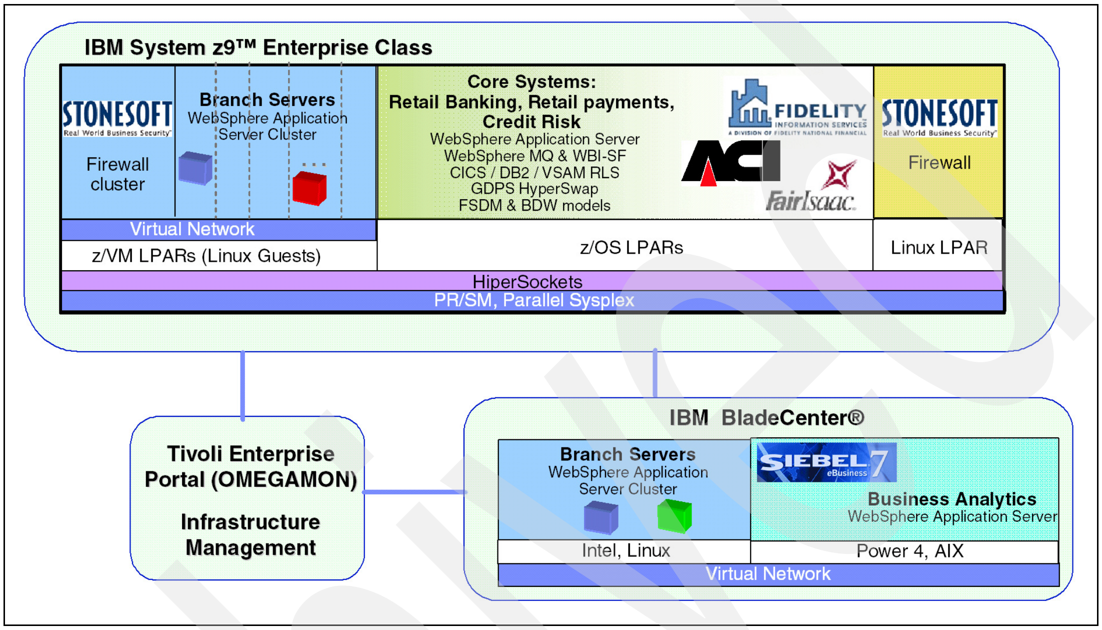


Figure 1-1 Smart Bank architecture

BASE24-eps is the enterprise payments system for the Smart Bank architecture, which manages all of the incoming retail payments that come from our Automated Teller Machine (ATM) network and from the VISA network. BASE24-eps logs the transactions and also integrates with Fidelity National Information Service's (FIS) Corebank V4.2 (Core System) to authorize the payments. The core system runs on CICS and processes these transactions in real-time, updating a DB2® database on z/OS where the DB2 is implemented in data sharing across two z/OS LPARs. The BASE24-eps system is deployed across the same two LPARs but in a separate CICSPlex environment.

IBM Tivoli® Monitoring V6.1 products, including Tivoli OMEGAMON® and Tivoli Monitoring agents, provide a heterogeneous system management capability of the runtime environments. Refer to *A Guide to Using ACI Worldwide's BASE24-es on z/OS*, SG24-7268 for more details. There was no specific Tivoli monitoring agent created for either BASE24-eps or the core system. In our solution, we used the z/OS, CICSm and WebSphere® MQ Omegamon agents to monitor these applications.

Figure 1-2 illustrates the role of BASE24-eps in more detail.

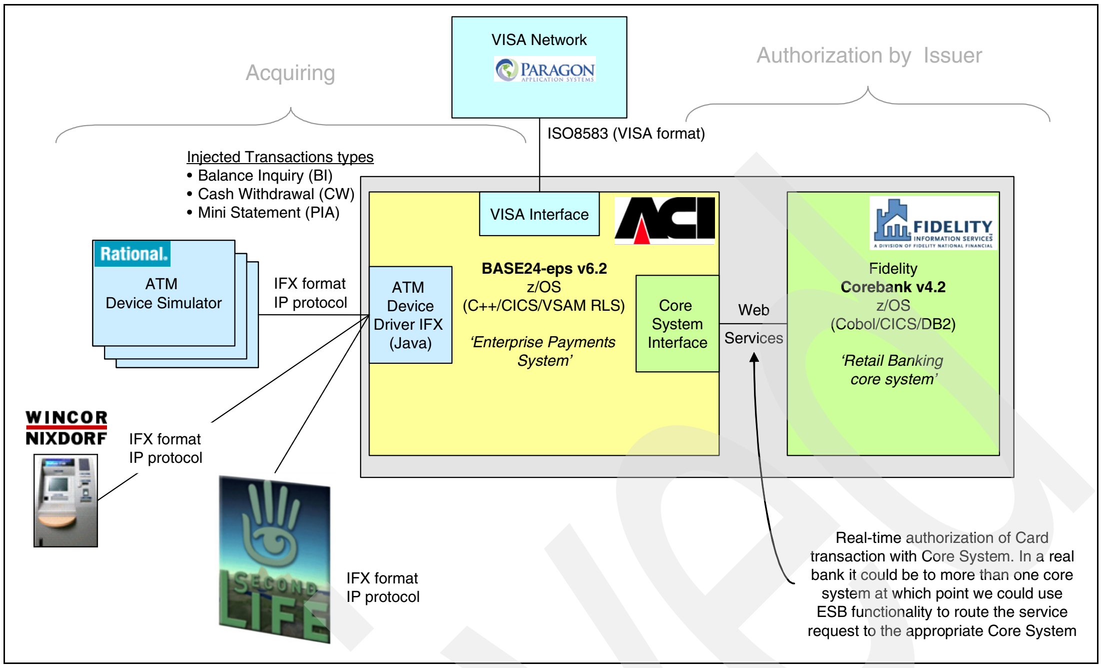


Figure 1-2 BASE24-eps integration to Smart Bank

In Figure 1-2, the channels on the left side represent the sources of our electronic payments. In our configuration, we used a real ATM provided by Wincor Nixdorf, which is operational against the BASE24-eps application. Since we could not acquire an entire ATM network, we simulate additional logical ATMs using Rational® Performance Tester. Partly for demonstration reasons, we also added a virtual ATM in Second Life within the Smart Bank area in IBM-8, IBM Advanced Client Technology Center hosted by Montpellier (France) and Poughkeepsie (NY, USA). Remote teams in different geographies can access this demonstration and Second Life, which provides a convenient collaboration tool for these teams to perform virtual ATM transactions. The connection from Second Life uses http rather than https and is hosted on the Second Life servers (not owned by this configuration), which is why, at the time of writing this Redpaper, we cannot recommend a virtual ATM in Second Life as a realistic proposition to a bank.

Figure 1-3 on page 4 shows the virtual ATM and one of the Tivoli Monitoring views that we created to monitor the CICS workload generated by the ATM and VISA channels. The three ATM channels all use the IFX protocol to talk to BASE24-eps using the IFX ATM Device driver from ACI Worldwide Ltd.

The other channel, which is at the top of the diagram, represents the VISA network that we simulated using FASTress from Paragon Application Systems, Inc. We injected VISA Base 1 ISO8583 messages into BASE24-eps. This interface then required the VISA Interface driver from ACI Worldwide Ltd.

The third key interface is the authorization link between BASE24-eps and the Core System, which required the Host Interface driver from BASE24-eps and a separate module that we created to request the authorization through Web Services.

Later, we discuss the Web Service interface. We also recorded the main configuration changes in Appendix C, “CICS Web Services configuration” on page 39.

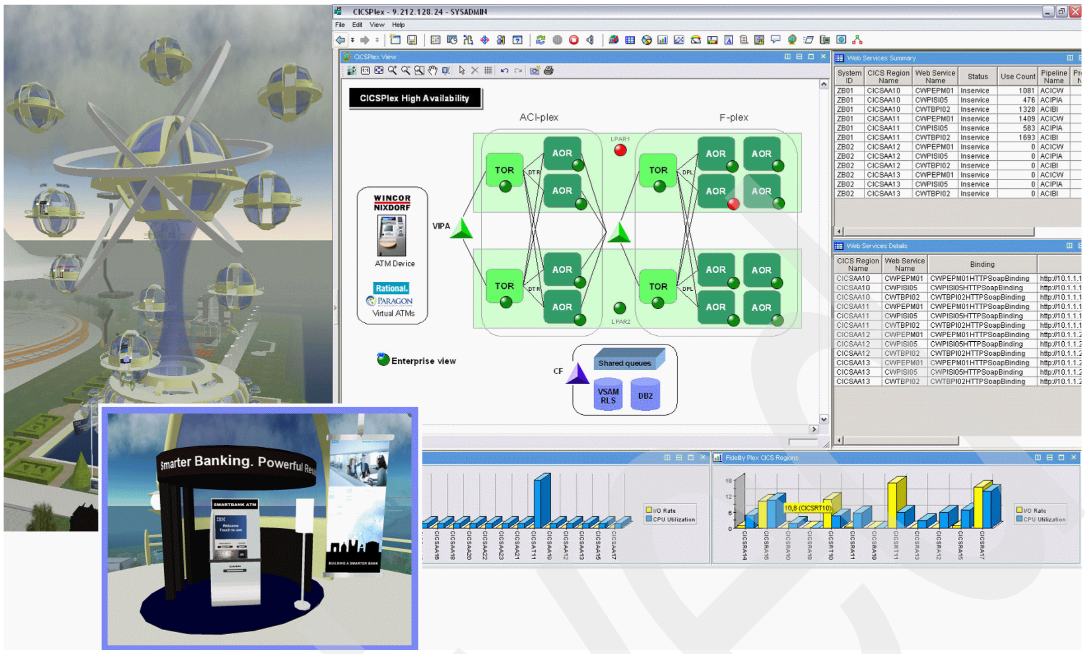


Figure 1-3 Virtual ATM linked to BASE24-eps and Core System

Another aspect to the integration picture, which we do not cover in this paper, is the near-real time update of payment data that flows through BASE24-eps to our Analytical Database on DB2 z/OS. BASE24-eps facilitates near-real time operational reporting and analysis.

