Smarter Banking with CICS Transaction Server

Unlock business data in the core banking system

Discover CICS TS V4.1 support for business events and Web 2.0

Learn about the IBM Smarter Banking Showcase

Chris Rayns
Neil Ashworth
Peter Beevers
Vesna Eibel
Fabrice Jarassat
Claus T Jensen
Alison Lucas
Anthony Papageorgiou
Alain Roessle
Nigel Williams

Redbooks

ibm.com/redbooks

International Technical Support Organization

Smarter Banking with CICS Transaction Server

April 2010

SG24-7815-00

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

First Edition (March 2010)

This edition applies to Version 4, Release 1, CICS Transaction Server.

Contents

Notices

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:

- AIX®
- BladeCenter®
- CICS Explorer™
- CICSPlex®
- CICS®
- Cognos®
- DataPower®
- DB2®
- developerWorks®
- DRDA®
- FICON®
- GDPS®
- Geographically Dispersed Parallel Sysplex™
- HiperSockets™
- HyperSwap®
- IBM®
- IMS™
- InfoSphere™
- Lotus®
- OMEGAMON®
- Parallel Sysplex®
- POWER®
- RACF®
- Rational®
- Redbooks®
- Redpaper™
- Redbooks (logo) ®
- System z10™
- System z®
- Tivoli®
- WebSphere®
- z/OS®
- z/VM®

The following terms are trademarks of other companies:

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

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

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

It goes without saying that 2009 was a year of unprecedented change in global banking. The challenges that financial institutions are facing require them to cut costs but also to regain trust and improve the service that they provide to an increasingly sophisticated and demanding set of customers.

In the past, siloed and rigid IT systems often inhibited banks in their attempts to re-engineer their business processes. The IBM® smarter banking initiative highlights how more intelligent software can be used to significantly improve the end-to-end integration of banking processes.

In this IBM Redbooks® publication, we aim to show how software technologies, such as SOA, Web 2.0 and event driven architectures, can be used to implement smarter banking solutions. Our focus is on CICS® Transaction Server, which is at the heart of most bank's core banking implementations.

The first part of the book is aimed at business leaders and introduces smarter banking, provides an overview the IBM Banking Framework, and then illustrates the value proposition with a set of smarter banking business scenarios.

In the second part of the book, we address the bank's technical leaders by providing an overview of the key technologies and describing how CICS supports these technologies today.

We use the IBM Montpellier smarter banking showcase in the final part of the book to illustrate how we enabled business event processing and Web 2.0 technologies in a non-invasive way without changing the applications:

- We show how we enabled the CICS core banking system to emit events for large transactions. We also show how these events are consumed by WebSphere® Business Events, which is configured to take a predefined action when a certain pattern of events occurs.
- We also show how the CICS core banking data, such as account transaction information, can be exposed as a mashable Web 2.0 feed so that the account holder has instant access to the recent transaction history.

The team who wrote this book

This IBM Redbooks publication was produced by a team of specialists from around the world working with the IBM Design Centre and Banking Centre of Excellence at the IBM location in Montpellier, France.

Chris Rayns is an IT Specialist and the CICS project leader at the International Technical Support Organization, Poughkeepsie Center. He writes extensively on all areas of CICS. Before joining the ITSO, Chris worked in IBM Global Services in the United Kingdom (UK) as a CICS IT Specialist.

Neil Ashworth is a Senior Certified IT Architect working in the IBM Design Centre, Montpellier, France. He has been with IBM since 2001. He has over 17 years of experience in the financial services sector, working on international banking client projects. Neil specializes in integration technologies and the IBM System z® platform.

Peter Beevers is an Executive IT Architect in the United Kingdom. He has over 18 years of experience in IT architecture and design, mainly in the Financial Services industry and has been with IBM since 2001. Pete has a Masters degree in Computer Science from Newcastle University and an MBA from the Henley School of Management. Pete has extensive experience in Banking systems and has worked with many of the leading Financial Services companies in the UK.

Vesna Eibel is a Certified IT Specialist working in the smarter banking showcase team, Montpellier, France. Her areas of expertise include z/OS®, Parallel Sysplex®, and the WebSphere family of products.

Fabrice Jarassat is a Certified IT Specialist working in the smarter banking showcase team, Montpellier, France. Before joining IBM in 2000, he worked for a large distribution company and managed the transaction systems that were based on CICS. He has a degree in computing from Ecole Superieure Informatique Professionnel, Paris. His areas of expertise include CICS, CICSPlex® SM, and CICS Tools.

Claus T Jensen is a Senior Technical Staff Member on the SOA Foundation team. The SOA Foundation team has architectural responsibility for all of the IBM software products to ensure that they support the key principles of SOA from a client perspective. This role requires both a broad and deep expertise on SOA concepts and practical use. Claus is the Chief Architect of Architectural synergies and lifecycles, driving the alignment between SOA, Business Process Management, Enterprise Architecture, and the interlock between SOA and Information Management. Prior to joining IBM in March 2008, Claus was Group Chief Architect, VP of Architecture and Business Development, in Danske Bank, which is a regional european bank. He drove Danske Bank's SOA initiative and

SOA center of excellence since its inception in 1999 and is known as an SOA expert and evangelist. Claus holds a PhD in Computer Science from Aarhus University, Denmark.

Alison Lucas is a Software Engineer in the CICS Transaction Server development organization based at the Hursley laboratory in the UK. She has over twelve years of software development experience of z/OS products, including WebSphere MQ and CICS TS.

Anthony Papageorgiou is a Software Developer in the CICS Transaction Server development organization based at the Hursley laboratory in the UK. Before he joined IBM in 2007 he worked for Mintel International Group Ltd as a Web Developer. Anthony has a degree in Computer Science from the University of Warwick. His areas of expertise include Web 2.0, Event Processing, and Mobile Technologies. He is also closely involved with the design and development of their support in CICS Transaction Server V4.1.

Alain Roessle is a Certified IT Specialist working in the IBM Design Centre, Montpellier, France. Before joining IBM in 2000, he worked for a large distribution company for 20 years. His areas of expertise include WebSphere, CICS, DB2®, and Parallel Sysplex.

Nigel Williams is a Certified IT Specialist working in the IBM Design Centre, Montpellier, France. He specializes in CICS integration technologies and service-oriented architectures. He is the author of many papers and IBM Redbooks publications, and he speaks frequently on CICS and WebSphere topics.

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

John Knutson, System z Product Marketing Manager - CICS Tools
IBM Hursley, for the original idea of writing this book.

Catherine Moxey, CICS Development, IBM Hursley, for her comments and assistance and guidance with CICS business event processing.

James Goethals, IBM Banking Center of Excellence, for his review comments and information on the smarter banking showcase.

Nick Garrod, CICS Transaction Server Marketing Manager
IBM Hursley, for assistance and guidance on the evolution of CICS and the content of CICS TS V4.1.

Roger Brooks, CICS Transaction Server Development Manager
IBM Hursley, for supporting the book by providing development resources and for his insights about business event processing scenarios.

Jean-luc Lepesant, Smart Banking environment team, IBM Montpellier, for assistance with running the CICS workload.

Santhosh Kumaran, IBM Banking Solutions, for information on the IBM Banking Framework.

Dr. Ali Arsanjani, IBM CTO, SOA and Emerging Technologies, for information on the SOA Reference Architecture.

Steve Wall, Yannick Le Floch, Guillaume Arnould and Marie-Angele Grilli, IBM Montpellier, for their support in reviewing the book.

Jonathan Lawrence, CICS Development, IBM Hursley, for his support in reviewing the book.

Peter Havercan, CICS Development, IBM Hursley, for information provided about CICS Web 2.0 support.

Helene Lyon, IBM France, for her comments and contributions on IMS™.

Isabelle Maudru, IBM Design Centre, Montpellier, for her support in planning the book’s residency.

Andrew Bates, CICS Development, IBM Hursley, for information on CICS fundamentals and history.

Ella Buslovich, IBM ITSO Graphics Editor.
KaTrina Love Abram, IBM ITSO Technical Editor.

Manon Jarassat, College Louise Michel de Ganges, for assistance in designing the virtual Smart Banking environment branch in OpenSim.

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

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications 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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/pages/IBM-Redbooks/178023492563?ref=ts

- Follow us on twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

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

Part 1

Smarter Banking

In part 1, we provide a business perspective on *smarter banking*, which is the IBM viewpoint on how financial organizations can implement smarter IT solutions.

We start with an overview of the principles of smarter banking and a short introduction to the key smarter banking capabilities, end-to-end process visibility and optimization, and an integrated user experience. In Chapter 1, “Introduction to Smarter Banking” on page 3, we provide an executive summary that you can read in isolation from the rest of the book.

We then discuss the modern banking imperatives and introduce the IBM Banking Framework, which provides a toolkit for smarter banking evolution.

We complete part 1 with a set of business scenarios that highlight how the key smarter banking capabilities apply to typical banking scenarios.

Part 1 is aimed mainly at executives, business leaders, and architects.

1

Introduction to Smarter Banking

Today we are witnessing an acceleration of market shifts. End-to-end visibility and horizontal integration across historically siloed systems are becoming critical enablers for business agility. For the finance industry, it is imperative that the classical computing environments, such as mainframes, evolve to embrace such horizontal integration capabilities.

Re-architecting decades of existing solutions is simply not a viable approach, so how do we achieve end-to-end process visibility and optimization in a non-invasive fashion? The answer is that the computing environment itself must provide capabilities, such as business event processing and integration on the glass, in a holistic fashion across vertical solution towers.

By re-inventing the classical mainframe as a modern process-aware platform, it becomes a full fledged participant in environments that are based on architecture paradigms, such as service-oriented architecture (SOA), Web 2.0, and event-driven architecture. Merging the classical mainframe virtues with modern architecture flexibility combines the best of both worlds and provides a natural evolution path for the valuable portfolios of existing finance solutions.

In this chapter, we describe the principles for smarter banking from a business perspective and discuss how to evolve existing environments towards a smarter banking paradigm. The primary audiences are executives, leaders, and architects who must understand how to effectively use classical finance industry computing environments as smarter banking differentiators.

1.1 The need for Smarter Banking

Economies and commodity markets are swinging rapidly, barriers to global competition are disappearing, and empowered customers are changing preferences and expectations faster than businesses can respond. At the same time personal, organizational, and business networks are becoming more interconnected, instrumented and intelligent, and our planet is literally becoming smarter. This new world presents tremendous opportunities, but to capture them financial companies must embrace the concept of Smarter Banking, as defined in Figure 1-1.

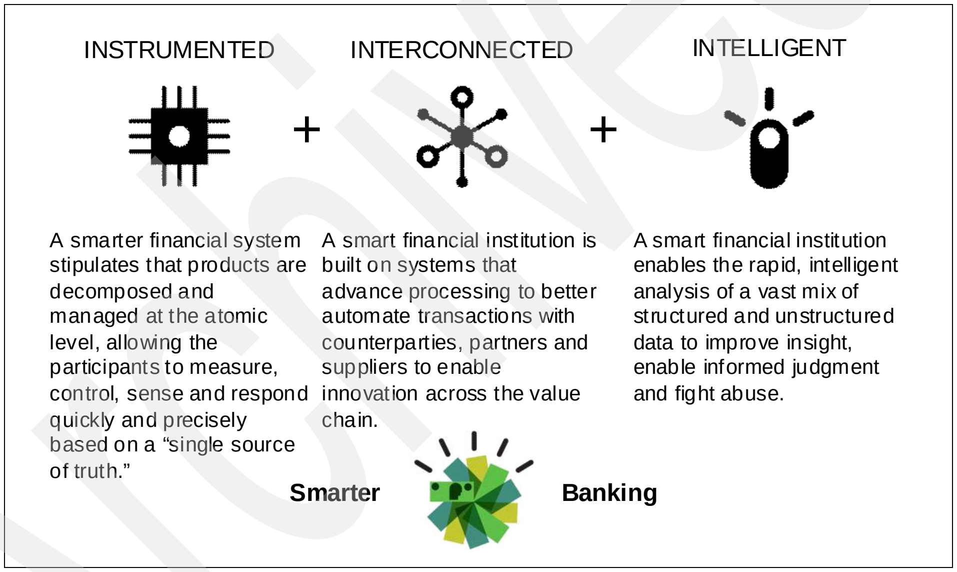

INSTRUMENTED
+
INTERCONNECTED
+
INTELLIGENT
A smarter financial system stipulates that products are decomposed and managed at the atomic level, allowing the participants to measure, control, sense and respond quickly and precisely based on a "single source of truth."
A smart financial institution is built on systems that advance processing to better automate transactions with counterparties, partners and suppliers to enable innovation across the value chain.
A smart financial institution enables the rapid, intelligent analysis of a vast mix of structured and unstructured data to improve insight, enable informed judgment and fight abuse.
Smarter
Banking

Figure 1-1 Smarter Banking

Important: A smart bank anticipates client needs and delivers innovative products more quickly and consistently than the competition. It can respond nimbly to changes in market conditions.

Too often businesses find themselves restrained from meeting the goals of smarter banking by siloed processes and rigid IT systems that inhibit collaboration and dramatically slow the process of change. In fact, as illustrated in Figure 1-2, inflexible and complex operations in many cases prohibit financial companies from focusing on their clients in a holistic fashion.

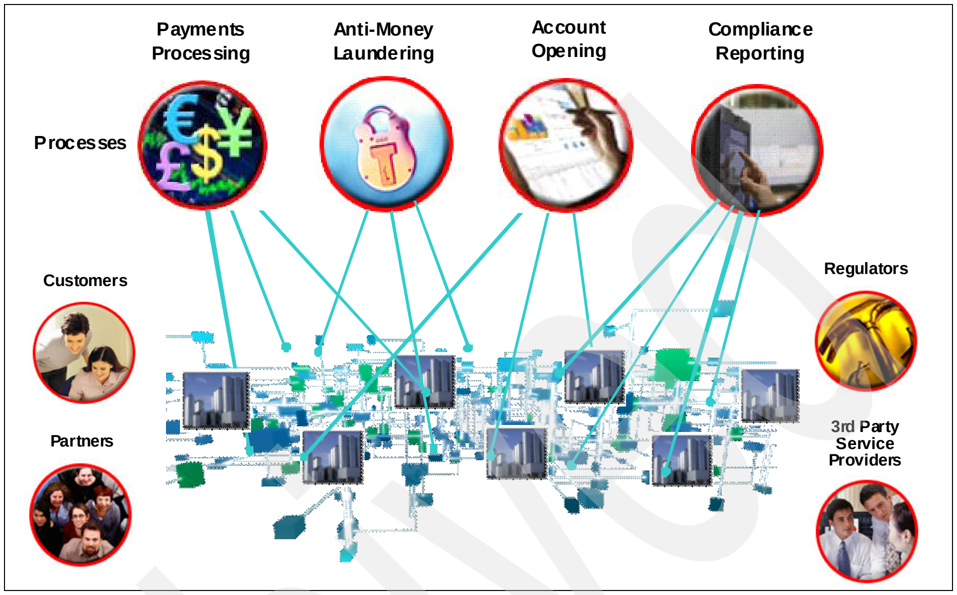


Figure 1-2 Siloed processes and rigid IT systems

Clearly smarter banking solutions must become more instrumented, interconnected, and intelligent, yet how do we achieve that end goal without disruptive change?

Historically, banking processes are bound by the IT systems that drive them, and integration is implemented through batch processing and data warehousing. While adequate from a transaction processing perspective, this approach does not cater to the emerging need for horizontal integration of business processes. Horizontal transaction processing is the ability to define and manage long running business processes in a transactional fashion, horizontally orchestrating the valuable portfolios of existing finance solutions.

From a horizontal integration perspective what becomes critical are end-to-end visibility and optimization, an integrated user experience, and a robust and scalable integration environment. Whether based on a classical mainframe platform or hosted in a distributed environment, any modern banking solution must provide all three capabilities, embracing technologies and standards, such as Web services, process orchestration, business event processing, Web 2.0, and so on. Most of these technologies are already enabled in CICS Transaction Server, and a similar evolution is occurring on other traditional computing platforms, such as IMS Transaction Manager.

1.2 End-to-end process visibility

According to Gartner, process improvements have been the number one concern of CEO's for the past four years. Studies, such as the recent McKinsey survey[1] and analysis of 100 companies in France, Germany, UK, and the US, show that aligning business and IT efforts results in double the productivity gains of those efforts in isolation. Yet smarter banking requires more than simple alignment of efforts; moreover, it requires a deep understanding of the business processes of the enterprise and the ability to execute change on these processes in collaboration between business and IT.

Re-architecting decades of existing solutions is simply not a viable approach to process improvement. So how do we achieve end-to-end process visibility and optimization in a non-invasive fashion? First, we must enable businesses to understand what is happening in time to actually make a difference. There are two distinct aspects of such understanding:

- Business event processing: Providing timely visibility into the state and progress of business processes
- Integration *on the glass*[2]: Providing a user-oriented view point with just enough information to make appropriate decisions

Both of these aspects must be provided by the horizontal integration environment in a holistic fashion across all vertical solution towers.

An Event Aware Enterprise processes event information continuously for business insight and action. Figure 1-3 on page 7 illustrates the evolution towards an Event Aware Enterprise.

1 [McKinsey]: London School of Economics, McKinsey survey and analysis of 100 companies in France, Germany, UK and the US.

2 The concept of integration on the glass means that the user can mash together various user interface components for a particular purpose without needing to change the underlying functionality.

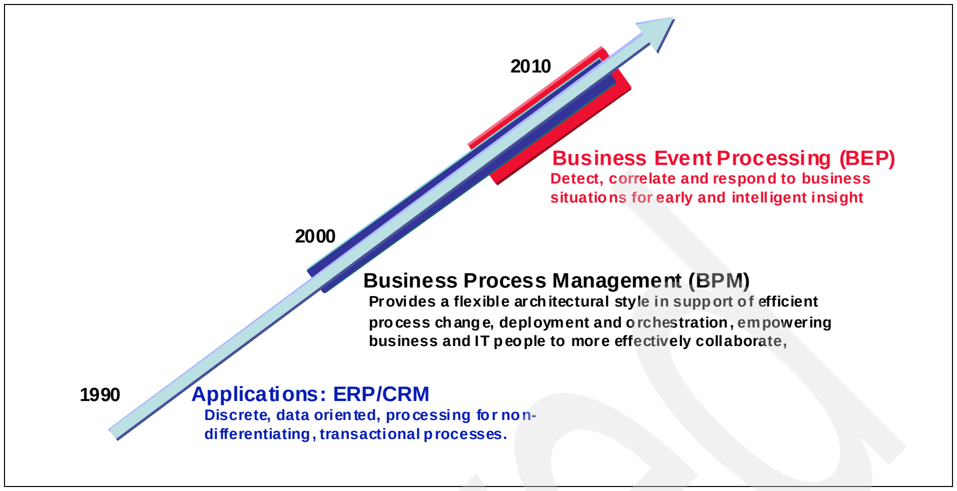

2010
Business Event Processing (BEP)
Detect, correlate and respond to business situations for early and intelligent insight
2000
Business Process Management (BPM)
Provides a flexible architectural style in support of efficient process change, deployment and orchestration, empowering business and IT people to more effectively collaborate,
1990
Applications: ERP/CRM
Discrete, data oriented, processing for non-differentiating, transactional processes.

Figure 1-3 Evolution of the Event Aware Enterprise

The ability to instrument existing solutions, allowing them to easily emit business events without disruptive change, is a critical enabler for the Event Aware Enterprise. A good example of how this can be done is the recently added business event capturing capabilities in CICS Transaction Server where the middleware itself automatically collects and filters events without requiring any change to the application. After the existing solutions are instrumented, we can collect the emitted events and use them to both drive horizontal business processes and to support intelligent decisions.

When the necessary operational visibility is in place, the next step is to provide the right information in the right context and tailored to individual user needs. The concept of a mashup originates with Web 2.0 as an *on the glass* heterogeneous integration of information, functionality, and user interface. A mashup by its nature provides a horizontally-integrated user experience, as illustrated in Figure 1-4 on page 8.

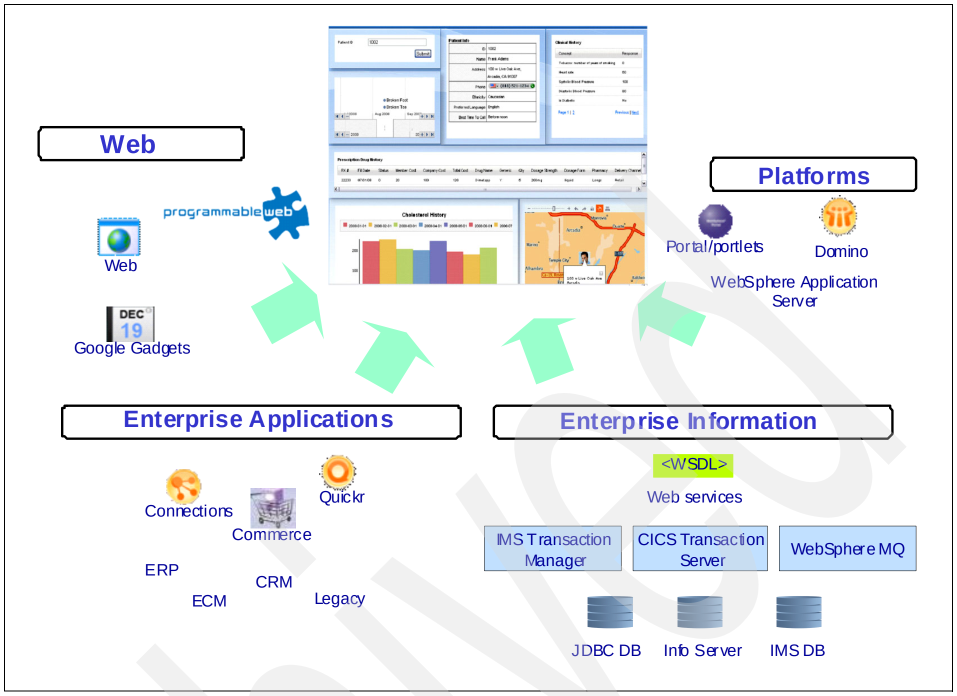


Figure 1-4 "On the glass" horizontal integration

Existing solutions that run on a multitude of various technology platforms, must expose themselves as mashable resources, which is not an easy task because many of these solutions were not built for such integration. In fact, the only non-invasive way of achieving the desired result is by having middleware and infrastructure transparently expose existing information sources and transactions in a mashable format that can be understood by whatever Web 2.0 environment was deployed, which is exactly the purpose of the recent Web 2.0 enablement of data on CICS.

After end-to-end process visibility is achieved, the foundation is in place for process improvement both in terms of current operations and in terms of process re-engineering. Without process visibility as a pre-requisite, any process improvement is part guess-work. In particular, in many cases the business processes that is inherent in existing solutions are not documented anywhere, hence the need to be discovered before they can be improved.

1.3 Robust and scalable information systems

Process improvement inevitably leads to change. Yet smarter banking is not just about change itself. Smarter banking is about embracing change and simultaneously retaining the classical virtues of well-managed enterprises. The IBM white paper, *Achieving business agility with BPM and SOA together*[3], introduces the notion of business agility, business performance, and business integrity as the three key differentiators for today's enterprises. IBM believes that to succeed, smarter banking must be based on robust and scalable information systems that ensure business performance and business integrity. If not, then business execution inevitably suffers.

The notion of transactional strength in information systems, which is the ability to execute and manage large numbers of concurrent transactions both consistently and efficiently, has been around much longer than the concept of smarter banking. Yet how do we form a smarter banking perspective mirror vertical enterprise solution characteristics in the horizontal business integration domain?

When properly executed, (horizontal) process improvement relies on the robustness and scaling characteristics of the classical enterprise platform. In other words, the classical computing environments of the Finance industry must re-position themselves in support of horizontal transaction processing. In fact, merging the classical mainframe virtues with modern architecture flexibility combines the best of both worlds and also provides a natural evolution path for the valuable portfolios of existing finance solutions. It is important to realize that there is no value to the business if optimized processes are not reliable, cannot scale to the demands of use, or leave critical business information vulnerable to corruption or misuse. Maintaining business performance and integrity in the face of change requires a reliable, adaptable, and scalable environment, technologically and organizationally.

The organizational scalability aspect is particularly interesting from a positioning perspective. In many cases, the specialized skill requirements for mainframe environments are used as an argument against their continued viability as a strategic platform. Yet the skill requirements are really related to the available development and management tools and not related to the underlying transactional strength of the platform itself. With the emergence of next generation tooling for the mainframe platform, such as IBM CICS Explorer™, this argument for perceiving the mainframe as an old school platform disappears.

[3] Achieving business agility with BPM and SOA together (WSW14078-USEN-00).

1.4 Summary

Successfully scaling a smarter banking initiative requires careful thought and consideration. From an organizational perspective, the enterprise needs horizontal visibility and appropriate skills to manage the business and IT operational environments. From a technological perspective, the enterprise must establish a platform that scales with the success of the smarter banking initiative and continuously ensures the integrity and reliability of business processes and services.

While much is being said in the marketplace about the need for business agility, the complementary need for business visibility, robustness, and scalability is often understated and undervalued. Agile change at the cost of operational excellence is a fragile value proposition at best. To ensure success, smarter banking must be based on transparent, robust, and scalable information systems. If not, then business execution inevitably suffers.

Providing end-to-end process visibility on a high-performance horizontal transaction processing platform will be a key differentiator for successful enterprises in their drive toward Smarter Banking. The IBM scalable, integrated tools and infrastructure are explicitly built to support transaction processing fundamentals, hence providing a good starting point and a solid foundation for the future.

2

Modern banking imperatives

The past two years were very traumatic for the financial services industry. Fundamental shifts in the nature and structure of banking require that banks throughout the industry rethink their relationships with clients, their costs base, and the flexibility of their platforms as they look to organic growth and new business models to regain profitability.

In this chapter, we outline several imperatives for banks, including the need to focus on clients, reduce costs, increase agility, and improve the management of risks. We then introduce the IBM Banking Framework as an approach that banks can adopt to help meet these imperatives. The framework provides a roadmap towards smarter banking and a set of prescriptive solutions that address the key areas of focus for banks: core banking transformation, payments and securities, integrated risk management, and customer care and insight.

We conclude the chapter with a look at how business event processing and Web 2.0 technologies can assist banks in improving end-to-end business process visibility and by providing a more integrated user experience.

2.1 Banking challenges

It goes without saying that 2008 and 2009 were a period of unprecedented change in global finance, whether retail, corporate, or wholesale banking. The numbers are stark:

- $10 trillion in government support globally for banks
- 51% loss in market capitalization of Financial Services sector globally. In various cases large banks lost over 90% for their market capitalization
- 150,000 financial sector job losses announced globally, which is likely to grow
- From July 2007 to March 2009 asset writedowns exceeded capital raised worldwide by 16 percent[1]

The political challenges are also significant, and everything from bonus remuneration to organizational structure and market dynamics are coming under scrutiny from politicians and social commentators, which can well lead to forced demergers and divestment by large banks. To add to the challenges that financial institutions are facing, their customers, both retail and corporate, are also under pressure. Their situation is exacerbated by the crisis. Unemployment rates are at multiyear highs, as are home repossessions, business failures, and loan impairments.

To see the current status in better context requires a review of banking strategy and operational behaviors over the past 20 years. The IBM Institute of Business Value identified a number of specific waves of activity and strategy within the global banking industry, as shown in Figure 2-1 on page 13. It is important to be aware of these phases to understand the current state.

[1] Global Financial Stability Report (GFSR).” International Monetary Fund. October 2009

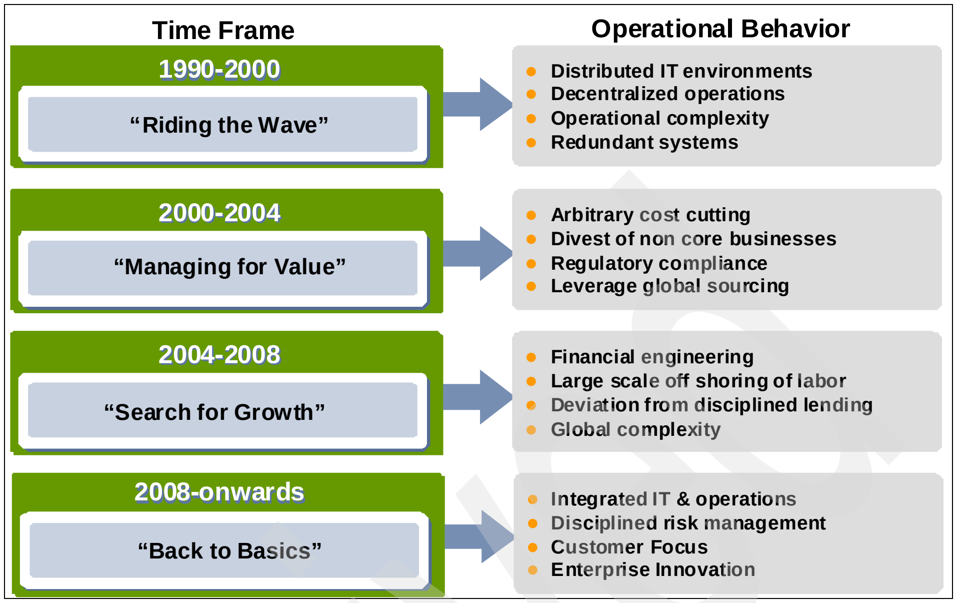


Figure 2-1 Banking industry phases

The various phases and changing banking strategies drive multiple approaches to integration and investment priorities:

- 1990-2000: Scale Growth

  This was the age of large scale-based mergers and acquisitions. Many banks made significant acquisitions of competitors and other organizations within the financial services value chain to increase scale and to get access to a greater market share. This activity was accompanied by a shift to non interest income and benefited from perceived risk transference through securitization.

  Operational behaviors drove decentralized operations, distributed processing, complexity, and system and process redundancy. A specific customer focus was not a primary concern during this period because banks sought to get bigger and more universal. The Internet revolution and the changing attitude of customers was creeping up on banks unaware.

- 2000-2004: Managing for shareholder value

  This period was characterized by dramatic cost cutting. Banks sought to drive out efficiency in operations by leveraging global sourcing and divesting of non-core businesses. Again, customer centricity and effective management of complexity was not core.

- 2004-2008: Searching for growth

  This period is characterized by financial engineering, using increasingly exotic derivatives and trading instruments, which increased the level of complexity and globalization in finance. These instruments allowed banks to lend significant amounts of money to companies and individuals, which fueled a credit-based economic expansion and housing boom in many countries. It can be claimed that this period was characterized by a deviation from disciplined lending, especially in financial markets. Many universal banks with retail divisions focused on the highly profitable derivatives markets rather than on traditional retail banking.

- 2008 onwards

  After the crash comes a desire to reinvest in the retail and commercial banking franchise, *getting back to basics* with a twist. The twist being an increased urgency for product and service innovation, coupled with consolidation and large-scale disruption in the marketplace. The present demands a refocus on the customer and on change and innovation in business models. In particular, banks must provide their customers with a more integrated user experience, which in turn requires that banking business processes are more horizontally integrated.

It is clear that banks of all types are operating in difficult times and face a new operating environment and a new set of strategic imperatives as a result.

2.2 New imperatives for banking

For banks to return to growth they require greater flexibility in their strategies, technology, and in their processes. The way the world works is changing and Banks must change accordingly. In a recent study by IBM, 90% of senior banking executives felt that the returns of the past were over and 67% ranked themselves as moderate to poor in business and technology agility[2]. Stiffer competition for incumbent high street banks can already be seen in various markets from supermarkets, other non-banks, and newly entrant multinational banks. All threaten to take business and margin from incumbents if they hesitate to embrace change.

The world is flatter, riskier, and smarter. Connections between firms and economies are more complex and more pervasive, and open trade and emerging economies are changing the traditional economic landscape. System complexity has increased along with the speed of information dissemination, which leads to a riskier environment that is more difficult to control, and as the gap between the

[2] IBM / CFA Institute Survey 2009. IBM Institute for Business Value analysis

availability of information and the ability to manage and horizontally integrate that information grows. Emerging technological advancements enabled the vision of a more instrumented, interconnected, and smarter world.

To address these challenges and improve their ability to compete in the post crisis world, IBM considers that banks must address the three emerging imperatives, shown in Figure 2-2.

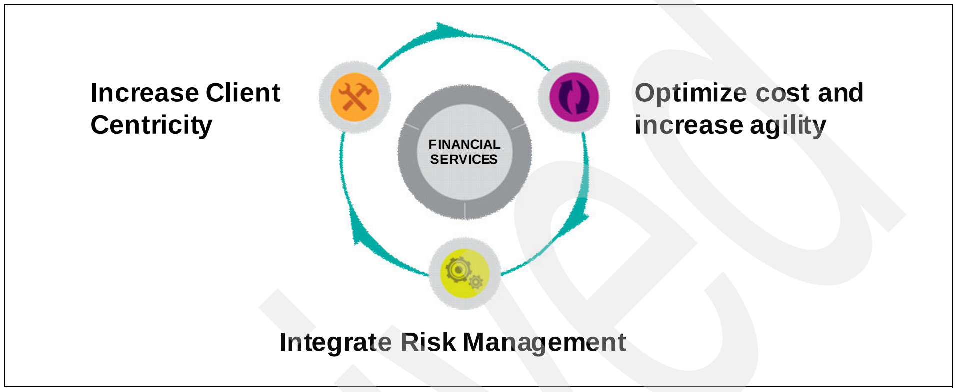

Increase Client Centricity
Optimize cost and increase agility
FINANCIAL SERVICES
Integrate Risk Management

Figure 2-2 Banking imperatives

The banking imperatives are:

- Increase client centricity: Reestablish trust and focus on developing clients as advocates.
- Optimize cost and increase agility: Reset the base costs and improve process and systems agility to address emerging business models.
- Improve Risk Management: Take a holistic view of risk management across business units, processes, and risk areas.

Meeting these imperatives requires horizontal integration and visibility across the enterprise.

2.2.1 Increase customer centricity

The banking industry has work to do to re-establish trust in client relationships. Recent analysis by IBM shows that clients do not trust banks to offer products that are in their best interests, and that there exists clear disconnects between what customers want and what banks think they want[3].

[3] Fit, focused, and ready to fight, IBM Institute of Business Value, 2009

To close this trust gap and to drive revenue and growth Banks must gain a deeper understanding of their clients. Customers today are more active and empowered and are much more willing to go elsewhere for financial products and services. Smart banks will invest in new solutions to gain insights into the wants and needs of their existing customers, which requires better integration across the banks' existing systems, which we discussed in 1.2, "End-to-end process visibility" on page 6.

Improved end-to-end process visibility helps banks to develop new products and pricing models that match what customers value and want from a financial services company. Customers want products at various price points and over multiple channel delivery mechanisms. Banks must understand and capitalize on this diversity.

As well as gaining new insights, banks must improve the way that they serve existing customers. They must develop and improve their customer care to ensure that they hang on to their most profitable customers and truly understand the profitability of a customer of the lifetime of their relationship with a Bank and invest accordingly. The IBM contention is that organic growth in retail and corporate banking requires a focus on customer advocacy. Classic drivers for growth focused on scale, pricing and fees, or cost containment and efficiency.

IBM believes Banks must focus more on developing capabilities for customer care, which develops clients who are advocates. Advocates are more likely to recommend an institution and will look to that institution first for new services. A recent survey has shown that advocates are more likely to take up cross sell offers (22% more products on average) and to develop longer term more profitable relationships with customers. By developing a more customer centric approach - investing in insight and care programs and focusing on the customer as an individual banks can close the "Trust Gap" and capitalise on organic growth opportunities.

2.2.2 Optimize costs and increase agility

Banking operations today are complex and inflexible. Systems and applications tend to be interlinked meaning one-off direct connections between functions are made among back-end data repositories and business processes. The result is costly inflexibility, where a change to one system or process creates the need to change the whole business process. Customer information is spread throughout the organization and can reside in various systems, which makes it difficult to correlate and analyze effectively.

Implementing new solutions in this environment can be risky and time consuming. As a result, banks can lose opportunities because they cannot change quickly enough to adapt to new market conditions. Banks must rethink their business models and radically simplify their core system's architecture to yield improved business agility to respond to a rapidly changing environment while ensuring that they aggressively take out cost and eliminate inefficiencies across the enterprise. Banks can develop more agile operating models and product engines to enable client responsiveness by leveraging common IT and operations platform to reduce costs to server and improve modernisation.

As their current revenue streams decline, banks will look to new sources of revenue to grow. Banks must improve their agility to take advantage of new selling opportunities. Smart banks must also streamline operations and renovate their heritage infrastructures and processes to achieve improved cycle times, which allows them to come to market faster with new products at lower costs.

Banks must better integrate front and back office functions, particularly in the areas of collections and recovery, the lending process, transaction processing, core banking system transformation, account opening, and client data management.