In the next section, we look at the basic architecture choices that we made in setting up BASE24-eps for Smart Bank. When we started planning for high availability configurations of these integration channels, we realized that we needed to make some configuration changes and additions to the BASE24-eps implementation.

1.1.1 BASE24-eps application summary

BASE24-eps is a largely C++ application that is deployed on CICS on z/OS and accesses VSAM RLS (record level sharing) files for data and configuration. For queuing mechanisms, BASE24-eps can use the native CICS transient data queues (TDQs) or external WebSphere MQ queues. The system configuration is performed both by a graphical user interface (GUI) that is called the ACI Desktop. The ACI Desktop remotely connects to CICS using the XMLS transaction service to change VSAM RLS files directly using a set of CICS 3270 transactions (R3nn). See Appendix A, “CICS configuration” on page 21 and Appendix B, “ACI Desktop based configuration” on page 29 for more details on the configuration.

The queuing configuration mechanisms fundamentally update five key VSAM RLS files, which influence the way that BASE24-eps functions in the terminal-owning region (TOR) and the application-owning region (AOR) region.

Table 1-1 on page 5 summarizes the VSAM files, or sometimes called the BASE24-eps Message Delivery System configuration files, which is an extract from *A Guide to Using ACI Worldwide's BASE24-es on z/OS*, SG24-7268.

Table 1-1 Message Delivery System Files

VSAM RLS file name	R3nn CICS Tran	Used by	Description
SOCKRECS	R309, R310	TOR	**Socket Records file** One record for each configured endpoint that the IP Server associates with a unique BASE24-eps symbolic name. Here we define all the physical and virtual ATMs, plus the VISA stations.
TCPIPCFG	R301, R302	TOR	**TCP/IP Control file** Contains one record per BASE24-eps TCP/IP communications handler. The TCP/IP Control file: ► Indicates whether the handler is a client or server. ► Indicates whether we verify the endpoint IP address or not. ► Verifies the port numbers used by the handlers to listen for requests from the endpoints.
SDMF	R303, R304	TOR	**Static Destination Map File** Contains one record per CICS transaction in the BASE24-eps system, plus one record per SOCKRECS entry. Contains static information about an endpoint. Contains the configuration data for each BASE24-eps task. Transaction codes refer to either a CICS TRANSID or TDQ.
DDMF	R305, R306	TOR and AOR	**Dynamic Destination Map File** Contains one record per routable endpoint in the BASE24-eps system that supports asynchronous requests. Transaction codes refer to either a CICS TRANSID or TDQ.
SYDMF	R307, R308	AOR	**Synchronous Destination Map File** Contains one record per routable endpoint in the BASE24-eps system that supports synchronous requests. When a transaction must get a reply back from a service before a determination can be made to complete processing, a synchronous reply is needed. In our case this is the external “Authorization” request to our Core System.
DTRCTL	R311, R312	TOR & AOR	**Dynamic Transaction Routing Control** Used with workload balancing to indicate which AORs and TORs are available for the BASE24-eps system to use.

1.2 Key architectural choices for implementing BASE24-eps for Smart Bank

The following list is a summary of the choices that we made in building the architecture that hosts the ACI BASE24-eps application. We assume that you are familiar with the deployment of BASE24-eps on a z/OS system. For details or additional information, refer to *A Guide to Using ACI Worldwide's BASE24-es on z/OS*, SG24-7268.

The following list summarizes our choices in building the architecture that hosts the ACI BASE24-eps application:

- ATM protocol

  Currently, many ATM and point-of-sale (POS) networks use proprietary message formats. We wanted, if possible, to choose an industry standard to ease the potential integration and interoperability issues that our institution could face in the future. We also wanted our institution to benefit from the extensible nature of such a standard from a functional point-of-view. For this reason, we chose the Interactive Financial eXchange™ (IFX) protocol for the ATM/POS industry. IFX represents a mature and well-designed business message specification for the industry, and not just for ATM/POS messages.

  – IFX is an open standard and is both vendor and platform independent.
  – With our demonstration, we wanted to explore some of the new and potential solutions for our customers.

- VISA network interface

  To make our demonstration closer to a real environment, we injected a transaction workload that simulated payments coming from an external network. We chose the VISA network using the ISO8583 protocol because it is a very common message format that is found in many Financial Institutions. The VISA network is also the base of the internal payments format that BASE24-eps uses.

- Authorization

  We had a choice to authorize with BASE24-eps or with our Core System. One of the advantages of implementing our payments system onto z/OS was the proximity to our Core System. Consequently, for reasons of efficiency of operations and functionality, we decided to authorize using the Core System directly, which required a synchronous mechanism into the BASE24-eps transaction path that extends response times.

  We chose to use Web Services to facilitate the synchronous host authorization for the following reasons:

  – To show how BASE24-eps can integrate into a simple service-oriented architecture.
  – To reflect that often in Banks, there is more than one core system, often on different platforms. Also, a decoupled service-oriented interface is quite an attractive and efficient way to integrate, in the same way, to many different core systems. Perhaps placing some sort of routing in an enterprise service bus, for example to determine which core system should provide the service.
  – To show that a CICS system can participate in a service-oriented architecture, both as a consumer and as a provider of services.

- Workload management

  We chose to manage the CICS workload using the CICSPlex System Management (CPSM) capability (CPSM), which was already in place to manage the Smart Bank environment with the existing Core System CICSPlex environment. This gave us greater control and an easy implementation.

- TCP/IP client/server configuration of BASE24-eps

  In the original Redbooks publication, *A Guide to Using ACI Worldwide's BASE24-es on z/OS*, SG24-7268, two choices were indicated on how to configure BASE24-eps from a TCP/IP point-of-view, as a client or as a server. In our case, we use BASE24-eps as a TCP/IP client for the ATM network. We kept the same configuration for the VISA network.

  As an IP Server, BASE24-eps retains ownership of the sockets on which it accepts connections. BASE24-eps TCP/IP server binds to and accepts connections on a specific TCP port. To accept connections on more than one port, it is necessary to define additional copies of the server. We have, for example, two TCP/IP servers for the ATMs, one for the logical ATMs, and one for the physical.

- WebSphere MQ versus CICS transient data queues (TDQs) for messaging

  In the original Redbooks publication, *A Guide to Using ACI Worldwide's BASE24-es on z/OS*, SG24-7268, the authors discussed which asynchronous queuing mechanism to use. BASE24-eps can use either TDQs or WebSphere MQ.

  CICS TDQs is the default messaging mechanism with BASE24-eps because it is native to CICS, and we use TDQs for all the processes that invoke C++ modules in CICS.

  We use WebSphere MQ for the ATM workload only because the ATM device drivers are written in JAVA and not in C++. BASE24-eps runs the JAVA processes under UNIX® Systems Service on z/OS. To interact with the JAVA process, we use WebSphere MQ queues because they are external to CICS.

1.3 BASE24-eps for Smart Bank configuration

The following tables and diagrams illustrate the Smart Bank specific implementation of BASE24-eps. Our set up follows the suggested diagram structures described in *A Guide to Using ACI Worldwide's BASE24-es on z/OS*, SG24-7268, which you can refer to for more information.

We based our configuration on two z/OS logical partitions (LPARs), which initially hosted WebSphere z/OS (used for channel and service integration purposes) and the Core System. We added another CICSPlex to host BASE24-eps with the following TORs and AORs running on the same two LPARs.

Table 1-2 displays the logical partition definitions with BASE24-eps in Smart Bank.

Table 1-2 Logical Partition definitions with BASE24-eps in Smart Bank

LPAR	TOR	AOR	LPAR specific definitions	System wide
LPAR1	AT10	AA10 AA11 AA14 AA16 AA18 AA20 AA22	**VSAM RLS files** ► ATMVSAM.ES62.TCPIPCFG.BA01 ► ATMVSAM.ES62.SDMF.BA01 ► ATMVSAM.ES62.DDMF.BA01 **WebSphere MQ shared queues** ► AT10.IFXRQST ► AT10.ATMISO ► AT10.JDHRPLY ► AT10.IFXRPLY ► BT10.IFXRQST ► BT10.IFXRPLY **JAVA Device Handlers for IFX ATM** JDH2 process	**VSAM RLS files** ► ATMVSAM.ES62.SOCKRECS ► ATMVSAM.ES62.SYDMF ► ATMVSAM.ES62.DTRCTL
LPAR2	AT11	AA12 AA13 AA15 AA17 AA19 AA21 AA23	**VSAM RLS files** ► ATMVSAM.ES62.TCPIPCFG.BA02 ► ATMVSAM.ES62.SDMF.BA02 ► ATMVSAM.ES62.DDMF.BA02 **WebSphere MQ shared queues** ► AT11.IFXRQST ► AT11.ATMISO ► AT11.JDHRPLY ► AT11.IFXRPLY ► BT11.IFXRQST ► BT11.IFXRPLY **JAVA Device Handlers for IFX ATM** JDH3 process	

In Chapter 2, “Configuration changes for high availability” on page 15, we provide the reasons why some VSAM RLS files and some WebSphere MQ queues have LPAR-specific implementations.

In addition to the basic BASE24-eps TCP/IP handlers that BASE24-eps needs for the ACI Desktop, we defined four additional handlers to handle the ATM and the VISA workloads.

Table 1-3 displays the TCP/IP handlers we defined for the ATM and VISA workloads in BASE24-eps.

Table 1-3 TCP/IP handlers defined for the ATM and VISA workloads in BASE24-eps

Service name	Destination	CICS TRANSID	Local Port Number	Description
ATM-IFX	ATMIFX	AIFX	4012	ATM IFX TCP/IP handler listening at port 4012 on both LPARs. This server was defined for the logical ATM traffic injected by Rational Performance Tester. They do not have individual IP addresses. 200+ logical ATMs all have the same IP address.
ATM-IFX2	ATMIFX2	WIFX	4014	ATM IFX TCP/IP handler listening at port 4014 on both LPARs. This server was defined for the physical ATMs, for example, those ATMs that do have individual IP addresses.
VISA-STA-01	INTFVISA	TC40	9008	VISA TCP/IP handler listening at port 9008 from AT10 in LPAR1.
VISA-STA-02	INTFVISA	TC40	9008	VISA TCP/IP handler listening at port 9008 from AT11 in LPAR2

We explain the reasoning for our choices in Chapter 2, "Configuration changes for high availability" on page 15, although we started introducing some of the complexities because we were not connected to a real-world environment and because we configured the system for high availability.

Figure 1-4 shows the TCP/IP handlers generically depicted in the TOR at the top left.

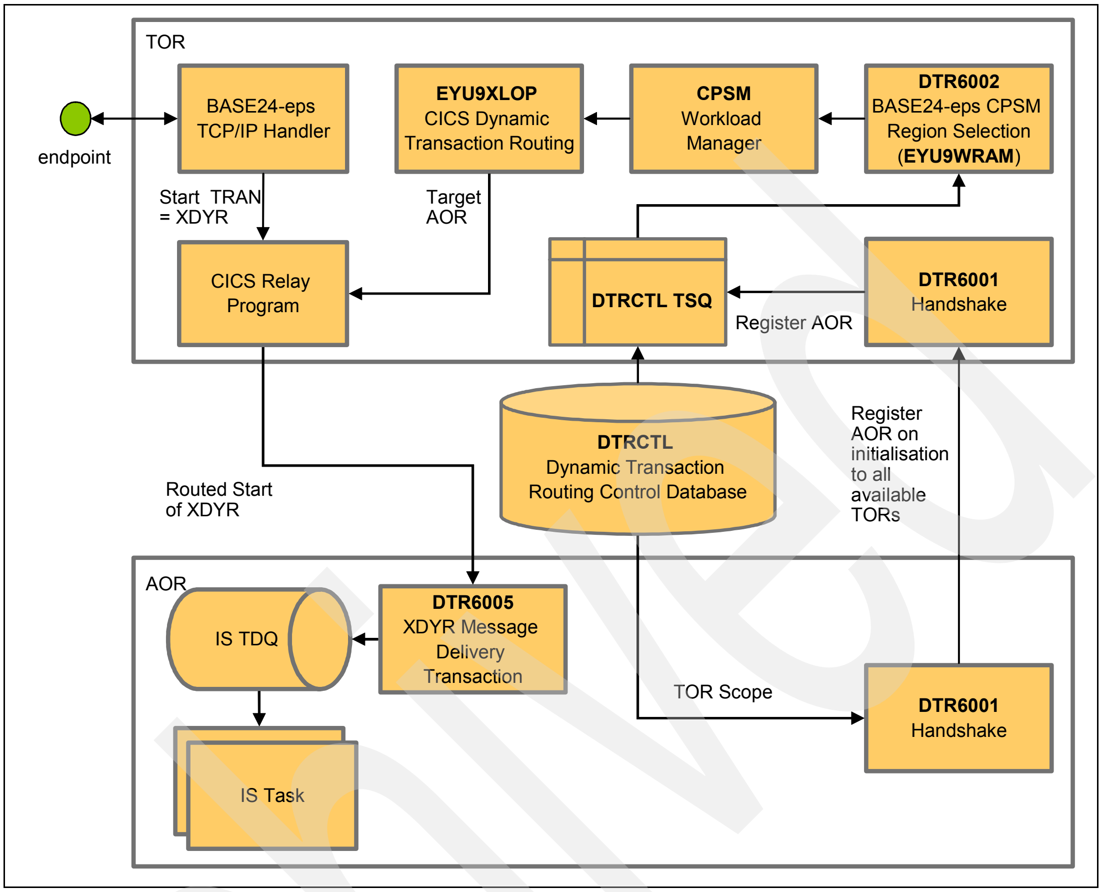


Figure 1-4 Workload management with CPSM

As discussed in *A Guide to Using ACI Worldwide's BASE24-es on z/OS*, SG24-7268, asynchronous messaging from the TOR to the AOR, along with the long-running nature of the Integrated Servers, cannot benefit from the standard CICS workload management mechanism; therefore, BASE24-eps provides a dynamic transaction routing program (DTR6002) that you can link-edit in the CICSPlex System Manager (CPSM) exit to work with CPSM in balancing the workload. An example of this is when:

- The TCP/IP handler receives the TC40 (or VISA) transaction.
- The TCP/IP handler invokes a CICS START of TRANSID XDYR if any of the indicated AORs CPSM and BASE24-eps are ready to receive it.
- The XDYR transaction then places the message on a TDQ called IS06 for that AOR.
- The long-running CICS transaction IS06 (specifically defined for VISA), which is constantly polling the IS06 queue, takes the transaction and initiates the authorization request with our Core System.
- The response is returned into the TC40 TDQ, which the TC40 TCP/IP handler retrieves.
- The message must return to the TC40 TCP/IP handler in the TOR that initiated the request. Figure 1-5 on page 11 illustrates this process.

This same process is used for all of the TDQ-based messaging transactions.

Figure 1-5 illustrates the VISA TCP/IP handler and integrated server interaction.

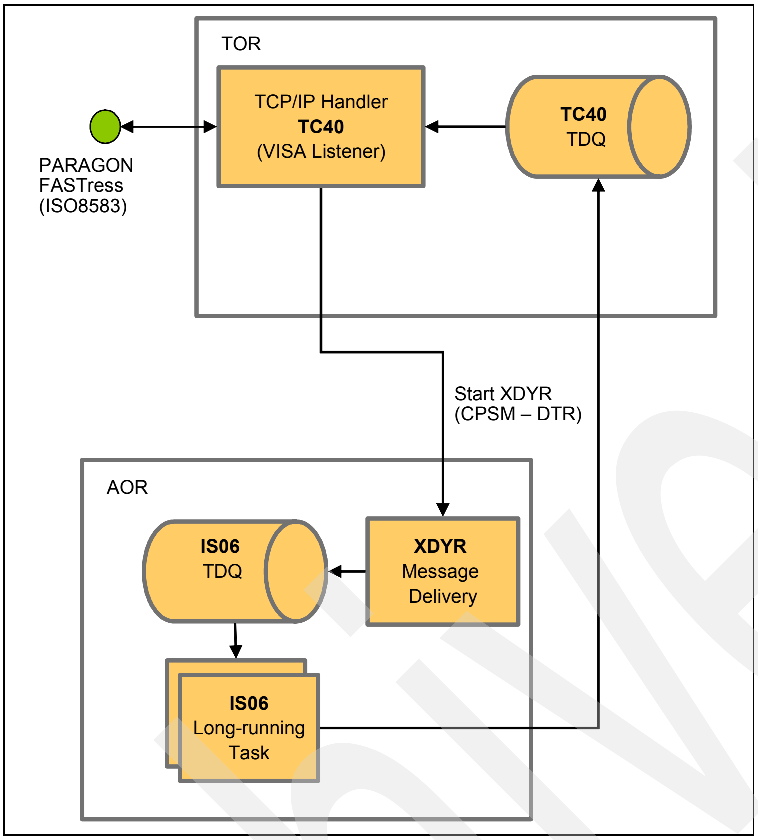


Figure 1-5 VISA TCP/IP handler and integrated server interaction

The mechanism to balance the ATM workload is different since we use WebSphere MQ queues instead of the Transient Data Queues, as shown in Figure 1-5. The ATM device driver for the IFX protocol is written in JAVA, and we use WebSphere MQ within BASE24-eps to interact with this process. We use the TDQ messaging because the device driver for VISA is written in C++, which runs under CICS with BASE24-eps.

Figure 1-6 on page 12 illustrates the WebSphere MQ transaction routing using the JAVA Device Handler (JDH) process that runs in UNIX System Services on z/OS. Table 1-2 on page 8 shows the two different JDH processes that we defined to run in each LPAR. One of the functions of the JDH process is to parse and convert the IFX XML into the internal ISO format that the BASE24-eps Integrated Server uses. We defined two queues for use between the JDH and the TOR for the messages in IFX format versus the ATM. The other two queues are used between the JDH and the AOR for the messages in the internal ISO format.

The Integrated Servers IS02 and IS04 are defined for the ATM workload and are both long-running tasks that listen to the shared WebSphere MQ queues AT10.ATMISO and AT11.ATMISO.

Figure 1-6 illustrates the WebSphere MQ transaction routing using the JAVA Device Handler (JDH) process that runs in UNIX System Services on z/OS.