2.2.3 Improve risk management

The crisis of 2008 demonstrated in the clearest way imaginable that all banks, whether it be the Retail Banking divisions of complex universal banks or the smaller regional pure play retail banking organizations, must rethink the way that they measure, track, and control credit risk. Banks must develop smarter and more complete risk management systems to optimize operational, system, and credit risks. In particular, banks must improve their risk models and their approach to risk analysis, which includes more timely reporting on business risk across the organization and more transparency in risk reporting and capital adequacy.

Accurate, consistent, and transparent risk information across business units prevent unexpected losses, the need for higher capital requirements, and civil or criminal penalties. Integrated risk management enables proactive identification and management of emerging risks. Winners eliminate redundancies and improve risk data quality and speed of access within and across silos, vastly improve cycle-times of complex risk calculations, remove infrastructure inhibitors to information sharing, and improve the timeliness and quality of information that is available to senior decision makers across the enterprise.

2.3 The IBM Banking Framework

As the world becomes more interconnected and instrumented, all companies face new IT challenges. Each industry faces its own particular challenges, for example, the challenges facing the health care industry are different from those facing the retail or banking sectors. To address these differences IBM developed a number of industry-specific strategies to enable specific businesses to address the challenges that they face. In Chapter 1, “Introduction to Smarter Banking” on page 3, we introduced smarter banking as a vision and specific strategy for the banking industry.

An IBM industry framework is a recommended set of prescriptive solutions, services, and roadmaps that address specific business problems. They are composed of the currently available software assets, pre-integrated solution accelerators, partner applications, IBM services, reference architectures, and proven methodologies. The IBM Banking Framework is specifically designed to help organizations address the emerging imperatives around client care, agility, and risk. It is intended to help financial organizations optimize process change and to ensure that technology investment meets business needs quickly. The framework can be regarded as a roadmap for the smarter banking vision.

A framework approach has several advantages:

- It enables past experiences to be developed into repeatable deployments
- It provides a tool to simplify the analysis of complex problems and situations
- It reduces or eliminates repetition
- It speeds time to value for solutions

The IBM Banking Framework includes specific, proven solutions that are developed through working directly with several of the largest and most innovative clients of IBM to meet the strategic objectives of the Banking industry. The framework provides the necessary flexibility to deploy various solutions at the pace that makes sense for a particular business and allows for reuse over time and across business units. IBM believes this makes implementation faster and lowers risk compared to alternative approaches. Figure 2-3 on page 19 shows the framework offerings organized into four domains.

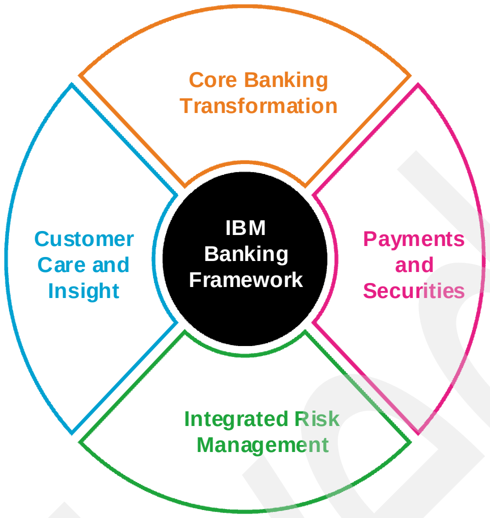

Core Banking Transformation
Customer Care and Insight
IBM Banking Framework
Payments and Securities
Integrated Risk Management

Figure 2-3 IBM Banking Framework

The four domains are:

- Core Banking Transformation
- Payments and Securities
- Integrated Risk Management
- Customer Care and Insight

These four domains reflect the key areas of focus for banks, that is, areas where both significant opportunities and need for change reside.

Note: The IBM Banking Framework does not cover every challenge that a bank faces; instead, it is a prioritized set of concerns.

2.3.1 Core banking transformation

Core banking is the bread and butter of retail banking. In mature financial markets, core banking is non-differentiating yet still accounts for a major part of revenue for most retail banks. One can argue that core banking has been commoditized to a degree where all that matters is buying and deploying an appropriate core banking package, which does not take into account the need for optimizing core banking processes, and it does not prepare the enterprise for things, such as product innovation, differentiated pricing, and customer personalization.

In the IBM Banking Framework, the core banking transformation domain is concerned with modernizing and renovating the mature applications that support core banking functions to align with and support the changing needs of the business. At the heart of core banking transformation is customer differentiation and business process agility and optimization, horizontally integrated across the portfolio of core banking solutions.

Figure 2-4 shows the various focus areas of the core banking transformation domain.

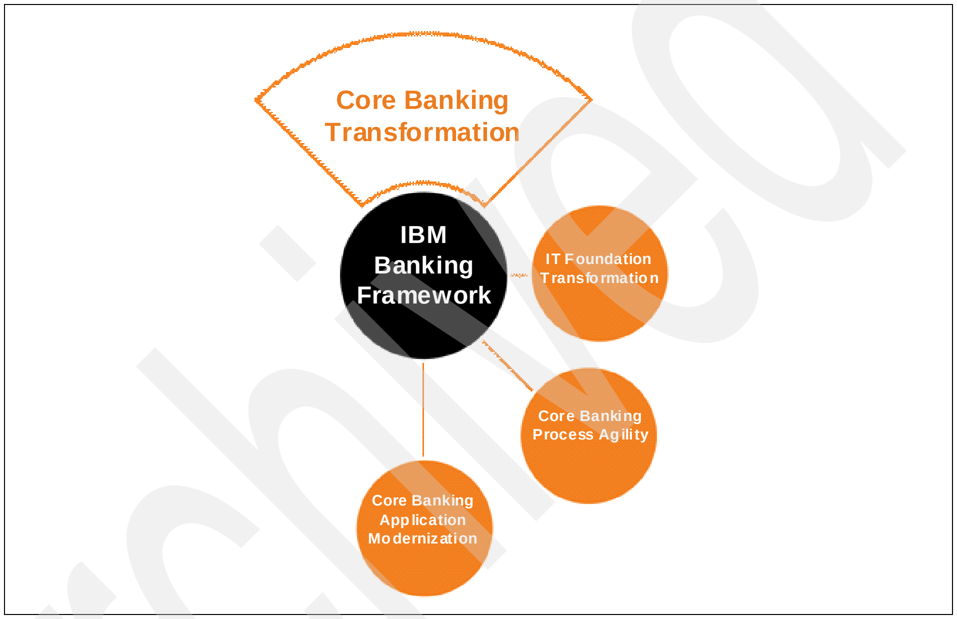

Core Banking Transformation
IBM Banking Framework
IT Foundation Transformation
Core Banking Process Agility
Core Banking Application Modernization

Figure 2-4 IBM Banking Framework: Core banking transformation

The areas of this framework domain are:

- IT Foundational Transformation

  Transform the core banking platform, reduce operational cost and risk, and improve core banking process efficiency with projects that address fundamental capabilities including a simplified IT infrastructure, platform scalability, enterprise-wide master data management for customer contract, product data, and model-driven development to build business service components.

- Core Banking Process Agility

  Improve profitability by expanding into new markets faster, bringing innovative products to market quickly, and differentiating pricing and terms for maximum customer satisfaction with flexible and efficient processes for account opening and management, product bundling, and dynamic relationship pricing.

- Core Banking Application Modernization

  Transition from existing core banking applications in a staged and modular way with near term payback and reduced risk and disruption, integrate best-of-breed application components, buy new packaged applications and integrate with enterprise systems, or build new SOA components for a customized, lower cost, and less risky approach.

2.3.2 Payments and securities

Payments and securities is a mixed domain in that traditional payment and security transactions are largely commoditized, and cash management products and complex derivatives are competitive differentiators. The revenue for payments and securities is mostly fee based, which means that volume and cost efficiency are critical earnings factors.

In the IBM Banking Framework, the payments and securities domain is specifically concerned with progressively transforming payments operations to become more flexible and efficient.

Figure 2-5 shows the various focus areas of the payments and securities domain.

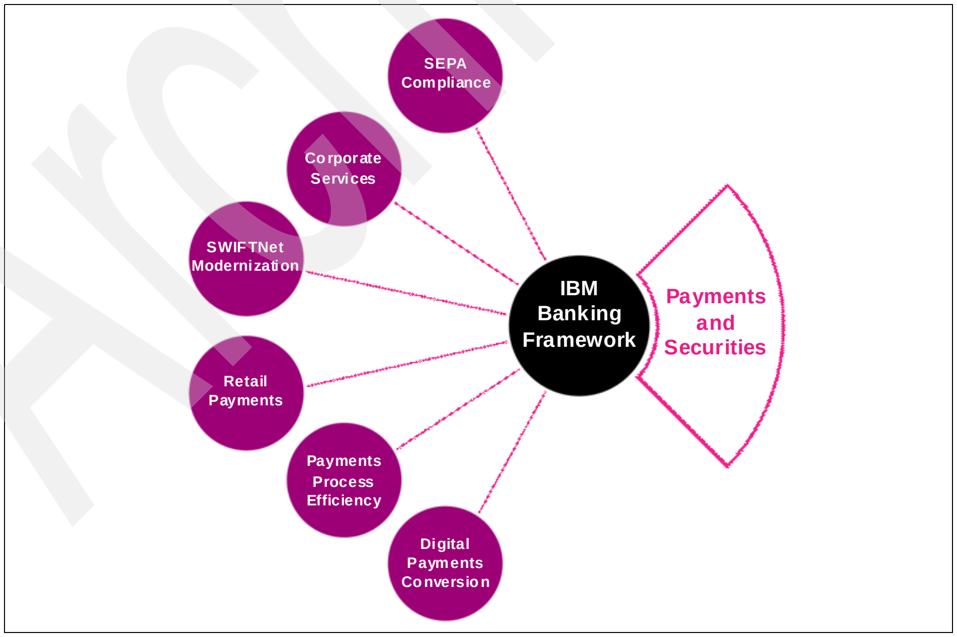


Figure 2-5 IBM Banking Framework: Payments and securities

The areas of this framework domain are:

- Corporate Services

  Integrate with your corporate customers' treasury operations to reduce working capital requirements and improve supply chain management.
- Retail Payments

  Migrate to a modern, more secure platform for ATM, mobile, and card switch functionality.
- Digital Payments Conversion

  Migrate from paper-based to digital processing capabilities to reduce costs.
- SEPA Compliance

  Restructure payments operations to comply with the European SEPA Direct Debit and SEPA Credit Transfer schemes.
- Payments Process Efficiency

  Drive efficiency in payments processing with business process management.
- SWIFTNet Modernization

  Modernize and upgrade your SWIFTNet operations for more efficient processing of high volume payments messages.

2.3.3 Integrated risk management

Risk assessment and management is an integral part of banking, and risk assessing individual transactions is routine in retail banks. The concept of integrated risk management, on the other hand, is still evolving. With financial products and transactions becoming more and more complex over time, the need for horizontal and integrated risk management is becoming ever greater. As evidenced by the current financial crisis, lack of integrated risk management can have dire consequences both from an enterprise and an industry perspective.

In the IBM Banking Framework, the integrated risk management domain takes a holistic approach to managing financial risk, financial crimes detection and prevention, operational and IT risk, and governance and compliance for more intelligent and useful insights. This domain integrates financial risk, operational risk governance, and financial crimes.

Figure 2-6 on page 23 shows the various focus areas of the integrated risk management domain.

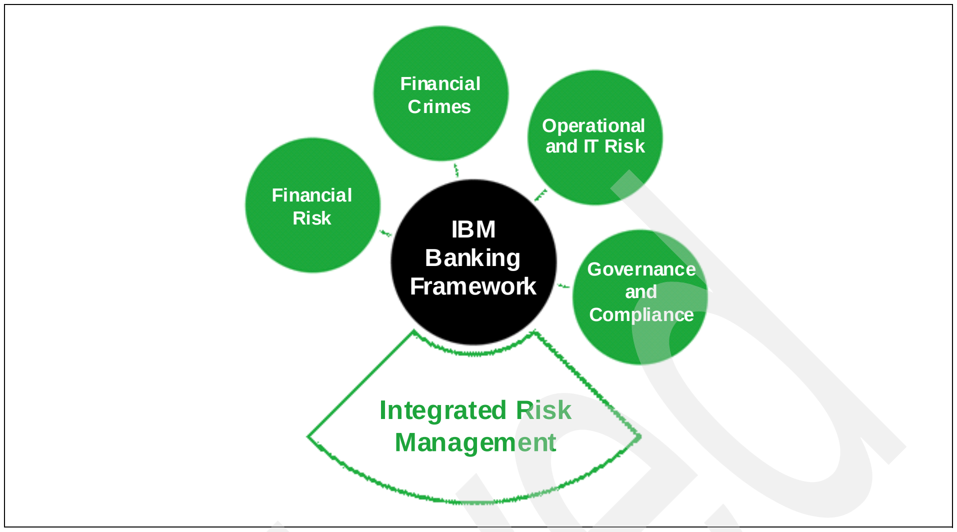


Figure 2-6 IBM Banking Framework: Integrated Risk Management

The areas of this framework domain:

- Financial Risk:
  - Connect market, credit counter-party, and liquidity risk to business and finance information systems
  - Enable risk reporting at various levels of the enterprise to enable better decision-making
- Financial Crimes:
  - Integrate disparate fraud platforms to move towards real-time and proactive monitoring and improve investigations
  - Leverage tools to identify fraudulent activity before it happens
  - Automate fraud prevention to reduce costs
- Operational and IT Risk:
  - Enable monitoring of internal processes, people, and systems for improved operational risk management
  - Implement early warning systems, IT and crises management processes, and business recovery planning
- Governance and Compliance:
  - Improve your ability to respond to regulatory scrutiny
  - Accurately report risk exposure

– Standardize and automate compliance processes
– Get the flexibility to respond dynamically to changing regulatory requirements

2.3.4 Customer care and insight

Customer care and insight is an emerging imperative in an environment where customers shop around more than ever before, and easy product comparison is enabled by the Internet. There are both defensive and offensive aspects of customer care and insight.

At a minimum, a financial company must protect engagement with existing customers by offering the right service in the right way at the right time. Having understood such customer dynamics, the insight can then be turned around for marketing and sales purposes, both for customer acquisition and for additional business with existing customers.

In the IBM Banking Framework, the customer care and insight domain is concerned with building the foundation for a single view of the customer and enabling more effective and efficient sales and service. Both of these are critical enablers for professional customer management.

Figure 2-7 shows the various focus areas of the customer care and insight domain.

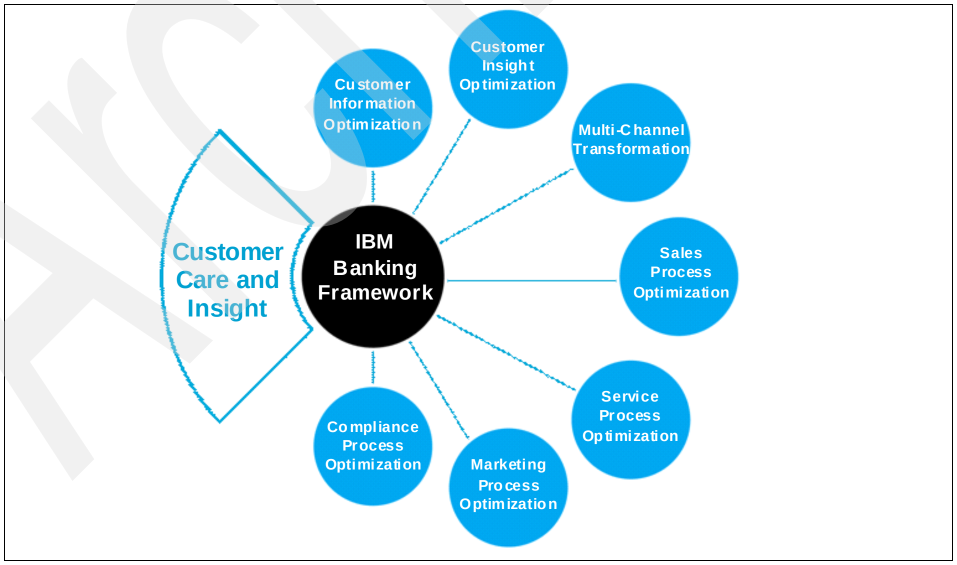

Customer Information Optimization
Customer Insight Optimization
Multi-Channel Transformation
Customer Care and Insight
IBM Banking Framework
Sales Process Optimization
Service Process Optimization
Compliance Process Optimization
Marketing Process Optimization

Figure 2-7 IBM Banking Framework: Customer care and insight

The areas of this framework domain are:

- Customer Information Optimization

  Create common data definitions, a common data warehouse, and integrate sources of disparate data to create an enterprise view of the customer
- Customer Insight Optimization

  Use business analytics to optimize insight at the point of interaction and better understand customer needs and preferences
- Service Process Optimization

  Leverage customer insight to optimize service processes, including case management, dispute management, and event-based decision-making
- Sales Process Optimization

  Leverage customer insight to optimize sales processes, such as account opening, lending, cross/up-selling, and dynamic product bundling
- Marketing Process Optimization

  Leverage customer insight for customer segmentation, targeted marketing communications, and campaign management
- Compliance Process Optimization

  Leverage customer insight for customer identification and customer preference processes
- Multi-Channel Transformation

  Transform your front office channels, including branch, internet, call center, kiosk, ATM, and mobile

2.3.5 Banking reference architecture

Figure 2-8 on page 26 outlines the reference architecture for the Banking Framework. Mapping solution areas to a reference architecture for Banking provides a mechanism for better understanding the framework and the capabilities that it provides because they can be aligned to various functional areas on the architecture.

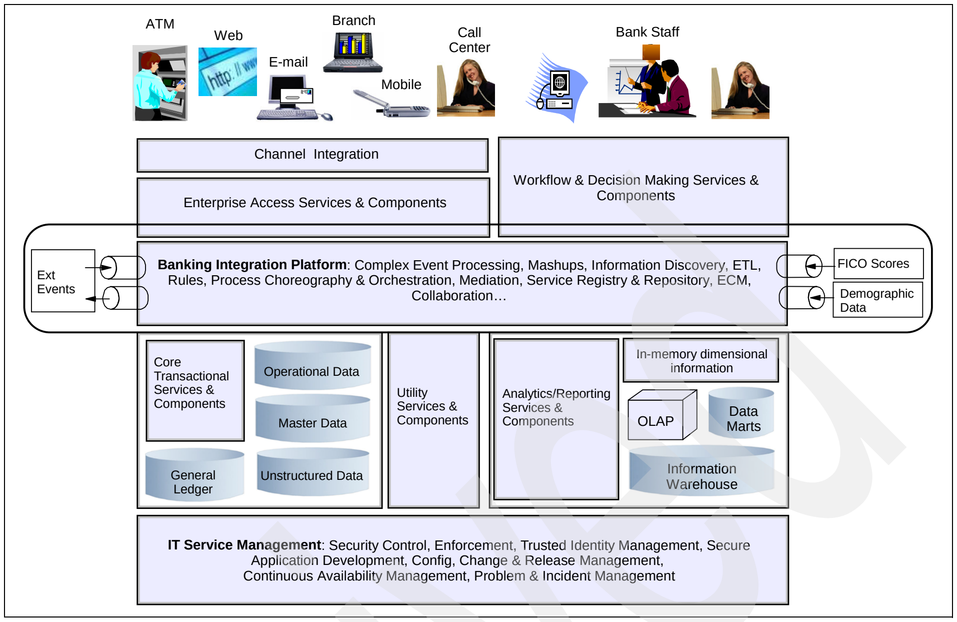

ATM
Web
E-mail
Branch
Mobile
Call Center
Bank Staff
Channel Integration
Enterprise Access Services & Components
Workflow & Decision Making Services & Components
Ext Events
Banking Integration Platform: Complex Event Processing, Mashups, Information Discovery, ETL, Rules, Process Choreography & Orchestration, Mediation, Service Registry & Repository, ECM, Collaboration…
FICO Scores
Demographic Data
Core Transactional Services & Components
Operational Data
Master Data
General Ledger
Unstructured Data
Utility Services & Components
Analytics/Reporting Services & Components
In-memory dimensional information
OLAP
Data Marts
Information Warehouse
IT Service Management: Security Control, Enforcement, Trusted Identity Management, Secure Application Development, Config, Change & Release Management, Continuous Availability Management, Problem & Incident Management

Figure 2-8 Banking Reference Architecture

The reference architecture is based on a service-oriented approach with each of the components providing a set of banking services. The major components of the architecture are:

- Banking Integration Platform

  The Banking Integration Platform forms the core part of any solution that is based on the IBM Banking Framework. It is a set of technologies that allow the other components to be integrated in an efficient and consistent manner. These technologies include process choreography, mediation and information discovery, and complex event processing and Web 2.0 style mashups.

- Core Transactional Services and Components

  These components provide transactional services that support the core banking business operations. Some examples of these components are Deposit and Loan product processors. An example of services provided are debit and credit transactions on a deposit account.

  Note: Many of these components exist today in transaction processing systems, such as CICS and IMS.

- Data components and services

  These are the primary three categories:

  - Operational Data

    Operational Data Stores are typically co-located with the components that execute business operations, for example, CICS core banking applications typically manage their own operational data stores in DB2.

  - Master Data

    In contrast, master data is externalized from the operational systems and managed across the enterprise. The master data management component manages master data that is associated with products, customers, and contracts. This component exposes services to create, read, and modify master data.

  - Unstructured Data

    All other data that is of a more unstructured nature.

- Analytical and Reporting Services and Components

  There are four levels of services in this category:

  - Reporting services
  - Data mining services that identify past trends from the history data
  - Predictive analytics that predict future behavior and simulate what-if scenarios
  - Business optimization that helps the bank to make the right decisions to optimize business performance

- IT Service Management

  These components provide an extensive set of services to monitor and manage the services and components in a framework-based solution, which includes security, trusted identity management, configuration, change and release management, continuous availability, performance management, and problem management.

- Workflow and Decision Making Services and Components

  These components provide services that enable the bank staff to perform the business tasks that are associated with the business operations of the bank, for example, a campaign management component provides services that enable the marketing, product development, and sales teams to collaborate on designing, launching, and managing campaigns.

- Customer Access Services and Components

  These components enable multi-channel delivery of the banking services to the customers. Some examples of customer access services are account opening, account access, account modification, mobile payments, and so on.
- Channel Integration

  The reference architecture supports access across a range of physical channels.

The focus of this IBM Redbooks publication is on the new capabilities of mainframe systems like CICS, such as business event processing and Web 2.0 capabilities, that can enable smarter banking. These systems are highlighted in the reference architecture and occupy part of the core Banking Integration Platform. As a core transactional component, CICS can enable banks to improve business processes across all four of the framework domains. However, before looking at the CICS capabilities in particular, it is useful to consider the general role of business event processing and Web 2.0 in banking.

2.4 Business event processing in banking architectures

An event is something that happens (or does not happen). A business event is anything that happens that is significant to your business. In IT terms, it has a name and normally a data payload, and it is generally produced and responded to asynchronously.

Some examples of business events are:

- A call to a help desk
- A sale is made
- A pin number is changed
- A delivery is made
- An inquiry is made on a product
- A customer's address is changed

Events can be simple, singular, and meaningful in themselves, or they can be complex in which case a complex event processing system is required to detect and respond to event patterns.

Enabling existing systems to emit business events provides valuable insight into business activities and is a cost efficient way to enable horizontal integration because it does not involve time consuming application changes. Whether it is the emission of simple events for monitoring purposes or complex event processing for correlation with events from across the organization, it is about understanding when actionable situations occur that extend the value of event processing across the information infrastructure.

Important: You can now enable CICS systems to emit events in a non-invasive way.

Complex event processing (CEP) involves looking at an event when it occurs and determining if the defined pattern has or has not taken place, for example, determining if other events or actions occurred, did not occur, or occurred within a certain time frame. CEP can be used for extracting information and meaningful insights from transactional or message-orientated systems. Until recently, CEP was the domain of low latency trading and real-time systems; however, technology change and business challenges are making it a more relevant technology for retail banking, which is driven by the same market forces that made the technology invaluable in capital markets, that is, reducing latency on actions and the exploding data volume.

The business value of event processing in retail banking is decreased latency in obtaining insights into customer behaviors, speed, and clarity of decisions that are made on those insights and automation and speed in execution of the inferred decisions.

Events from multiple systems can be processed and analyzed to look for particular patterns that might indicate potential breaches of regulations, fraudulent activity, or fleeting business opportunities. The event processing system must be able to receive data in various formats and turn it into actionable information, in real time, unencumbered by the volume of the data.

Event processing can provide a great deal of assistance and opportunities to banks as they try and meet the imperatives of the post crisis economy. A smart financial institution must constantly absorb, measure, and model large volumes of data from a wide array of sources. It must also apply this data quickly to ensure a streamlined, customer-centric experience.

Figure 2-9 on page 30 shows the complex event processing.

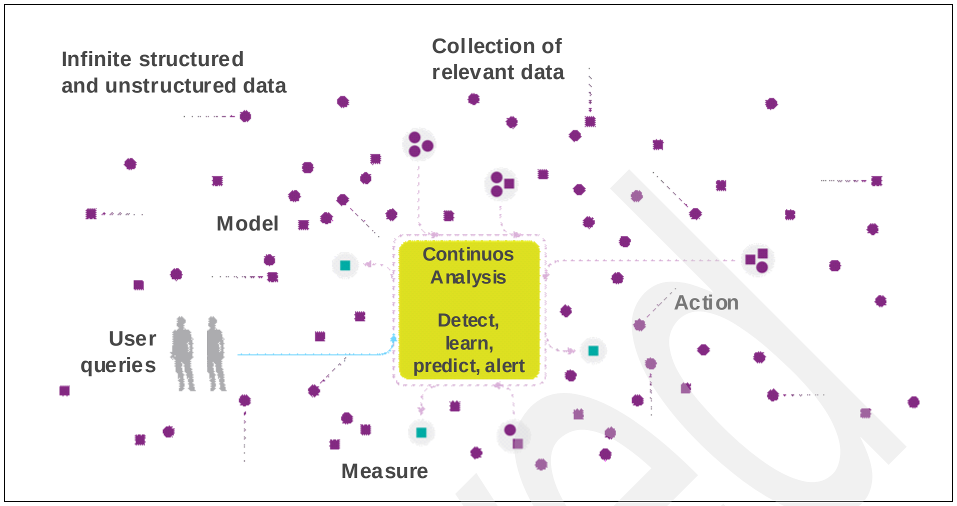

Infinite structured and unstructured data
Collection of relevant data
Model
Continuos Analysis
Detect, learn, predict, alert
Action
User queries
Measure

Figure 2-9 Complex event processing

2.5 The role of Web 2.0 in banking

Web 2.0 is the next frontier in customer relationship management and customer interactions. Customers are increasingly comfortable with the precepts, concepts, and capabilities of this environment and will start to demand that their banks interact with them in this way, for example, banks can provide additional value by opening up information in the form of feeds that their customers can compose with other data sources and display on new platforms and portable devices.

Banks that are intent on reconnecting with an increasingly sophisticated, fragmented, and untrusting customer must bare all[4] and use the power of Web 2.0 (on the glass integration) that is now at their disposal, which is especially the case when considering that banks must re-establish strong lasting relationships and trust. Customers are connecting, collaborating, sharing information and opinions in new ways, and are increasingly comfortable with read-write interaction through social networking venues. They are soaking up information about product, services, and experiences, which are not necessarily to the advantage of the financial institutions.

An ever increasing number of people visit a form of social networking or blogging site on a regular basis. In a recent survey, 60% of organizations see revenue benefits in Web 2.0 through attracting new customers and retaining existing

[4] Undressing in public: Harnessing the power of Web 2.0 to rebuild trust in banking, IBM 2009

ones. While 30% see savings that can be made in account management and servicing costs through use of Web 2.0 technologies.

In 2009, 16% of US households are estimated to have used an online financial planning tool and more are expected to use this in the future (estimated at 33% by 2016). Such interactions are increasingly likely to be delivered on smart phones. Some banks moved beyond the typical, offering more convenience and adding value through budgeting and cash flow applications, providing customers with a total view of their accounts (even those with other institutions) through a portal, which again is available when and where the customer chooses, for example, sites, such as Wesabe, marry the benefits of social networking and convenient, one-stop financial management to create added value and convenience for customers. Wesabe is an online site where users can upload information from banks and credit card accounts to monitor their spending and network with other people with the same financial goals. Banks increasingly must understand what it takes to be the intermediary of choice in the online marketplace. Web 2.0 provides numerous opportunities to attract customers.

Some institutions are already experimenting with a number of tools and techniques to create a channel-less customer experience. Contrary to the traditional online model, which is largely a transactional extension of the bank, the new online model uses all available technologies to nurture the customer experience. Banks and online providers are using mashable Web 2.0 style feeds and other capabilities to expose data in new ways. Some are moving toward full financial dashboards for expenditure analysis and wealth planning. But banks must start to move even further, thinking beyond channel and device, to envisioning an environment and culture that is defined and driven by the flexibility of the *anywhere*, *anytime* mobile experience, which reaches across customer segments, industries, and geography.

Important: You can now expose data from CICS core banking systems as mashable Web 2.0 style feeds.

2.6 Summary

To flourish in the *new normal* banks must focus on developing clients as advocates, increase agility, and at the same time optimize costs and improve risk management.

The IBM Banking Framework provides a roadmap towards smarter banking and a set of solutions that address the key areas of focus for banks. The framework combines industry knowledge, best practices, and software and tools, which can be used to build more flexible and horizontally integrated banking solutions.

3

Smarter Banking scenarios

In Chapter 1, “Introduction to Smarter Banking” on page 3, we introduced the powerful concept of Smarter Banking. It is a compelling value proposition, but clearly smarter banking is a long term journey, so where does that journey begin? The answer really depends on the most pressing needs of any given environment.

Our purpose in this chapter is to describe a structured set of scenarios that represent stepping stones towards smarter banking. While obviously not complete, these scenarios can serve as inspiration for a financial company that is planning a smarter banking roadmap.

3.1 Introduction

Leveraging the IBM Banking Framework, for each domain in the framework, we outline scenarios with a brief business case and an indication of how the key smarter banking capabilities (end-to-end visibility and optimization and an integrated user experience) apply as enablers.

In chapter 2, we introduced the four domains of the IBM Banking Framework:

- Core Banking Transformation

  Provides the tools and a business-driven approach to renovating lending, mortgage, deposit, and other core processing systems.

  Components of this domain include IT foundational transformation, core banking process agility, and core banking application modernization.

- Payments and Securities

  Progressively transform your payments and securities processing operations to become more flexible and efficient.

  Key components in this domain include SEPA compliance, SWIFTnet modernization, payment process efficiency, retail payment transformation, and corporate services.

- Integrated Risk Management

  Provides a framework to take a holistic approach to managing financial risk, financial crimes detection and prevention, operational and IT risk, and governance and compliance for more intelligent and useful insights.

  This domain integrates financial risk, operational risk governance and financial crimes.

- Customer Care and Insight

  Provides the ability to create a single view of the customer, enable sales, marketing and service and drive effective multi-channel delivery.

  Components of this domain include customer insight and service process optimization, compliance, multichannel transformation, and marketing.

In picking any one scenario, in particular in each domain, it is not to say that other equally important business cases do not apply, rather it is simply a practical measure to keep the total number of scenarios to a manageable number.

3.2 Core banking transformation

In the IBM Banking Framework, the core banking transformation domain is concerned with modernizing and renovating the mature applications that support core banking functions to align with and support the changing needs of the business. At the heart of core banking, transformation is customer differentiation and business process agility and optimization that is horizontally integrated across the portfolio of core banking solutions.

3.2.1 Product packages with differentiated pricing

The business case for this scenario is centered on the ability to define packages of existing financial products. A typical product package includes a checking account, credit card, Internet banking, and perhaps mobile banking. Such packages are marketed and sold as a unit. From the perspective of the Bank, the advantage is additional sales and reduced operational cost.

From the customer perspective, the value is in ease of use and having a pre-integrated set of products available to match their needs. Adding a differentiated pricing component significantly increases the attraction of the business base because optimized pricing on the packages can help drive sales and shape customer behavior.

Smarter Banking perspective

End-to-end visibility is needed across multiple separate core banking products—products that often reside in separate solution towers. The integration needs to be non-invasive and flexible enough for the product packages to evolve over time. A good approach to the required horizontal integration is to instrument the individual product systems to emit significant events and accept external orchestration of end-to-end business processes at the product package level.

From a customer viewpoint, the product package must appear as a unit, which often means re-crafting the related UIs. An attractive approach in such cases is to provide data feeds from the individual product systems, and then to do on-the-glass integration of the various feeds for an integrated product package experience, which is possible only if the existing systems that support financial products can expose their resources as mashable resources.

3.2.2 Legislation on consumer loans

This scenario is mostly defensive. There is no particular business case that is associated with this scenario because legal compliance is simply part of the cost of doing business. An example of consumer loan legislation is that a bank must provide the customer with certain information before it is legal to contractually establish a loan. In the case where the customer is in a branch office, it is relatively straightforward to implement operational procedures that has a bank advisor in the branch office provide the customer with the required information as part of the loans process. Only, it is becoming more and more common that loan processes are done over the Internet or even executed in a multi channel environment where several of the activities occur over the Internet, using the phone, and still others physically exchange documents. In such a heterogeneous environment, how does a bank prove beyond reasonable doubt that the consumer loan legislation was complied with?

Smarter Banking perspective

End-to-end visibility is needed across all of the various channels that are potentially involved in the loan process. It is virtually impossible to tightly orchestrate highly heterogeneous processes because there is simply no mechanism for controlling which channel the customer will use next. The fact that typically different IT solutions are being used to support various channels increases the challenge. Consequently horizontal visibility must be implemented in a way that is non-invasive from a channel perspective, retaining free choice of which channel a customer will use next.

A good approach is to instrument all transactions that are related to the loan process to emit events of completion, and then in a real time business event processing system set up filters and rules that provide real time alerts if and when there is danger of breaking the legislation guidelines. While not a complete substitute for embedding hard control logic in all loan-related transactions, the event based approach is both more lightweight and more flexible, in particular, if legislation regulations change frequently.

3.2.3 Accounts in multiple banks

The business case is centered on the trend towards customers having multiple accounts in multiple banks. What can be done for customers who want to see all of their bank accounts in one consolidated view?

From the perspective of a financial company, there are two reasons to be interested. First, if such multi account solutions become popular, it will be a significant competitive disadvantage for a bank not to participate in the mashup of multiple accounts. Second, in today's fragmented marketplace there is a continuous struggle to be perceived as the primary bank for any given customer, and becoming the provider of the multi-account mashup is one interesting way of attaining such status.

Smarter Banking perspective

A financial company must be able to act both as a provider of account information and as the on-the-glass integrator of the multi account overview. From a technology perspective, a good approach is to expose the account information as a mashable Web 2.0 feed. That feed can then be leveraged either by someone else's multi account overview solution, in the customers own mashups, or in a multi-account overview solution that the bank implements itself.

Web 2.0 account feed: We show an example of how to implement a Web 2.0 account feed in Chapter 9, "Enabling Web 2.0" on page 167.

A nice side benefit is that a Web 2.0-enabled account information feed can be used for other purposes than the originally intended multi-account overview, thereby providing additional value for Web 2.0-aware customers. As an example of this see 3.5.4, "Integrated budgeting" on page 44 for a scenario about integrated budgeting.

Obviously security is a concern when exposing account information in any format; therefore, we must make sure that account feeds are secure and can only be accessed from the account holder.

3.3 Payments and securities

Payments and securities is a mixed domain in that traditional payment and security transactions are largely commoditized and cash management products and complex derivatives are competitive differentiators. The revenue for payments and securities is mostly fee based, which means that volume and cost efficiency are critical earnings factors.

In the IBM Banking Framework, the payments and securities domain is specifically concerned with progressively transforming payments operations to become more flexible and efficient.

3.3.1 Notification for corporate payments

This business case is centered on providing real time updates to corporate banking and private banking customers over a mobile channel. Many corporate customers require real time updates on their accounts, especially at month end when payroll runs are processed, and they require the ability to view and authorize payment transactions on the move. These customers are travelling executives who might need to approve vendor payments. Mobile authorization workflow can aid in not only providing mobile information but also authenticating decisions, such as purchasing and selling financial assets.

By surfacing events in a secure way to mobile devices, banks can offer differentiated payment services to particular customers who need up to the minute information to control company cashflow and liquidity.

Smarter Banking perspective

While an event based architecture is not required to address this requirement, using such a solution offers a *light touch* and avoids application changes to systems that are often hard to maintain. It also offers a common approach across multiple disparate systems. Payment events can be emitted, correlated based on certain conditions, and an action to send a secure notification to the customer can be automated.

3.3.2 SEPA compliance

The Single Euro Payments Area (SEPA) initiative involves the creation of a zone for the euro in which all electronic payments are considered domestic and where a difference between national and intra-European cross border payments does not exist. The project aims to improve the efficiency of cross border payments and turn the fragmented national markets for euro payments into a single domestic one.