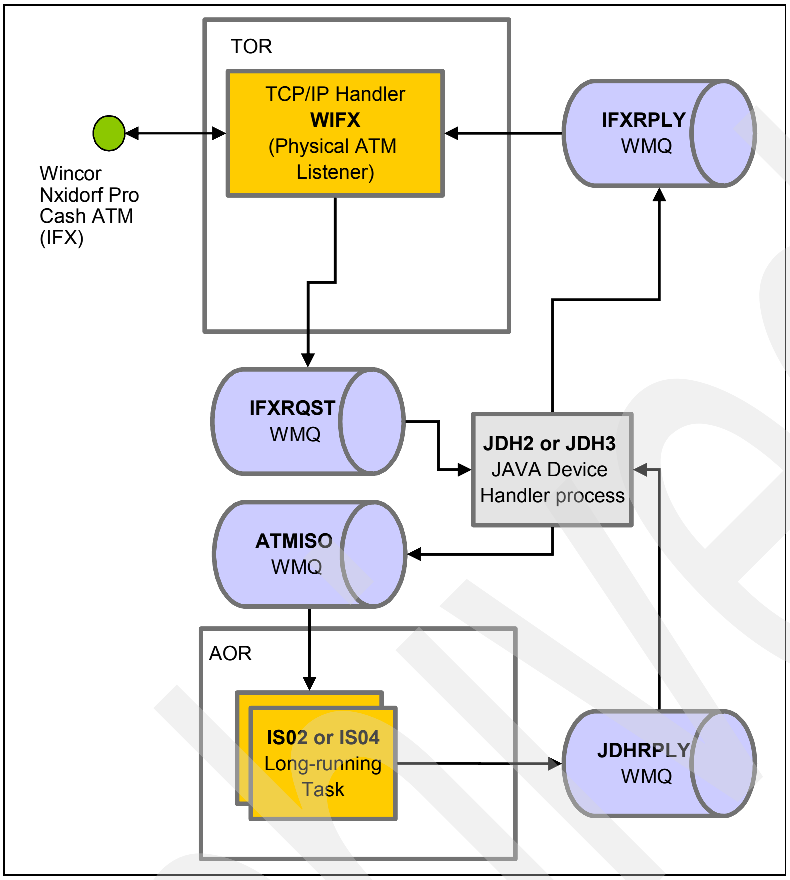


Figure 1-6 IFX ATM TCP/IP handler and Integrated Server interaction

Another important implementation part is the host interface to the Core System to authorize the payments, both ATM and VISA. Both channels use the same authorization mechanism.

Figure 1-7 illustrates the authorization and host interface.

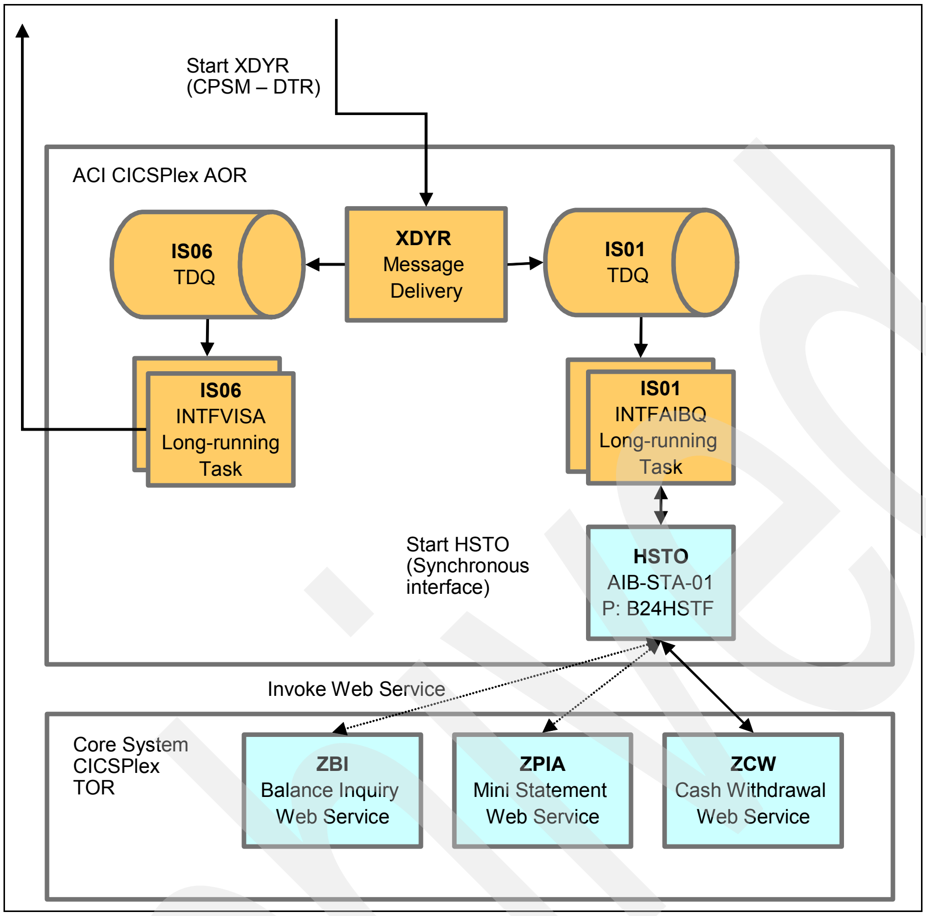


Figure 1-7 Authorization – Host Interface

We could have performed this integration from BASE24-eps to our HSTO transaction either with an EXEC CICS LINK or with an EXEC CICS START. We chose the START capability because it is easier to monitor the transaction flow by looking at the HSTO transaction, which is accomplished through a Web Service call. In case the Web Service provider is within the same CICS region, using the CICS Host Optimization, CICS translates this call into an EXEC CICS LINK. However, since the provider, in our case, is in a separate CICSPlex, CICS has no way of knowing where the provider is located until it resolves the Web Service binding. See Appendix C, “CICS Web Services configuration” on page 39 for more set up information.

Table 1-4 shows the main Integrated Servers configured for our system: ATM, VISA, and Host interface.

Table 1-4 Integrated Servers

Integrated Server – Service Name	CICS TRANSID	Description
INTFAIBQ	IS01	Host Interface to our Core system for Authorization (DDMF)
ISATM	IS02	ATM IFX TCP/IP handler listening at port 4014 on both LPARs. This server was defined for the physical ATMs, for example, those ATMs that do have individual IP addresses. (SDMF)
ISATM4	IS04	VISA TCP/IP handler listening at port 9008 from AT10 in LPAR1 (SDMF)
ISVISA	IS06	VISA TCP/IP handler listening at port 9008 from AT11 in LPAR2 (SDMF)

2

Configuration changes for high availability

In this chapter, we describe the specific BASE24-eps configuration changes that we made to CICS resource definitions and to BASE24-eps to handle the following scenarios:

- High availability of ATM IFX transactions
- High availability of VISA transactions

The principles behind both of the scenarios are the same; however, the techniques are different, largely because the ATM channel uses WebSphere MQ and Java™ Device Handlers, whereas the VISA channel uses CICS Transient Data Queues.

2.1 High availability

In our project, we focused on the following aspects to provide a highly available solution for our application:

- *Redundancy*: The ability to automatically distribute workload to other available resources if one component becomes unavailable without an impact on the quality-of-service of that transaction (response time, success, and so on), for example, that one component is made redundant without a significant impact on the overall system.
- *Recovery*: The ability to recover resources and workload in the event of a failure.
- *Hardening:* Through security and system resiliency, ensure that the problem does not occur in the first place.

Within the Smart Bank project, we demonstrate these aspects by re-routing workload across different system components for workload balancing purposes and to manage an unplanned outage, where the system needs to react automatically to a disaster situation. We use the inherent capabilities of z/OS, GDPS® HyperSwap™, Parallel Sysplex®, and data sharing in a System z environment to facilitate this functionality. However, hardware and software cannot be the only players in this solution. The applications must also be able to operate in these environments. BASE24-eps has this capability if correctly configured.

The main factor in designing the high availability solution is that the system component that receives a transaction is the same one that needs to provide a response due to the nature of ATM and VISA network interfaces.

Typically, ATMs are permanently connected to one IP address socket, and they expect the response back from the same port number and IP address socket. The VISA interface works in the same way, although in other respects it is a much simpler interface than the conversational ATM interface. Even when we created logical ATMs within Rational Performance Tester, we still had to maintain the affinity between Terminal ID (specific ATM) with a particular socket/IP address. BASE24-eps naturally does not expect virtual ATMs, so if we did not maintain this affinity, we experienced multiple IP Socket connections and disconnections that resulted in long response times and transaction issues.

So there is an affinity between the source and the system component that receives the request.

If the TOR, where the ATM (endpoint) is connected, disappears, the ATM retries and reconnects with another TOR. We define a virtual IP address (VIPA) within the z/OS Sysplex, so that the Sysplex Distributor component routes the workload to the LPAR (and TOR) that is ready to receive transactions.

The TOR that receives the transaction has, within its scope, all the AORs in the BASE24-eps CICSPlex. The resulting AOR that carries out the work, can be on a different LPAR. That AOR needs to return the transaction to the original TOR that has the connection to the ATM.

2.2 High availability for ATM IFX transactions

In this section, we describe the final application architecture, where:

- The ATM IFX transactions are received by a long running CICS Transaction (Listener) in the Terminal Owning Region (TOR), and then being placed on a queue.

- WIFX listens to port 4014 on a physical IP address, which is linked to LPAR1 in our Sysplex configuration, and AIFX listens to port 4012 on the same IP address for LPAR1.
- On LPAR2 we have the same configuration, but the IP address is obviously different. The VIPA is the same for the two.
- Rational Performance Tester injects workload for the virtual ATMs using the VIPA.
- After it is connected to BASE24-eps, the real physical ATM always uses the same IP address as soon as the connection stays open. The same is true for the individual virtual ATMs.
- Sysplex Distributor routes to different resources based on each new connection. However, for an ATM there is one connection and then many different transactions because it maintains the socket open. If the LPAR was to disappear, then the ATM tries to reconnect to a different socket, and create a new connection. At this point, Sysplex Distributor again allocates to the most available LPAR.

Figure 2-1 shows the final application architecture.

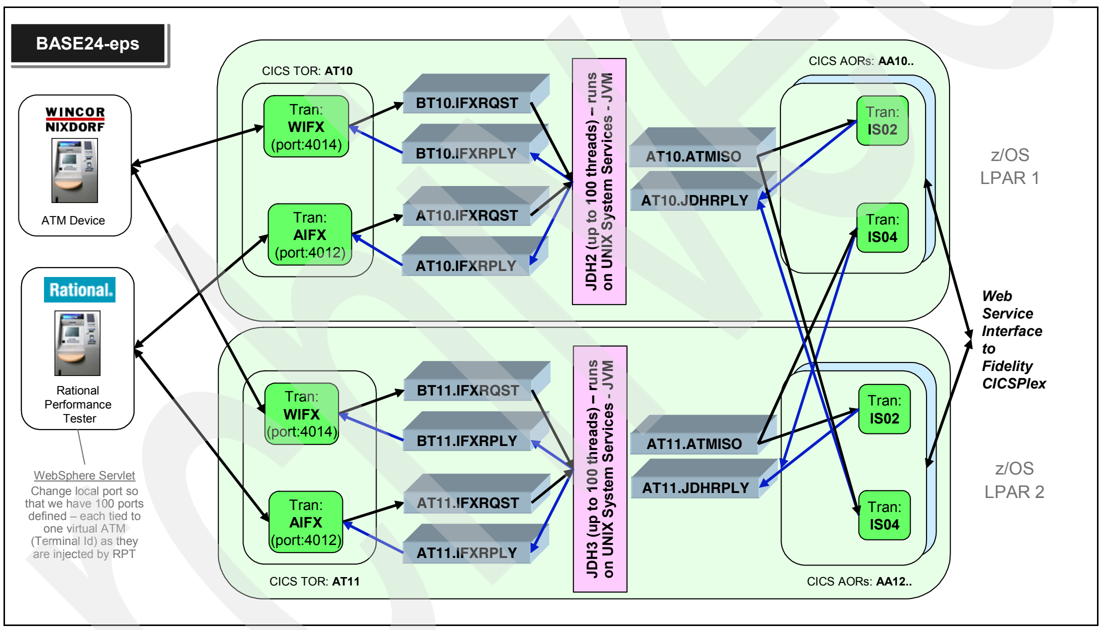


Figure 2-1 The ATM IFX system architecture within BASE24-eps

We specified the separate WebSphere MQ Queues to isolate the AIFX and WIFX workloads, so that the reply always finds its way back to the correct listener, which is also why we have BT10 and AT10 queues on the same LPAR.

For the same reason, we specified IS02 on the AOR for local requests. IS02 always responds back to the queue in its own LPAR, whereas IS04 is for requests that originate on the other LPAR. IS04 always responds back to the other LPAR queue, which is why we need WebSphere MQ sharing, where the queues exist in the coupling facility.

In order to set up these configuration parameters in the BASE24-eps, we created local copies of the dynamic destination map file (DDMF) and the static destination map file (SDMF).

2.2.1 Configuration settings for ATM IFX

In this section, Table 2-1 and Table 2-2 summarize and simplify the relationship between the different CICS, WebSphere MQ, and file definitions.

Static destination map file

Table 2-1 summarizes the relationship between the different CICS, WebSphere MQ, and file definitions.

Table 2-1 Static destination map files

LPAR	Service name	CICS TRANSID	Destination Type	MQ Support Flag	Queue Name
1	ATM-IFX	AIFX	0012	Y	AT10.IFXRPLY
2	ATM-IFX	AIFX	0012	Y	AT11.IFXRPLY
1	ATM-IFX2	WIFX	0012	Y	BT10.IFXRPLY
2	ATM-IFX2	WIFX	0012	Y	BT11.IFXRPLY
1	ISATM	IS02	0011	Y	AT10.ATMISO
2	ISATM	IS02	0011	Y	AT11.ATMISO
1	ISATM4	IS04	0011	Y	AT11.ATMISO
2	ISATM4	IS04	0011	Y	AT10.ATMISO

Dynamic destination map file

Table 2-2 summarizes the relationship between the different CICS, WebSphere MQ, and file definitions.

Table 2-2 Dynamic destination map file

LPAR	Service name	CICS TRANSID	Destination Type	Queue Name
1	ATM-IFX	AIFX	0013	AT10.IFXRQST
2	ATM-IFX	AIFX	0013	AT11.IFXRQST
1	ATM-IFX2	WIFX	0013	BT10.IFXRQST
2	ATM-IFX2	WIFX	0013	BT11.IFXRQST
1	JDH2	-----------------	0011	AT10.JDHRPLY
2	JDH2	-----------------	0011	AT10.JDHRPLY
1	JDH3	-----------------	0011	AT11.JDHRPLY
2	JDH3	-----------------	0011	AT11.JDHRPLY

We defined the JDH2 and JDH3 service names individually on each LPAR for completeness. There are also queues for the JTIMER process and the ATMCUTOVER.

Server flag setting to verify local IP address of the ATM

Another notable configuration setting we made was whether or not to verify the local (source/endpoint) IP address that is held on the SOCKRECS configuration file. If the ATM is a real ATM for WIFX, then we turned this flag on, because there is one-to-one mapping of physical device to IP address. However, for the logical ATMs where many ATMs have the same local IP address (Rational Performance Tester), we turned this flag off. We made this configuration on the R302 3270 CICS transaction and stored it in the TCPIPCFG configuration file. The change is to set “Srv Type” to enable verification of the local IP address.

2.3 High availability of VISA transactions

The issues for the VISA workload are similar to the issues that we discussed for the ATM, in 2.2, “High availability for ATM IFX transactions” on page 16, with regard to using the Transient Data Queues instead of the WebSphere MQ queues.

We defined another Station ID on the ACI Desktop and created two different entries on the TCPIPCFG configuration file through the R301 3270 CICS transaction. We created local copies of the TCPIPCFG file to make specific configurations for each LPAR.

2.3.1 Configuration settings for VISA

In this section, we provide some configuration settings for the VISA transactions.

Table 2-3 displays the TCPIPCFG configuration file settings for the Visa transactions.

Table 2-3 TCPIPCFG—TCP IP Configuration File

LPAR	Service name	Port	Destination type
1	VISA-STA-01	9008	INTFVISA (set-up to run on AT10 only)
2	VISA-STA-02	9008	INTFVISA (set-up to run on AT11 only)

Table 2-4 shows the static destination map file and dynamic destination map file for the Visa transactions.

Table 2-4 SDMF and DDMF

LPAR	Service name	CICS TRANSID	Region
1	VISA-STA-01	TC40	AT10
1	VISA-STA-01	TC40	AT10
2	VISA-STA-02	TC40	AT11
2	VISA-STA-02	TC40	AT11

When the message is put into a TDQ, a CICS transaction gets triggered. If the trigger level is greater than one, then the presence of one or more transactions trigger a transaction associated with the TDQ. The BASE24-eps AORs, which were started by the TC40 VISA TCP/IP handler in TORs, writes the response back into the TDQ called “TC40” in the originating TOR. BASE24-eps specifies the region for the TDQ in the SYSID.

When a TDQ is configured, you specify its type. TYPE=Intra for intra partition or local. Specifying the SYSID on a TDQ in CICS indicates that the queue is remote in any case and overrides the TDQ definition. We defined the two VISA stations in such a way that the AOR knows where to send the response. For example, if the VISA transaction arrived on LPAR2 in

AT11 TOR, then we assign this transaction the service name VISA-STA-02. Even if the AOR that processes the request happens to be on LPAR1, we still know that it originated from VSIA-STA-02. TC40 is the TDQ and TRANSID associated with this service. TC40 is common to both TORs. However, with the knowledge that we are running with VISA-STA-02 we ensure that the return TDQ WRITE has the SYSID (AT11) specified.

ACI Desktop graphical user interface change

The change that you make using the ACI-DESKTOP graphical user interface is to add VISA-STA-02 to the VISA_NET_Interface definitions:

1. Select, Configure → Interface → Network → VISA.
2. Go to the STATIONS tab, and use the Insert option to add the new station – STA-01.
3. Save the change.
4. Update OLTP, which refreshes the VSAM RLS file through the XMLS transaction. For BASE24-eps to pick up the change, restart TC40 and IS06.

A

CICS configuration

In this appendix, we describe the configuration definitions from LPAR1 (AT10, AA10, and AA11). Where the configuration applies to a specific LPAR, we note it by referencing the local CICS TOR. These configuration settings apply to the BASE24-eps 3270 CICS transactions.

Table A-1 on page 22 through Table A-11 on page 28, provides a summary and a cross reference of the configuration definitions that are required to set up CICS.

R301 (DTR6301) – TCP/IP control file record list

Table A-1 R301 TCP/IP control file record list

Service name	Local Port Number	Destination
ACI-DESKTOP	09007	XMLI
ATM-CUTOVER	09020	ATMCUTOVER
ATM-IFX	04012	ATMIFX
ATM-IFX2	04014	ATMIFX2
JAVA-EVENTS	09014	LOGGER
TSEC	09015	TSEC-HSM
VISA-STA-01	09008	INTFVISA (AT10 only)
VISA-STA-02	09008	INTFVISA (AT11 only)

VISA-STA-01 and 02 are created for the high availability set up. VISA requests that originate in LPAR1 (AT10) can be distributed to the most available AOR by CICSPlex (potentially on LPAR2) and still respond back to AT10 on LPAR1. This session affinity is required for the VISA and ATM interfaces. The ATM interfaces are different and require different configuration using WebSphere MQ.

R302 (DTR6302) – TCP/IP control file record maintenance

If you browse, you get directed to *R302* where the valid commands you can issue are:

- i – insert
- e – edit
- d – delete
- c – copy
- b – browse

Table A-2 R302 TCP/IP control file record maintenance

Service name	Max sockets	Backlog	Hdr type	Srv type	Evt Destination TD Queue	Time out value	Log flag	Trace flag
ACI-DESKTOP	010	002	0	0	EVTS	9	N	N
ATM-CUTOVER	010	002	0	0	EVTS	9	Y	T
ATM-IFX	500	100	0	0	EVTS	1	Y	T
ATM-IFX2	500	100	0	1	EVTS	1	Y	T
JAVA-EVENTS	010	010	0	0	EVTS	9	N	N
TSEC	010	010	0	0	EVTS	9	Y	T
VISA-STA-01	210	100	0	0	EVTS	1	N	N
VISA-STA-02	210	100	0	0	EVTS	1	N	N

Srv Type**:** 0=server (do not verify), 1=Verify with IP address – can send an unsolicited request to the ATM, 2=client, 3=stclient

When 0, BASE24-eps does not check the IP address from the R309 window.