SEPA enables customers to make cashless euro payments to anyone who is located anywhere in the area using only a single bank account and a single set of payment instruments. Being SEPA compliant is a must for most European banks, hence there is no particular business case that is associated with it.

SEPA is based on a set of XML message formats that are defined by the ISO 20022 (UNIFI) message schemas, for example, for credit transfer and direct debits.

Smarter Banking perspective

Some banks implemented vendor SEPA packages, several changed existing payment systems, and others chose a mix of approaches. In most cases, new SEPA payment components must integrate with existing payments solutions that run in transaction processing systems, such as CICS. This implementation accelerated the number of mature integration projects in recent years.

Because SEPA solutions are oriented towards processing batch files of XML-based payments, many customers choose to use the CICS Web services support to integrate new SEPA components and existing CICS payment modules.

There is a horizontal visibility requirement because various types of payments all need to be SEPA compliant, yet it is not clear how to realize that component in a smarter fashion. Ultimately, end-to-end process visibility, while a necessary element of a SEPA solution, is not sufficient in and of itself. Technologies, such as business event processing, data feeds, and on the glass integration, can all play a role in a SEPA implementation, yet in the end must be combined with changes in core payment formatting and processing capabilities.

Note: We included this scenario to illustrate that smarter banking must include the skills to appropriately apply multiple approaches and technologies to any particular business scenario and to point out that often there is no uniquely correct answer to the question of what is the best solution or approach.

3.4 Integrated risk management

With financial products and transactions becoming more and more complex over time, the need for horizontal and integrated risk management is becoming ever greater.

In the IBM Banking Framework, the integrated risk management domain supports taking a holistic approach to managing financial risk, financial crimes, and operational and IT risk and compliance.

3.4.1 Improved credit card fraud detection

The business case is centered on the ability to detect credit card fraud either better or faster. From the perspective of the financial company, the advantage is obvious, reduction of losses due to fraud. From the perspective of the customer, there is (indirect) value too in terms of an increased feeling of security and a faster handling of fraud cases.

Credit card fraud detection needs are not new, and there are many existing solutions that cater to such needs. On the other hand, the level of inventiveness and complexity in credit card fraud schemes has continuously risen over the years and as a consequence losses have mounted.

Smarter Banking perspective

The challenge is two-fold:

- First, the growing need for fast reaction.
- Second, the level of dynamicity needed in fraud detection rules is also increasing as attempted fraud schemes change faster.

One good way to address both needs is to switch to an event-based paradigm for fraud detection instead of the typical transaction-based paradigm. The advantage of an event-based solution is that events can be collected in real time from multiple sources and then correlated (also in real time) in a business event processing system using adaptable filters and rules. The business can change these filters and rules without IT involvement.

3.4.2 Detecting check kiting

Check kiting is the illegal act of taking advantage of the float to make use of non-existent funds in a checking or other bank account. It is commonly defined as writing a check from one bank knowingly with non-sufficient funds and then writing a check to the other bank, also with non-sufficient funds, to cover the absence. The purpose of check kiting is to falsely inflate the balance of a checking account to allow checks that were written, which otherwise bounces, to clear essentially “robbing Peter to pay Paul”.

While check kiting, in many cases, can be detected by transaction patterns on a single customer account, this is by no means a foolproof method.

Smarter Banking perspective

A more efficient way to detect and defeat check kiting is to provide cross bank visibility, horizontally integrating knowledge about certain types of transactions across account boundaries. Clearly, classical application or data integration schemes do not apply to such horizontal integration across banks. Standardized event processing on the other hand can be a good fit due to its non-invasive nature and the ability to standardize event formats and consolidate in a common event processing system.

Imagine, for instance, a case where the finance sector in a given country implemented a common standard for check processing events and established a centralized event processing system for identifying check kiting patterns. In such an environment, it is easy to scan for parallel (otherwise uncovered) checks and take appropriate action.

3.4.3 Market risk management

Market risk is one of three key risk parameters for a financial company (the others are credit risk and operational risk). Market risk management systems have existed for decades and performed relatively adequately over time. Yet, as witnessed by the credit crisis that began in 2008, various types of financial collaterals are becoming more and more interconnected, and a clean separation into the three main types of risk is becoming harder and harder. Bonds are based on mortgages that are essentially a credit risk. Complex derivatives have dependencies on counter-party credit and liquidity risk. The examples of interdependent risk factors are numerous, and that situation is not getting better.

Smarter Banking perspective

It is imperative to provide risk dashboards with horizontal integration of the three classical risk types: financial, operational, and criminal. Floor traders with better integrated risk information can set sharper equity prices and avoid overly risky trade transactions, thereby providing additional top-line and bottom-line growth, which requires both integrated risk-related data feeds and on-the-glass integration into a risk Dashboard. Most traders have their own preference on how to set up the combination of information about their main trading window.

3.5 Customer care and insight

Banks are trying to change the relationships that they have with their customers, which is imperative both to close the trust gap that exists between banks and their customers and also to ensure that Banks understand the wants and desires of customers sufficiently to offer them appropriate and valuable products.

In the IBM Banking Framework, the customer care and insight domain is concerned with building the foundation for a single view of the customer and enabling more effective and efficient sales and service. Both of these are critical enablers for professional customer management.

3.5.1 Account overdraft protection

The business case is centered on improving account overdraft protection, either simply as an improved service for existing customers or in several cases potentially as a competitive differentiator. Typically, overdraft charges are calculated at the end of each day. Where an account has just gone overdrawn the bank might want to take immediate (and automatic) intraday action to inform the customer that they can avoid bank charges by topping up the account immediately.

Smarter Banking perspective

There are two key aspects of this scenario:

- Horizontal visibility across all channels
- Timely action either before or immediately after an overdraft occurs

Clearly a batch-based solution is not viable because there is an inherent need to implement real time consolidation of all transactions on the same core account.

In the case where all channels already have real-time integration with the core account, a solution can be devised using traditional means to trigger a notification (through email, SMS, and so on) or to issue a real time alert before the offending transaction is even processed.

In the case where such integration does not exist, for instance many Internet banking systems are not integrated in real time with the bank's core account systems, an alternative solution is an event-based approach to horizontal visibility. Such a solution is attractive in that it provides non-invasive real time visibility, yet it does have the drawback that the event processing system is forced to keep track of currently available funds, and keeping track of historical sums is a task that is better suited to business and data logic than event logic.

3.5.2 Vacation-related credit card sales

The business case is centered on the ability to provide the right product for the right need, at the right time. More often than not customers are not aware of all the options that are available to them in terms of financial products; therefore, they apply for non-optimal products, explore multiple options, overall behaving in a fashion that if detectable indicates that some financial advice is in order. Detecting such situations is difficult because the signs are often insidious; nevertheless, being able to propose the right product for the current customer need is a competitive differentiator that leads to both more satisfied customers and additional sales.

As a particular instance of this scenario, consider the case where a customer is planning a big vacation for the entire family. Cash is a bit short, so the customer goes online to take out a loan, and initiates an application for a $5000 loan over 12 months. The customer is interrupted and never completes the loan application. A couple of weeks later the customer goes into a local branch and wants to change some travel money into euros. $2000 is changed, and this money is taken from the current customer account. The next day, early morning, the customer returns online to again pursue the loan application, but again does not complete the application.

From the perspective of the bank, what is going on here? If we can track events like these and correlate events, such as the two aborted loan applications and currency transfer, we can make informed decisions that lead to happier customers and increased sales.

In this particular scenario, it is a reasonable guess that travel plans are involved, yet a consumer loan is not really the optimal way of funding international travel, so why not use an international credit card? Clearly the customer is not aware of that option, having not used an international credit card before; consequently, that same day and immediately after the second aborted loan application, a call center agent contacts the customer and based on a dialogue in real time suggests the most appropriate credit card for what was in fact (as guessed) an impending family vacation. The customer is happy, did not incur loan charges, and gets more flexible payment options to use when he travels. The bank is happy because of the credit card sale and because of the increased customer satisfaction.

Smarter Banking perspective

Emitting business events to horizontally collect and correlate seemingly unrelated customer actions (in a non-invasive fashion) is a good way of addressing the challenge of better understanding and anticipating customer wants. Furthermore, applying an event processing system to set up filters and rules on the emitted events allows the business organization a much greater degree of independent control and flexibility than if such logic had to be built into traditional IT transaction systems.

After the customer's event pattern is captured, a call center agent can be scheduled to call the customer and, based on the dialogue, suggest the most appropriate credit card for the impending family vacation.

3.5.3 Customers of interest

The business case is centered on the ability to detect and contact customers of interest when they are in a branch office. There can be various reasons for identifying a customer as of particular interest, for example:

- A policy to encourage customers to perform transactions without visiting the branch.
- A bank internal policy that a branch manager must personally greet high-value customers.
- An advisor wanting to follow up on previous unfinished business, for example, an outstanding request for a mortgage.

Smarter Banking perspective

Detecting and acting on customers of interest when they are in the branch office requires both horizontal visibility and real time action. Customers of interest are often not initially in contact with the person in the branch that really needs to speak with them. Also, they are not in contact with any branch personnel when they are simply in the branch office to use an ATM, for example, consider the case where a customer does repeated teller account transfers over a period of a couple of months. The bank wants to be able to identify this behavior pattern and suggest an Internet banking solution the next time the customer is in the branch for an account transfer. Technically this can be done using traditional technology solutions, running queries in a data warehouse environment, and setting up leads for the teller to act on. From a smarter banking perspective, however, it is desirable to be able to set up and continually adjust multiple policies on when to suggest each self-service solution. A good approach is to instrument all service transactions (across all teller applications) to emit an event that is fed into a central event processing system that contains the filters and rules for self-service solution suggestions. The event processing system can then be integrated into the teller desk top to provide real-time suggestions based on both previous behavior and the event that is emitted from the currently executing service transaction.

Another example is a customer that changes his typical banking pattern and starts to regularly perform high-value transactions. When such a customer arrives at the branch, perhaps to deposit a large cheque, it might be appropriate for the branch manager, or the customer's account advisor, to discuss new sales opportunities.

Advanced technology solutions, such as having RFID tags in credit cards, detecting the RFID in the card, and emitting an event when the customer enters a branch are feasible. As are non-invasive instrumentation of traditional ATM and teller systems to emit customer transaction related events. All of these events can then be fed into an event processing system for real time correlation and action.

Note: We provide an example of how to implement a customer of interest scenario in Chapter 8, "Enabling business events" on page 139.

3.5.4 Integrated budgeting

The business case is centered on the customer need for feeding account transaction information into their favorite budgeting solution instead of manually typing in all of the transactions. Traditionally, such budgeting solutions have run on PCs, providing a range of import options. Yet with the proliferation of Smartphones and other computationally capable devices, transaction information can now be consumed by a wide range of devices. In particular, the portable devices are an interesting case because traditional file-based solutions for importing transaction data simply do not work in this case.

Smarter Banking perspective

At a minimum, a bank must be able to act as a provider of device-neutral account information. From a technology perspective, a good and flexible approach is to expose the account information as a mashable Web 2.0 feed. That feed can then be leveraged by any budgeting solution that is Web 2.0 enabled.

A nice side benefit is that a Web 2.0-enabled account information feed can be used for other purposes than the originally intended budgeting solution, which provides additional value for Web 2.0 aware customers, for example, see the scenario 3.2.3, "Accounts in multiple banks" on page 36.

3.6 Summary

From a business perspective, the scenarios that we presented in this chapter illustrate the breadth of concerns for Smarter Banking. Realizing any of them provides value to a financial company on a smarter banking journey. What is important to observe is that all of these scenarios fundamentally rely on one or both of the key smarter banking capabilities, end-to-end visibility and optimization, and an integrated user experience.

From a technology perspective, the scenarios illustrate how business event processing and Web 2.0 feeds can serve as critical enablers for end-to-end visibility and the integrated user experience respectively. Thus it is no coincidence that these technologies are key components in the reinvention of the classical mainframe as a modern process aware platform.

Part 2

CICS Transaction Server

In part 2, we focus on CICS Transaction Server and its role in the evolution towards Smarter Banking.

We start with a review of the traditional role of CICS as a transaction processing environment and a short look at its 40 year history at the core of many financial systems. We then introduce the major capabilities in the latest version, CICS TS V4.1. We focus on the major technologies that enable smarter banking, namely CICS support for event processing and Web 2.0 enablement.

4

CICS and banking

In this chapter, we focus on the role of CICS both from its position as a dependable and cost-effective platform for building modern banking applications, and also as an enabler for the evolution towards smarter banking.

4.1 Traditional role of CICS in banking

Why do a large number of banks bet their businesses on CICS? There is no single answer to this question. The answer is partly historical (better the devil you know), but it is mostly due to the proven abilities of CICS as the IBM flagship TP monitor coupled with the unique qualities of the mainframe.

4.1.1 Why the mainframe

In 1.3, “Robust and scalable information systems” on page 9, we explain that it is no good introducing new technologies that improve end-to-end business process visibility if the deployment of these solutions are not based on robust and scalable information systems. The System z mainframe is the platform of choice for the banking industry because it has exactly these unique qualities that make it ideal for the types of high volume mixed workloads run by banks.

Banks today are being driven by multiple forces to change their IT infrastructures. Cost pressures are continually rising as a result of under utilized IT assets and increasing operational complexity. Service expectations have never been higher with customers demanding continuous service availability and higher quality user experiences across a range of application services. New security risks and threats are emerging from the accelerated pace of business change and the blurring of traditional infrastructure boundaries, and a host of smarter and more adaptive technologies, including cloud computing, virtualization, and Web 2.0, can be leveraged to drive innovation. These challenges demand a dynamic infrastructure with capabilities to help reduce costs, improve service, and manage risk.

System z technology can play an important role in meeting today’s infrastructure challenges:

- To reduce costs, for example, System z offers industry-leading virtualization and large scale consolidation capabilities to drive down the costs of managing and maintaining the proliferation of servers. Business process management software for System z can streamline business activities to improve operating efficiency and lower costs.
- To improve service, System z allows just-in-time deployment of resources to respond dynamically to changing business priorities. Enterprise modernization software specifically enables more rapid and flexible responses by allowing System z users to create and reuse mainframe services and applications as Web services. Sophisticated data management capabilities help to ensure that information is consistently available when needed without compromising data security.

- To manage risk, System z with comprehensive security capabilities builds on a long-established reputation as the platform of choice for mission critical workloads.

Dynamic infrastructure: A *dynamic infrastructure* provides the flexibility that businesses need to quickly add capacity and workloads without disruptions – while reducing costs, managing risks, and improving service.

Some of the main System z unique qualities that enable a dynamic infrastructure are:

- Architecture

  The System z architecture embodies a balanced design that delivers maximum overall throughput with embedded reliability, availability, and serviceability (RAS) capabilities right in the processor chips themselves.

  Although the latest System z servers have one of the fastest commercially available processors on the market, they also dedicate a significant amount of silicon chip space to processor memory, and they devote additional processor resources entirely to I/O bandwidth.

  The System z architecture dramatically reduces the time that the system takes to move data in and out of memory and the time that it takes for that data to travel to and from peripheral devices, such as disk. As a result, the throughput of real business applications that submit high volumes of create, read, update, and delete requests against very large databases (as is usual in banking applications) is unparalleled on the System z platform.

  This balanced performance makes it much easier to run multiple business applications and databases concurrently on the same system. System z customers routinely run their machines at over 80 or 90 percent utilization all year round.

  The System z architecture continues to evolve, for example, the System z10™ processor adds numerous new instructions, primarily concentrated on improving the efficiency and performance of compiled code. The z/OS Java™ SDK now exploits these additional instructions when running on a z10.

- Virtualization and scalability

  System z servers are inherently highly virtualized environments and by consolidating applications and data on System z, organizations can dramatically improve utilization and related cost savings. Its advanced virtualization capabilities support the highest utilization rates, and its modular and efficient design results in a much smaller footprint. A single System z10 can do the workload of up to hundreds of distributed servers with greatly reduced energy demands.

It is the partitioning and hypervisor technologies that make this virtualization unique to System z. The partitioning approach allows a subset of the computer's hardware resources on a physical machine (processor resources, memory, and I/O bandwidth) to be partitioned across multiple logical partitions (LPARs) with each housing a separate operating system. System resources can be dynamically balanced across LPARs with the system automatically assigning more processor resources or I/O channel bandwidth to LPARs according to predefined service level agreements.

Parallel Sysplex technology enables software and applications that run in separate LPARs to conduct parallel processing using the same cluster wide state data (for both read and write access) using a direct memory access-like mechanism, even across physically separate System z servers. This Parallel Sysplex is perceived as a single system image that can be managed from a single point-of-control.

In recent years, distributed platforms began to implement partitioning techniques. However, the ability to create a truly shared computing-system complex is unique to the System z platform.

- Security

  System z security is one of the main reasons why the world's top banks rely on the IBM mainframe to help secure sensitive business transactions.

  For a bank to remain flexible and responsive, it must be able to give access to its systems to internal employees, customers, and partners and still require the proper authorization to access financial systems and data. The bank must provide access to the data that is required for the business transaction, but also be able to secure other data from unauthorized access. The bank must prevent rogue data from being replicated throughout the system and to protect the data of the trusted partners. In summary, the bank must be open and secure at the same time.

  The System z environment has the security concept deeply designed in the operating system. The ability to run multiple applications concurrently on the same server demands isolating and protecting each application environment. The system must be able to control access and allow users to get to only the applications and data that they need and keep them out of those applications that they are not authorized to use.

  Hardware components, such as those for the cryptographic function that is implemented on each central processor, deliver support to the System z platform for encrypting and decrypting data and for scaling up the security throughput of the system.

  In addition, other security components, such as Resource Access Control Facility (RACF®), provide centralized security functions, such as user identification and authentication, access control to specific resources, and the auditing functions that can help provide protection and meet the business security objectives.

- Middleware

  IBM middleware products are explicitly designed to exploit the z/OS operating system and Parallel Sysplex technology:

  – CICS regions can run across multiple LPARs, for multiple production environments (for example, banks run sets of LPARs for core banking and sets of LPARs for credit card processing), and separate LPARs for development, test, and quality assurance.

    Workloads can be evenly balanced across a set of cloned CICS regions, delivering a highly available and highly scalable infrastructure and significantly simplified systems management.

  – DB2 on z/OS in a Parallel Sysplex configuration is unique in that it can provide a truly *shared everything* architecture by using the coupling facility (CF) to share database state information (locks and disk cache). All systems and applications that have access to the CF have read and write access to it at memory-like speed, regardless of their logical or physical location.

    Using *shared everything* technology, DB2 for z/OS can deliver unparalleled levels of near-linear scalability, performance, and availability.

  – IMS Transaction Manager and IMS Database Manager provide a highly available and scalable configuration, for example, an IMS transactional workload can be processed across several LPARs with performance goals based on workload classification that is managed by MVS WLM. IMS databases can be shared by IMS regions, CICS regions, and also by remote Java-based applications.

  – WebSphere MQ is the market leader in enterprise messaging. It can provide assured, once-and-once-only delivery of messages between CICS and over 80 other platform configurations. Uniquely on z/OS WebSphere MQ delivers the highest levels of availability possible by taking advantage of the CF to hold shared queues, thus removing any single point-of-failure.

4.1.2 Why CICS

CICS handles billions of transactions a week. In Martin Campbell-Kelly's history of the software industry, *From Airline Reservations to Sonic the Hedgehog*[1], he states that, "Although most people are blissfully unaware of CICS, they probably

[1] *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry*, Martin Campbell-Kelly

make use of it several times a week for almost every commercial electronic transaction they make."

CICS Transaction Server continues to be one of the most scalable, secure, and reliable application environments because of its deep integration with the underlying z/OS operating system. This ability to execute large numbers of concurrent transactions, in a consistent and efficient manner, is at the center of the past success of CICS. It is equally important as an enabler of the horizontal business integration requirements that the banking industry faces today.

The value of CICS is discussed in depth in the IBM whitepaper *Why to chose CICS Transaction Server for new IT projects*[2]. In this paper, the authors outline why, even after forty years of existence, CICS remains an excellent choice for the IBM customers in the banking sector and in other sectors.

Business transactions (particularly banking transactions) tend to be short and repetitive. Common characteristics include concurrent access by many users, through multiple interfaces, to shared information without compromising data integrity. Some examples of such financial transactions include a customer information inquiry, a financial transfer, a Point of Sale (PoS) transaction, an insurance claim, stock trading, a credit check, or any number of similar operations. Because processing these transactions represents a direct cost to the business, the cost for each transaction must be minimized.

It is possible to write standalone applications to perform these tasks in shared, multiple-user environments; however, it is complex to do so because application developers must address such issues as the communications with the user interface, ensuring that only authorized users access the system, preventing concurrent changes, handling failures and backing out partial updates, managing the database connectivity, and numerous other complicated tasks, each of which must be rewritten for each new application. All of this is extra and unnecessary work in addition to writing the actual business logic that is required to process the business transaction.

CICS takes the load off application programmers by taking care of the application's non-functional requirements, such as transactionality, security, and so on. It does this for applications that are written in a variety of languages including COBOL, PL/I, C/C++, and Java. It can process complex and demanding application workloads with optimal performance. It provides a rich set of functions for application developers and system administrators that enable them to build and manage highly efficient applications that can use standard interfaces, such as database queries, Web services, Java connectivity, and WebSphere MQ messaging.

[2] *Why to choose CICS Transaction Server for new IT projects*, Andrew Bates and Timothy Sipples, September 2008.

CICS has sophisticated virtualization capabilities that integrate with the System z Parallel Sysplex technology that we discussed in 4.1.1, “Why the mainframe” on page 50. These capabilities deliver the almost unlimited levels of scalability that are required to cost-effectively manage business growth.

A key question today is whether JEE servers are better placed than traditional TP monitors, such as CICS, for running banking workloads. For various banks this is an attractive proposition because it allows them to create a common processing environment for all applications, whether they are mission critical core banking applications or less critical applications. However, most medium to large sized banks are choosing to maintain their TP monitors and are opting for mixed IT environments that integrate across their existing CICS-based banking systems and new JEE implementations.

CICS usage continues to grow across all industries, including the finance industry, and IBM continues to invest in CICS to ensure that it takes advantage of new technologies when they become available and when the technology fits well with CICS.

4.2 CICS evolution

CICS recently celebrated 40 years of existence, having first become a program product in 1969. It was originally designed to avoid duplication and the proliferation of centralized repositories for the Public Utility industry sector. CICS was originally called the Public Utility CICS or PUCICS.

Soon after its arrival, CICS became a significant product in the IBM mainframe software portfolio, and a critical component in the IT infrastructure for many high-street banks.

In the 1970s, CICS became a major force in banking, insurance, and securities used for such applications as cheque clearing, customer record database updates, and orders.

In the 1980s, ATMs began to proliferate and customers could withdraw cash from hole in the wall ATMs for the first time over weekends and evenings.

By the 1990s, with the advent of CICS Transaction Server, and its new support for the Web, green panel applications were being replaced by Web applications. This transition allowed e-business and e-commerce to become a reality with trading stocks and shares taking place over Internet connections to back-end CICS systems. This re-facing was done without major disruption to the core banking applications.

In the early 2000s, CICS TS V2 put support behind Java and allowed Java programs to interact with CICS applications using the CICS Transaction Gateway, making the previous static Web pages much more dynamic. A new generation of programmers can now interface to traditional CICS applications irrespective of platform or industry.

From the mid-2000s, the SOA revolution takes over, and the service-oriented architecture style of developing and integrating software brings about significant changes in all mainframe software products, including CICS. We focus on the role of CICS in an SOA in 4.2.1, “CICS support for SOA” on page 57.

Figure 4-1 shows the significant milestones in the life of CICS.

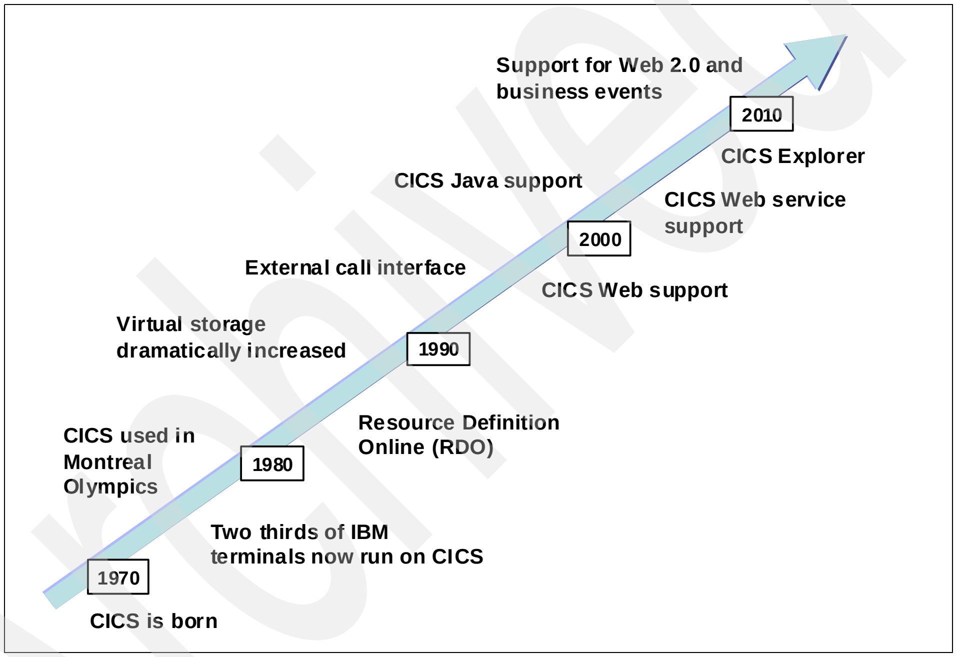

Support for Web 2.0 and business events
2010
CICS Explorer
CICS Java support
CICS Web service support
2000
External call interface
CICS Web support
Virtual storage dramatically increased
1990
Resource Definition Online (RDO)
CICS used in Montreal Olympics
1980
Two thirds of IBM terminals now run on CICS
1970
CICS is born

Figure 4-1 Potted history of CICS

1969	On July 8 CICS was born two weeks before man lands on the moon
1975	IBM determines that two-thirds of all IBM machine terminals now run on CICS
1976	CICS is used in the Montreal Olympics
1982	CICS OS/VS Version 1.6 allows online definition of CICS resources

1989	CICS MVS/ESA Version 3.1 introduces 8,000 times more virtual storage and dynamic workload balancing
1994	With the release of CICS MVS/ESA 4.1, IBM introduces a new external call interface to allow CICS programs to be called from platforms other than the mainframe
1997	CICS rebrands to Transaction Server (TS) and CICS TS allows users to ‘take CICS to the Web in 15 minutes’
2001	CICS begins its journey on the Java roadmap with CICS TS V2.1 Java support
2005	Support for the simple object access protocol (SOAP) is integrated with CICS TS V3.1, which allows CICS applications to be exposed directly as Web services and equally allows CICS applications to make Web service calls to other service providers
2009	CICS Explorer is introduced as the new face of CICS, and CICS TS V4.1 provides support for Web 2.0 and business events

4.2.1 CICS support for SOA

IT solutions are rarely deployed on a single IT system, so integration between heterogeneous systems was always a requirement. The IT industry converged on a set of common standards and principles that led to the emergence and maturity of the service-oriented architecture model of IT integration.

The SOA style of integration involves breaking an application down into common, repeatable “services” that can be used by other applications, both internal and external, in an organization, independent of the applications and computing platforms on which the business and its partners rely.

As well as more flexible integration, the other major benefit of a service-oriented architecture is the alignment of business with IT. The service contract facilitates more flexible IT integration but it also promotes the clear definition of business-aligned services. In other words, it provides a common mechanism of service definition across the business and IT communities.

Evolution towards the SOA model remains an important part of most banks IT strategies to attain the flexibility, simplicity, shared services, and business and IT alignment that they need to grapple with the more complex business environments of the future.

In recent years CICS added support for SOA and now provides near seamless connectivity with other IT environments. CICS TS V3 was very popular with customers because of its in-built Web services support, which led to it becoming the fastest take up of a new CICS version in its history. CICS now plays a full part in the IBM SOA portfolio of products. Similarly, the IBM other major transaction processing system, IMS Transaction Manager, also has full SOA support (see the IBM Redbooks publication *Powering SOA Solutions with IMS,* SG24-7662).

The Open Group[3] published a draft SOA Reference Architecture, as shown in Figure 4-2.

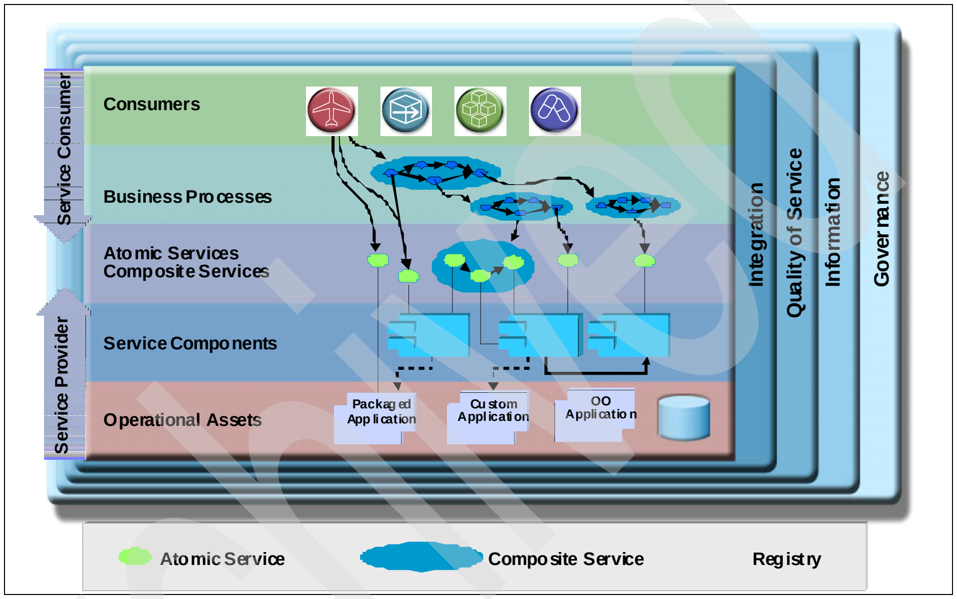


Figure 4-2 SOA reference architecture

Figure 4-2 shows the logical layers of an SOA deployment. In Table 4-1, we provide the layers and show how CICS can play its part across the architecture.

Table 4-1 SOA layers and the role of CICS

Layer	Definition	Role of CICS
Consumers	Provides multi-channel access for service consumers. Web 2.0 components tend to reside here.	A CICS application can use the EXEC CICS INVOKE SERVICE API to consume services that other systems provide.

[3] The Open Group is a vendor-neutral and technology-neutral consortium. For more information about the Open Group, see http://www.opengroup.org.

Layer	Definition	Role of CICS
Business processes	The executable processes, business rules, and state machines that deliver business-oriented services to each other and the consumer layer by choreographing services.	An activity of a business process can be implemented as a CICS application using the CICS Web services support or the JCA support that the CICS Transaction Gateway provides.
Services	Service descriptions and container information for atomic and composite services.	You can describe and publish CICS atomic or composite services. CICS supports the Service Component Architecture (SCA), which provides a standard way for defining and programming composite applications.
Service components	Software components that provide the realization for a service or an operation on a service. Service components bind the service contract to the implementation of the service in the operational systems layer.	CICS can host service components directly (using the CICS Web services support), or service components that run in other containers can connect to CICS applications.
Operational systems	The IT assets (CICS applications, JEE applications, databases, and so forth) and infrastructure that supports the SOA solution.	CICS applications are among the most valuable assets of many large companies, particularly in the finance industry.

The other layers of the SOA Reference Architecture support general concerns of a more non-functional nature and include a set of infrastructure qualities that aim to ensure that an SOA-based solution is robust and scalable:

- Integration layer

  Service integration can be based on point-to-point connections, or it can provide the capability to mediate, transform, route, and transport service requests from the service consumer to the correct service provider.

  An Enterprise Service Bus (ESB) allows a decoupling of the service consumer from the service provider. This decoupling of service endpoints can greatly increase the flexibility of an SOA solution, for example, a CICS application that is connected to other IT systems through an ESB can be upgraded, changed, moved, or replaced without necessarily requiring the connected applications themselves to be changed.

Most ESB solutions support both request-response interactions and also event-driven architectures.

- Quality of service layer

  Using SOA development tools you can create business services very quickly; however, it normally takes a lot longer to address all of the non-functional requirements of a project. An SOA infrastructure provides support for the commonly required qualities of service, such as security (it is no good publishing a financial service if it is not secure), performance (or does not perform), and RAS (reliability, availability, and scalability) qualities.

- Information architecture layer

  This layer includes information architecture, business intelligence, meta-data considerations and ensures the inclusion of key considerations that pertain to data architecture and information architectures that can also be used as the basis for creating business intelligence through data marts and data warehouses.

- Governance

  The governance layer includes both SOA governance (governance of processes for policy definition and enforcement) and service governance (service life-cycle). Service governance covers the entire life cycle of the services and the management aspects, for example, SLA, capacity, performance, security, and monitoring.

SOA in banking

In many banks, factors, such as merger and acquisition activity, regulatory requirements, and globalization resulted in an inflexible and complicated IT environment—an environment with lots of point-to-point connections.

In an executive brief from the IBM Institute of Business Value, *Service-oriented architecture: Revolutionizing today's banking systems*[4], the authors explain how SOA can help with this problem by reducing redundancy and inflexibility in crucial banking processes, such as payments, multichannel integration, and account opening, which is done by creating a single set of services that are re-used across all channels and internal support applications, through a common integration layer (an ESB, for example). A single “get customer information” service, for instance, can be applied to any product where it is relevant, including core banking, credit card, wealth management, and brokerage applications.

[4] Service-oriented architecture: Revolutionizing today's banking systems, IBM Institute of Business Value

Figure 4-3 shows one such customer organization, for a large European banking group, that handles business service requests from a number of requesters, including branch offices, customers, suppliers, and trusted business partners.

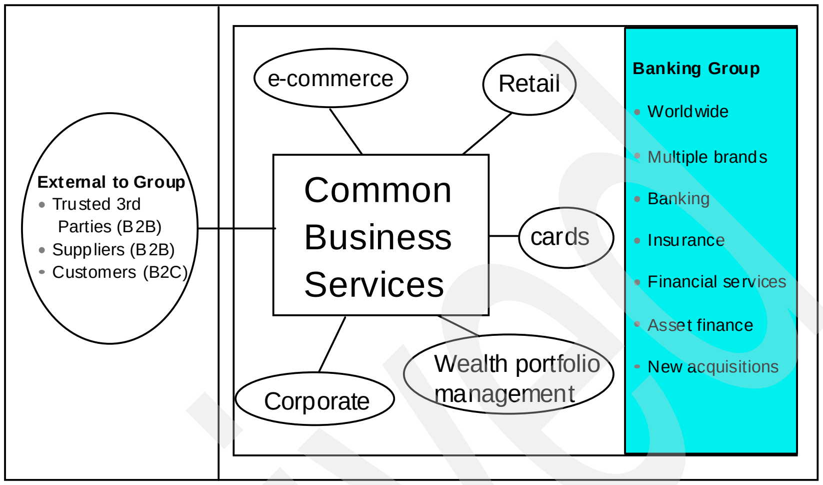


Figure 4-3 Common business services

The IBM whitepaper *Deploying CICS Web services to preserve IT investments in the banking industry*[5] describes how the specific QoS requirements (specifically in the areas of security, availability, performance, and monitoring) for this customer are met by an SOA solution based on using CICS Web services with WebSphere DataPower® used as an ESB.

Continued SOA support in CICSTS V4.1

CICS TS V4.1 extends the CICS SOA roadmap with its improved support for Web services and ability to deploy service components and business services to CICS in more standardized and flexible ways.

Enhanced Web services support

CICS TS V4.1 continues to enhance its support for SOA with data mapping enhancements and support for the Web Services Addressing (WS-Addressing) specification.

The improvements to data mapping provide a faster and enhanced conversion between XML and language structures for all Web services and include new APIs to access these services.

[5] Deploying CICS Web services to preserve IT investments in the banking industry, IBM Design Centre.

WS-Addressing enables the normalization of information that is usually provided by transport or messaging systems and allows the inclusion of message routing data within SOAP headers, which eliminates reliance on network-level transport to carry routing information.

Service Component Architecture

To make it easier to publish and use CICS applications in an SOA, CICS TS V4.1 provides infrastructure and run-time support for deploying and managing application components. These components provide a common programming interface for service invocation and a service description that is compliant with the Service Component Architecture (SCA).

SCA extends and complements previous approaches to implementing services and builds on open standards, such as Web services. The specifications describe how to create *composite* applications. A composite application is created by combining one or more components that together implement the business logic of the new application.