Logging Flag: Y=log on

Tracing Flag: T=all, R=receive, S=send

R303 (DTR6303) static destination map file record list

Table A-3 R303 static destination map file record list

Service name	Transaction	Region	Notes
ACI-DESKTOP	TC90	AA10	Development (10.1.1.108) Production (Blade 9)
AIB-STA-01	HSTI	AT10	Fidelity Web Service Interface
ATM-CUTOVER	TC20	AT10	---
ATM-IFX	AIFX	AT10	Service name for virtual ATMs controlled by Rational injection.
ATM-IFX2	WIFX	AT10	Service name for single machines simulating real ATMs.
ATMT001	WIFX	AT10	Test injection Servlet machine
ATMT002	WIFX	AT10	Wincor Nixdorf real ATM
ATMT003	WIFX	AT10	ACI IFX Simulator running with Desktop
ATMT004	AIFX	AT10	1st Virtual ATM
ATMT005	AIFX	AT10	2nd Virtual ATM
ATMT006	AIFX	AT10	3rd Virtual ATM
…			
ATMT300	AIFX	AT10	297th Virtual ATM
EOPP	EOPP		
IS	IS01		
ISATM	IS02		
ISATM4	IS04		
ISTSEC	IS03		
ISVISA	IS06		
JAVA-EVENTS	TC00	AT10	
RFSH	RFSH		
RSTR	RSTR		

Service name	Transaction	Region	Notes
RTF-STA-01	BDWI		Real Time Feed Interface
SAFM	SAFM	AA10	Store and Forward (limit region for RTF)
TBP	TBP1		
TSEC	TC30	AT10	
VISA-STA-01	TC40	AT10	
VISA-STA-02	TC40	AT11	
XMLS	XMLS		CICS transaction side of the ACI Desktop process

R304 (DTR6304) static destination map file record maintenance

Table A-4 R304 static destination map file record maintenance

Service name	TRAN	Dest type	Max servers	Depth Srv ratio	Dyn Srv flag	MQ Sup flg	Queue name
ACI-DESKTOP	TC90	0003	0001	0001	L	N	
AIB-STA-01	HSTI	0003	0005	0001	Y	N	
ATM-CUTOVER	TC20	0009	0001	0001	L	N	
ATM-IFX	AIFX	0012	0001	0001	L	Y	AT10.IFXRPLY
ATM-IFX2	WIFX	0012	0001	0001	L	Y	BT10.IFXRPLY
ATMT001	WIFX	0011	0001	0001	N	N	
ATMT002	WIFX	0011	0001	0001	N	N	
ATMT003	WIFX	0011	0001	0001	N	N	
ATMT004	AIFX	0011	0001	0001	N	N	
ATMT005	AIFX	0011	0001	0001	N	N	
ATMT006	AIFX	0011	0001	0001	N	N	
…							
ATMT100	AIFX	0011	0001	0001	N	N	
EOPP	EOPP	0001	0001	0001	Y	N	
IS	IS01	0001	0010	0001	L	N	
ISATM	IS02	0011	0010	0001	L	Y	AT10.ATMISO
ISATM4	IS04	0011	0010	0001	L	Y	AT11.ATMISO
ISTSEC	IS03	0011	0001	0001	Y	Y	AT10.TSEC
ISVISA	IS06	0001	0010	0001	L	N	

Service name	TRAN	Dest type	Max servers	Depth Srv ratio	Dyn Srv flag	MQ Sup flg	Queue name
JAVA-EVENTS	TC00	0009	0001	0001	L	N	
RFSH	RFSH	0001	0001	0001	L	N	
RSTR	RSTR	0001	0001	0001	N	N	
RTF-STA-01	BDWI	0003	0003	0001	N	N	
SAFM	SAFM	0001	0001	0001	L	N	
TBP	TBP1	0001	0001	0001	Y	N	
TSEC	TC30	0003	0001	0001	L	N	
VISA-STA-01	TC40	0007	0003	0001	L	N	
VISA-STA-02	TC40	0007	0003	0001	L	N	
XMLS	XMLS	0001	0001	0001	L	N	

Destination Type:

- 0001 – Started Task (where we do the downloads from)
- 0003 – TDQ task

Dynamic Server Flag:

- L – long running task

R305 (DTR6305) DDMF record and R306 (DTR6306) DDMF record maintenance

Table A-5 R305 DDMF record and R306 DDMF record maintenance

Service name	TRAN	Region	Dest type	Region list	Queue name
TIMER-SERVICE	TIMR		0013		
ACI-DESKTOP	TC90		0003		
AIB-STA-01	HSTO		0009		
ATMCUTOVER	TC20		0011		AT10.ATMCUTOVER
ATMIFX	AIFX		0013		AT10.IFXRQST
ATMIFX2	WIFX		0013		BT10.IFXRQST
EOPP	EOPP		0006		
IFMGR	IS01		0001		
INTFAIBQ	IS01		0003		
INTFRTF	IS01		0003		
INTFRTF	IS01		0003		

Service name	TRAN	Region	Dest type	Region list	Queue name
INTFVISA	IS06		0006		
IS	IS01		0003		
JDH1			0011		AT10.JDH1RPLY
JDH2			0011		AT10.JDH1RPLY
JDH3			0011		AT11.JDH1RPLY
JNLBTCH	R004		0001		
JTIMER			0011		AT10.JDHTIMER1
JTIMER2			0011		AT10.JDHTIMER1
LOGGER	TC00		0003		
PCTL	XMLS		0008		
RFSH	RFSH		0003		
RTF-STA-01	BDWO		0009		
SAFM	SAFM		0001		
TBP	TPB1		0003		
TSEC	TC30		0003		
TSEC-HSM	TC30		0003		
VISA-STA-01	TC40	AT10	0003		
VISA-STA-02	TC40	AT11	0003		

- Destination type – 11, 13 for queue name
- Destination 9: CICS START with data

R307 (DTR6307) Synch destination map (SYDMF) file record and R308 (DTR6308) SYDMF record maintenance

Table A-6 R307 and R308 SYDMF file record and record maintenance

Service name	Program	Header flag	Xadapter flag
AIB-STA-01	B24HSTF	Y	N
EOPP	EOPPLO	N	N
IS	ISLO	N	N
IS01	ISLO	N	N
IS02	ISLO	N	N
SAFM	SAFMLO	N	N

Header flag: Y if destination requires header-less messages

R309 (DTR6309) SOCKRECS file record list

Socket records: Table A-7 is the table that BASE24-eps verifies against with the WIFX transactions – real ATMs.

Table A-7 R309 SOCKRECS file record list

Service name	Port	TCP/IP Address	Destination	Notes
ATMT001	00000	10.001.001.204	ATMIFX2	Development station
ATMT002	00000	10.001.001.110	ATMIFX2	Wincor ATM
ATMT003	00000	10.001.001.108	ATMIFX2	IFX Simulator
ATMT004	00000	10.001.001.020	ATMIFX	Second Life
ATMT005	00000	10.001.001.020	ATMIFX	Virtual - VIPA
ATMT006	00000	10.001.001.020	ATMIFX	--
…				--
ATMT100	00000	10.001.001.020	ATMIFX	--
VISA-STA-01	09008	10.001.001.132	INTFVISA	Blade 9 - PARAGON
VISA-STA-02	09008	10.001.001.132	INTFVISA	Blade 9 - PARAGON

R310 (DTR6310) SOCKRECS file record maintenance

Table A-8 R310 SOCKRECS file record maintenance

Service name	Connect type	Disconnect type	TCP/IP Hdr type
ATMT001	Y	N	0
ATMT002	Y	N	0
ATMT003	Y	N	0
ATMT004	Y	N	0
ATMT005	Y	N	0
ATMT006	Y	N	0
…			
ATMT100	Y	N	0
VISA-STA-01	Y	N	0

Connect type:

- N—delayed connect
- Y—immediate

Disconnect type:

N—do not disconnect after connect

R311 (DTR6311) – DTRCTL file record list

Table A-9 R311 DTRCTL file record list

Key	Name	Apps	TORs	AORs
CTLR	DTRCTL	0002	------	------
EOPP	EPSDFLT	------	0	1
XDYR	EPSDFLT	------	2	14

R312 (DTR6312) – DTRCTL record maintenance

- Control key: 0000
- Control record name: DTRCTL
- CPSM version: 0310
- CPSM User ID: CPSMUSER
- CPSM Signon: CPSMUSER
- TRAN: EOPP TSQ#: 002
- TRAN: XDYR TSQ#: 003

Table A-10 R312 DTRCTL record maintenance

Appl. key	Appl. name	CPSM Workload name	API flag	Default seq flg	Handshake flag	Init trans	TOR scope	AOR scope
EOPP	EPSDFLT	WLSACI	N	Y	N	EOPP	------	AA10
XDYR	EPSDFLT	WLSACI	Y	Y	Y		AT10 AT11	AA10 AA11 AA12 AA13 -- AA23

R314 (DTR6314) TERMINAL (TERMCFG) file record list and R315 (DTR6315) Term Config map file record maintenance

Table A-11 R314 and R315 TERMCFG

Service name	Destination name
BDWI	INTFRTF
HSTI	INTFAIBQ

B

ACI Desktop based configuration

In this appendix, we describe the ACI Desktop configuration.

Physical cards created in the card file

Table B-1 contains the physical cards that are created in the card file.

Table B-1 Physical cards created in the card file

Number	Track2	Track3
1 (Wincor)	4988248000000000=0912	
2 (Wincor)	4988248000004800=0912	
3 (Wincor)	4988248000007201=0912	
4 (Wincor)	4508768000003000=0912	
5 (Wincor)	4508768000004801=0912	
6 (Fabrice Hertz)	4988248000000001=0912	

Table B-2 displays the BIN numbers that are configured in BASE24-eps.

Table B-2 BIN numbers configured in BASE24-eps

BIN number	Description
498824	Smart Bank current accounts
450876	Smart Bank savings accounts
430056	OASIS Bank of the Sahara
451298	Community Bank debit card
497422	Forest Regeneration Bank
563002	Yachting Island Bank
572198	Pharmaceutical Bank

To start the ACI Desktop (10.1.1.132 / 9.212.128.32 – 2003 server)

Use the following steps to start the ACI Desktop:

1. Open Version checker.
2. Run User Security, which communicates with CICS using IP port - 3031.
3. Click StartesWeb, which starts the Tomcat Web server and allows desktop communications with CICS – XMLS transaction.
4. Start ACI Desktop, and enter the logon user name: SysAdmin (sysadmin01).
5. In the ISS-FCBK field, choose Fidelity Corebank.

Viewing ATM traffic

Use the following steps to view ATM traffic:

1. Select, System Operations → ATM Framework → ATM Status Query (SMARTBANK – IFX – binoculars).
2. Select, System Operations → ATM Framework → ATM Channel Management → Selected Channels.

Viewing denied and approved transactions

Go to System Operations → Active Script Statistics, to view denied and approved transactions.

Viewing the status of physical ATMT002 ATM

To view the status of physical ATMT002 ATM, select System Operations → ATM Framework → ATM Channel Administration → choose ATMT002.

Viewing sockets and listener configuration

Use the following steps to view sockets and listener configurations:

1. Select, System Operations → Server Management → Server Admin → Socket Administration.
2. Select, System Operations → Server Management → Listener.

Adding cards to the card file

Use the following steps to add cards to the card file:

1. Select, Customer Management → Card.
2. Select, Institution ID: SMARTBANK.
3. Next to Card Number, click the Find icon (binoculars). The new Card Management window is displayed.
4. Click the Find button. If no from and to values are entered, we get the full details of all cards defined to the file. Select a value, and click OK. This pulls the specific card details into the original Card Management window and allows us to change the Card number and Customer ID to add a new card by clicking the Save to disk icon (top left). You need to change the Customer ID as well.

Block Cards

Use the following steps to block cards:

1. Select, Customer Management → Card → Select Institution ID: SMARTBANK.
2. Find the card you want to block using the steps in “Adding cards to the card file” on page 31.
3. Click the Status tab, select the required status, and click Save. Due to VSAM RLS all CICS regions should be updated.

4. To block the card, choose from the following options:
   - Denied-Lost (60)
   - Denied-Stolen (70)
   - Denied-Closed (80)
5. To reopen select, Open (00).

Verifying action codes

Use the following steps to verify action codes:

1. Select, Configuration → Action Code Profile.
2. Select, our profile: **ATM_ACT_CDE_PRFL**.
3. Click the Outbound Mapping tab to see our return codes.
   - 114 – Deny, no account type requested → 014 external
   - 125 – Deny, card not effective → 014 external code sent back to ATM
   - 908 – Transaction destination cannot be found for routing – bad BIN number
   - 912 – Card issuer unavailable – bad BIN number

Verifying journal records

To verify journal records, select View → Journal Perusal.

Viewing transaction formats – VISA

To view transaction formats for VISA, click View → Journal Perusal.

Adding a new script from a copy on another platform

Use the following steps to add a new script from a copy of the script from another platform (Development to production):

1. Select, Configuration → Script → Script repository → Script editor, and choose the script.
2. Double-click Script Editor.

To add a new script:

1. Put your cursor into the script area, and click Edit → Select → Copy →Edit → Copy.
2. Move to Blade 9, select the new script from the Script Repository. Paste the script into the new script.
3. Save to Repository with the script name and a description.
4. From the Script Editor, submit the script to the server for compilation by clicking the third icon from the right.

Note on file changes using the ACI Desktop

In this section, we describe how to handle changes on the VSAM files. We also provide the best practice to optimize the resources:

- The ACI Desktop points to one region – AA10 and consequently the file changes that it initiates are made to the VSAM RLS files from that region.

- Because we have VSAM RLS these changes are made available to all regions within that CICSPlex.
- The reason to point the ACI Desktop to one region is to limit the configuration traffic to one CICS region, which can be quite heavy.
- However, about 85% of the VSAM RLS files are loaded to memory at CICS Start up time during a process called LOAD OLTP. This will happen with a warm or cold restart. It is necessary to re-perform this reboot to refresh memory.
- The 15% that are not in memory include the following files and consequently do not require a reboot/recycle of CICS.
  - Card file
  - Interface files
  - Journal files
- For Scripts it is necessary to recycle the CICS regions.

ACI Desktop configuration

The following sections describe how we configured our installations. You can also use this information as samples or guidelines.

Configuring the Directory

The following information is related to the Desktop directory configuration:

Configuring the interface

Perform the following steps to configure the interface. In our environment, we used the definitions summarized in Table B-2 on page 30:

1. Select Configure → Interface → Host → Real Time Feed (disabled in production).
2. Select Configure → Interface → Host → ATM Device Handler.

Table B-3 Interface sample definitions

Interface Name	RTF_FCBK	ACQ_IFX
Extract Script Name	JRNL_DATA_RTF	
Remote DB Server Format	TLFX	
Processing Context	IFX-CONTEXT	
SAF Delay & Retry interval	150s / 60s	
Country ID		FRA
ISO Msg		ATMDH_DEFAULT
Action Code		ATM_ACT_CDE_PRFL
Stations Input & Output	RTF-STA-01	JDH1, JDH2, JDH3
Stations Logon type	Combined Acquirer / Issuer	Combined Acquirer / Issuer

3. Select Configure → Interface → Network → VisaNet. In our configuration, we used the definitions in Table B-4.

Table B-4 Network interface definitions

Interface Name	ACQ_VISA
Source routing	ACQRTE_VISA
Acquirer transaction	ACQ_TRX_VISA
Issuer transaction	ISS_TRX_FCBK
Context	VISA_CONTEXT
ISO Msg	MSG_VISA
Action Code	ACT_CDE_VISA
Stations Input & Output	VISA-STA-01 / VISA-STA-02
Stations Logon type	Combined Acquirer / Issuer
Visa Station	123456

4. Select Configure → Interface → Context Profile. Table B-5 provides the definitions we used in our installation.

Table B-5 Context Profile definitions

AIB-CONTEXT	CXD_HISO
IFX-CONTEXT	CXD
VISA-CONTEXT	CXD_VISA

Routing configuration

The following tables describe the definitions we used for the routing environment.

1. Select, Configure → Routing → Destinations.

Table B-6 Routing configuration destinations

Routing Profile	Route Code	Cust. Auth.	Floor Limit Display	Pre-screen component and name	Auth. component and name
ISS_FCBK	RTE_BALINQ	PIN	Under and Over	IAUT CRD_CHECK	INTFAIBQ FCBK_HOST1
ISS_FCBK	RTE_MINISTMNT	PIN	Under and Over	IAUT CRD_CHECK	INTFAIBQ FCBK_HOST1
ISS_FCBK	RTE_WITHDRAWAL	PIN	Under and Over	IAUT CRD_CHECK	INTFAIBQ FCBK_HOST1

2. Select, Configure → Routing → Profile Description.

Table B-7 Routing configuration profile descriptions

Source Routing Profile	Profile description
ACQRTE_IFX	IFX ATM DH SOURCE ROUTING
ACQRTE_VISA	VISA NET SOURCE ROUTING

3. Select, Configure → Routing → Destination/Source Relationship.

Table B-8 Routing configuration destination and source relationship definitions

Destination Routing Profile	Source Routing Profile	Transaction Code	Route Code
ISS_FCBK	ACQRTE_IFX	Balance Inquiry (31)	RTE_BALINQ
ISS_FCBK	ACQRTE_IFX	Mini Statement Print (3B)	RTE_MINISTMNT
ISS_FCBK	ACQRTE_IFX	Debit Cash (ATM) (01)	RTE_WITHDRAWAL
ISS_FCBK	ACQRTE_VISA	Purchase (00)	RTE_WITHDRAWAL
ISS_FCBK	ACQRTE_VISA	Balance Inquiry (31)	RTE_BALINQ
ISS_FCBK	ACQRTE_VISA	Mini Statement Print (3B)	RTE_MINISTMNT
ISS_FCBK	ACQRTE_VISA	Debit Cash (ATM) (01)	RTE_WITHDRAWAL

Script configuration

Table B-9 and Table B-10 summarize the definitions we made to configure the Script environment.

Select Configure → Script → Authorization Script.

Table B-9 Script configuration definitions

Authorization Name	Message Type	Script Name	Enabled & Monitored
CRD_CHECK	Authorization Request	CRD_ST_CHK_PS_RQ	Y / Y
CRD_CHECK	Financial Request	CRD_ST_CHK_PS_RQ	Y / Y

Table B-10 Script configuration summary

Script Name	Description
JRNL_DATA_RTF	Real Time Feed Script (not currently used)
CRD_ST_CHK_PS_RQ	Card Status pre-screen request – block card script

Transactions Allowed configuration

Table B-11, Table B-12, and Table B-13 on page 37, summarize the definitions that we made to configure the Transactions Allowed environment.