CICS supports the SCA Assembly Model 1.0 specification, which describes how service components can be assembled to form composites. A composite is the unit of deployment in SCA and is described in an XML language called Service Component Definition Language (SCDL). Composites can contain components, services, references, property declarations, and the wiring that describes the connections between these elements. CICS composites can be developed using tooling that IBM provides with Rational® Developer for System z (RDz) and are deployed to CICS in a new resource type called a *bundle*.

Support for business services

CICS TS V4.1 includes the CICS Service Flow Runtime, which allows the deployment of CICS business services (or service flows) created by the Service Flow Modeller component of RDz.

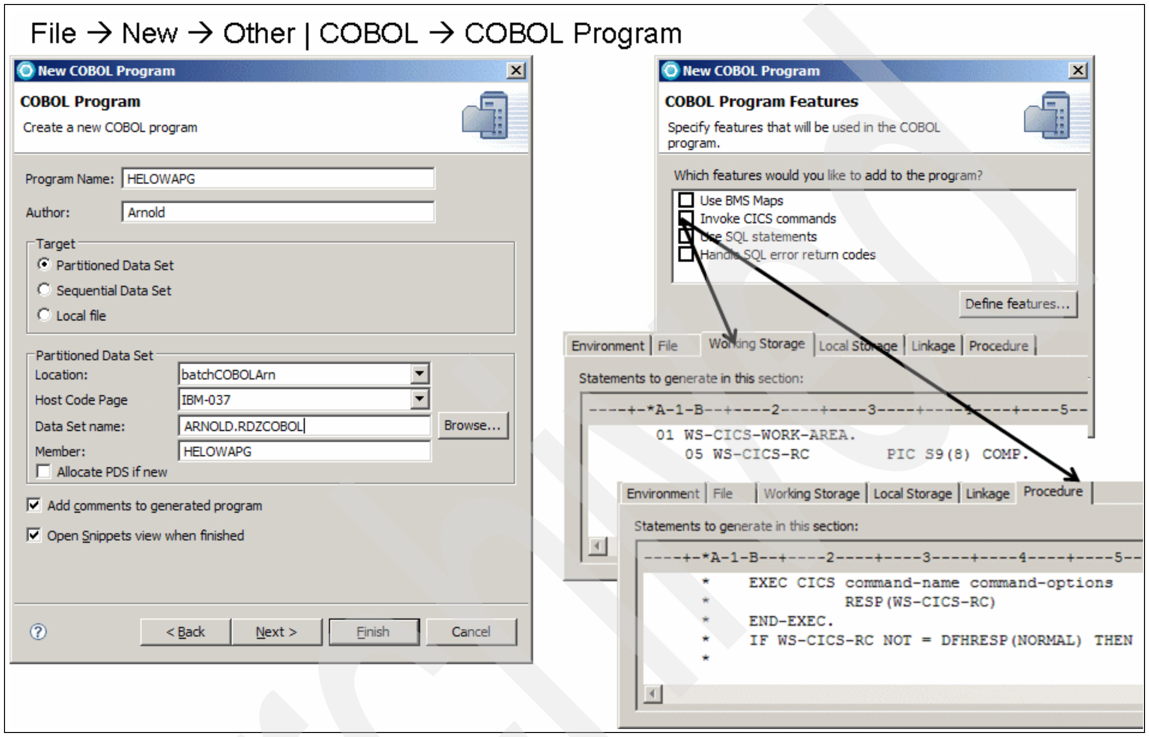
4.3 Application development

In the past, the specific skill requirements for mainframe application development was used as an argument against the deployment of new applications on the mainframe. Yet the skills that are required for developing mainframe applications with the latest integrated development environments (IDEs) are now very similar to other environments.

For CICS application developers, RDz provides a single, integrated IDE for all supported languages and architectural styles whether you are creating new

CICS applications from scratch or are wrapping, refactoring, or otherwise reusing your CICS application assets.

Figure 4-4 shows a pictorial view of how RDz can assist in the development of CICS COBOL programs.


File → New → Other | COBOL → COBOL Program

New COBOL Program

COBOL Program

Create a new COBOL program

Program Name: HELOWAPG

Author: Arnold

Target
- (●) Partitioned Data Set
- ( ) Sequential Data Set
- ( ) Local file

Partitioned Data Set
- Location: batchCOBOLArn
- Host Code Page: IBM-037
- Data Set name: ARNOLD.RDZCOBOL
- Member: HELOWAPG
- [ ] Allocate PDS if new

- [✓] Add comments to generated program
- [✓] Open Snippets view when finished

< Back | Next > | Finish | Cancel

New COBOL Program

COBOL Program Features

Specify features that will be used in the COBOL program.

Which features would you like to add to the program?
- [ ] Use BMS Maps
- [ ] Invoke CICS commands
- [ ] Use SQL statements
- [ ] Handle SQL error return codes

Define features...

Environment | File | Working Storage | Local Storage | Linkage | Procedure

Statements to generate in this section:

```
----+-*A-1-B--+----2----+----3----+----4----+----5--
       01 WS-CICS-WORK-AREA.
           05 WS-CICS-RC         PIC S9(8) COMP.
```

Environment | File | Working Storage | Local Storage | Linkage | Procedure

Statements to generate in this section:

```
----+-*A-1-B--+----2----+----3----+----4----+----5--
      *     EXEC CICS command-name command-options
      *               RESP(WS-CICS-RC)
      *     END-EXEC.
      *     IF WS-CICS-RC NOT = DFHRESP(NORMAL) THEN
      *
```

Figure 4-4 Using RDz to develop CICS COBOL programs

RDz supports end-to-end, model-based development, runtime testing, and rapid deployment of mixed-language applications. Application components in various programming languages can be created and deployed to a range of runtime environments, including CICS and WebSphere Application Server.

Using RDz, core business applications can be developed more easily in one of the supported languages: COBOL, PL/I, C/C++, and Java. These applications can then be deployed as service components and integrated using standard interfaces, for example, Web services.

The latest version of RDz includes support for the CICS Explorer (see 4.4.3, “CICS Explorer” on page 66).

4.4 Compete, control, and comply

CICS TS V4.1 delivers important new functions to help organizations to position themselves to take advantage of new growth opportunities, respond to the demands of increased regulatory scrutiny, and to control costs.

The main objectives for CICS TS V4.1 are to assist customers to:

Compete	Competing for new opportunities by gaining insight into business processes and responding by modifying key business applications quickly and with confidence.
Comply	Complying with corporate, industry, and government policies to manage business risks of critical business applications.
Control	Controlling costs and resources by simplifying IT infrastructure and improving development and operations productivity through easier-to-use interfaces and functions.

The highlights of the CICS TS V4.1 enhancements are:

- Non-invasive detection and emission of business events from CICS applications that provides better insight into business behavior.
- Atom feeds and RESTful interfaces that enable existing CICS programs and resources to participate in mashups, Web 2.0, and other situational applications.
- The CICS Explorer, which simplifies the development and management of traditional and modern applications and provides an integration point for IBM and third-party tools.

These key functions support two or more of the themes (compete, comply, and control) simultaneously, for example, in support of multiple themes in this version of CICS TS, the ability to generate business events without changing application programs both reduces cost and complexity and delivers compliant flexible business solutions.

CICS TS 4.1 also contains significant performance improvements, including better Web services performance and enhanced workload management.

We summarize the major new enhancements in CICS TS V4.1 in the next section.

4.4.1 Support for event processing

You can specify business events and then capture and emit them from a CICS application without changing the application. You can use these business events in many ways, such as providing insight into business activity and driving new processing to respond to business opportunities or threats, for example, they can feed into another CICS application or be placed on a WebSphere MQ queue. After they are on the queue, they can be consumed in a variety of ways:

- By a business event processing engine, such as WebSphere Business Events.
- By business monitors and dashboards, such as WebSphere Business Monitor.
- Using methods, such as reading the event in another program.

CICS event processing is focused on application events of significance to the business rather than systems management or IT events. The primary intent of this function allows existing applications to be non-invasively instrumented for events so that events can be produced by the applications without any code changes.

The CICS Explorer includes an event binding editor that allows event specifications to be created and edited within event bindings. The event specifications define the events to be emitted by CICS, how the occurrence of the events can be detected in applications at runtime, and where and in what format the event is to be emitted. Event bindings can be deployed into CICS using the new CICS bundle resource type and enabled or disabled as required.

For more information about CICS support for event processing, see Chapter 5, “Event processing support” on page 71.

4.4.2 Atom feeds from CICS

CICS TS V4.1 provides access to CICS resources and application programs in a RESTful style by exposing them as Atom feeds or collections that are structured according to the Atom Syndication Format and the Atom Publishing Protocol. This implementation of REST (Representation State Transfer) permits the HTTP methods GET, PUT, POST, and DELETE to be used to read and update the contents of CICS resources from an external HTTP client application.

Atom feeds can be enabled in CICS in a non-invasive way, that is, without changes to CICS applications, which allows CICS resources to be viewed and manipulated using feed readers, mashups, and other Web 2.0 and situational applications in a non-invasive way.

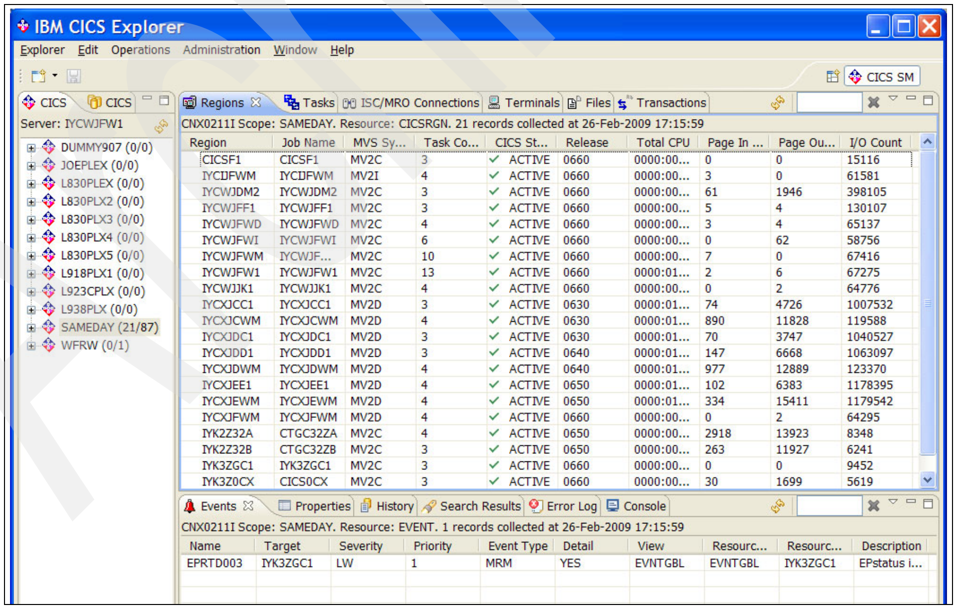
You can quickly create powerful new applications using IBM tools, such as Lotus® Mashups, WebSphere sMash, Rational Developer for System z (RDz), and tools from other vendors.

For more information about CICS support for Atom and Web 2.0, see Chapter 6, “Web 2.0 support” on page 87.

4.4.3 CICS Explorer

Prior to the CICS Explorer, performing each CICS systems programming task required the use of various disconnected interfaces. These applications were often ISPF-based panels that were written to do a specific function. Each application required custom knowledge to operate, and if the completion of a specific task crossed application boundaries, the user had to switch back and forth between the applications, which had its own commands and user interfaces.

With the CICS Explorer, CICS now gives users a much more powerful tooling environment to help experienced, skilled IT staff to be more productive and provide a more intuitive experience for less experienced team members. It also plays a vital role in establishing and handing over best practices to the next generation of IT staff. Figure 4-5 shows how CICS Explorer can simplify the day-to-day operations of a large CICS environment.



CNX0211I Scope: SAMEDAY. Resource: CICSRGN. 21 records collected at 26-Feb-2009 17:15:59

Region	Job Name	MVS Sy...	Task Co...	CICS St...	Release	Total CPU	Page In ...	Page Ou...	I/O Count
CICSF1	CICSF1	MV2C	3	✓ ACTIVE	0660	0000:00...	0	0	15116
IYCIJFWM	IYCIJFWM	MV2I	4	✓ ACTIVE	0660	0000:00...	3	0	61581
IYCWJDM2	IYCWJDM2	MV2C	3	✓ ACTIVE	0660	0000:00...	61	1946	398105
IYCWJFF1	IYCWJFF1	MV2C	3	✓ ACTIVE	0660	0000:00...	5	4	130107
IYCWJFWD	IYCWJFWD	MV2C	4	✓ ACTIVE	0660	0000:00...	3	4	65137
IYCWJFWI	IYCWJFWI	MV2C	6	✓ ACTIVE	0660	0000:00...	0	62	58756
IYCWJFWM	IYCWJF...	MV2C	10	✓ ACTIVE	0660	0000:00...	7	0	67416
IYCWJFW1	IYCWJFW1	MV2C	13	✓ ACTIVE	0660	0000:01...	2	6	67275
IYCWJJK1	IYCWJJK1	MV2C	4	✓ ACTIVE	0660	0000:00...	0	2	64776
IYCXJCC1	IYCXJCC1	MV2D	3	✓ ACTIVE	0630	0000:01...	74	4726	1007532
IYCXJCWM	IYCXJCWM	MV2D	4	✓ ACTIVE	0630	0000:01...	890	11828	119588
IYCXJDC1	IYCXJDC1	MV2D	3	✓ ACTIVE	0630	0000:01...	70	3747	1040527
IYCXJDD1	IYCXJDD1	MV2D	3	✓ ACTIVE	0640	0000:01...	147	6668	1063097
IYCXJDWM	IYCXJDWM	MV2D	4	✓ ACTIVE	0640	0000:01...	977	12889	123370
IYCXJEE1	IYCXJEE1	MV2D	4	✓ ACTIVE	0650	0000:01...	102	6383	1178395
IYCXJEWM	IYCXJEWM	MV2D	4	✓ ACTIVE	0650	0000:01...	334	15411	1179542
IYCXJFWM	IYCXJFWM	MV2D	4	✓ ACTIVE	0660	0000:00...	0	2	64295
IYK2Z32A	CTGC32ZA	MV2C	4	✓ ACTIVE	0650	0000:00...	2918	13923	8348
IYK2Z32B	CTGC32ZB	MV2C	3	✓ ACTIVE	0650	0000:00...	263	11927	6241
IYK3ZGC1	IYK3ZGC1	MV2C	3	✓ ACTIVE	0660	0000:00...	0	0	9452
IYK3Z0CX	CICS0CX	MV2C	3	✓ ACTIVE	0660	0000:00...	30	1699	5619

CNX0211I Scope: SAMEDAY. Resource: EVENT. 1 records collected at 26-Feb-2009 17:15:59

Name	Target	Severity	Priority	Event Type	Detail	View	Resourc...	Resourc...	Description
EPRTD003	IYK3ZGC1	LW	1	MRM	YES	EVNTGBL	EVNTGBL	IYK3ZGC1	EPstatus i...

Figure 4-5 CICS Explorer

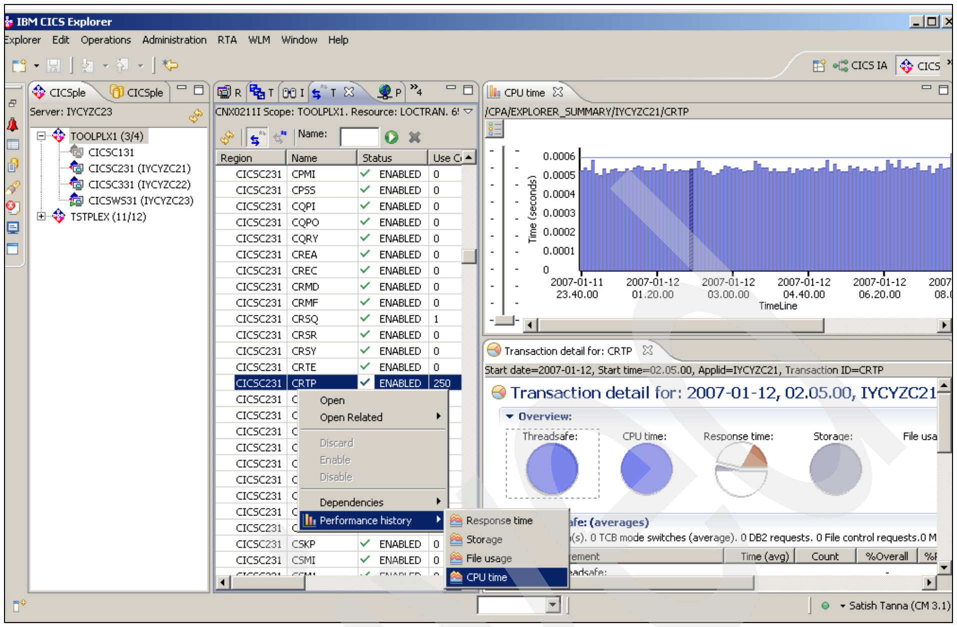
As a system management tool, the CICS Explorer is designed to provide an intuitive way to manage a single CICS region, a collection of regions within a CICSplex, or a collection of CICSplexes. Experienced system programmers can quickly identify and tune their systems, and newer users are guided through the intuitive interface to learn how to perform standard tasks. Having a consolidated approach to system management reduces the number of interfaces that are required by CICS system administrators to learn to become effective, which helps to make the effort of attracting and training new developers and system programmers easier and less expensive.

With CICS TS V4.1, CICS Explorer is also used to create new resource types, such as CICS event bindings and CICS bundles, for example, we show how an application analyst or programmer uses the CICS Explorer to configure CICS to emit business events in 8.2, “Enabling business events in the core banking system” on page 142.

As well as basic CICS system management, the definition and operation of regions and their installed resources, a typical CICS shop uses a number of tools for tasks, such as system configuration, debugging, and performance monitoring. These tools are typically obtained from IBM, third-party vendors, or developed in-house. Each tool has its own interfaces and method of presenting the CICS runtime.

One of the driving forces behind the creation of the CICS Explorer was providing an integration platform where all of the tools and applications that a user needs to perform a business task work together coherently in an integrated environment. The benefit of this integration is to help new users become familiar with the platform, to present a single and re-usable way of accessing and manipulating CICS to promote consistency, and to allow tools that are built by IBM, vendors, and customers to be integrated within a single environment, for example, Figure 4-6 on page 68 shows how the CICS Performance Analyzer tool can be plugged in to CICS Explorer so that the CICS systems programmer can use the same tool for operations and performance analysis.

System programmers and developers can also use CICS Performance Analyzer for z/OS (CICS PA) to understand the performance of new or changed CICS applications and to identify performance improvements, including the benefits of running applications in threadsafe mode prior to deployment, which helps to get the most benefit from your CICS application investments with reduced risk.



Server: IYCYZC23

- TOOLPLX1 (3/4)
  - CICSC131
  - CICSC231 (IYCYZC21)
  - CICSC331 (IYCYZC22)
  - CICSWS31 (IYCYZC23)
- TSTPLEX (11/12)

CNX0211I Scope: TOOLPLX1. Resource: LOCTRAN. 6!

Name:

Region	Name	Status	Use C
CICSC231	CPMI	✓ ENABLED	0
CICSC231	CPSS	✓ ENABLED	0
CICSC231	CQPI	✓ ENABLED	0
CICSC231	CQPO	✓ ENABLED	0
CICSC231	CQRY	✓ ENABLED	0
CICSC231	CREA	✓ ENABLED	0
CICSC231	CREC	✓ ENABLED	0
CICSC231	CRMD	✓ ENABLED	0
CICSC231	CRMF	✓ ENABLED	0
CICSC231	CRSQ	✓ ENABLED	1
CICSC231	CRSR	✓ ENABLED	0
CICSC231	CRSY	✓ ENABLED	0
CICSC231	CRTE	✓ ENABLED	0
CICSC231	CRTP	✓ ENABLED	250
CICSC231	C		
CICSC231	C		
CICSC231	C		
CICSC231	C		
CICSC231	C		
CICSC231	C		
CICSC231	C		
CICSC231	C		
CICSC231	C		
CICSC231	CSKP	✓ ENABLED	0
CICSC231	CSMI	✓ ENABLED	0

/CPA/EXPLORER_SUMMARY/IYCYZC21/CRTP

Time (seconds)
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0

2007-01-11 23.40.00	2007-01-12 01.20.00	2007-01-12 03.00.00	2007-01-12 04.40.00	2007-01-12 06.20.00	2007 08.(

Transaction detail for: CRTP

Start date=2007-01-12, Start time=02.05.00, Applid=IYCYZC21, Transaction ID=CRTP

Transaction detail for: 2007-01-12, 02.05.00, IYCYZC21

afe: (averages)

(s). 0 TCB mode switches (average). 0 DB2 requests. 0 File control requests.0 M

ement	Time (avg)	Count	%Overall	%F
eadsafe:			-	

Figure 4-6 CICS PA integrated with CICS Explorer

The CICS Explorer is based on the Eclipse Rich Client Platform (RCP). Eclipse RCP is a cross platform framework for building and deploying PC desktop applications. One of the features of Eclipse that makes it a good choice of technology for the CICS Explorer is its plug-in architecture that allows components to be added that have the ability extend and complement each other.

Today, many plug-ins exist that can be integrated into Eclipse environments, such as workflow, messaging clients, or development tools. One of these products is IBM Rational Developer for System z (RDz), which we discuss in 4.3, “Application development” on page 62. Given the open nature of the Eclipse framework, customers and Business Partners can also develop plug-ins for the CICS Explorer using the CICS Explorer SDK.

With the emergence of the IBM CICS Explorer, the specialized skill requirements for supporting CICS environments is no longer an inhibitor to new application deployments.

4.5 Summary

CICS has been, and continues to be, a critical solution enabler for the finance industry. In this chapter, we charted the history of CICS within the context of banking. We also introduced the most recent capabilities of the latest version of CICS Transaction Server, V4.1.

In the next two chapters of this part, we further expand on the major enhancements to CICS TS V4.1 that enable banks to evolve towards the smarter banking paradigm.

5

Event processing support

The ability to detect events in CICS application processing and to emit those events for consumption in a variety of ways without making changes to the existing applications opens up a number of new opportunities for businesses.

Events from CICS Transaction Server V4.1 allow processing within CICS to be monitored, which enables insight into CICS applications and the business processes that they support without making application changes. It is also possible to extend the processing in new ways by using CICS events, which enable flexible and timely responses to business opportunities or threats. Events from CICS can also be used to look for particular patterns of events, either from CICS alone or from CICS and other sources, which might indicate potential breaches of corporate, industry, or government regulations or other undesirable situations.

5.1 Introduction to event processing

Event processing and event-based systems have been around for a long time, used in particular in managing and monitoring IT systems. An aspect of event processing that is now gaining considerable momentum is a focus on the business value that is obtained from events. The processing is based on the growing need to react and make decisions that are closer to real time, to gain insight into business processing in response to the introduction of compliance regulations, and the desire to respond rapidly to changes in the business without entailing long development cycles.

In Chapter 3, "Smarter Banking scenarios" on page 33, we outline a number of business scenarios and highlight how you can use business event enablement to improve business process agility across the four domains of the IBM Banking Framework. In this chapter, we closely review how you can enable CICS to emit business events.

Companies around the world run their key business processing within CICS. However, there is a great deal of information that is locked up in this processing, to which they do not currently have access. This locked up information ranges from understanding the business decisions that are encapsulated in the applications to detecting business opportunities and threats.

Using the new event processing technology you gain this business understanding and insight. With new support that enables CICS to act as a source of business events, business monitor dashboards, such as IBM WebSphere Business Monitor, can be quickly and easily configured to derive value from the information in CICS applications, and business event engines, such as IBM WebSphere Business Events, can provide enhanced insight and action by detecting patterns among events.

5.2 Leveraging events to address business needs

Business managers and analysts understand the key business events and the desired actions to be taken. However, they have not previously had the solutions available to enable them to identify and respond to the volume and complexity of these situations themselves. At the same time, although millions of potentially actionable events are flowing freely through the IT infrastructure today, support for advanced event-driven solutions previously required long development and test cycles.

The challenge of closer alignment between business and IT is addressed by business event processing, which combines event processing with capabilities that enable users to make use of events and to define the event processing behavior themselves. Business event processing provides IT with the functionality to support advanced event processing requirements in a high-performance, manageable, scalable environment.

The event support in CICS is designed to significantly reduce the development time for introducing event-driven solutions and to allow IT to respond more rapidly to requests from the business.

5.3 Solutions for integrated business event processing

The IBM software portfolio enables a range of options for processing business events. In this section, we provide an introduction to the two key products that interoperate with events from CICS: WebSphere Business Monitor and WebSphere Business Events:

- WebSphere Business Monitor

  WebSphere Business Monitor is comprehensive business activity monitoring (BAM) software that provides business users with a real-time and end-to-end view of business processes, events, and operations. It is a core part of the IBM WebSphere Dynamic Process Edition foundational offering of the IBM Business Process Management (BPM) suite, and is also available as a standalone product.

- WebSphere Business Events

  WebSphere Business Events helps businesses to detect, evaluate, and respond to the impact of business events based on the discovery of action able event patterns. WebSphere Business Events is specifically designed to support business event processing by meeting the high-volume demands and processing that is required across industries and application domains. Equally important is the extensive use of graphical, codeless user interfaces that greatly simplify implementation and empower business users to directly develop and maintain event processing logic.

5.4 Event capability in CICS

In December 2008, IBM enabled support for interoperation between WebSphere Business Events V6 and CICS Transaction Server for z/OS V3 through a new SupportPac capability that is available for download from the CICS Web site for all licensed users of CICS Transaction Server for z/OS V3.2. Using this CICS SupportPac you can implement event points in CICS applications, which positions CICS as a key source for emitting business events in a format that is suitable for consumption by WebSphere Business Events.

CICS TS V4.1 extends this capability in a number of ways, of which the two most significant are support for non-invasive event detection and emission and interoperation with a wider range of event consumers. Using this non-invasive event detection you can capture events without the changing the application code.

CICS systems run an enormous amount of existing business logic, which carries out processing and makes business decisions that represent interesting business events. Because of the critical nature of these applications and a growing skills gap, there is a reluctance to directly enhance these applications. The event-based approach in CICS TS V4.1 makes it possible to gain insight into processing in CICS and to easily introduce additional extensions to applications in a dynamic and decoupled fashion.

With CICS event processing support existing business logic is enabled to emit events. Tooling defines events and the data that is associated with them and is also used to deploy the events to CICS. The tooling creates *event specifications* that include information about how the events can be detected by the CICS runtime and indicates how a related group of events are to be formatted and routed. This tooling is the Event Binding Editor, which is provided with CICS TS V4.1 as part of the CICS Explorer.

Figure 5-1 on page 75 shows an overview of the CICS event processing support.

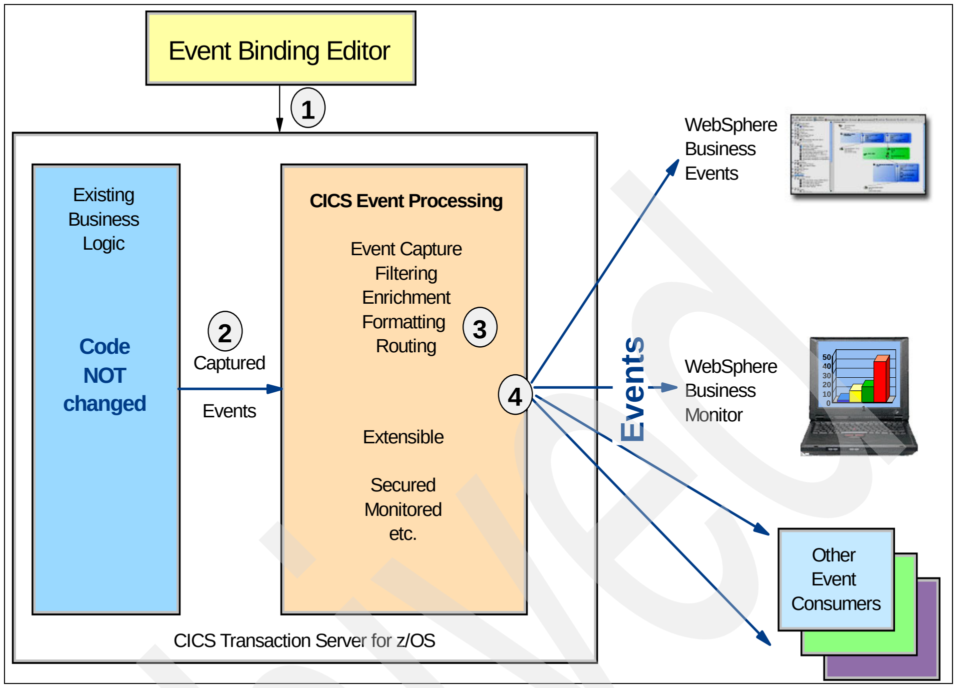


Figure 5-1 Overview of CICS event processing support

In Figure 5-1:

1. Using the Event Binding Editor Application Analysts can quickly create the information that CICS requires to identify when events of interest occur and to collect the required data.
2. The CICS runtime detects events that are described by currently enabled event specifications and captures the events to enable rapid, easy deployment of event-based solutions. CICS event processing is a core component of the CICS runtime and provides all of the qualities of service that you expect of CICS.
3. When CICS captures events, it can carry out specified filtering, enrich the event with information about the application context in which it occurred, and format the event.
4. CICS then routes the event to the appropriate event consumer, which can be WebSphere Business Events, WebSphere Business Monitor, or other event consumers (including another CICS system).

5.4.1 CICS event specifications hierarchy

Figure 5-2 shows the logical hierarchy of event specifications for CICS events.

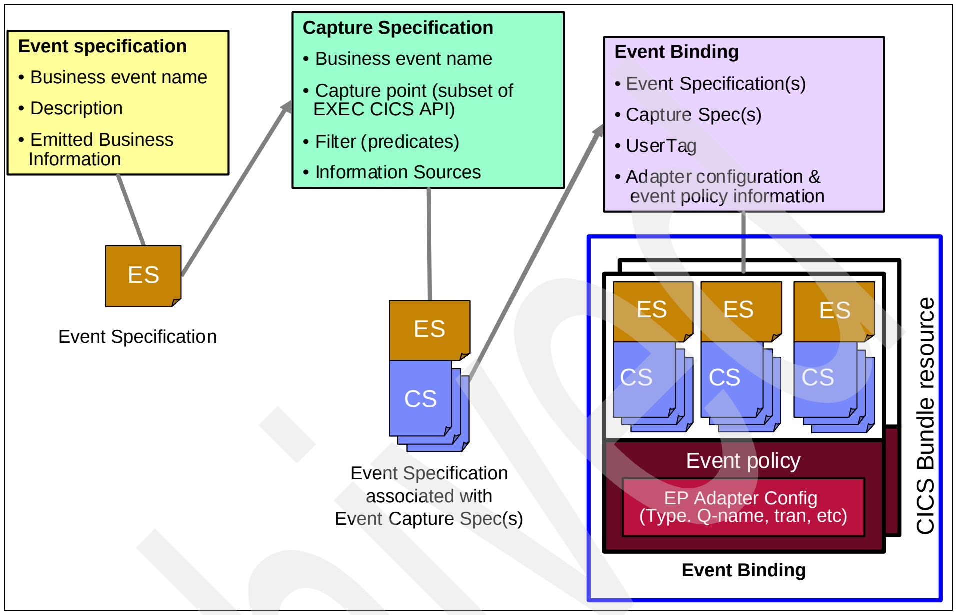


Figure 5-2 CICS event specifications

In Figure 5-2:

- The *event specification* is a statement of the event required, such as Request to Open Account, and the business information to be emitted as part of the event, such as customer and account type.
- Normally one *capture specification* is associated with an event specification. The capture specification provides the information that CICS uses to detect the event in application processes that are running within the system, for example, when the Accounts program is linked through an EXEC CICS LINK command, and data in a container passed to the program indicates that a new account is being opened, this indicates that the event of interest occurred.

Note: In most cases, you only associate a single capture specification with an event specification; however, there can be cases where a specific event must be captured in more that one place in the application processing, for example, an open account request that is processed using various channels, such as the Internet channel or in the branch. In this case, the path through the application might be different for each channel.

The capture specification also relates information that is available in the application at this point to the business information to be emitted as part of the event. Using our Accounts program as an example, the customer and account type might be obtained from data in other containers in the channel.

- Related event specifications and their associated capture specifications are grouped together into an *event binding*. The event binding also provides information about how (what format) and to where (such as to which WebSphere MQ queue) the event is to be emitted. This information is contained in the event processing adapter configuration.

  An event binding is the unit of enablement for a group of related events. It is defined to CICS and deployed as part of a *CICS bundle resource*, which is a new resource type introduced in CICS TS V4.1. The CICS bundle can contain several related event bindings and other new CICS entities that are installed and managed using CICS bundle resources.

The main focus for CICS event processing support is on the ability for CICS to capture events without changing the application code, which is referred to as non-invasive event capture. Table 5-1 provides the subset of the EXEC CICS API that is supported for event capture.

Table 5-1 Capture points

CONVERSE	DELETE FILE	DELETEQ TD
DELETEQ TS	INVOKE SERVICE	LINK PROGRAM
PUT CONTAINER	READ	READNEXT
READPREV	READQ TD	READQ TS
RECEIVE	RECEIVE MAP	RETRIEVE
RETURN	REWRITE	SEND
SEND MAP	SEND TEXT	SIGNAL EVENT
START	WEB READ	WEB READNEXT
WRITE FILE	WRITEQ TD	WRITEQ TS
XCTL		

Note: There is one event capture point that is not an EXEC CICS command. This is program initiation, which allows an event to be specified when a program starts.

The subset of the EXEC CICS API that is supported for event capture was selected to give the best chance that users can specify where events occur in their applications in this non-invasive manner. However, there are situations where explicit control over capturing events is desired, and to meet this need, a new EXEC CICS SIGNAL EVENT command was introduced into the CICS API. Think of this new command as an event opportunity: It does not cause the event to be automatically emitted every time; instead, it allows the command to be included within an event capture specification, which gives full flexibility to enable and disable the events and to vary the way in which they are used without further change to the application.

Restriction: CICS does not support event capture for data-oriented commands that can be evented in other ways, such as through the database. This restriction includes SQL and WMQ commands.

5.4.2 Event processing adapters

When the CICCS runtime captures an event, it is dispatched to an event processing adapter (EP adapter). Figure 5-3 shows the EP adapters that can be configured in the CICS event binding.

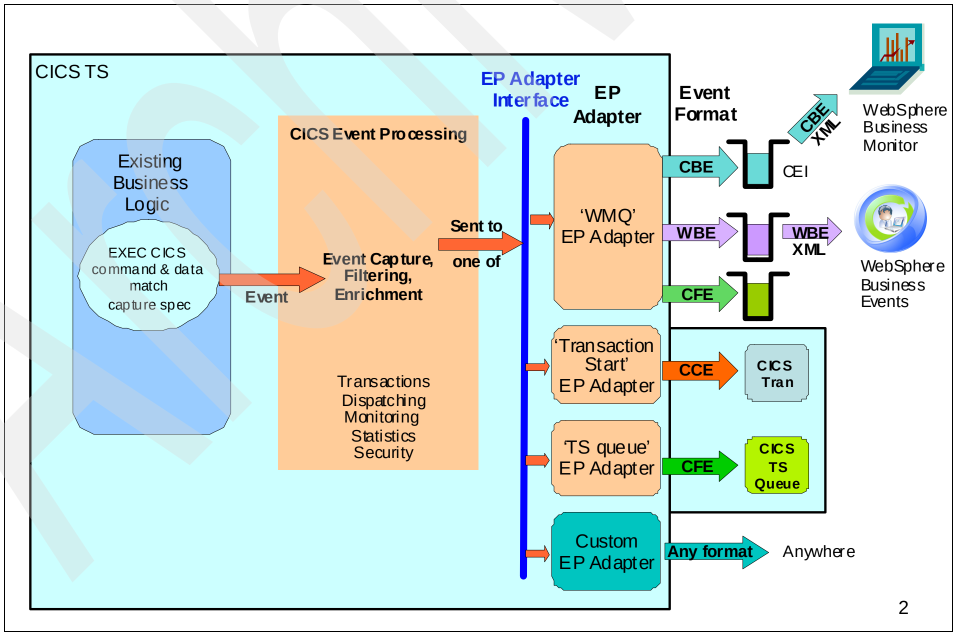


Figure 5-3 CICS Event processing adapters

The role of an EP adapter is to format the event and route it to event consumers. The EP adapters that CICS enable provides are:

- Emitting the event over WebSphere MQ, in one of three formats:
  - The standard Common Base Event (CBE) format for consumption over the Common Event Infrastructure by products, such as WebSphere Business Monitor
  - The WebSphere Business Events (WBE) XML format that WebSphere Business Events recognizes
  - A non-XML text-based format called CICS flattened event (CFE) format for consumption by user programs that get the event from the WebSphere MQ queue
- Starting a new CICS transaction as a result of an event to drive new work in this or another CICS system, providing the event details in containers within a channel in the CICS channel-based event format (CCE).
- Writing the event to a CICS temporary storage queue, in the text-based CFE format, primarily used to test that events are emitted when expected and that they contain the correct data.

All EP adapters are invoked using a standard EP adapter interface, which provides an opportunity for custom-written EP adapters.

See 8.2, “Enabling business events in the core banking system” on page 142 for detailed information about enabling CICS events.

5.5 Business process monitoring with CICS events

CICS TS V4.1 provides for interoperation with WebSphere Business Monitor using events from CICS. Providing the ability to consume such events in a WebSphere Business Monitor dashboard can provide insight into CICS applications in a way that was not previously possible. Events that CICS applications provide carry information about the processing within CICS, and by consuming those events using WebSphere Business Monitor, or other event monitoring products, you can quickly and easily get value out of that information.

To produce events from CICS that can be consumed by WebSphere Business Monitor, the EP adapter that is specified in the event binding must be the WebSphere MQ Queue EP adapter with the CBE format selected. Other configuration details, such as the WebSphere MQ queue to which the event must be emitted, are also specified. Figure 5-4 on page 80 provides a high-level view of CICS and WebSphere Business Monitor integration.

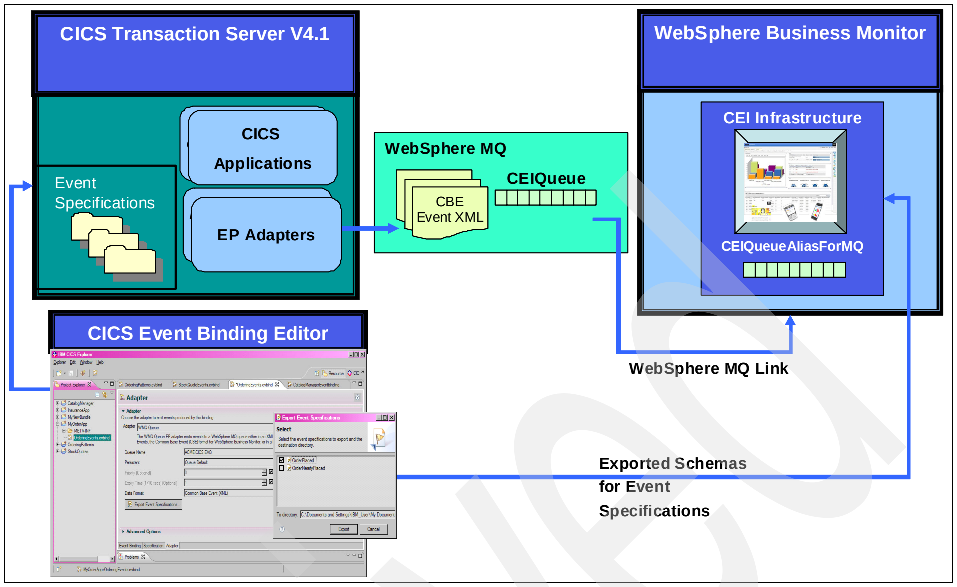


Figure 5-4   CICS integration with WebSphere Business Monitor

Figure 5-4 shows how the CICS event binding editor is used to create event specifications that are enabled in CICS. The schemas for the event specifications are then exported and used by WebSphere Business Monitor to interpret the events that are emitted from CICS.

Business monitoring can be used to gather useful business metrics, for example, users can obtain monitoring information from a sequence of single events, such as loan requests, mortgage applications, and the issue of credit cards. From that information, it is possible to create reports about a specific customer's actions or trends in business performance.

Business monitoring dashboards that are created in Business Space that is powered by WebSphere provide a number of ways to display this information, from a simple counter that moves each time an event occurs to charts that show trends.

You can also use business monitoring to track whether key performance indicators (KPIs) are being met, for example, by producing events from CICS whenever a mortgage application is placed, when it is fulfilled, and when it is cancelled, it is possible to obtain insight into:

- How long it is taking to process mortgage requests
- How often requests are cancelled after they are placed

Figure 5-5 shows KPIs for another scenario that relates to the issue of credit cards, which includes how frequently cards are reissued because the previous cards were lost or stolen. A CARDISSUE event is emitted when a card is issued because the previous card was either lost or stolen with the event including information that indicates the reason. The capture specification for the CARDISSUE event is defined such that issuing a completely new card does not emit an event.

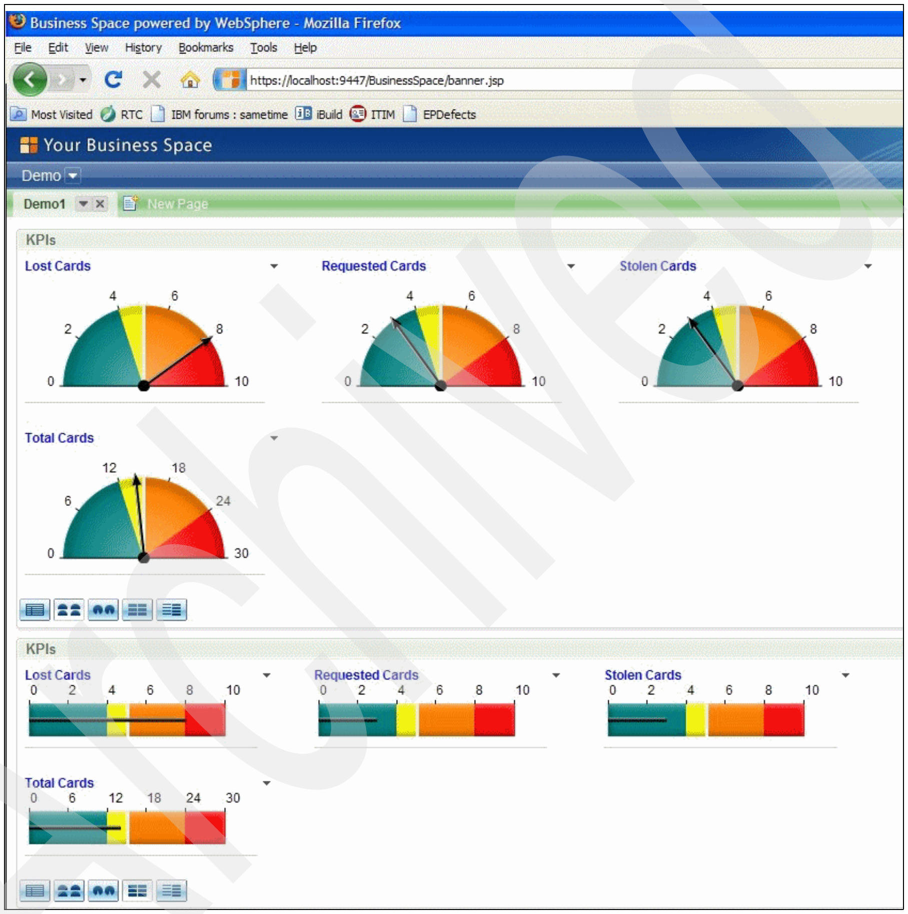

Business Space powered by WebSphere - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://localhost:9447/BusinessSpace/banner.jsp
Most Visited RTC IBM forums : sametime iBuild ITIM EPDefects
Your Business Space
Demo
Demo1
New Page
KPIs
Lost Cards
Requested Cards
Stolen Cards
Total Cards
KPIs
Lost Cards
Requested Cards
Stolen Cards
Total Cards

Figure 5-5 Business Space powered by WebSphere

Business Space powered by WebSphere provides a user-friendly, browser-based interface. It is the unifying front end for the IBM BPM products and is included with products, such as WebSphere Business Monitor, WebSphere Business Events, WebSphere Business Services Fabric, and WebSphere Process Server. Using Business Space powered by WebSphere business users can build their own dashboards using widgets.

5.6 Detecting event patterns and taking action with CICS events

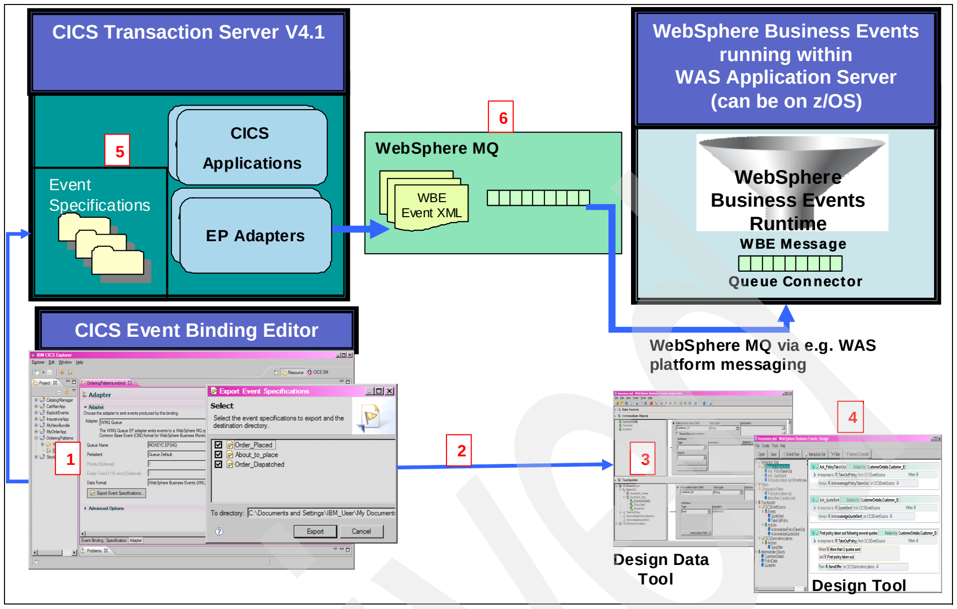
WebSphere Business Events can be used to derive useful information and actions from CICS events. CICS events can be used to provide both responsiveness and business flexibility and to facilitate the detection of fraudulent situations and help to satisfy governance and compliance regulations.

There are a number of regulations that affect many industries, which require such situations to be detected in real time, for example, the U.S. Fair and Accurate Credit Transactions (FACT) Act requires companies to have a red flag policy for detecting potential instances of identity theft.

New and revised financial regulations are a part of banking life. Adherence to them can be difficult because existing applications are not always extensible in a short period of time. CICS events can be employed to capture and emit data to other processes. The important data (otherwise locked into the CICS application) can be utilized by these processes to enable adherence to the regulations. Giving customers good advice and treating them fairly are aspects of retail banking regulations.

Business event processing can also be used to deliver a business advantage. Customer trust is very important in the banking industry. Improving a customers' perception of a bank and winning back trust is important. Emitting CICS events to WebSphere Business Events so that a customer's patterns of behavior is monitored allows a bank to propose a tailored product package to the customer.

Figure 5-6 on page 83 provides a high-level view of CICS and WebSphere Business Events integration.



CICS Transaction Server V4.1
5
Event Specifications
CICS Applications
EP Adapters
6
WebSphere MQ
WBE Event XML
WebSphere Business Events running within WAS Application Server (can be on z/OS)
WebSphere Business Events Runtime
WBE Message
Queue Connector
WebSphere MQ via e.g. WAS platform messaging
CICS Event Binding Editor
1
2
3
4
Export Event Specifications
Order_Placed
About_to_place
Order_Dispatched
Export
Cancel
Design Data Tool
Design Tool

Figure 5-6 CICS integration with WebSphere Business Events

In Figure 5-6:

1. The CICS event binding editor creates event specifications.
2. The schemas for the event specifications are then exported.
3. The CICS event specifications are used by the WebSphere Business Events Application Developer to define the events data model using the WBE Design Data tool. The data model is the representation of the data that is processed by the WebSphere Business Events application.
4. The business analyst uses the WBE Design tool to define the business event logic based on the specific business event correlation rules and the data model that is defined in the Design Data tool.
5. An event binding for a group of related CICS events is deployed as part of a CICS bundle resource.
6. Events are emitted from CICS in WBE format to a message queue.
7. The events are consumed by WebSphere Business Events and used to initiate actions that are based on specific event patterns.

See 8.3, “Working with WebSphere Business Events” on page 150 for detailed information about integrating CICS events with WebSphere Business Events.

5.6.1 Credit card fraud and CICS events

Credit card fraud detection needs are not new, and all banks have implemented solutions to protect against misuse of credit cards. However, the risk of fraud continues to rise, and there remains room for improvements and extensions to existing fraud solutions.

In 3.4.1, "Improved credit card fraud detection" on page 39, we discussed the requirement for banks to detect credit card fraud either better or faster. Enabling events in CICS can help with this, and it can also be useful in helping to ensure that regulations that require banks to detect potentially fraudulent situations are satisfied. Events that relate to bank card usage can be emitted from CICS to WebSphere Business Events, and then used to check for unusual patterns of behavior, such as a new card ordered within a week of an address change request.

Much earlier recognition of fraud is hugely valuable because it enables the bank to take earlier actions to address the fraud and to help prevent further fraud:

- Credit card agencies typically identify four suspicious banking transactions before they realize something is wrong. The agencies bear the cost of fraud, so CICS events can be employed to recognize a suspected fraud in a fewer number of suspicious transactions, perhaps within two suspicious transactions, which of course can save the credit card company from bearing the cost of the next two transactions.
- Credit card authorization must be fast. Where there is off-line or separate processing within the authorization process, the identification of suspicious banking transactions can take a long time, hours in most cases. CICS events can be employed to reduce this period. An in-flight banking transaction can be checked, in near real-time, by emitting events from a CICS application to WebSphere Business Events. If the pattern of events suggests that a potential fraud occurred, a set of predetermined actions can be initiated.

WebSphere Business Events provides tooling for IT users to define building blocks that represent the events to be received and processed by the system and their data. Business users can then use previously defined building blocks in the definitions that they create for conditions (also known as *filters*) and *interactions*. Defining interactions (event processing logic) involves defining the events, conditions, and actions groupings.

Figure 5-7 on page 85 shows an interaction set that we defined using WebSphere Business Events tooling that builds on the credit card fraud scenario that we previously discussed.

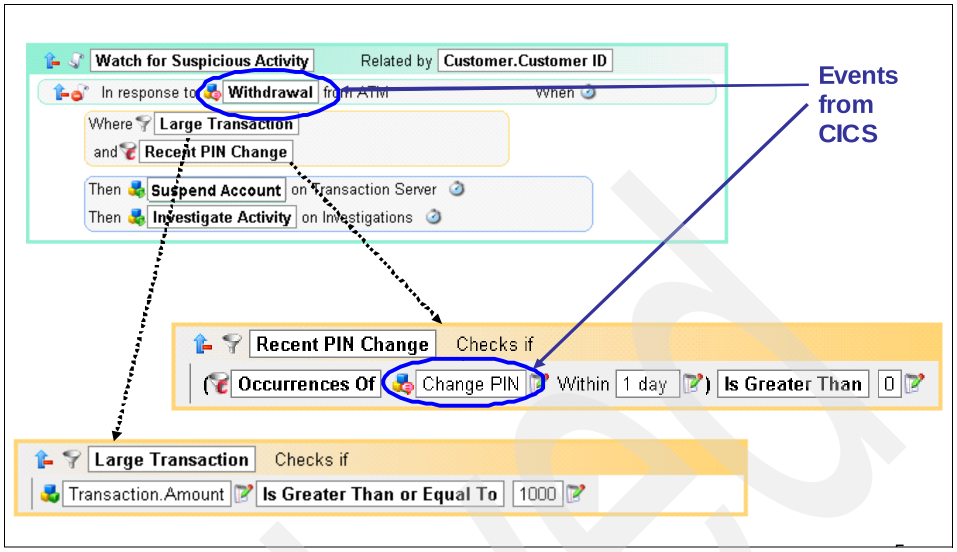

Watch for Suspicious Activity
Related by Customer.Customer ID
In response to Withdrawal from ATM
When
Where Large Transaction
and Recent PIN Change
Then Suspend Account on Transaction Server
Then Investigate Activity on Investigations
Recent PIN Change Checks if
(Occurrences Of Change PIN Within 1 day) Is Greater Than 0
Large Transaction Checks if
Transaction.Amount Is Greater Than or Equal To 1000
Events from CICS

Figure 5-7 Matching event patterns with WebSphere Business Events

The interaction is for "Watch for a Suspicious Activity," which specifies that when a Withdrawal event occurs, check whether it is a large transaction (as defined by a Large Transaction filter). If it is a large transaction, also check whether there was an occurrence of a Change Pin event for this customer recently (within one day). If so, this is considered as a suspicious activity pattern.

Consider a situation in which a CICS application carries out PIN processing and processing of withdrawals from ATMs and CICS event processing support is enabled. Where the PIN processing transaction writes a new PIN to a CICS file, a Change PIN event is triggered that passes information about the customer making the PIN change. In addition, the program that processes ATM withdrawals emits a Withdrawal event that passes information about the withdrawal transaction and the customer who makes the withdrawal.

If WebSphere Business Events detects a large withdrawal and PIN change event pattern occurring within a one-day period, it creates and emits action messages for suspicious activity and fraud investigation:

1. A Suspend Account message is sent to the bank's transaction processing system to suspend the account.

   Note: This can be done by sending a Web service request to CICS.

2. An Investigate Activity is sent to the fraud department requesting that they investigate the case.

In Chapter 8, “Enabling business events” on page 139, we show how we enabled event processing in the CICS core banking component of the IBM Montpellier smarter banking showcase. We also show how these events are consumed by WebSphere Business Events and how WebSphere Business Events initiates automatic actions as a result of specific event patterns.

5.7 Summary

The support for business event processing in CICS TS V4.1 provides the strategic direction for integration with event processing products in the WebSphere portfolio. CICS support for events allows CICS applications to emit business events in a non-invasive way, and a new SIGNAL EVENT API is also provided to add explicit event-enabling points into applications where an additional level of control is required.

6

Web 2.0 support

The Internet was designed for sharing and browsing documents as Web pages through a Web browser. Web 2.0 builds on the technology and availability capabilities of the traditional Web (Web 1.0) to allow users to run software applications entirely through a browser. Users can own the data on a Web 2.0 site and exercise a level of control over the data that they do not have in the Web 1.0 world.

You can use the new functionality in CICS TS 4.1 to make CICS data and processing available to the Web 2.0 world, which allows you to make better use of existing resources and maintain the safety and security of these resources within CICS.

6.1 Web 2.0 introduction

Web 2.0 has a varied meaning to most people. To many it is social media, such as Facebook, tagging, wikis, and blogs. Others believe it is a philosophy of user participation. More technical users see it as a platform on which rich Internet applications (RIAs) can run, providing the performance and usability characteristics of traditional desktop applications as it runs in a Web browser. These applications can be mashups or RIAs where dynamic information from multiple sources is combined, often producing novel results, or dashboards, which is another type of RIA where information from many sources is shown together in one place, often for ease of tracking and managing projects. The dashboard information can be from mashups or direct data and is often provided with options about how (and whether) it is displayed.

Interesting fact: The term Web 2.0 recently became the one millionth word or term in the English language.

6.1.1 Web 1.0 versus Web 2.0

The Internet is no longer just a collection of Web pages. It is now also a collection of *information sources* and *services*, which form the foundations for Web 2.0 applications.

To highlight the benefit of a Web 2.0 application over a Web 1.0 one, let us consider the example of a user who wants to check up on the latest news stories.

In a Web 1.0 world, they check a number of their favorite news Web sites one or two times a day, which is not ideal because they must visit a number of news pages. Visiting the Web sites is time consuming, and they might not be able to check as often as they want and they occasionally miss critical updates.

In a Web 2.0 world, things are better. The news sites have published feeds *(information sources)* of their latest stories, and the user has a feed reader *(Web 2.0 application)* that can poll the feeds for updates much more often. The feed reader can alert the user to new stories as soon as they are published, and the user only has to look in one place to see all of their news. Also, the user can configure his feed reader to show him only stories that are local to him or only those on a subject of interest *(customizable experience)*. The feeds from the news sites can be consumed by a number of users for many purposes. Some might mash the feed with a map *(service)* to show the locations of the news stories, and others might combine the news feed with a feed of information about the people or places that are involved to augment the story *(mashup)*.

The point is that the control is in the hands of the user. The news sites made their information available in a consumable way; therefore, users can control what information they want to see by selecting feeds of interest, and users can control how the information is displayed and actioned by choosing the feed reader application that meets their needs.

Enabling Web 2.0

Enabling a business for Web 2.0 is not as simple as providing support for certain technologies or standards, for example, a business that redesigns their Web site to use PHP and JavaScript with AJAX is no more Web 2.0 enabled than it was before. Customers still come to the Web site and consume it in the way that the business dictates. Even if the Web site is prettier and more usable than it was before, it is still very much Web 1.0. Figure 6-1 shows how in a Web 1.0 application the user has no choice but to use the business' Web site to access its information and services.

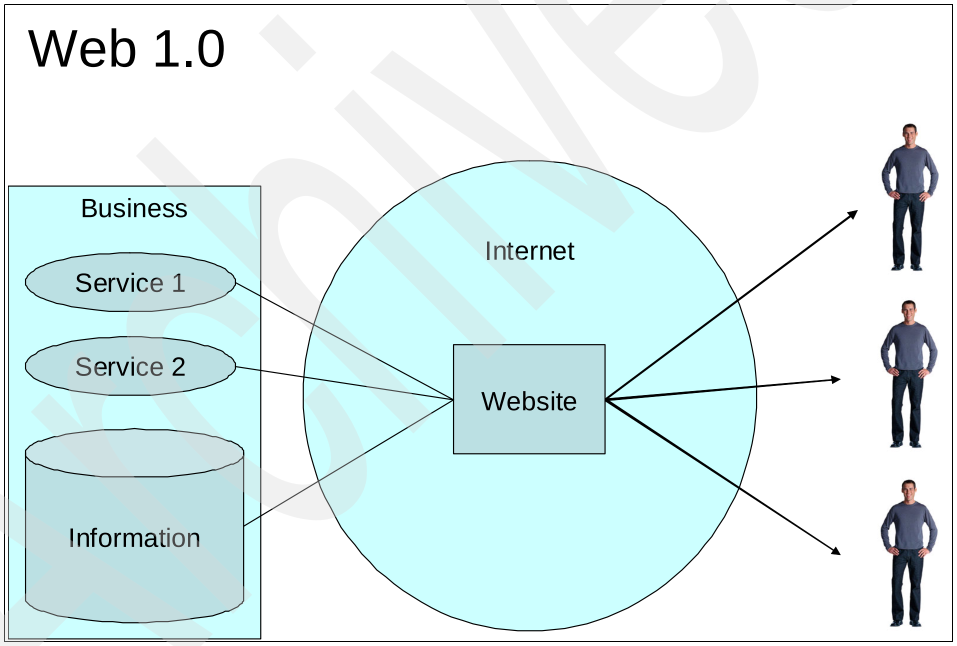


Figure 6-1 Web 1.0

To truly enable a business for Web 2.0, we must start to think about its online presence not only being made up of Web pages but also of information sources and services that can be consumed in a Web 2.0 fashion. Figure 6-2 on page 90 shows how, after published, the business' information and services can be consumed in a number of ways. Users interact with the business using a method that suits their needs and can even draw value from integration with third-party applications and services, something that was impossible in a Web 1.0 world.

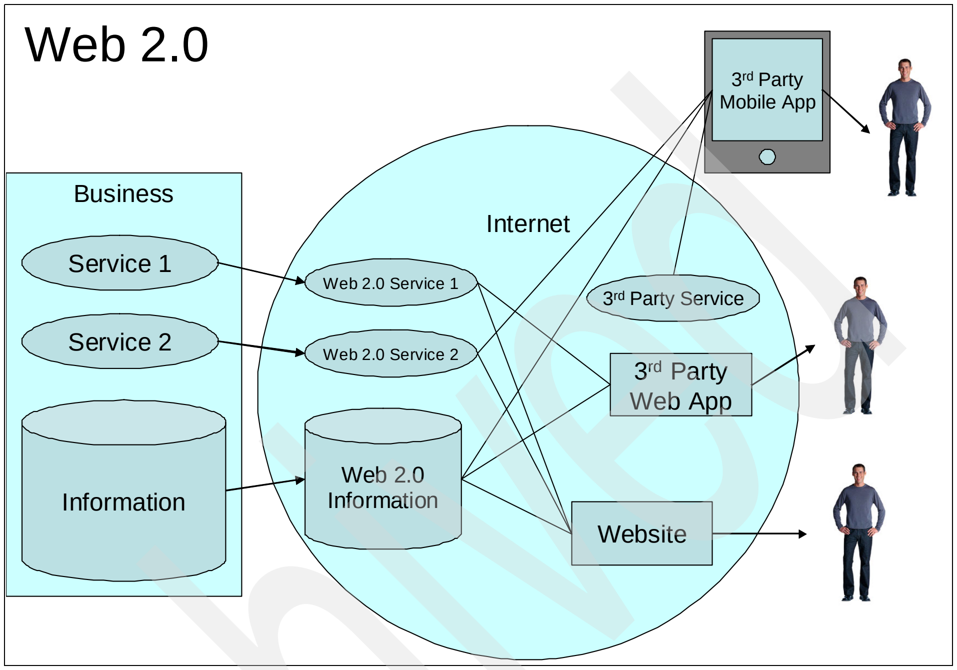


Figure 6-2 Web 2.0

6.1.2 Web 2.0 technologies

A collection of technologies is used in Web 2.0. Representational State Transfer (REST), feeds, XML, JSON, and AJAX are used to provide information in standard formats so that other applications can understand and use them. Other Web 2.0 technologies include blogs and wikis.

REST

Representational State Transfer (REST) is an architectural style that is used extensively in Web 2.0 technologies. It describes a way for client applications to access resources that are hosted by a server. A resource is any discrete item that is hosted by the server to which the client requires access, for example, a Web page, a file, a record from a file, a program, or a Web service.

The key principles of REST are that all of the resources to be made available to the client are given unique identifiers, and they are accessed by a uniform interface. The Internet is an example of a RESTful architecture. Its resources are

Web pages that are given unique identifiers (their URLs), and they are accessed over TCP/IP using the standard HTTP requests of GET, PUT, POST, and DELETE.

In Web 2.0 applications, HTTP is commonly used as the protocol for naming and taking action on the resources, and the resource details are usually returned as XML or JavaScript Object Notation (JSON), which is a lightweight data-exchange format that is less verbose than XML.

A bank might want to make a list of its ATM location records available in a RESTful manner over the Internet. To do this, the bank must assign both the list of ATM locations and each individual ATM its own unique identifier. Because the protocol to be used is HTTP, the identifiers are URLs. One sensible scheme might be identifier http://www.Showcase.com/ATMs for the list, and this URL with the termID appended as the identifier for each ATM resource, for example, http://www.Showcase.com/ATMs/ATM0001 for an ATM with termid ATM0001, http://www.Showcase.com/ATMs/ATM0002 for an ATM with termid ATM0002, and so on. Then standard HTTP requests can be used to manage the list, as shown in Table 6-3 on page 95.

Note: www.Showcase.com in our examples refers to the smarter banking showcase. See Chapter 9, “Enabling Web 2.0” on page 167 for detailed information about how we enabled the smarter banking showcase for Web 2.0.

RESTful architectures are also required to be *stateless*. Any client application wanting to access the resources must pass all state information with each request, for example, if the application is browsing through a list of items, the client must not rely on the server to know where it got to in the list on its last request, but in each request to the server, it must pass enough information to ask for the next record, and the server must make sure that the client has the information it needs to do this.

Feeds

Feeds are the basis of many Web 2.0 applications. They make collections of data available in a standard XML format; therefore, when new entries are added to the collection (or existing entries are changed) interested parties are automatically informed.

With feeds, information comes to the user rather than the user searching for it. Feed reader software (*aggregator* or *news reader*) checks for new feed entries at user-definable intervals. Collections of data that change frequently make good use of feed technologies, so one common use is for news sites to make their stories available as a feed. In this way, when a new story breaks, it is simply added to the collection and interested parties know as soon as their feed reader next checks for updates.

Because a feed is made available in a standard format, it can be understood by more than feed readers. A more complex use of a feed, depending on its content and suitability, include:

- Processing it using JavaScript to produce an appealing Web page
- Displaying it with other information in a dashboard or RIA, and perhaps making it available using a widget that can be included and customized as the user wants
- Combining it with information from other sources in a mashup

Feeds are available in two main formats: Really Simple Syndication (RSS) and Atom. Mashup software, feed aggregators, and blogs generally support both RSS and Atom formats.

RSS

RSS, which is commonly expanded to Really Simple Syndication, was the first type of feed format commonly available. Within RSS are two major and several minor commonly used formats. The most recent format at the time of writing is RSS 2.0.1.

Atom

The newer type of feed format, Atom, is defined by the Internet Engineering Task Force (IETF). Atom feeds are defined by two specifications:

1. The first specification is the Atom Syndication Format, defined by RFC 4287, which defines feeds and entries and is targeted at providing read-only access to resources. Table 6-1 shows access is through the HTTP GET method.

Table 6-1 RESTful access to a feed

HTTP request	URL to which the request is directed	Action on feed
GET	Entry	Returns representation of entry
	Feed	Returns a list of representations of entries

2. The second specification is the Atom Publishing Protocol, defined by RFC 5023, which defines an "editable" feed called a collection. The "editing" is done by RESTful HTTP methods, as shown in Table 6-2 on page 93. RFC 5023 also covers the idea of partial lists, providing segmented feeds so that there is no need to get an entire collection in one go. Each partial list links to

the next partial list and, optionally, the first, last, and previous partial lists in the collection.

Table 6-2 RESTful access to a collection

HTTP request	URL to which the request is directed	Action on feed
GET	Entry	Returns representation of entry
	Collection	Returns a list of representations of entries (whole or partial list)
POST	Collection	Creates a new entry in the collection
PUT	Existing entry	Updates the existing entry
DELETE	Existing entry	Deletes the existing entry from the collection

An Atom feed document consists of one *feed* element that contains one or more child entry elements. It also contains the entries, and a feed element provides information about the feed in general. It has several required elements:

- Title element to specify the title of the feed
- ID element to uniquely and permanently identify the feed
- URL to be used to retrieve the feed, which is specified as a link element with the rel attribute value of self
- Updated element to show the date and time that the feed was last significantly altered
- Author element to display details that relate to the author of the feed. Individual entries can specify their own authors. If every entry specifies its own, there is no need to specify an author in the feed element

The feed element can also contain a number of optional elements, such as a subtitle to provide subscribers more information about the feed's purpose, and a generator to supply the agent that was used to generate a feed. Feed documents that relate to collections can also contain information that relates to the first and last entries in the collection, and, if the document only contains part of the collection, links to the next and previous pages.

Each entry is also required to have its own ID, title, link, and updated elements. Optional entry elements include a summary of the entry and its contributor(s). An entry also contains a content element to contain the data that the feed provides. This content element specifies what type of data the feed contains, which can be normal text (the default), HTML, XML, and so on. The content can be a link to the data or the data itself.

Example 6-1 shows an example of a simple Atom feed.

Example 6-1 Example of a simple Atom feed

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title type="text">Showcase ATM Feed</title>
  <subtitle>A feed of location data for Showcase ATMs</subtitle>

  <link rel="self" href="http://www.Showcase.com/ATMs"/>
  <id>tag:Showcase.com,2009-12-17:ATMs</id>
  <updated>2009-12-17T17:05:07+00:00</updated>

  <author>
    <name>Showcase</name>
    <email>inquiries@Showcase.com</email>
  </author>

  <entry>
    <title>ATMT001 ROMSEY</title>
    <link rel="self" href="http://www.Showcase.com/ATMs/ATM0001"/>
    <id>tag:Showcase.com,2009-12-17:ATMs:ATMT001</id>
    <updated>2009-12-17T17:05:07+00:00</updated>
    <published>2009-12-17T17:05:07+00:00</published>

    <content type="application/xml">
      <ATMLOCB xmlns="http://www.ibm.com/xmlns/cics/atmlocb">
        <address>68, The Hundred, Romsey, Hants</address>
        <country>UK</country>
      </ATMLOCB>
    </content>
  </entry>

  <entry>
    <title>ATMT002 MONTPELLIER</title>
    <link rel="self" href="http://www.Showcase.com/ATMs/ATM0002"/>
    <id>tag:Showcase.com,2009-12-17:ATMs:ATMT002</id>
    <updated>2009-12-17T17:05:07+00:00</updated>
    <published>2009-12-17T17:05:07+00:00</published>

    <content type="application/xml">
      <ATMLOCB xmlns="http://www.ibm.com/xmlns/cics/atmlocb">
        <address>Rue de la Vieille Poste, Montpellier, France</address>
        <country>France</country>
      </ATMLOCB>
  </entry>
```