1. Select, Configure → Acquirer Transactions Allowed → ACQ_TRX_IFX.

Table B-11 Transaction Allowed

Message Category	Processing Code	Tran	Admin	On-Us
Authorization	Withdrawal from Checking	012000	N	Internationally
Authorization	Mini Statement Print from Default Account	3B0000	N	Internationally
Authorization	Mini Statement Print from Savings	3B1000	N	Internationally
Authorization	Mini Statement Print from Checking	3B2000	N	Internationally
Authorization	Balance Inquiry from Default Account	310000	N	Internationally
Authorization	Balance Inquiry from Savings	311000	N	Internationally
Authorization	Balance Inquiry from Checking	312000	N	Internationally
Authorization	Balance Inquiry from Credit	313000	N	Internationally
Financial	Withdrawal from Default Account	010000	N	Internationally
Financial	Withdrawal from Savings	011000	N	Internationally
Financial	Withdrawal from Checking	012000	N	Internationally
Financial	Withdrawal from Line of Credit	013800	N	Internationally
Financial	Fast Cash from Default Account	170000	N	Internationally

2. Select, Configure → Acquirer Transactions Allowed → ACQ_TRX_VISA.

Table B-12 Acquirer Transaction Allowed

Message Category	Processing Code	Tran	Admin	On-Us
Authorization	Purchase from Savings	001000	N	Internationally
Authorization	Purchase from Checking	002000	N	Internationally

Message Category	Processing Code	Tran	Admin	On-Us
Authorization	Withdrawal from Checking	012000	N	Internationally
Authorization	Balance Inquiry from Default Account	310000	N	Internationally
Authorization	Balance Inquiry from Savings	311000	N	Internationally
Authorization	Balance Inquiry from Checking	312000	N	Internationally
Financial	Withdrawal from Default Account	010000	N	Internationally
Financial	Withdrawal from Savings	011000	N	Internationally
Financial	Withdrawal from Checking	012000	N	Internationally

3. Select, Configure → Issuer Transactions Allowed → ISS_TRX_FCBK.

Table B-13 Issuer Transaction Allowed

Message Category	Processing Code	Tran	Settlement impact	On-Us or Not-on-Us
Authorization	Purchase from Savings	001000	Y	Internationally
Authorization	Purchase from Checking	002000	Y	Internationally
Authorization	Withdrawal from Checking	012000	Y	Internationally
Authorization	Mini Statement Print from Default Account	3B0000	Y	Internationally
Authorization	Mini Statement Print from Checking	3B2000	Y	Internationally
Authorization	Balance Inquiry from Savings	311000	Y	Internationally
Authorization	Balance Inquiry from Checking	312000	Y	Internationally
Financial	Withdrawal from Default Account	010000	Y	Internationally
Financial	Withdrawal from Savings	011000	Y	Internationally
Financial	Withdrawal from Checking	012000	Y	Internationally

Action Code Profile configuration

Table B-14 provides the definitions that we used for the Action Code Profile.

Select, Configure → Action Code Profile.

Table B-14 Action Code Profile

Action Code Profile
ACT_CDE_VISA
AIBQ_ACT_PRFL

Customer Management directory

The following tasks refer to the Customer management directory configuration.

Card file configuration

We used the definitions in Table B-15 for our Card file configuration.

Select, Customer Management → Card Transactions Allowed → ISS_TRX_FCBK.

Table B-15 Card file configuration

Card Number	Customer ID	Card Type	Accounts	Account Type
4988248000000000	Current Account 1	Visa Classic	8000000000	Checking (20)
4988248000004800	Current Account 2	Visa Classic	8000004800	Checking (20)
4988248000007201	Current Account 3	Visa Classic	8000007201	Checking (20)
4508768000003000	Savings Account 1	Visa Classic	8000003000	Savings (10)
4508768000004801	Savings Account 2	Visa Classic	8000004801	Savings (10)
4988248000000001	Fabrice's Hertz Card	Visa Classic	8000000001	Checking (20)

C

CICS Web Services configuration

In this appendix, we summarize the CICS Web Services definitions. Use the summarized parameters in Table C-1 on page 40 through Table C-9 on page 42, for your reference.

CICS Web Services requester configuration

For the CICSPlex ACI BASE24-eps, we used the definitions in Table C-1, Table C-2, Table C-3, and Table C-4 on page 41.

Table C-1 TCP/IP definitions

CICS TCP/IP services definitions	
TCP/IP	Yes

Table C-2 CICS Web services requester definition

CWTBPI02	Web Services	Auto-Install for Balance Inquiry AOR WSDL & Bind /u/fabrice/aciwsbi/ Endpoint: (http://10.1.1.9:51001/u/fabrice/cwtbpi02)
CWPEPM01	Web Services	Auto-Install for Cash withdraw AOR WSDL & Bind /u/fabrice/aciwscw/ Endpoint: (http://10.1.1.9:51001/u/fabrice/cwpepm01)
CWPISI05	Web Services	Auto-Install for Mini statements AOR WSDL & Bind /u/fabrice/aciwspia/ Endpoint: (http://10.1.1.9:51001/u/fabrice/cwpisi05)

Table C-3 CICS Pipelines for Requester definitions

ACIBI	Pipeline	GR(FABRICE): AOR Config: /usr/lpp/cicsts/cicsts31/samples/pipelines/basicsoap11requester.xml
ACICW	Pipeline	GR(FABRICE): AOR Config: /usr/lpp/cicsts/cicsts31/samples/pipelines/basicsoap11requester.xml
ACIPIA	Pipeline	GR(FABRICE): AOR Config: /usr/lpp/cicsts/cicsts31/samples/pipelines/basicsoap11requester.xml

Table C-4 CICS Transaction and programs definitions

BSTR	Transaction	GR(FABRICE) test tx for ACI RSTR load tx
STRTRSTR	Program	GR(FABRICE) program PLTPI TOR & AOR for auto-start of RSTR ACI tx

CICS Web Services provider configuration

For the CICSPlex Fidelity Corebank, we used the definitions in Table C-5 through Table C-9 on page 42.

Table C-5 CICS TCP/IP services definitions

TCP/IP	Yes
CRBWEBS1	9.212.128.2 (Port 51201) CICSRT11 - Lpar2
CRBWEBS1	9.212.128.1 (Port 51201) CICSRT10 - Lpar1
CRBWEBS	(VIPA address) 10.1.1.9 (Port 51001) CICSRT10 & CICSRT11

Table C-6 CICS Web Services Provider definitions

CWTBPI02	Web Services	GR(FABWEBS) - Balance Inquiry WSDL & Bind /u/fabrice/cwtbpi02/
CWPEPM01	Web Services	GR(FABWEBS) - Mini statements WSDL & Bind /u/fabrice/cwpepm01/
CWPISI05	Web Services	GR(FABWEBS) – Cash withdraw WSDL & Bind /u/fabrice/cwpepm01/

Table C-7 CICS Pipelines for Provider definitions

PIPWSBI	Pipeline	GR(FABPIPE): Config: /u/fabrice/pipelines/spBIsoap11provider.xml
PIPWSCW	Pipeline	GR(FABPIPE) Config: /u/fabrice/pipelines/spCWsoap11provider.xml
PIPWSPIA	Pipeline	GR(FABRICE) Config: /u/fabrice/pipelines/spPIAsoap11provider.xml

Table C-8 CICS URIMap for Provider definitions

URIZCW	URIMAP	GR(FABURI) URI: /u/fabrice/cwpepm01 Pipeline → PIPWSCW Webservice → CWPEPM01 Tcpipservice → CRBWEBS
URIZPIA	URIMAP	GR(FABURI) URI: /u/fabrice/cwpisi05 Pipeline → PIPWSPIA Webservice → CWPISI05 Tcpipservice → CRBWEBS
URIZBI	URIMAP	GR(FABURI) URI: /u/fabrice/cwtbpi02 Pipeline → PIPWSBI Webservice → CWTBPI02 Tcpipservice → CRBWEBS

Table C-9 CICS Transaction and program definitions

ZBI	Transaction Balance Inquire	GR(FABURI) TOR → Remote AOR → Local
ZZBI	Transaction	GR(FABURI) TOR → Local AOR → N/A
ZCW	Transaction Cash withdraw	GR(FABURI) TOR → Remote AOR → Local
ZZCW	Transaction	GR(FABURI) TOR → Local AOR → N/A
ZPIA	Transaction Mini statement	GR(FABURI) TOR → Remote AOR → Local
ZZPI	Transaction	GR(FABURI) TOR → Local AOR → N/A
COTBPI02	Program	GR(JCAPGM) Switch Container / Commarea Call Fidelity Balance Inquire transaction
CWPEPM01	Program	GR(JCAPGM) Switch Container / Commarea Call Fidelity Cash withdraw transaction
CWPISI05	Program	GR(JCAPGM) Switch Container / Commarea Call Fidelity mini statement transaction

HANDLBI	Program	GR(FABRICE) TOR Switch Transaction ZBI
HANDLCW	Program	GR(FABRICE) TOR Switch Transaction ZCW
HANDLPIA	Program	GR(FABRICE) TOR Switch Transaction ZPIA

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

IBM Redbooks

For information about ordering these publications, see "How to get Redbooks" on page 45. Note that some of the documents referenced here may be available in softcopy only.

- *A Guide to Using ACI Worldwide's BASE24-es on z/OS*, SG24-7268

Online resources

The following Web site is also relevant as further information sources:

ACI Worlwide website

http://www.aciworldwide.com/

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

ACI Worldwide's BASE24-eps V6.2: A Supplement to SG24-7268

Redpaper™

Practical installation experience

ATM, VISA, and payment networks integration

Sample environment definitions

In this paper, we document our configuration findings from an implementation scenario that we performed from the IBM Redbooks publication, A Guide to Using ACI Worldwide's BASE24-es on z/OS®, SG247268. We based this Redpaper on a project that was undertaken by the Product and Solutions Support Center (PSSC) in IBM® Montpellier, France.

One of the considerations is that, since the publication date of the Redbooks publication, the product name changed from BASE24-es to BASE24-eps (enterprise payments system).

The Smart Bank project, hosted in Montpellier, implemented BASE24-eps on z/OS, and then integrated it with both simulated and real ATM devices. A simulated VISA network authorized the payments against an independent host-core system from another independent software vendor (ISV). As a result of this work, we covered some new ground in configuring BASE24-eps on z/OS, which is what we cover in this Redpaper.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

REDP-4338-00