```
</feed>
```

Table 6-3 shows an example of how HTTP requests are used to manage the list of ATM locations that are displayed in Example 6-1 on page 94.

Table 6-3 RESTful access to ATM location resources

HTTP request	**URL to which the request is directed**	**Action**
GET	http://www.Showcase.com/ATMs/ATM0001	Returns ATM location details for ATM0001
	http://www.Showcase.com/ATMs	Returns a list of ATM location records
POST	http://www.Showcase.com/ATMs	Creates a new ATM location record
PUT	http://www.Showcase.com/ATMs/ATM0001	Updates the existing ATM0001 location record
DELETE	http://www.Showcase.com/ATMs/ATM0001	Deletes the existing ATM0001 location record

See "Terminal Location Feed: Atom feed from VSAM file" on page 171 for information about how to enable Atom feeds with CICS.

Blogs

Blogs (Web logs) are Web sites that can be considered as publicly accessible journals or diaries. Entries are frequently added to the blog and are displayed with the most recent entry first and an archive of older entries following. Usually readers can interactively leave comments on what is written, which the owner of the blog or other readers can answer, as applicable.

Blogs are used both for personal and corporate purposes, and access can be entirely public or limited to a specific group of people. Blogs tend to focus on one specific area, for example, a product, a hobby, or a person. Generally, blog entries combine text, links to other blogs and Web pages, and other media, such as images.

Blogs can be made available as feeds so that interested parties are automatically informed when a new entry is made.

Wikis

A wiki is a database of interlinked Web pages that are managed over the Internet. Users collaboratively create a body of information that can be read and refined by other users. A prime example of a wiki is wikipedia, which is an online encyclopedia. Internet users can update the content of the encyclopedia's pages and create new pages that are referred by and referred to by other pages in the wiki. Stubs are pages that are identified as necessary, but are not yet written. With wikis, one user can create a stub and leave it for another user to populate with information.

Different levels of access can be provided to wikis, for example, most allow anyone to create a new page, but often the ability to delete pages is limited. Wiki pages can be set so that administrators review the updates before they are made available to other users of the wiki.

PHP

PHP is a dynamic scripting language that is used to implement Web sites. It is not strictly a Web 2.0 technology; however, PHP is often used for the server side scripting of Web 2.0 applications. It is used by millions of developers world wide and is the third most popular language (after Java and C), according to the TIOBE Programming Community Index (as of December 2009).

PHP is easy to use, has a gentle learning curve, efficient syntax, and an extensive open source library of functions that can be reused, which saves developers from having to write and debug their own common functions. Impressive results can be achieved with very little code, and the language is suited to rapid incremental prototyping.

6.1.3 Web 2.0 in banking

A great number of applications are required to run a bank. Figure 6-3 on page 97 shows the range and use of the applications that are involved.

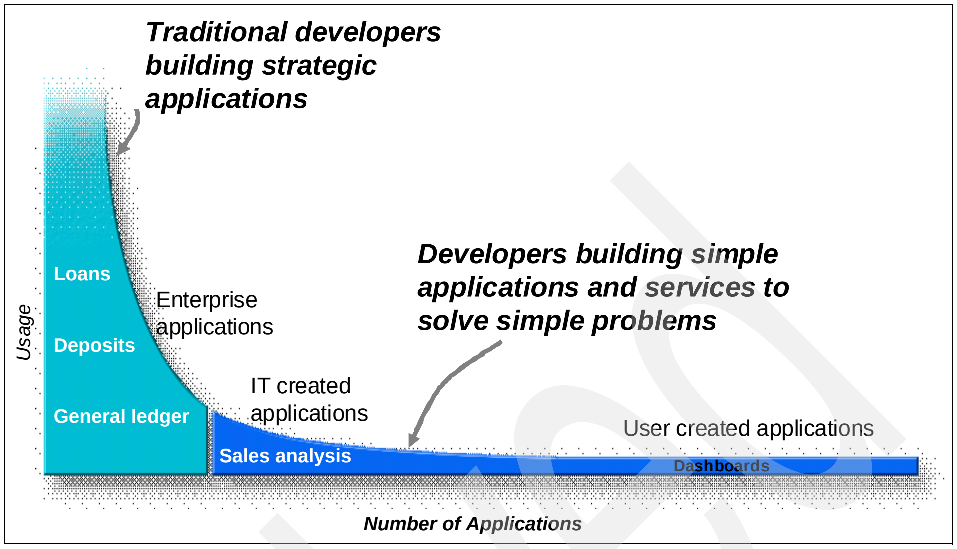


Figure 6-3 Business applications

A small number of these applications are the traditional applications that are required to run the bank and are extensively used, business critical, meticulously designed, and change-controlled applications, such as Loans, Deposits, and General Ledger.

In addition to these enterprise applications are a *long tail* of numerous supporting applications that are not vital for the core business. These supporting applications are far less heavily used and might only be required by a small number of users or only for a short duration, such as for a special promotion.

It is often difficult to justify the resource that is needed to write and maintain these supporting applications because skilled IT staff are needed for the traditional applications that run the enterprise, which can mean that opportunities are lost because of the applications not being written. Web 2.0 technologies can enable less skilled staff to write these supporting applications, which allows new opportunities to be found.

6.2 Why Web 2.0

Web 2.0 technologies allow existing resources and services to be made available using standard Web 2.0 formats, which unlocks their value and provides real-time access to trusted sources of data to new people and applications. The ease of use of the available tooling makes it straightforward and simple for business people to use these resources and write their own long tail applications, which means that the IT department can concentrate on the core business applications and, because they are written by the users themselves, these applications can be designed and updated to do exactly what is required, as it is required. Because these Web 2.0 applications are standards based, they can be combined with other services and resources, which allows reuse of existing work.

Making existing services available over Web 2.0 provides:

- Control

  The services are still kept safe and secure on the server with defined access control.

- Flexibility

  Simple access is provided to the resources, which allows faster and less skilled users to use them, saving both time and money, with no need to wait for the IT department.

- Value

  When sources of information are combined the end result can be more valuable than the sum of the individual resources, so it makes sense to provide them in Web 2.0 standard formats so that they are easily integrated on the glass and encouraging reuse.

  A good example of the value of integrating multiple Web 2.0 feeds is the scenario 3.2.3, “Accounts in multiple banks” on page 36. In this scenario, we see how if banks provide account feeds in a Web 2.0 standard format, a customer can see a consolidated view of his finances; whereas, today he probably has to consult multiple Web sites or pieces of paper.

Web 2.0 technologies also allow functionality to be made available through Web browsers that previously required applications to be installed and maintained on client computers. With zero footprint on the user’s computer, now the only maintenance each application requires is on the server, which saves both time and money.

6.2.1 Using Web 2.0

Business uses of new Web 2.0 technologies are:

- Blogs that are used both internally and externally:
  - Internally they are usually used for communication and collaboration between employees.

   - Externally, corporate blogs are used for advertising new and updated products or for other marketing purposes because they allow interested parties to see the updated information. Blogs tend to be less formal than press releases, which provides a more user-friendly face to a company, which can improve customer loyalty. Access can even be given to allow customers to comment on the company's products.

     For CICS TS, for example, there is the Master Terminal blog to keep interested parties up to date with news on CICS TS:

     `http://themasterterminal.com/`

     A bank might choose to blog about updates to account terms and conditions, new accounts and special offers, and even news of new branches or branch refurbishments to keep their customers informed and engaged.

- Wikis that are mostly used internally to store and distribute information and allow collaboration. Sometimes wikis are available externally, for example, the IBM developerWorks® wikis are provided to allow IBMers and customers to share knowledge and expertise on topics of interest to developers:

  `https://www.ibm.com/developerworks/wikis/dashboard.action`

- Feeds that are used both internally and externally:
   - Internally feeds are used to track and share up-to-the-minute data with interested parties. Some example uses are real-time software test results, the status of server machines, and the location of the company minibus.
   - Externally, feeds provide all sorts of information to customers. IBM feeds, for example, provide information that can include news headlines and press releases, software support, and the latest technical documentation from sources, such as developerWorks and IBM Redbooks Publications.

After information is available in a standard Web 2.0 format it can be found to have far more uses than those for which it was first intended, which creates extra value from the resource, and its unexpected uses can sometimes reap large rewards, for example, if a bank's ATM locations are made available as a feed, this data can have many uses:

- By combining the ATM's feed with Google maps it is easy to see exactly where the ATMs are located within a region, which allows:
   - Customers to find their nearest ATM and a route to it, if required.
   - Staff to see where their ATMs are located to determine whether they need more machines in a particular area or whether they can reorganize to de-commission a machine.

To help with these decisions, the ATM data can be mashed up with other sources of information, for example, feeds from other banks on the location of their ATMs.

- This ATM data can also be provided to other companies, for example, home rentals, sales agencies, or hotel chains can use the data to provide information about the location and distance of the nearest ATM as part of the details that they provide about the location of a house or hotel.

So, after information is available, even simple information in a simple feed, it can provide a variety of uses, quickly and simply, that are available to anyone with access to Web 2.0 tools. All of this can be done without Web 2.0, of course, but with Web 2.0, it can be done quickly and easily, and the data can be reused and combined because of the standards and tools that are available. Because the data is the original, real-time information there is no need to update other sources manually or using custom-written automation or mirroring.

6.2.2 Web 2.0 tools

Many tools exist to make use of Web 2.0 resources, for example:

- Feed readers allow users to subscribe to, read, and often aggregate feeds. Various feed readers also provide filtering and sorting capabilities too, which can be built into Web browsers or standalone applications. Feed readers are also built into mobile devices, such as iPhones or those mobile devices that run the Android[1] operating system.

  See 9.4, “Creating the GPS-enabled mobile application” on page 180 for information about how we developed an Android mobile application to consume Atom feeds.
- Dashboards allow information from multiple sources to be displayed in one view, often with optional widgets and fancy formatting.

A number of IBM tools make it easier for business people to write their own Web 2.0 applications:

- WebSphere sMash (which started out as Project Zero) is a development and execution environment for dynamic Web applications. It provides both a development environment for developers and a way for business people to combine existing services on the glass, for example, Figure 6-4 on page 101 shows how straightforward it is for business people to combine feeds using sMash Assemble flow.

[1] Android is a mobile operating system that runs on the Linux® kernel. It allows developers to write managed code in the Java language, controlling the device using Google-developed Java libraries.

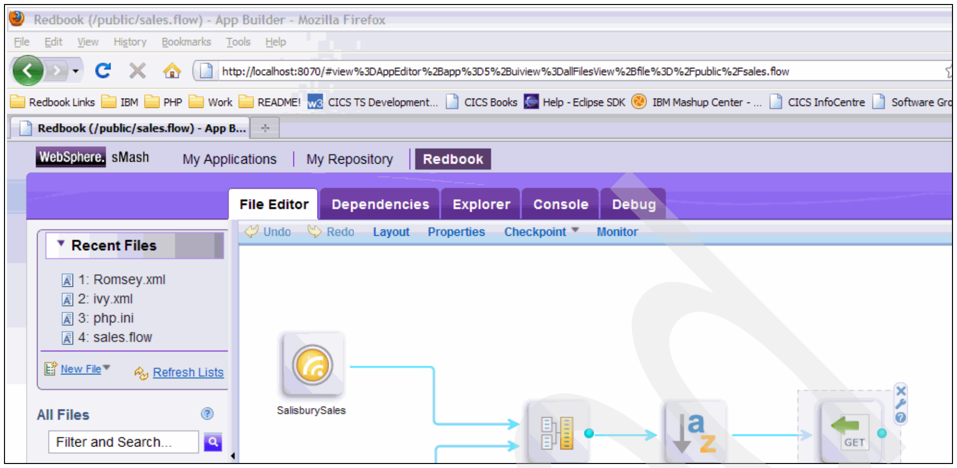

Redbook (/public/sales.flow) - App Builder - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8070/#view%3DAppEditor%2Bapp%3D5%2Buiview%3DallFilesView%2Bfile%3D%2Fpublic%2Fsales.flow
Redbook Links IBM PHP Work README! CICS TS Development... CICS Books Help - Eclipse SDK IBM Mashup Center - ... CICS InfoCentre Software Gro
Redbook (/public/sales.flow) - App B...
WebSphere. sMash
My Applications
My Repository
Redbook
File Editor
Dependencies
Explorer
Console
Debug
Undo
Redo
Layout
Properties
Checkpoint
Monitor
Recent Files
1: Romsey.xml
2: ivy.xml
3: php.ini
4: sales.flow
New File
Refresh Lists
All Files
Filter and Search...
SalisburySales
GET

Figure 6-4 Aggregating feeds with WebSphere sMash

- Lotus Mashups provide a lightweight-mashup environment for assembling personal, enterprise, and Web content into simple, flexible, and dynamic applications.
- InfoSphere™ MashupHub (formerly known as IBM Mashup Hub) provides a lightweight information management environment for IT and business professionals who want to unlock and share Web, departmental, personal, and enterprise information for use in Web 2.0 applications and mashups. It includes visual tools for creating, storing, transforming, and remixing feeds to be used in mashup and situational applications. It also includes a central catalog where users can tag, rate, and share mashable assets.
- IBM Mashup Center is an end-to-end enterprise mashup platform that supports rapid assembly of dynamic Web applications that meet specific management, security, and governance requirements. It is designed around the primary goal of making it fast and easy for users of all skill levels to create simple Web applications from existing information sources by dragging and dropping widgets onto the page and then wiring them together on-the-glass.
- Enterprise Generation Language (EGL) is a business-oriented programming language that provides a simple way to build Web 2.0 applications. EGL Rich UI is available in IBM Rational Business Developer 7.5.1.

6.3 Web 2.0 in CICS Transaction Server for z/OS

By unlocking existing CICS resources through standard Web 2.0 interfaces, using CICS TS V4.1 you can better use existing enterprise data. This feature allows business people to roll their own long tail applications based on CICS resources. Some of the ways that CICS makes its resources available through Web 2.0 interfaces are:

- Atom feeds

  The Atom support in CICS makes files and temporary storage queues available as RESTful resources over HTTP. It also allows programs to be written to make any resource available to CICS available as a feed, for example DB2.

- REST support in CICS using PHP SupportPac CA1S

  This SupportPac allows PHP scripts to be run in CICS in response to HTTP requests, and these scripts can make use of CICS resources. This also permits PHP scripting language capabilities to extend the range of CICS assets that can be exposed as REST services.

- RESTful interface into CICSPlex SM and CICS

  The CICS management client interface (CMCI) allows CICS and CICSPlex SM resources to be managed over HTTP. This RESTful interface is used internally by the CICS Explorer.

The CICS implementation leverages the existing CICS HTTP support, including security capabilities, such as, basic authentication and SSL, so that resources are protected from unauthorized access.

6.3.1 Atom feeds

Atom feeds provide a standard way to publish the data that is locked inside CICS. After this data is RESTfully available as XML, it can participate in the Web 2.0 world that we previously described. You can even store blogs in CICS by using blogware that conforms to the Atom publishing protocol, for example, Windows® Live Writer.

To support Atom feeds, a new ATOMSERVICE resource was introduced to CICS. Various types of Atom documents can be obtained using the four types of ATOMSERVICE:

- Atom entry documents contain a single item of information, which is known as an entry. These documents are available from both ATOMTYPE(FEED) and ATOMTYPE(COLLECTION) ATOMSERVICEs.
- Feed documents can contain several entries and implement paging so that partial lists of entries can be returned, which is useful when large resources are being queried. An ATOMTYPE(FEED) ATOMSERVICE provides read-only access to its resource, and an ATOMTYPE(COLLECTION).

ATOMSERVICE allows the resource that it represents to be updated using RESTful HTTP requests.

- Service documents, which list the editable collections (those with ATOMTYPE(COLLECTION)) that are available from a server, are provided by ATOMTYPE(SERVICE) ATOMSERVICEs.
- ATOMTYPE(CATEGORY) ATOMSERVICEs provide category documents, which contain lists of categories for the entries in a collection. Categories can be specified in a service document, but separate category documents are useful if you want to use the same categories to define multiple Atom feeds.

Table 6-4 illustrates the difference in access between ATOMTYPE(FEED) and ATOMTYPE(COLLECTION) resources. The file API commands show the action that the HTTP request has in CICS if the ATOMSERVICE resource supports that type of request and represents a file.

Table 6-4 Actions on ATOMTYPE(FEED)s and ATOMTYPE(COLLECTION)s

HTTP request	URL to which the request is directed	Action on ATOMTYPE FEED	Action on ATOMTYPE COLLECTION	File API
GET	Entry	Returns representation of entry	Returns representation of entry	EXEC CICS READ FILE
	Collection	Returns a list of representations of entries (whole or partial list)	Returns a list of representations of entries (whole or partial list)	EXEC CICS READ FILE
POST	Collection	Not allowed	Creates a new entry in the collection	EXEC CICS WRITE FILE
PUT	Existing entry	Not allowed	Updates the existing entry	EXEC CICS READ FILE UPDATE then REWRITE FILE
DELETE	Existing entry	Not allowed	Deletes the existing entry from the collection	EXEC CICS DELETE FILE (where necessary this is a logical delete that replaces the record with one byte of high bytes)

To expose a file or temporary storage queue as an Atom feed, all that is needed is the creation of the CICS resources, as shown in Figure 6-5.

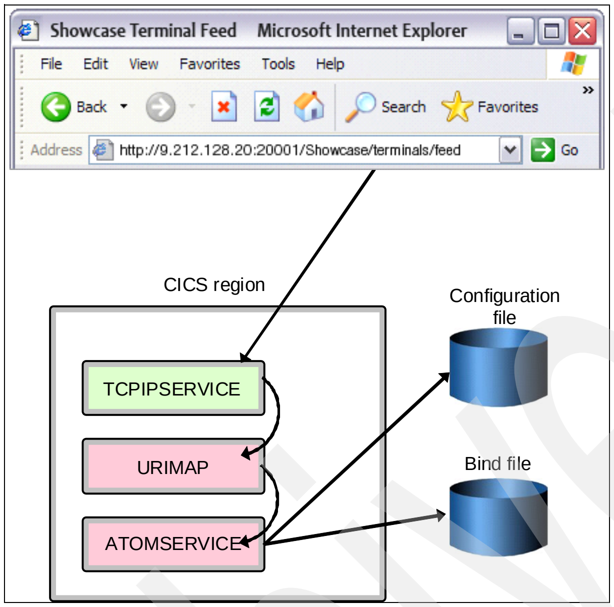

Showcase Terminal Feed   Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://9.212.128.20:20001/Showcase/terminals/feed Go
CICS region
TCPIPSERVICE
URIMAP
ATOMSERVICE
Configuration file
Bind file

Figure 6-5   Example resources for a FILE ATOMSERVICE

We provide a summary of the required resource definitions in the next immediate sections. See “Terminal Location Feed: Atom feed from VSAM file” on page 171, for detailed information about how to configure an Atom feed based on a VSAM file.

TCPIPSERVICE resource

The TCPIPSERVICE defines the port on which CICS can receive HTTP requests for ATOMSERVICEs, which is where you specify the security measures, such as SSL, that protect your Atom feeds and collections from unauthorized access.

URIMAP resource

The URIMAP maps the URL that is specified in the HTTP request to the ATOMSERVICE that is to be used to service the request.

ATOMSERVICE resource

The ATOMSERVICE defines the Atom service, feed, collection, or category document that CICS delivers in response to the HTTP request. This resource specifies the CICS resource that provides the data for the feed, the Atom configuration file that defines the feed document, and usually the XML binding that describes the layout of the resource.

ATOMSERVICE Configuration file

The Atom configuration file is stored in z/OS UNIX® System Services. It contains XML that specifies metadata and field names for the Atom document that is returned for this resource definition. If the whole resource is not to be made available in the feed, the configuration file is where the fields that are to be exposed are defined.

ATOMSERVICE Bind file

The bind file is also stored in z/OS UNIX System Services. This file automatically converts the EBCDIC binary or packed decimal data in the file or on the temporary storage queue into the ASCII string data that is returned in the Atom feed XML. You can use the CICS XML assistant utility DFHLS2SC to create this file from the language structure that describes the file or queue record. When the ATOMSERVICE is installed, this binding is incorporated in a new XMLTRANSFORM resource. At run time, the parsing is carried out by IBM z/OS XML System Services and so is eligible for zAAP offload.

The PROGRAM resource does not require a bind file because it does its own conversion. It can choose to do this by using the new XMLTRANSFORM functionality.

FILE or TSQUEUE resource

New or existing files and temporary storage queues can be exposed as feeds, which means that any application can easily make its data available just by writing it to a temporary storage queue.

To access a specific record from a file or queue, specify a selector in the HTTP request's URL. CICS identifies a suitable selector value depending on the type of resource. For a temporary storage queue, the selector value is the number that identifies the record in the temporary storage queue. For a file, the selector value is the key for the file, which must be unique.

Important: You can expose CICS files and temporary storage queues as Atom feeds without writing application programs.

PROGRAM resource

It is not just files and queues that can be made available as Atom feeds. New service programs can be written to expose other resources, such as DB2 databases or existing programs that are available in a RESTful way.

CICS links to the service program for each entry that is needed to satisfy the HTTP request, communicating with it through containers. To help with writing service programs, the following changes were made to the CICS application programming interface for CICS TS V4.1:

- New TRANSFORM DATATOXML and TRANSFORM XMLTODATA commands convert application data to XML and convert XML back to application data.
- A new BIF DIGEST command calculates the SHA-1 digest of a string of data. You can use the result as a strong entity tag (ETag HTTP header) on HTTP messages that are sent out from CICS or to make an HTTP PUT request conditional.
- New WEB READ QUERYPARM and browse QUERYPARM commands read or browse keyword parameters that consist of name and value pairs, from a query string in the HTTP request's URL.
- The existing CONVERTTIME and FORMATTIME commands now understand the RFC 3339 format that Atom requires.
- The ABSTIME value that is returned by the ASKTIME command is no longer rounded to the nearest hundredth of a second, which allows the required accuracy for RFC 3339.

6.3.2 PHP

The REST support in CICS using PHP SupportPac (CA1S) extends access to CICS resources through Web 2.0 interfaces. This SupportPac enables PHP Programmers to:

- Write PHP scripts that are invoked in CICS in response to inbound HTTP requests
- Exploit existing CICS COMMAREA applications from PHP code
- Easily create RESTful services to widen and simplify Web 2.0 access to CICS applications and data
- Access DB2 tables through PHP Data Objects (PDO) using the CICS Java Database Connectivity (JDBC) driver

PHP scripts are quick to write and easy to alter, providing a method of prototyping CICS applications and perhaps a gentle introduction to the mainframe world for new starters. Running in CICS, the scripts can access any Java class using the Java bridge for PHP and manage units of work.

Support for the PHP scripting language is provided using a subset of the WebSphere sMash product, which supports almost all of the language features of PHP 5.2 and a subset of the functions that PHP extensions provide. Being Java based, it is eligible for zAAP offload.

Full installation instructions are provided in the SupportPac documentation, including information about the prerequisites and supplied samples that you can use to verify correct installation. These instructions assume that your CICS is already set up to run Java, so if this is not the case, set that up first.

Exploiting CICS COMMAREA applications from PHP code

After the SupportPac is installed, all that is needed to be able to LINK to an existing CICS PROGRAM is to create the CICS resources that are summarized in Figure 6-6. This resource can be a LINK to a PROGRAM that is local to the CICS on which the script is running or a DPL.

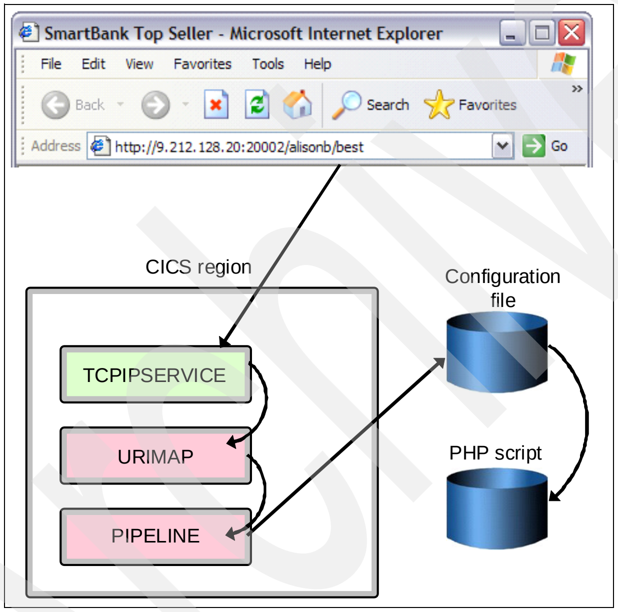

SmartBank Top Seller - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://9.212.128.20:20002/alisonb/best
Go
CICS region
TCPIPSERVICE
URIMAP
PIPELINE
Configuration file
PHP script

Figure 6-6 Example resources for a PHP script to LINK to an existing CICS PROGRAM

TCPIPSERVICE resource

The TCPIPSERVICE defines the port on which CICS can receive HTTP requests to run the scripts, which is where you specify the security measures, such as SSL, that protect your scripts from unauthorized use.

URIMAP resource

The URIMAP maps the URL that is specified in the HTTP request to the PIPELINE that is to be used to service the request.

PIPELINE resource

The PIPELINE specifies the configuration file that contains information about how the HTTP request is processed.

PIPELINE Configuration file

The PIPELINE configuration file is stored in z/OS UNIX System Services. It provides information about the message handler program that acts on the HTTP request and provides the response. If you want one specific script to be run in response to the request, specify it here to be passed to the handler program. If a specific script is not specified here, the script name or resource details can be provided in the URL, which allows many scripts or REST support for a number of resources to be provided by one set of CICS resources.

PHP script

The PHP script itself is also stored in z/OS UNIX System Services. This script can be in EBCDIC or ASCII, depending on your PHP configuration file setting. To alter the script, save the changes and re-run the request.

COMMAREA data conversion

COMMAREA data structures are represented in the PHP scripts by generated Java classes. To create the class parse binary ADATA files, created by the COBOL compiler, use the supplied JZOS tooling to produce a Java source file that must then be compiled. This class defines getter and setter methods for each field in the copybook and the data structure itself. These methods can be used by the script to read and write COMMAREA data to communicate with the CICS PROGRAM.

RESTful access to CICS resources through PHP

You can also use the SupportPac to make CICS PROGRAMs available RESTfully over HTTP.

As with the Atom support, the HTTP request can be directed at a URL that represents a single item in a collection or at a URL that represents the collection itself. For each HTTP method, for each of these types, Table 6-5 shows the event type that is returned to the script or the event handler method that is called.

Table 6-5 Events and handler methods for each type of request

HTTP request	URL to which the request is directed	Event	Event handler method
GET	Entry	retrieve	onRetrieve()
	Collection	list	onList()

HTTP request	URL to which the request is directed	Event	Event handler method
POST	Entry	postMember	onPostMember()
	Collection	create	onCreate()
PUT	Entry	update	onUpdate()
	Collection	putCollection	onPutCollection()
DELETE	Entry	delete	onDelete()
	Collection	deleteCollection	onDeleteCollection()

Supplied functions (zget and zlist) allow the script to obtain details of the HTTP request, such as its event type, and URL and input data, with which it was initiated. Alternatively, you can implement the event handler methods shown in Table 6-5 on page 108 (or the subset you require), which are automatically called for the appropriate combination of HTTP method and URL.

After, by whichever method, the script determines why and for what it was called, it can then LINK to an existing or new CICS application to service this request, passing appropriate information to it and receiving results back in a COMMAREA. The script can then format these results and return them in the HTTP response. Alternatively, instead of LINKing to a CICS PROGRAM, the program logic can be written in the script itself, perhaps using Java APIs or through PDO using the CICS JDBC driver to access DB2.

6.4 Summary

CICS TS V4.1 provides support for Atom feeds and Atom Publishing Protocol collections. Existing CICS business data can be exposed as a feed and therefore made available to Web 2.0 applications and mashups. This flexibility allows more value to be obtained from these existing resources. The REST support in CICS using PHP SupportPac (CA1S) extends CICS Web 2.0 support by providing direct or RESTful access to existing and new CICS programs.

Part 3

IBM Smarter Banking showcase

The IBM Smarter Banking showcase is a simulation of a real bank. In this part, we provide an overview of the showcase architecture and current infrastructure.

First, we show a typical retail banking workload that consists of banking transactions, such as cash withdrawals, cheque deposits, and account transfers.

We then use the smarter banking showcase to highlight how new technologies, such as complex events processing and Web 2.0, can be enabled, and how these technologies assist with the end-to-end visibility and the integrated-user experience challenges that banks face today.

We describe how we unlock valuable business data from the banking system non-invasively:

- By emitting business events from the core banking transactions, thus allowing quicker response to unusual account activity.
- By enabling real-time Atom feeds of account transactions, thus providing direct customer access to information that can be 'mashed up' with information from other sources.

7

The IBM Smarter Banking showcase environment

In this chapter, we describe the key business drivers for the smarter banking showcase and provide an overview of the showcase architecture and current infrastructure.

7.1 Introduction to the Smarter Banking showcase

The IBM Smarter Banking showcase is a simulation of a real bank. A team of banking and IT infrastructure specialists working in the Banking Centre of Excellence at the IBM Montpellier location in France created and developed the showcase.

The banking systems that are used in the showcase run a mixed workload of real-world financial transactions, including cash withdrawals, deposits, mortgages, and car loans. The COBOL core banking application runs on CICS and stores customer and account records in DB2 for z/OS.

The initial focus of the showcase team was to create a multi-channel core banking environment that could run a representative banking workload. As the showcase evolved, we adopted an approach based on service-oriented architecture (SOA) for re-using the CICS core banking system. We also developed a comprehensive monitoring solution to provide a real-time status of the health of the IT infrastructure.

The fundamental components of multi-channel integration, core banking renovation, and advanced systems management still form the basis of the showcase. However, the showcase also developed in other directions. In particular, we added new scenarios based on products from Independent Software Vendors (ISV), and we use virtual world technologies, such as Second Life and OpenSimulator to provide remote access to the ATM that is installed at the demonstration center in IBM Montpellier.

The evolution of the showcase continues with this book with the introduction of complex business event processing. We enabled business events in the CICS core banking system, and we implemented WebSphere Business Events and the use of Web 2.0 technologies.

7.1.1 Showcase objectives

The objectives of the smarter banking showcase project are to:

- Demonstrate a live banking operating environment (consisting of IBM and ISV products) that is based on the IBM Banking Framework and a dynamic infrastructure.
- Share a vision of a modern, efficient, smarter bank that can cope with the IT challenges of today and position itself for the challenges of tomorrow.

- Demonstrate the linkage between infrastructure and business value to CIO and Line of Business (LoB) management through a set of scenarios that represent banking pain points and opportunity areas.
- Use a realistic mix of workload and run this workload at operational volumes that are representative of a typical European bank:
  - Six million clients in z/OS DB2 database
  - Twelve million accounts in z/OS DB2 database
  - Average daily online transaction throughput of about 300 transactions per second, rising to a peak of up to 800 transactions per second

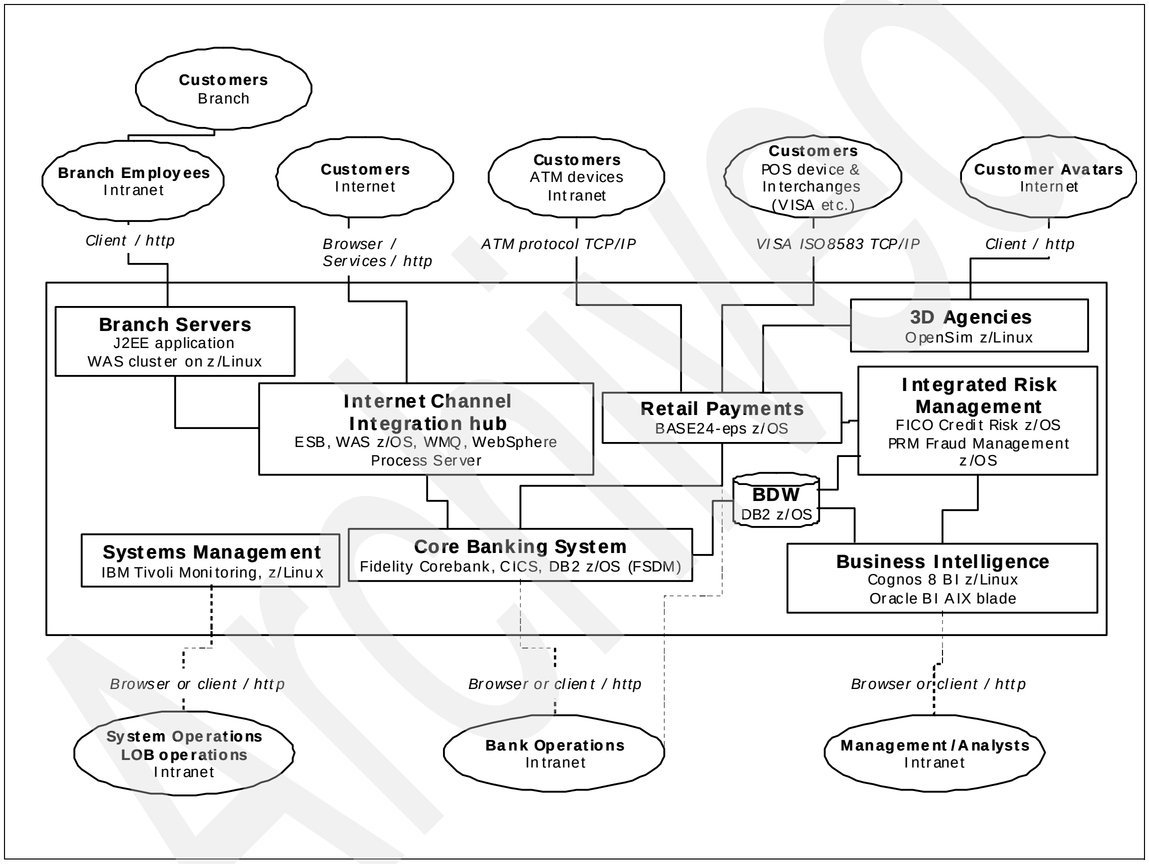
To achieve these objectives, we use the showcase to demonstrate the following proof-points:

- The ability of the IT infrastructure to provide optimized customer service (fast and consistent response times) across multiple channels and simultaneously respond to varying peaks in branch, Internet, Point of Sale (POS), and ATM traffic
- Consolidate and centralize multiple branch IT environments to increase availability and boost performance of teller support
- Key resiliency capabilities in planned and unplanned failure scenarios and simultaneously always retaining customer service
- How system capacity can be dynamically increased and removed to address peak workloads and immediate business opportunities
- Improved management of operational risk by defining a clear relationship between IT infrastructure and business services

7.2 Showcase architecture

In this section, we provide an overview of the showcase operational and logical architecture.

Figure 7-1 shows the banking IT architecture, including the various channels and major software components. It is a logical representation or map of the business components or building blocks of the bank.



Customers Branch
Branch Employees Intranet
Customers Internet
Customers ATM devices Intranet
Customers POS device & Interchanges (VISA etc.)
Customer Avatars Internet
Client / http
Browser / Services / http
ATM protocol TCP/IP
VISA ISO8583 TCP/IP
Client / http
Branch Servers J2EE application WAS cluster on z/Linux
3D Agencies OpenSim z/Linux
Internet Channel Integration hub ESB, WAS z/OS, WMQ, WebSphere Process Server
Retail Payments BASE24-eps z/OS
Integrated Risk Management FICO Credit Risk z/OS PRM Fraud Management z/OS
BDW DB2 z/OS
Systems Management IBM Tivoli Monitoring, z/Linux
Core Banking System Fidelity Corebank, CICS, DB2 z/OS (FSDM)
Business Intelligence Cognos 8 BI z/Linux Oracle BI AIX blade
Browser or client / http
Browser or client / http
Browser or client / http
System Operations LOB operations Intranet
Bank Operations Intranet
Management/Analysts Intranet

Figure 7-1 Architecture overview diagram

The key IBM and ISV software components that are used in the showcase are:

- Core banking system:
  - FIS Corebank V4.2 from Fidelity National Information Services is a real-time retail banking application that is based on a physical implementation of the IBM Information Framework Financial Services Data Model (FSDM). The showcase implementation of FIS Corebank is based on CICS, Cobol, and DB2 running on System z.

    It is within the FIS Corebank application database that we store the twelve million accounts and the six million customers. It was also the focus of our core banking transformation work (described in 7.3.1, "Core banking transformation" on page 120).

    The showcase uses a number of financial products that are available with FIS Corebank, including current accounts, savings accounts, term deposits, car loans, and mortgages, which represent a good selection of the core banking functionality that is normally found in a retail banking institution.

    Recently the CICS environment for FIS Corebank was upgraded to CICS TS V4.1 (see 7.4.3, "CICS configuration" on page 129 for details about the CICS environment).
- Retail payments:
  - ACI Worldwide Ltd BASE24-eps V8.2 is used as a payment hub for the retail payments. BASE24-eps can handle the device protocols for most commercially available ATM and point-of-sale devices and for the standard interchanges, such as Mastercard and VISA. BASE24-eps handles the routing and authorization of these card transactions. In the smarter banking showcase implementation, BASE24-eps authorizes in real-time against FIS Corebank. This authorization occurs using an External CICS Interface (EXCI) call from the C++ processes that run in UNIX System Service on z/OS to CICS.
  - BASE24-eps runs natively on z/OS UNIX System Services, is written in C++, and uses shared WebSphere MQ and DB2 structures in the System z coupling facility. A java client runs on WebSphere Application Server.
- Integrated Risk Management:
  - FICO TRIAD V8.0 is a risk calculation engine that runs in a Cobol batch environment on z/OS. FICO generates the credit risk and probability of default scores for the simulated banking customers. This information is then stored in the Banking Data Warehouse (BDW) that is used as an analytical database.
  - ACI Worldwide Ltd Proactive Risk Manager (PRM) rel7.1 SP6 v8.2 provides fraud management capabilities for payment transactions. It works

natively with BASE24-eps; however, it can also take feeds from any payment transaction source. PRM can calculate risk using a choice of two basic techniques: either with neural network algorithms to detect suspicious patterns (which is not used in the showcase) or with rules that the customer defines. The fraud detection processes can be invoked in real-time to stop the transaction or in near real-time so that analysts can block the card at a later time.

PRM is written mostly in C++ and runs on UNIX System Services on z/OS using WebSphere MQ and DB2. A java client runs on WebSphere Application Server.

- Business Intelligence:
  - The showcase Banking Data Warehouse (BDW) is implemented using DB2 for z/OS. Initially this database was built with the FICO interface to create a reporting solution around BASEL II[1]. Leveraging this customer information, we now use the same database for other purposes, such as operational statistics, operational transaction summaries, real-time general ledger positions, customer segmentation, and customer insight. We created near real-time feeds to the BDW from our operational channels.
  - Oracle Business Intelligence running on WebSphere Application Server for AIX® on a POWER® blade was our initial business intelligence tool, providing use with the online dashboard reporting capability for BASEL II and customer segmentation. This tool accesses the BDW on z/OS remotely using DRDA®.
  - Cognos® 8 Business Intelligence is a more recent addition to the showcase environment. Cognos 8 BI server runs on WebSphere Application Server on a z/Linux guest under z/VM® and accesses both the BDW and the PRM operational data stores to provide an Executive Dashboard on fraud and operational reporting for the payments infrastructure.
- Internet Channel and Integration Hub:
  - WebSphere Application Server and WebSphere Process Server run on z/OS and provide our central hub, which exposes the core functions of FIS Corebank as services and provides a standard way to create business processes. WebSphere Process Server natively provides us with an Enterprise Service Bus (ESB). We use the Service Component Architecture (SCA) to create mediation routines and JEE applications, which are used by the Internet Banking channel and across other channels.

[1] Basel II is the second of the Basel Accords, which are recommendations on banking laws and regulations issued by the Basel Committee on Banking Supervision.

Note: We recently deployed WebSphere Business Events as part of the showcase Integration Hub so that business events can be collected in a central location for improved business process visibility.

- Branch Servers:
  - WebSphere Application Server is also used on Linux for System z to provide a range of teller applications to the simulated branch employees. Branch applications reuse the core banking functions by making service requests to the Integration Hub.
  - A WebSphere cluster that runs within a Linux for System z and z/VM environment allows us to consolidate the branch servers.
- Systems management:
  - Tivoli® provides the systems management capability. The IBM Tivoli Monitoring (ITM) solution provides operational monitoring, which alerts us on technical events that might impact our service level agreements as a provider of banking services to the business.
  - We make a distinction between the infrastructure events that ITM manages and business events, which is one of the focus areas of this book. We recognize, however, that there is a grey area where an infrastructure event, such as a physical ATM being unavailable, can also be considered as a business event.
  - The Tivoli Enterprise Portal, OMEGAMON®, and Tivoli Enterprise Monitoring Agents monitor the banking infrastructure. See 7.4.5, “Systems monitoring” on page 134 for more information. These tools are typically used by Operational Analysts and Systems Programmers.
- 3D Agencies:
  - OpenSim 0.6.6 is implemented on a z/VM Linux guest running SUSE Linux SLES 10 to allow remote access to a virtual ATM. OpenSimulator, often referred to as OpenSim, is an open source server platform for hosting virtual worlds. See 8.4.2, “Customer enters branch” on page 163 for information about how we use OpenSim in our business events scenario.

7.3 The Smarter Banking showcase and the IBM Banking Framework

Within the smarter banking showcase we talk of *proof-points*, which are discreet scenarios that we use to illustrate ways to address the specific business problems that banks face. We structure the proof-points in relation to the four domains that constitute the IBM Banking Framework (2.3, “The IBM Banking Framework” on page 17).

In this section, we summarize how the showcase already provides proof-points across the framework domains. In the ensuing chapters, we describe how we use the new capabilities of CICS TS V4.1 to introduce new proof-points that address the need to improve end-to-end process visibility and horizontal integration.

7.3.1 Core banking transformation

The initial proof-points that the showcase team developed addressed many of the basic building blocks that are now found in the core banking transformation domain of the IBM Banking Framework:

- How to improve core banking process efficiency and reduce costs
- How to build flexible business processes
- How to efficiently reuse core banking functions

IT Foundation Transformation

The IT foundational transformation domain addresses capabilities, such as IT infrastructure simplification, platform scalability, and model-driven development. The showcase core banking platform runs on a simplified IT infrastructure that uses the System z operating system and middleware capabilities to reduce operational cost and risk. The core banking system runs in a CICSplex for high availability and uses DB2 data sharing so that there is just one single copy of customer and product data. Rational development tools are used to build business service components.

Core banking process agility

Using WebSphere Process Server we can quickly create business processes that are aligned to new business requirements. Business processes can access the existing core banking components as services. We created a number of business processes that allow our multi-channel architecture to reuse the same core functionality and to provide business intelligence by updating the BDW in near real-time.

Core banking application modernization

FIS Corebank is supplied with a comprehensive set of application programming interfaces that allow multiple banking channels to access the core banking system. We used these APIs to create a multi-channel architecture.

We evolved towards a service-oriented architecture, reusing the APIs that FIS Corebank provided as our service building blocks. We built composite applications using the Service Component Architecture (SCA). These composite applications provide additional functionality but at the same time they reuse core system functions that FIS Corebank provides. We use an ESB to provide intelligent routing and transformation. Using this flexible approach we can call other application functions that other ISVs provide to enrich our business processes without touching the FIS Corebank application.

7.3.2 Payments and securities

The showcase is focused on retail banking rather than wholesale banking; therefore, it does not deal with corporate services, derivative products, and the trading operations that are covered by the financial markets area. We do, however, deal with certain payment types, in particular card-based payments. Debit and Credit card payments, either with the card present (ATM or PoS) or not present (Internet or contactless), is a highly strategic area in retail banking that is driven by opportunity and new regulations.

Retail payments

We implemented ACI Worldwide's BASE24-eps application to handle card payments and the Proactive Risk Manager to monitor and alert on potential payment fraud. BASE24-eps authorizes each payment with FIS Corebank as the card issuer.

BASE24-eps is a secure, modern enterprise payment platform that can help a Financial Institution transform its payments strategy. In the showcase, we enabled an ATM channel and the VISA interchange channel, and we inject payments over both during a demonstration.

7.3.3 Integrated risk management

To address multiple aspects of risk, we implemented a number of proof-points in the showcase.

Financial risk

We installed the FICO TRIAD product to give us credit risk and probability of default scores on our loan portfolio.

Financial crimes

With the integration of ACI's Proactive Risk Management (PRM) with BASE24-eps, we can assess the risk of fraud to our enterprise from our card payment channels. PRM can receive feeds from any channel, and in the future, our other channels will also be assessed here. By using PRM, we can analyze transactions in real-time or in near real-time.

Fraud Analysts work the PRM queues to alert about potential fraud and have the ability to stop a card, if necessary. In real-time, PRM can block a transaction if it breaks specific rules.

We exposed the metrics and statistics that are stored within the PRM DB2 database to a Cognos Executive dashboard to give the banks executives a near real-time view of the current fraud exposure with the Key Performance Indicators (KPIs) that are normally reported to the LoB executives.

Operational and IT risk

Using a variety of System z capabilities, including Parallel Sysplex, GDPS®[2] Hyperswap, System Automation, and IBM Ti™voli Monitoring we can monitor the service level agreements of our workload and react to planned and unplanned outages to maintain continuous availability.

Governance and compliance

The loan portfolios are extracted from the FIS Corebank operational database and stored in the BDW to address the BASEL II regulatory requirements. Oracle Business Intelligence provides a dashboard analysis and reporting on BASEL II compliance.

Note: The BDW is part of the IBM Banking Industry Model, Information Framework (IFW).

The TRIAD product pulls data for our loan customers from the BDW and stores the resulting scores back into this analytical database. We then use analytical tools to access the BDW to create the reports. The reports fundamentally address credit risk. However, through our analytical tools we can also identify trends within the way the bank manages this risk, which then starts to address operational risk.

7.3.4 Customer care and insight

We previously discussed how we use the BDW for risk assessment. When creating an analytical warehouse it is best to tackle the problem project by project, that is, to have a clearly defined deliverable for each project and slowly

[2] Geographically Dispersed Parallel Sysplex (GDPS) is the ultimate Disaster Recovery and Continuous Availability solution for a System z multi-site enterprise.

build the warehouse, which is the approach that we took with the showcase BDW.

The BASEL II Credit Risk reporting was our first BDW project during which we loaded customers (Involved Parities in the IFW model) and loan accounts (arrangements in the IFW model) to the database along with a number of more static reference tables, such as periods, rates, classifications, and locations. Having completed this project, we then reused this data for other projects and slowly built more content, views, accumulations, and summaries into the database.

Creating the BDW helped us to build a single view of the simulated banking customer and to address many of the customer care and insight challenges.

Marketing process optimization

Using Oracle Business Intelligence and its marketing functions, we performed customer segmentation based on the customer data that we stored in the BDW, which allowed us to define a pool of customers with similar characteristics that we used, for example, to launch a new marketing campaign for a new product. In our case, we leveraged the credit risk scoring and probability of default from the risk analysis to help the segmentation process.

Customer information optimization

The BDW does not just store static data. We have a number of near real-time feeds that provide Extract Transform Load (ETL) processes to load operational data into the warehouse.

We use WebSphere MQ as a technique to separate the synchronous unit of work for the real banking channel response to the customer or branch, from the asynchronous, near real-time update of the warehouse, which provides us with very current up-to-date customer data.

We use Cognos 8 Business Intelligence (BI) to provide an executive dashboard that shows fraud KPIs. We use the same analytics server to also provide an operational dashboard for payments based on the near real-time feeds to the BDW.

Multi-channel transformation

We enabled multi-channel transformation by reusing the core banking services across all of the showcase channels. New channels can be added with a rich functionality after the new channel access protocols are established.

7.4 Showcase infrastructure

The smarter banking showcase highlights some of the System z unique qualities that enable a dynamic infrastructure, including service management, virtualization, business and information resiliency, and energy efficiency (see 4.1.1, “Why the mainframe” on page 50).

The showcase infrastructure is part of the Green Data Center at the IBM Montpellier site. The Green Data Center is a facility that we use to demonstrate innovative green technologies and energy management strategies, such as consolidation and virtualization and energy management tools and techniques. The Green Data Center at Montpellier monitors and manages environmental conditions, such as air pressure, temperature, and humidity in the separate machine rooms, and allows the showcase team to supervise energy consumption.

The showcase infrastructure is based primarily on System z10 as the operating platform. System z10 incorporates a number of features to support greener infrastructures, including a smaller footprint that reduces power and cooling requirements, which allows organizations to both conserve power and save on costs.

Figure 7-2 on page 125 shows a simplified view of the operational model and illustrates the main operating environments that we use in the showcase.

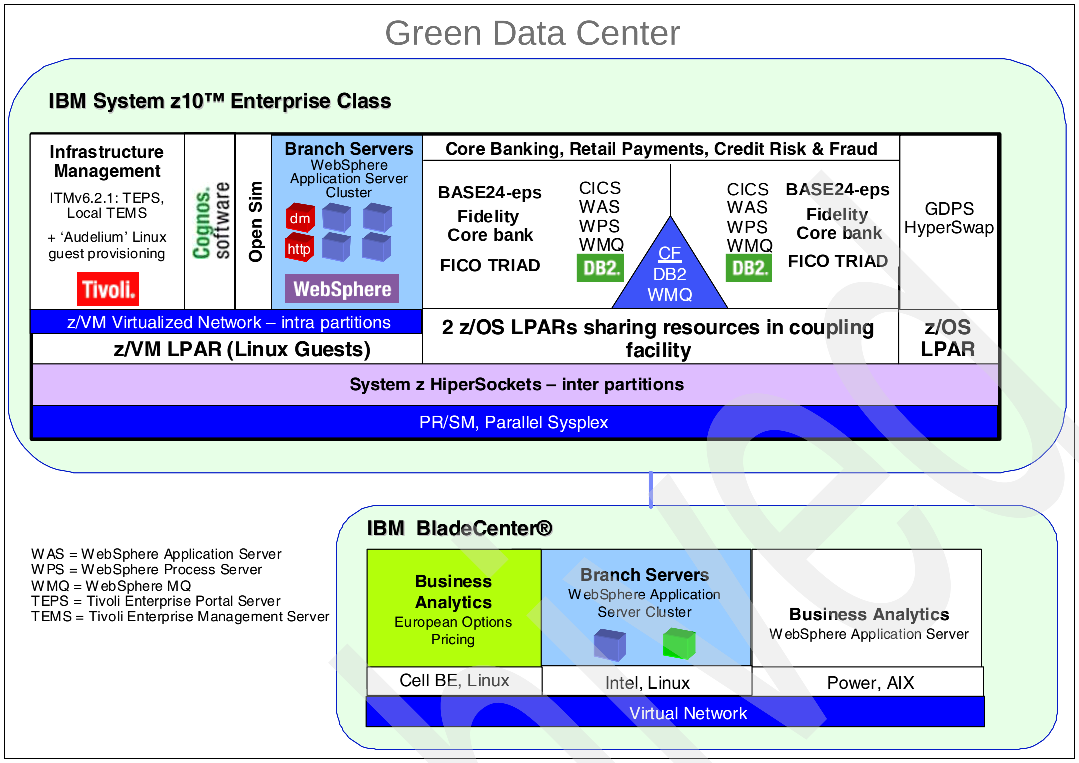

Green Data Center
IBM System z10™ Enterprise Class
Infrastructure Management
ITMv6.2.1: TEPS, Local TEMS
+ 'Audelium' Linux guest provisioning
Tivoli.
Cognos. software
Open Sim
Branch Servers
WebSphere Application Server Cluster
dm
http
WebSphere
Core Banking, Retail Payments, Credit Risk & Fraud
BASE24-eps
Fidelity Core bank
FICO TRIAD
CICS
WAS
WPS
WMQ
DB2.
CF
DB2
WMQ
CICS
WAS
WPS
WMQ
DB2.
BASE24-eps
Fidelity Core bank
FICO TRIAD
GDPS HyperSwap
z/VM Virtualized Network – intra partitions
z/VM LPAR (Linux Guests)
2 z/OS LPARs sharing resources in coupling facility
z/OS LPAR
System z HiperSockets – inter partitions
PR/SM, Parallel Sysplex
WAS = WebSphere Application Server
WPS = WebSphere Process Server
WMQ = WebSphere MQ
TEPS = Tivoli Enterprise Portal Server
TEMS = Tivoli Enterprise Management Server
IBM BladeCenter®
Business Analytics
European Options Pricing
Branch Servers
WebSphere Application Server Cluster
Business Analytics
WebSphere Application Server
Cell BE, Linux
Intel, Linux
Power, AIX
Virtual Network

Figure 7-2 Operational model

The System z10 machine is shared with other projects, and there are a number of LPARs and Sysplexes that are defined within this single machine. The showcase production environment uses three z/OS LPARs. One LPAR runs GDPS HyperSwap® software and the other two are the key application-owning LPARs where the banking workloads run. The systems that run within the parallel sysplex access the same shared operational data in DB2 using the coupling facility (CF) to share resources.

MVS Workload Manager (WLM) manages the workloads that are injected into the demonstration environment, and each request is classified and assigned performance goals based on channel and customer status. The performance goals are then monitored using the IBM Tivoli Monitoring infrastructure.

Increasingly, we use Linux on System z environments hosted under z/VM. Initially we established a WebSphere Network Deployment cluster of four application server nodes to represent our branch servers on Linux on System z. We added many of the major components of IBM Tivoli Monitoring to Linux, the Cognos 8 BI Server, and most recently the Open Sim 3D virtual world software.

Two IBM BladeCenters are used for running the Rational Performance Tester Controller, Tivoli infrastructure management, Cognos Framework Manager, Oracle BI Server, and other Graphical User Interfaces.

7.4.1 Hardware configuration

The showcase hardware configuration is based on an IBM System z10 Enterprise Class (EC) machine and two IBM BladeCenter® machines.

Figure 7-3 shows the hardware configuration, including our two storage devices. We use peer-to-peer remote copy (PPRC) between the two storage devices that mirrors primary disk updates to the secondary disk, so that an exact replica always exists in the event of an unplanned or planned outage, which allows us to swap to our secondary disk, if necessary.

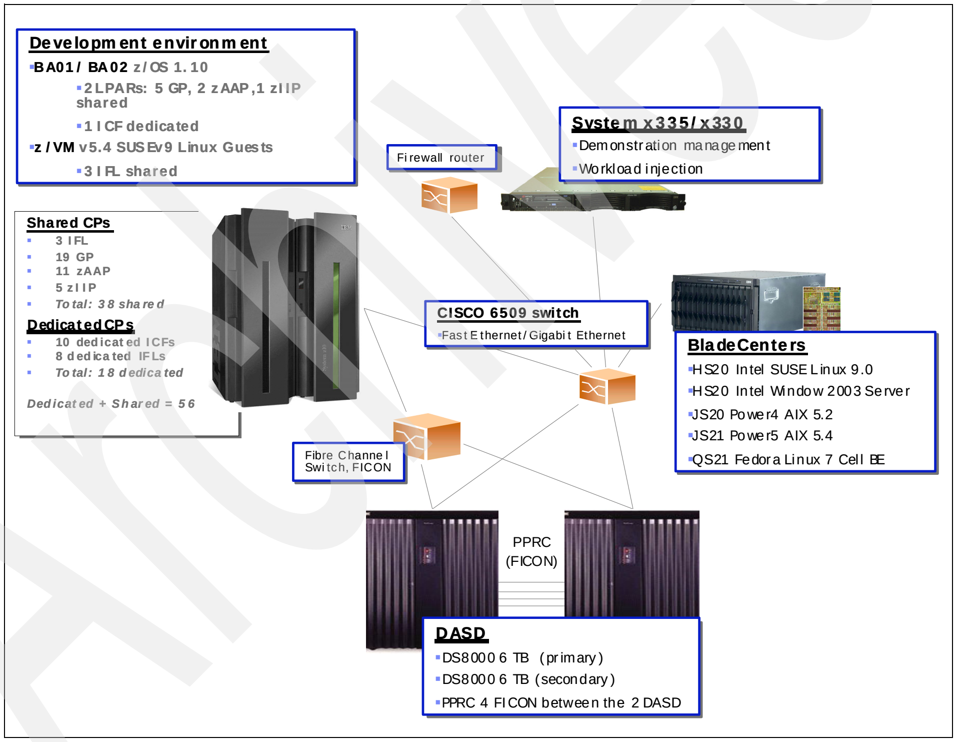


Figure 7-3 Hardware configuration

The hardware configuration components are:

- System z10 Model E56

  The processors of the z10 machine are shared across various LPARs, including the two showcase environments:

  – Showcase production sysplex (ZBPLEX), which consists of three z/OS 1.10 LPARs ZB01, ZB02, and ZB03.

  – Showcase development sysplex (BAPLEX), which consists of two z/OS 1.10 LPARs BA01 and BA02.

  > **Note:** The business events and Web 2.0 scenarios that are enabled as part of this book were configured on the showcase test environment.

- One z/VM 5.4 LPAR for development and one for production, each with the following Linux guests:

  – Six SUSE V9 Linux guests used for the WebSphere ND cluster (one HTTP server, one deployment manager, and four application servers) for the branch servers.

  – One SUSE V9 Linux guest for the Cognos 8 BI server.

  – One SUSE V9 Linux guest for the OpenSim 3D environment.

  – Two SUSE V9 Linux guests for the TEPS and TEMS (IBM Tivoli Monitoring).

- One coupling facility LPAR for development and two for production.
- Two direct access storage devices (DASD). A DS8300 and an ESS800 with 6.6 terabytes of data.
- Four FICON® channels between the DASD devices.
- Four IBM System X330 machines for the Rational Performance Tester workload injection.
- Two IBM BladeCenters with blades for Rational Performance Tester Controller, Tivoli infrastructure management, Cognos Framework Manager, Oracle BI Server, and other Graphical User Interfaces.
- One CISCO 6509 switch.

The operational environment is designed to be resilient and highly available using many of the technologies that are natively available with IBM System z, including:

- Geographically Dispersed Parallel Sysplex (GDPS) HyperSwap for disk resiliency

- Parallel Sysplex for server resiliency and data sharing
- Sysplex Distributor for TCP/IP connection workload distribution and high availability
- Virtualization to allow dynamic resource allocation and system resource sharing
- HiperSockets™ and virtual local area network (VLAN) for virtualization of the network within the IBM System z10 machine
- Workload Manager (WLM) for transaction-based workload management on z/OS and machine priorities on z/VM

7.4.2 Operating system and middleware configuration

In this section, we show the current versions of System z software that are used in the smarter banking showcase:

- Communication Server V1.10 as the network server with Systems Network Architecture (SNA) and TCP/IP configured for both LPARs.
- We use a private gigabit network for workload injection and for the Tivoli monitoring infrastructure. All of the other remote hubs and agents in the distributed platform use the dynamic virtual IP address (DVIPA) address:
  - 10.1.1.20 for BA01
  - 10.1.1.21 for BA02
  - 10.1.1.29 for the DVIPA address
- CICS Transaction Server V4.1 (CICS TS) is configured with:
  - Two terminal-owning regions (TOR), one per z/OS LPAR
  - Six application-owning regions (AOR), three per z/OS LPAR

  The CICSPlex SM component of CICS TS V4.1 is used for CICSPlex Single System Image management, providing a single point-of-control and dynamic workload transaction balancing over the core banking CICSplex.

  See 7.4.3, "CICS configuration" on page 129 for more information about the showcase CICS configuration.
- CICS Explorer V1.0.0.2.
- WebSphere Application Server V6.1.0.21 with WebSphere Business Events V6.2.1, and WebSphere Process Server 6.1.2.
- We use IBM DB2 V9 for the operational core banking application data store and BDW, and it is configured with two DB2 instances in data sharing mode, one per z/OS LPAR.

- We use WebSphere MQ for z/OS V6 as the classical access layer for core banking and for publishing. It is configured with:
  - Two queue managers, one per z/OS LPAR
  - Two channel initiators, one per z/OS LPAR
- CICS Transaction Gateway (CICS TG) V6.0.1 is used for JEE connectivity. It is configured with two CICS TG daemons, one per z/OS LPAR.

7.4.3 CICS configuration

When deploying CICS in a parallel sysplex, you can take advantage of the z/OS specific workload management capabilities, including Sysplex Distributor, MVS Workload Manager (WLM), and shared access to data, including DB2 and VSAM data.

Figure 7-4 shows the CICS core banking infrastructure for the showcase development environment.

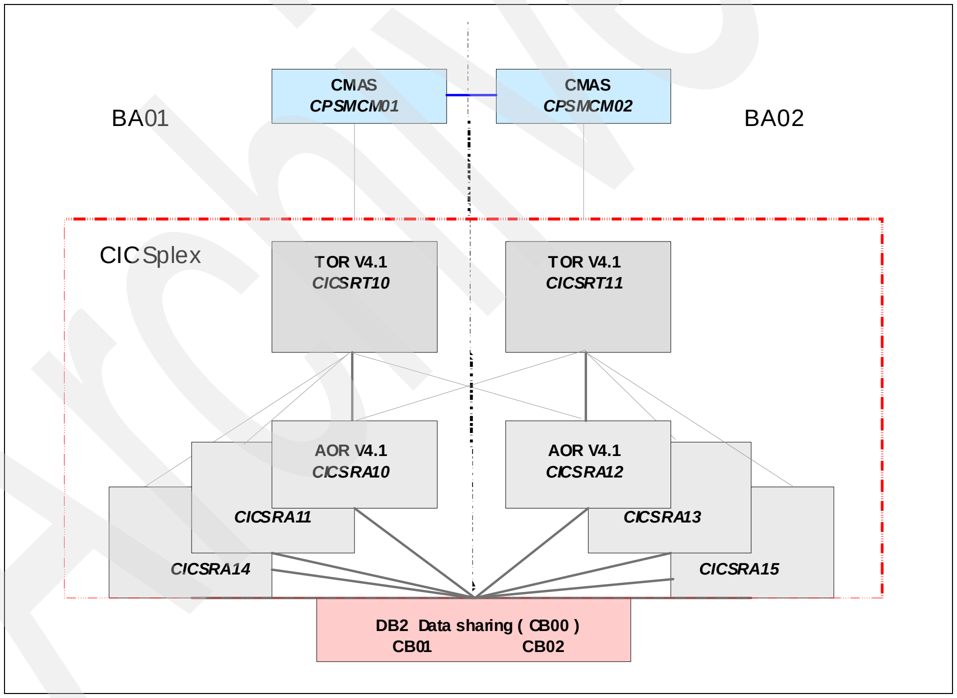


Figure 7-4 CICS core banking configuration

The CICS core banking configuration consists of:

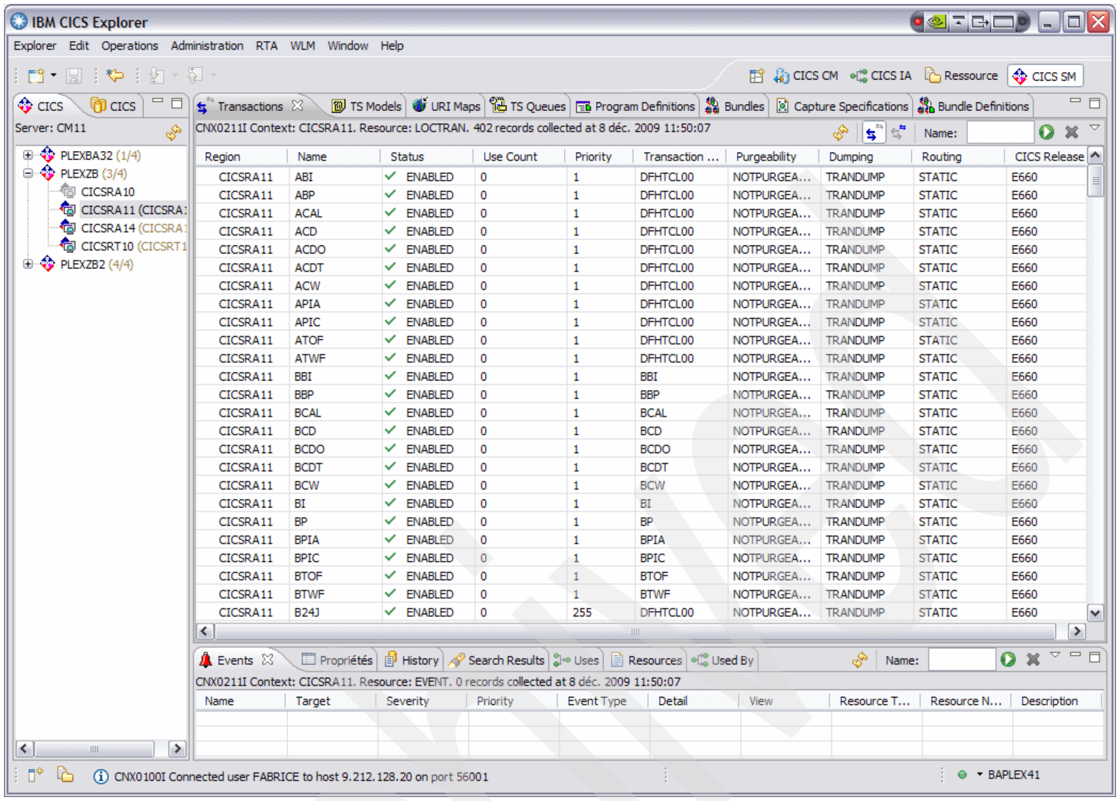
- Sysplex Distributor for workload management of TCP/IP connections across two LPARs (BA01 and BA02)
- A TOR on each LPAR listens on a shared TCP/IP port:
  - CICSRT10 on BA01
  - CICSRT11 on BA02
- Transaction requests that are dynamically routed to cloned AORs using CICSPlex SM:
  - CICSRA10, CICSRA11, and CICSRA14 on BA01
  - CICSRA12, CICSRA13, and CICSRA15 on BA02

  Note: CICS TS V4.1 can now use the coupling facility to store more detailed, accurate workload management data, which enables improved performance and overall transaction throughput.

- CICS core banking programs that run in the AORs share access to the customer and accounts operational database using DB2 data sharing.
- CICSPlex SM provides a single point-of-control and dynamic workload transaction balancing. CICSPlex SM is configured with two CICSPlex SM address spaces (CMASs):
  - CPSMCM01 on BA01
  - CPSMCM02 on BA02

CICS Explorer

CICS Explorer is used for CICSplex systems management. Figure 7-5 on page 131 shows a CICS Explorer view of the production CICSplex.



CNX0211I Context: CICSRA11. Resource: LOCTRAN. 402 records collected at 8 déc. 2009 11:50:07

Region	Name	Status	Use Count	Priority	Transaction ...	Purgeability	Dumping	Routing	CICS Release
CICSRA11	ABI	✓ ENABLED	0	1	DFHTCL00	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	ABP	✓ ENABLED	0	1	DFHTCL00	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	ACAL	✓ ENABLED	0	1	DFHTCL00	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	ACD	✓ ENABLED	0	1	DFHTCL00	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	ACDO	✓ ENABLED	0	1	DFHTCL00	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	ACDT	✓ ENABLED	0	1	DFHTCL00	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	ACW	✓ ENABLED	0	1	DFHTCL00	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	APIA	✓ ENABLED	0	1	DFHTCL00	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	APIC	✓ ENABLED	0	1	DFHTCL00	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	ATOF	✓ ENABLED	0	1	DFHTCL00	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	ATWF	✓ ENABLED	0	1	DFHTCL00	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	BBI	✓ ENABLED	0	1	BBI	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	BBP	✓ ENABLED	0	1	BBP	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	BCAL	✓ ENABLED	0	1	BCAL	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	BCD	✓ ENABLED	0	1	BCD	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	BCDO	✓ ENABLED	0	1	BCDO	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	BCDT	✓ ENABLED	0	1	BCDT	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	BCW	✓ ENABLED	0	1	BCW	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	BI	✓ ENABLED	0	1	BI	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	BP	✓ ENABLED	0	1	BP	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	BPIA	✓ ENABLED	0	1	BPIA	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	BPIC	✓ ENABLED	0	1	BPIC	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	BTOF	✓ ENABLED	0	1	BTOF	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	BTWF	✓ ENABLED	0	1	BTWF	NOTPURGEA...	TRANDUMP	STATIC	E660
CICSRA11	B24J	✓ ENABLED	0	255	DFHTCL00	NOTPURGEA...	TRANDUMP	STATIC	E660

CNX0211I Context: CICSRA11. Resource: EVENT. 0 records collected at 8 déc. 2009 11:50:07

Name	Target	Severity	Priority	Event Type	Detail	View	Resource T...	Resource N...	Description

CNX0100I Connected user FABRICE to host 9.212.128.20 on port 56001

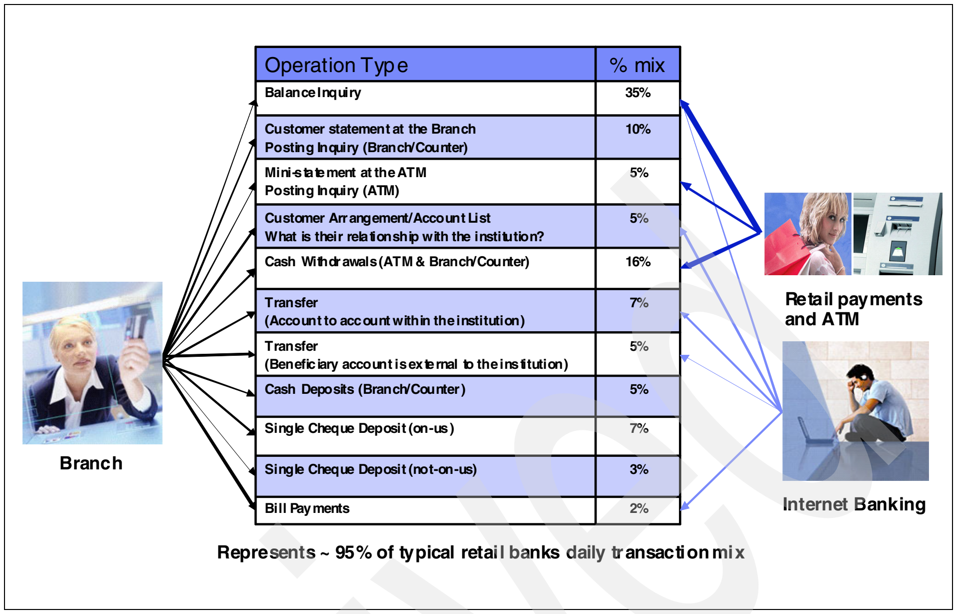
Figure 7-5 CICS configuration: CICS Explorer view

> **Note:** We also use the CICS Explorer to enable business event processing in CICS (see 8.2.1, “Application analyst role” on page 143).

7.4.4 Workload simulation

We use Rational Performance Tester to inject online workload into the core banking system. The online workload is derived from research into actual banking workloads and is designed to be a reasonable representation of a typical days online activity with a transaction mix that covers balance inquiries, statement requests, cash transactions, transfers, and cheque deposits.

Figure 7-6 on page 132 shows the exact transaction mix that we use in the demonstration and the various channels that are simulated: branch, Internet, retail payments, and ATM.


Branch
Retail payments and ATM
Internet Banking
Represents ~ 95% of typical retail banks daily transaction mix

Operation Type	% mix
Balance Inquiry	35%
Customer statement at the Branch Posting Inquiry (Branch/Counter)	10%
Mini-statement at the ATM Posting Inquiry (ATM)	5%
Customer Arrangement/Account List What is their relationship with the institution?	5%
Cash Withdrawals (ATM & Branch/Counter)	16%
Transfer (Account to account within the institution)	7%
Transfer (Beneficiary account is external to the institution)	5%
Cash Deposits (Branch/Counter)	5%
Single Cheque Deposit (on-us)	7%
Single Cheque Deposit (not-on-us)	3%
Bill Payments	2%

Figure 7-6 Online workload injected by Rational Performance Tester (RPT)

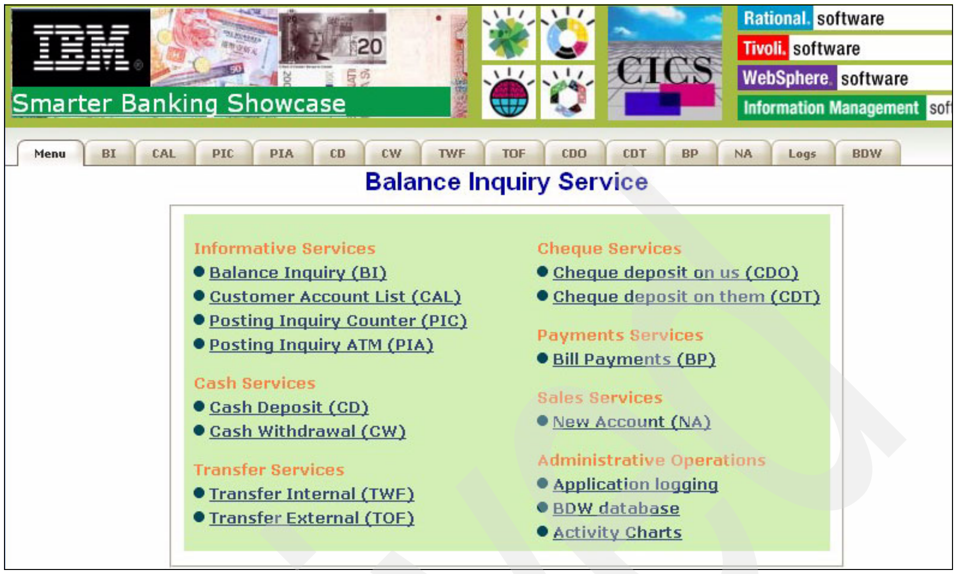
All online transactions, whatever the channel, are processed by the core banking system (FIS Corebank), which runs in CICS. Each transaction has a unique transaction type, as shown in Table 7-1 on page 133.

Table 7-1 Banking transaction types

Transaction Type	Description
BI	Balance inquiry.
PIC	Posting inquiry. Customer Statement at the branch. Shows the last 20 transactions in detail.
PIA	Posting inquiry. Mini statement at the ATM. Shows the last eight transactions in summary.
CAL	Customer arrangement list. Customer relationship with the bank.
CW	Cash withdrawal. ATM and counter.
TWF	Account-to-account transfer, where both accounts are within our financial institution.
TOF	Account-to-account transfer, where the beneficiary account is in another financial institution.
CD	Cash deposit at the branch.
CDO	Check deposit on us (drawn on our financial institution).
CDT	Check deposit on them (drawn on another financial institution).
BP	Bill payments.

Note: We use the transaction type when enabling business events in the CICS core banking system. See “Capture specifications” on page 147.

We also inject business transactions manually, either using a Web application or using a real ATM or a virtual ATM in OpenSim. Figure 7-7 on page 134 shows the range of financial transactions that are simulated by our Web application.



Balance Inquiry Service

Informative Services
- Balance Inquiry (BI)
- Customer Account List (CAL)
- Posting Inquiry Counter (PIC)
- Posting Inquiry ATM (PIA)

Cash Services
- Cash Deposit (CD)
- Cash Withdrawal (CW)

Transfer Services
- Transfer Internal (TWF)
- Transfer External (TOF)

Cheque Services
- Cheque deposit on us (CDO)
- Cheque deposit on them (CDT)

Payments Services
- Bill Payments (BP)

Sales Services
- New Account (NA)

Administrative Operations
- Application logging
- BDW database
- Activity Charts

Figure 7-7 Transaction simulation

During a typical demonstration we inject a workload of approximately 400 financial transactions per second.

7.4.5 Systems monitoring

The showcase uses many Tivoli System Management products and monitoring tools to track, monitor, and react to multiple infrastructure events that can occur in the course of a normal working day:

- IBM Tivoli Monitoring Services V6.2 is the foundation product and includes three components:
  - IBM Tivoli Enterprise Portal (TEP)
  - IBM Tivoli Enterprise Portal Server (TEPS)
  - IBM Tivoli Enterprise Monitoring Server (TEMS)

  Figure 7-8 on page 135 shows how we deployed these components, specifying one TEMS as the Hub to which all of the other TEMS send monitored data. The Tivoli Enterprise Monitoring Agents (TEMAs) and OMEGAMON agents (for z/OS) send metrics from the individual subsystems and operating systems to the TEMS. The portal server (TEPS) then extracts the data from the TEMS Hub.

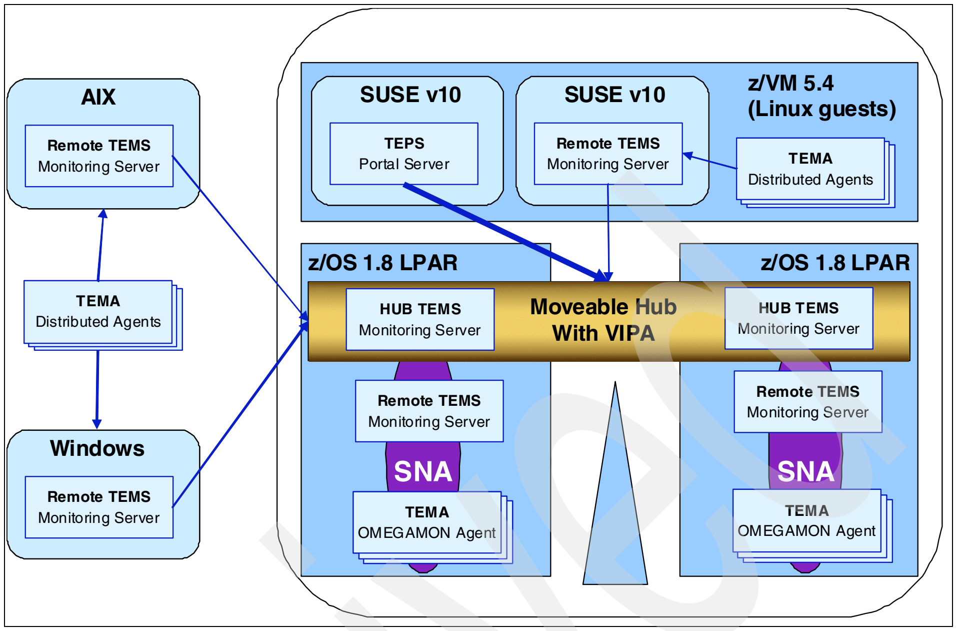


Figure 7-8 Systems monitoring architecture

- Figure 7-8 shows the moveable TEMS Hub that provides us with a more highly-available solution. If the TEMS Hub (a started task on z/OS) has a problem, Automatic Restart Manager (ARM) restarts another instance of the TEMS Hub on another z/OS LPAR.
- We implemented the following list of OMEGAMON agents and TEMAs as part of the showcase monitoring solution:
  - IBM Tivoli OMEGAMON XE for CICS on z/OS V4.1.0
  - IBM Tivoli OMEGAMON XE on z/OS V4.1.0
  - IBM Tivoli OMEGAMON XE for DB2 V4.1.0
  - IBM Tivoli OMEGAMON XE for Messaging V6.1
  - IBM Tivoli Composite Application Manager for WebSphere Agent v6.1
  - IBM Tivoli Composite Application Manager for WebSphere v6.1
  - IBM Tivoli Enterprise Monitoring Agent for Windows v6.2.1
  - IBM Tivoli Enterprise Monitoring Agent for AIX v6.2.1
  - IBM Tivoli Enterprise Monitoring Agent for Linux v6.2.1

The monitoring infrastructure flags technical infrastructure events, such as a CICS region failure or a network component failure and sends alert events to the Tivoli Enterprise Portal. Figure 7-9 on page 136 shows how the TEP is used to provide a high-level view of the current health of the banking infrastructure.

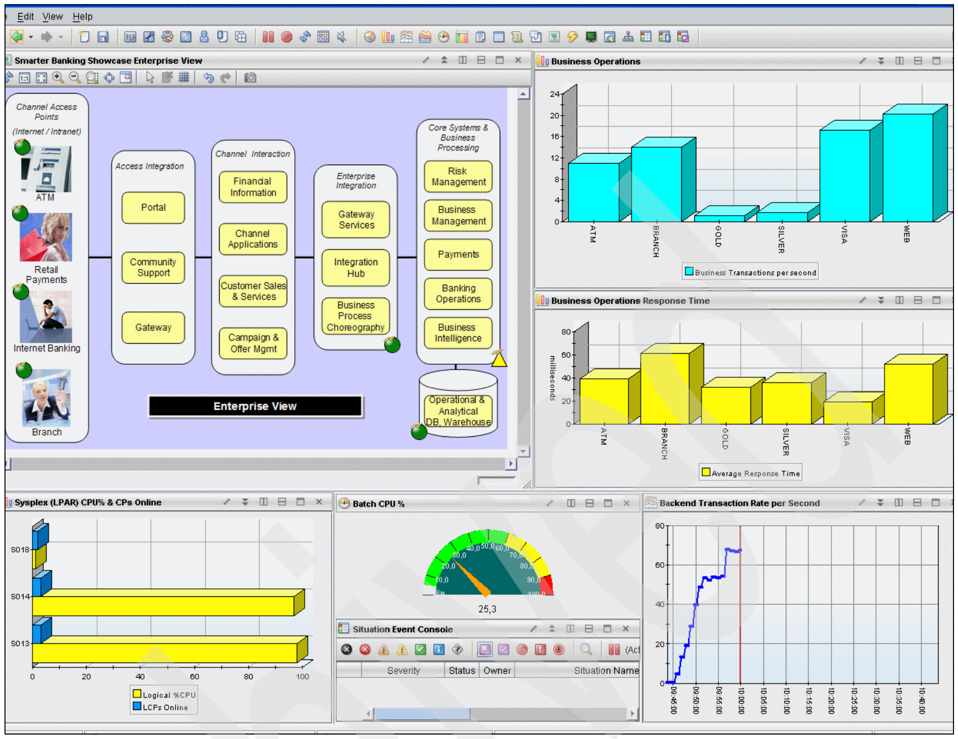


Figure 7-9 Infrastructure monitoring

Figure 7-9 shows monitoring information about:

- Channel status
- Transactions per second by channel
- Transaction response time by channel
- Transaction rates by time of day
- Batch CPU usage
- Infrastructure events, for example server failure
- Overall sysplex CPU usage

Important: We make a distinction between infrastructure events, which Tivoli monitoring captures and business events, which WebSphere Business events capture and can be monitored using WebSphere Business Monitor.

7.5 Summary

In this chapter, we showed how the smarter banking showcase provides proof-points across the four domains of the IBM Banking Framework.

In the subsequent chapters of this part of the book, we focus on new showcase proof-points that illustrate improved business process visibility and Web 2.0 style integration:

- In Chapter 8, “Enabling business events” on page 139, we show how we enabled event processing in the CICS core banking component of smarter banking showcase.
- In Chapter 9, “Enabling Web 2.0” on page 167, we show how we enabled account feeds from the CICS core banking component using the CICS Atom feed support.

8

Enabling business events

In this chapter, we show how we enabled the smarter banking showcase for business events.

First, we explain our hypothetical business scenario, which relates to events that are associated with *high-interest* customers. In 3.5.3, “Customers of interest” on page 43, we introduce the business case based on the ability to detect and contact customers of interest when they are in a branch office.

A high interest customer, in our scenario, is one that performs a number of very high-value transactions within a single month. We want to capture this information to monitor the account usage (whether these are credit or debit transactions) and possibly to take action. We might offer a new account package with additional benefits to the high-value customer, or we might want to discuss account management and budgeting with a customer whose account is significantly in the red.

We then describe how we enabled the multiple systems, CICS and WebSphere Business Events, for event processing. WebSphere Business Events performs actions based on events captured by CICS (account transactions) and other events. In our scenario, an event is emitted when the customer enters the bank branch.

8.1 Business events scenario

A bank runs millions of banking transactions each day. Each transaction can be considered as an event. How do we decide which transactions are significant business events? Which are significant from a business perspective is a choice made by the Line of Business managers and business analysts.

Some examples of transactions that can be significant business events are:

- Attempts to use credit cards fraudulently.

  These are significant events that need a quick response from the bank.
- Very large transactions.

  These might indicate that a customer is a good candidate for buying a new product.
- Failed transactions.

  These can indicate that customer satisfaction is at risk.

Single events on their own might not be that significant but combined with events that are emitted from other sources, they can be more interesting, for example, a customer that performs an unusual amount of account activity, perhaps because he is starting a new business venture or building a house, might present a new selling opportunity to the bank. When such a customer arrives at the branch to deposit a large amount of cash or to make a large withdrawal, it can be worthwhile for the branch manager to discuss such opportunities with the customer.

In this chapter, we describe how business event processing can be used to initiate an action (for example, inform the branch manager) when a specific event occurs (for example, the customer enters a branch) if the customer is of significant interest (for example, he performs a number of large transactions within the last month).

Important: Business events are enabled in the smarter banking showcase in a non-invasive way, that is, without changes to the core banking system.

8.1.1 Roles

The workflow that is required to enable event processing involves the participation of people with different job roles:

- The LoB manager or business analyst

  This person defines the business application events in terms that he is familiar with together with the data that is to be passed to the event consumer with the event.

  We provide an overview of the actions of the business analyst in 8.1.2, “Business analyst role” on page 141.

- The application analyst

  The application analyst takes the natural language specification of the event from the business analyst and converts it into formal specifications. He understands:

  – Where events are generated
  – How actions are performed
  – What protocols are used to transport events
  – What data mappings are used between multiple event formats

  We provide an overview of the actions of the application analyst in 8.2.1, “Application analyst role” on page 143.

- The systems programmer

  The systems programmer sets up the infrastructure in support of the event architecture. He configures the system so that events are routed to the required destinations, and he deploys the event bindings to the appropriate systems.

  We provide an overview of the actions of the systems programmer in 8.2.2, “Systems programmer role” on page 150.

8.1.2 Business analyst role

A business analyst understands the needs of the business, but does not necessarily know about the programs and computer systems that support that business. The business analyst expresses his requirement in business terms, for example, “I want to track every core banking transaction that exceeds 5000 euros, and I want to be informed when a customer that performs five or more large transactions visits my branch.”

The business analyst wants to track large transactions for the following types of transaction:

- Cash deposit
- Cash withdrawal
- Account transfer
- Cheque deposit

Note: A cheque or check is a negotiable instrument that instructs a financial institution to pay a specific amount of currency from a specified account. Within the smarter banking showcase we use the English spelling 'cheque'.

The business analyst also defines the information that is to be emitted when an event occurs, for example, the business analyst wants to emit the following information for large transactions:

- Account number
- Amount of transaction
- Type of transaction
- Channel (for example, branch, ATM or internet)

The application analyst uses this information (8.2.1, "Application analyst role" on page 143) when using CICS Explorer to configure the CICS event binding.

The business analyst also wants to track when a customer of interest enters a bank branch. When this happens, he wants to emit an event that contains the customer identifier and account identifier. See 8.4.2, "Customer enters branch" on page 163 for information about how we simulated this event using the virtual world OpenSim.

Finally, the business analyst defines the actions that result from a specific sequence of events. We provide an overview of how the business analyst makes these definitions using the WBE Design tool in 8.3, "Working with WebSphere Business Events" on page 150.

8.2 Enabling business events in the core banking system

In this section, we summarize how we enabled business events in the CICS core banking system. For detailed information about configuring CICS for business events, see the IBM Redbooks Publication *Implementing Event Processing with CICS*, SG24-7792.

8.2.1 Application analyst role

The application analyst needs knowledge of the application to identify the specific programs and program APIs at which events can be captured.

In our example, the application analyst must know which part of the core banking application processes cash deposits, cash withdrawals, account transfers, and cheque deposits. He also must understand where to find the transaction type and the value of the transaction because these are both used to filter the event.

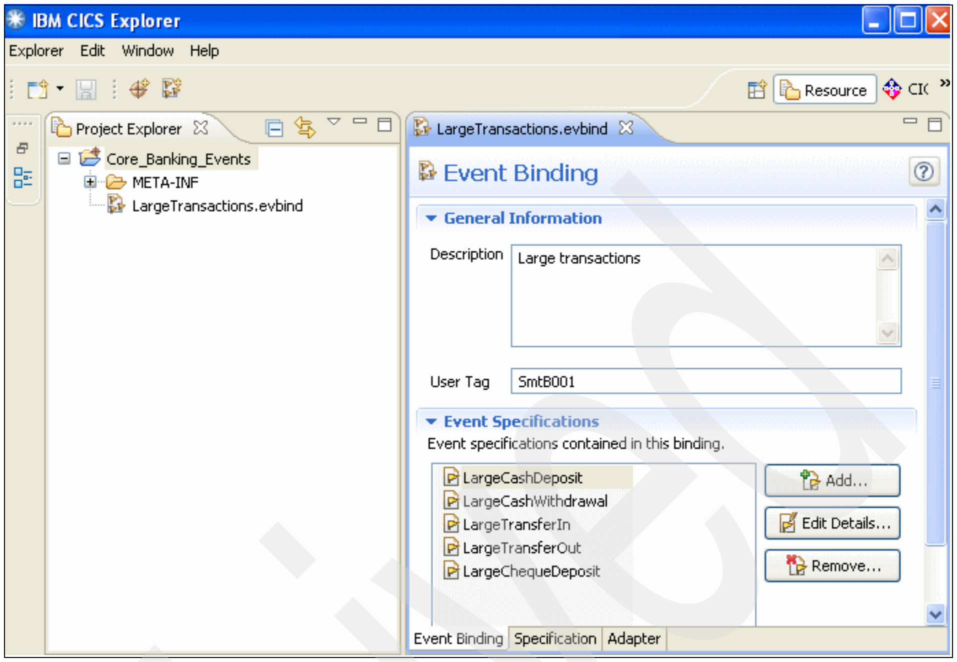
Note: The application analyst can use CICS Interdependency Analyzer for z/OS (CICS IA) to better understand the structure of the core banking application.

The application analyst also specifies how the data to be included in the event payload can be obtained from data that is available at the event capture point or from the context in which the event occurs.

The application analyst uses the CICS event binding editor, which is part of the CICS Explorer, to create an event binding for large transactions. We provide a short review of this configuration in the next sections.

Event binding

The event binding file is an XML file that defines a set of event specifications to CICS. The event binding consists of several parts, each edited in a separate tab in the editor. Figure 8-1 on page 144 shows the event binding that we created for capturing large transactions.



Description: Large transactions

User Tag: SmtB001

Event specifications contained in this binding.

- LargeCashDeposit
- LargeCashWithdrawal
- LargeTransferIn
- LargeTransferOut
- LargeChequeDeposit

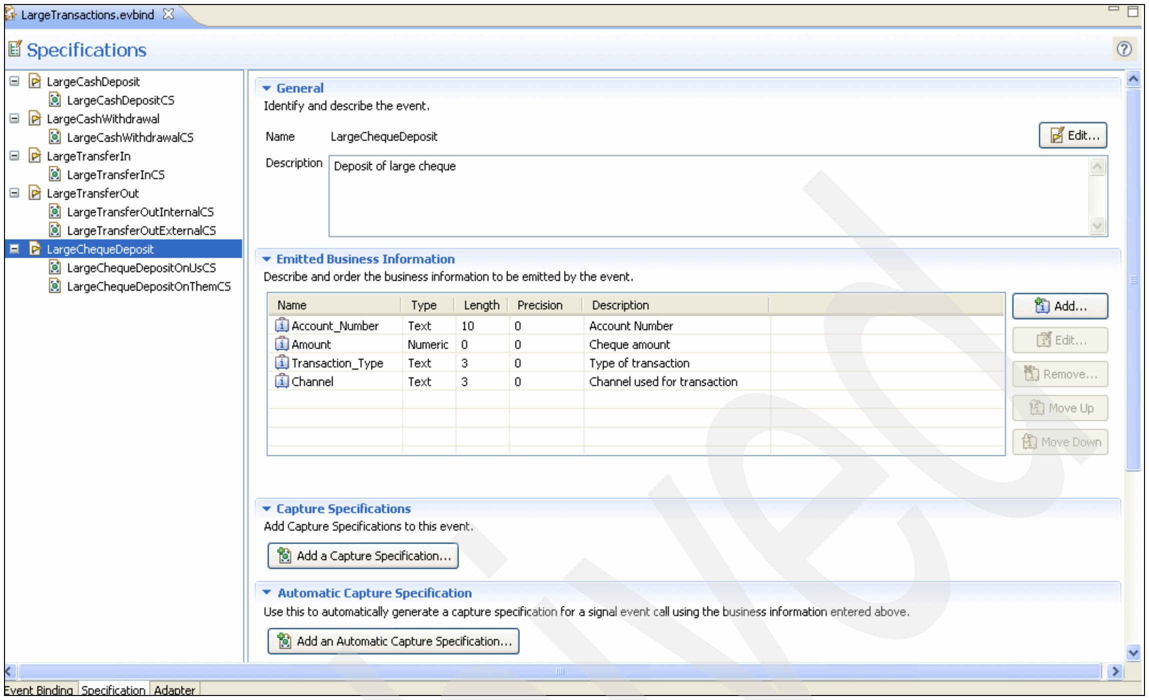
Figure 8-1 Event binding for large transactions

Event specifications

The application analyst takes the defined business event from the business analyst and converts it to several event specifications:

- ► LargeCashDeposit
- ► LargeCashWithdrawal
- ► LargeTransferIn
- ► LargeTransferOut
- ► LargeChequeDeposit

Each event specification describes an event and the business data or information to be contained in the event. Figure 8-2 on page 145 shows the event specification for the LargeChequeDeposit event.



LargeTransactions.evbind

Specifications

- LargeCashDeposit
  - LargeCashDepositCS
- LargeCashWithdrawal
  - LargeCashWithdrawalCS
- LargeTransferIn
  - LargeTransferInCS
- LargeTransferOut
  - LargeTransferOutInternalCS
  - LargeTransferOutExternalCS
- LargeChequeDeposit
  - LargeChequeDepositOnUsCS
  - LargeChequeDepositOnThemCS

General
Identify and describe the event.

Name LargeChequeDeposit

Description Deposit of large cheque

Emitted Business Information
Describe and order the business information to be emitted by the event.

Name	Type	Length	Precision	Description
Account_Number	Text	10	0	Account Number
Amount	Numeric	0	0	Cheque amount
Transaction_Type	Text	3	0	Type of transaction
Channel	Text	3	0	Channel used for transaction

Capture Specifications
Add Capture Specifications to this event.

Automatic Capture Specification
Use this to automatically generate a capture specification for a signal event call using the business information entered above.

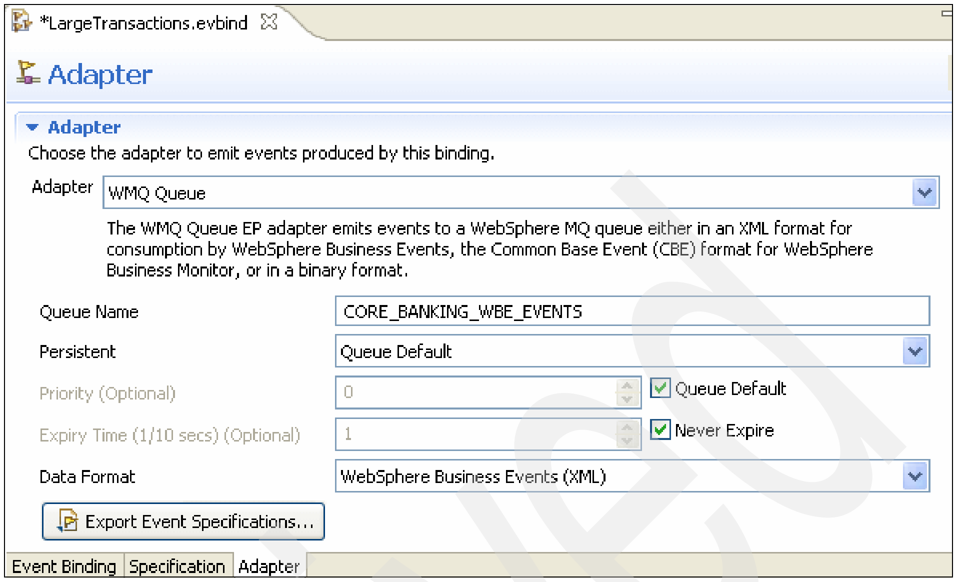
Figure 8-2 Event specification for large cheque deposits

The LargeChequeDeposit event specification describes the business information to be emitted by the event.

Event adapter

The application analyst uses the event processing adapter (EP adapter) pane to export the event specifications that are handed over to the WebSphere Business Events application developer. It contains the XML schema, which describes the format of the large transaction events.

Figure 8-3 on page 146 shows the EP adapter definition that is used for the LargeTransactions event binding.


*LargeTransactions.evbind

Adapter

Adapter

Choose the adapter to emit events produced by this binding.

Adapter: WMQ Queue

The WMQ Queue EP adapter emits events to a WebSphere MQ queue either in an XML format for consumption by WebSphere Business Events, the Common Base Event (CBE) format for WebSphere Business Monitor, or in a binary format.

Queue Name	CORE_BANKING_WBE_EVENTS	
Persistent	Queue Default	
Priority (Optional)	0	✓ Queue Default
Expiry Time (1/10 secs) (Optional)	1	✓ Never Expire
Data Format	WebSphere Business Events (XML)	

Figure 8-3 EP adapter for large transactions

In addition to the user-defined emitted data (in our example, account number, amount, transaction type, and channel), CICS automatically emits transaction context information, such as the unit of work identifier and the capture specification name. Example 8-1 shows the exported event specification for the LargeChequeDeposit event.

Example 8-1 Exported event specification for LargeChequeDeposit event

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="qualified"
            targetNamespace="http://cics.ibm.com/LargeChequeDeposit"
xmlns:tns="http://wbe.ibm.com/6.2/Event/LargeChequeDeposit">
  <xsd:element name="LargeChequeDeposit">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element maxOccurs="unbounded" minOccurs="0" name="LargeChequeDeposit_Context">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element minOccurs="0" maxOccurs="1" name="Binding user tag" type="xsd:string" />
              <xsd:element minOccurs="0" maxOccurs="1" name="Network UOWID" type="xsd:string" />
              <xsd:element minOccurs="0" maxOccurs="1" name="businessevent" type="xsd:string" />
              <xsd:element minOccurs="0" maxOccurs="1" name="Capture Spec Name" type="xsd:string" />
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element maxOccurs="unbounded" minOccurs="0" name="LargeChequeDeposit_Data">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element minOccurs="0" maxOccurs="1" name="Account_Number" type="xsd:string" />
              <xsd:element minOccurs="0" maxOccurs="1" name="Amount" type="xsd:decimal" />
              <xsd:element minOccurs="0" maxOccurs="1" name="Transaction_Type" type="xsd:string" />
              <xsd:element minOccurs="0" maxOccurs="1" name="Channel" type="xsd:string" />
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

In 8.3.1, “Application developer role” on page 151, we describe how the WebSphere Business Events application developer imports the CICS event specifications into the WBE Design Data tool.

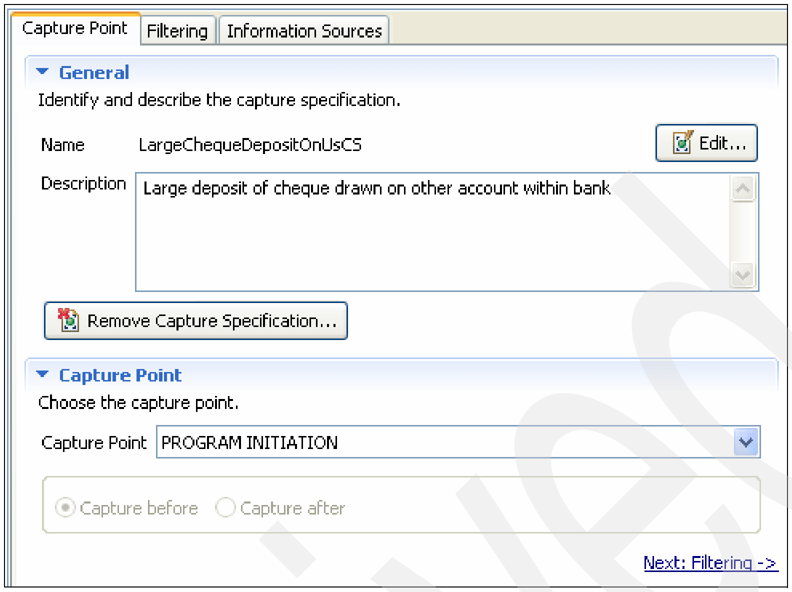
Capture specifications

A capture specification defines the place in a CICS application where a particular event can be captured. The application analyst defines two capture specifications for the types of cheque deposit transaction that the core banking system supports:

- LargeChequeDepositOnUsCS
- LargeChequeDepositOnThemCS

We specify two capture points because there are multiple paths through the application for the two types of cheque deposit: for a cheque issued by our bank (‘on us’) and a cheque issued by another bank (‘on them’). We define different filter data for each capture point based on the contents of the COMMAREA that are passed to the core banking program that handles cheque deposits. In this way, we can distinguish between the different types of cheque deposit.

The places in a CICS application that can be enabled as capture points consist of a number of the EXEC CICS API commands and program initiation. Because the core banking application does not use the CICS API extensively, program initiation is used as the capture point for each of the large transaction capture specifications. Figure 8-4 on page 148 shows the capture point for the LargeChequeDepositOnUsCS capture specification.



Name: LargeChequeDepositOnUsCS

Description: Large deposit of cheque drawn on other account within bank

Capture Point: PROGRAM INITIATION

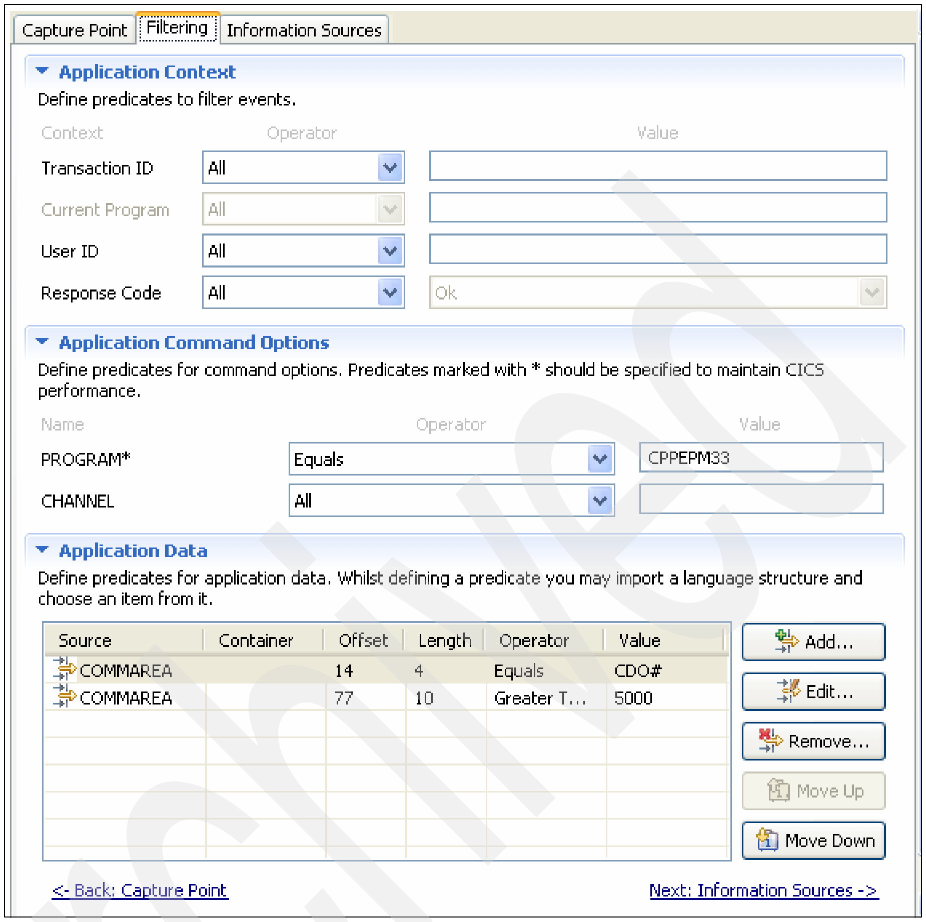
[✓ Capture before] Capture after

Figure 8-4 Capture point for large cheque deposit

The capture specification further refines the criteria for event emission by using filter data, such as the program name and data values in the COMMAREA. The capture specification defines the location of the data that is used to filter the event, for example, the application analyst knows that for all cheque deposits, the core banking program CPPEPM33 is called with a COMMAREA defined by the COBOL copybook PPEPM33. He also knows that the field name ws_id specifies the type of cheque deposit (CDO# for cheque deposit on us, and CDT# for cheque deposit on them).

Figure 8-5 on page 149 shows the filter that we used for the LargeChequeDepositOnUsCS capture specification. The application analyst:

1. Imports the PPEPM33I copybook into the event binding editor.
2. Selects which fields, and their corresponding offsets, are to be used in the data filter.
3. Specifies that an event is to be captured if the transaction type is CDO# and the amount of the transaction is greater than 5000.



Application Context

Define predicates to filter events.

Context	Operator	Value
Transaction ID	All	
Current Program	All	
User ID	All	
Response Code	All	Ok

Application Command Options

Define predicates for command options. Predicates marked with * should be specified to maintain CICS performance.

Name	Operator	Value
PROGRAM*	Equals	CPPEPM33
CHANNEL	All	

Application Data

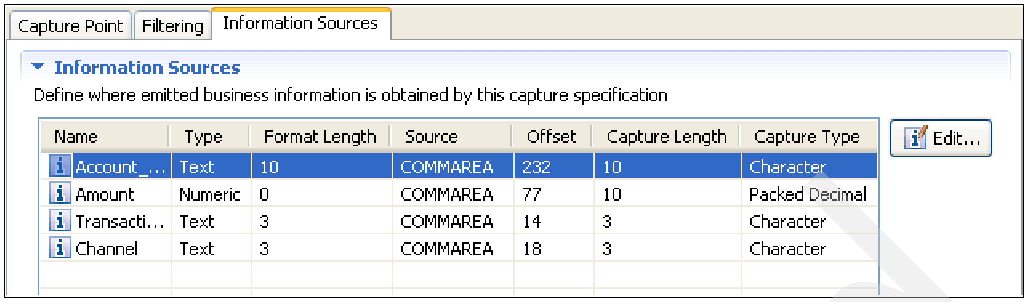
Define predicates for application data. Whilst defining a predicate you may import a language structure and choose an item from it.

Source	Container	Offset	Length	Operator	Value
COMMAREA		14	4	Equals	CDO#
COMMAREA		77	10	Greater T...	5000

Figure 8-5 Filter for large cheque deposit

Important: The CICS event binding editor uses an asterisk (*) to show the primary predicate for the capture point that is selected. For performance reasons, you must always use an Equals operator for the primary predicate.

The capture specification also describes the data to be emitted with the event and where the data is obtained. Figure 8-6 on page 150 shows the Information Sources that we used for the LargeChequeDepositOnUsCS capture specification. As for the filter, the PPEPM33I copybook is imported into the event binding editor and the application analyst selects which fields are to be emitted with the event.



Information Sources

Define where emitted business information is obtained by this capture specification

Name	Type	Format Length	Source	Offset	Capture Length	Capture Type
Account_...	Text	10	COMMAREA	232	10	Character
Amount	Numeric	0	COMMAREA	77	10	Packed Decimal
Transacti...	Text	3	COMMAREA	14	3	Character
Channel	Text	3	COMMAREA	18	3	Character

Figure 8-6 Information sources for large cheque deposit

8.2.2 Systems programmer role

The systems programmer understands how programs interact with CICS and can help the application analyst to deploy the new event processing resources. He can also monitor and optimize the performance of the system and its applications and debug and diagnose problems that are related to the application and its events.

EP adapter

CICS supports multiple EP adapters and event formats. Figure 8-3 on page 146 shows that in our scenario large transaction events are emitted from CICS in WBE format to a WebSphere MQ queue called CORE_BANKING_WBE_EVENTS. The systems programmer defines this queue and makes the necessary resource definitions in CICS to enable access to the queue.

Enabling event binding in CICS

To enable the LargeTransactions event binding, the systems programmer uses the CICS Explorer to export the Core_Banking_Events bundle that contains the event binding to the host machine. He then installs the resource object for the bundle in the core banking CICS application owning regions (AORs).

8.3 Working with WebSphere Business Events

In this section, we summarize how a Complex Event Processing logic is described within WebSphere Business Events to implement our business events scenario.

First, we explain the role of the application developer in the definition of the data model. We then explain how the business analyst builds complex event processing logic using the data model. Last, we explain the role of the systems programmer in setting the overall infrastructure.

8.3.1 Application developer role

The WebSphere Business Events application developer defines the underlying *data model* using the WBE Design data tool. The data model is the representation of the data that is processed by the WebSphere Business Events application, which we refer to as the Customer Insight application.

The data model consists of two main parts:

- Touchpoints, events, and actions that represent either in-coming events that are received or out-going actions that are sent on systems (touchpoint).
- Intermediate objects that represent business objects and that are used later during the design phase to set Context and to define Filter.

Touchpoints

A Touchpoint is a business system to be integrated with another business system as part of business event processing.

The application developer defines Touchpoints that initiate an activity (an Event) and Touchpoints that are affected by an activity (an Action).

To implement the business event logic that the business analyst defines the Customer Insight application uses the Touchpoints that are listed in Table 8-1.

Table 8-1 Application Touchpoints

Touchpoint	**Purpose**
CICS Core Banking	Receive CICS Large Transaction Events
Branch	Receive information about customer entering the Branch
Email Server	Notify Branch manager

Note: To notify the branch manager in our scenario we send an e-mail. In practice, we think a more immediate action is an SMS as a better alternative.

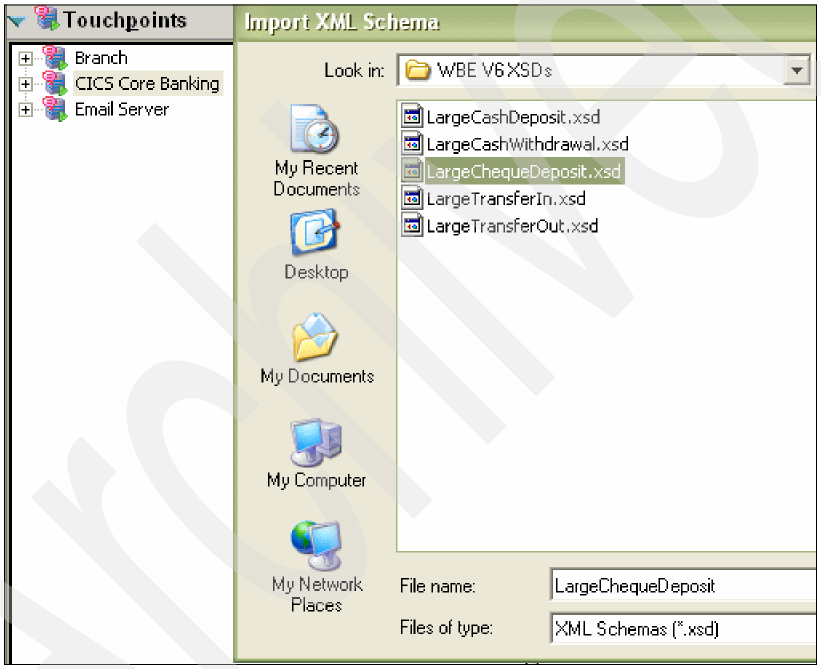
Events

The application developer identifies Events that arrive at Touchpoints. The events together with the actions (see “Actions” on page 154) are used by the Touchpoints to interact with each other.

For the Customer Insight application, the application developer uses the Import XML Schema pane to import the xsd file that the application analyst provides.

This file contains the XML schema that describes the format of the large transaction events, as shown in Example 8-1 on page 146.

Figure 8-7 illustrates importing the XML schema that describes the LargeChequeDeposit event.



Look in: WBE V6 XSDs

LargeCashDeposit.xsd
LargeCashWithdrawal.xsd
LargeChequeDeposit.xsd
LargeTransferIn.xsd
LargeTransferOut.xsd

File name: LargeChequeDeposit
Files of type: XML Schemas (*.xsd)

Figure 8-7 Large Cheque Deposit import

The same import process must be done for every event that CICS emits.

The events emitted by CICS are made up of two parts: the Context part and the Data part. These parts are modelled in WebSphere Business Events as two Event Objects, although they are associated with the same CICS LargeChequeDeposit event.

Figure 8-8 shows the Large Cheque Deposit event object corresponding to the data part of the emitted CICS event.

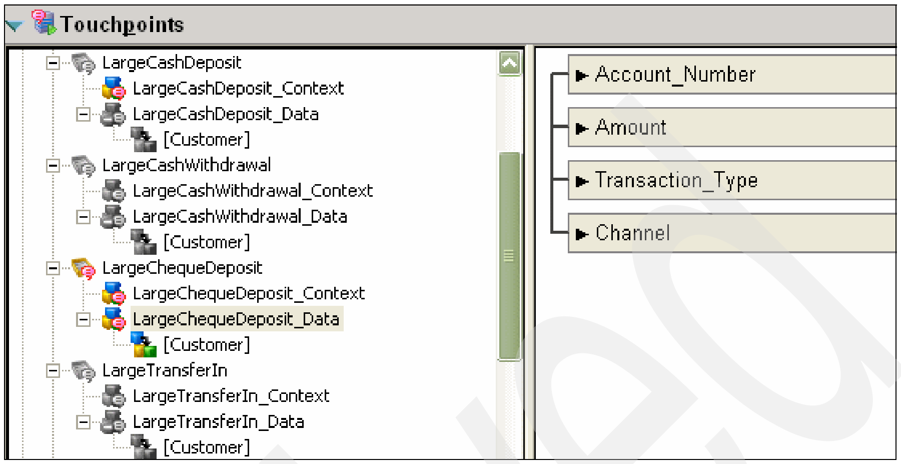


Figure 8-8 Large Cheque Deposit Event Object

Figure 8-8 also shows the other events that are defined under the CICS Core Banking Touchpoint. These are the CICS events that are defined in section 8.2, “Enabling business events in the core banking system” on page 142.

In our scenario, the Bank Branch event represents the occurrence of the customer entering a branch. As we explained in section 8.3, “Working with WebSphere Business Events” on page 150, an OpenSIM application simulates this event. The OpenSIM application sends an HTML form to WebSphere Business Events with the needed information: customer identification and account number.

Note: In a real banking scenario, an event can be emitted when the customer enters the bank branch based on an RFID that is present on his credit card.

The application developer uses the Insert pane to define the Event and the corresponding Event Object. Because no xsd file is provided by the OpenSIM application, the event object fields are manually defined.

Figure 8-9 on page 154 shows the description of Customer Enters the Branch Event Object.

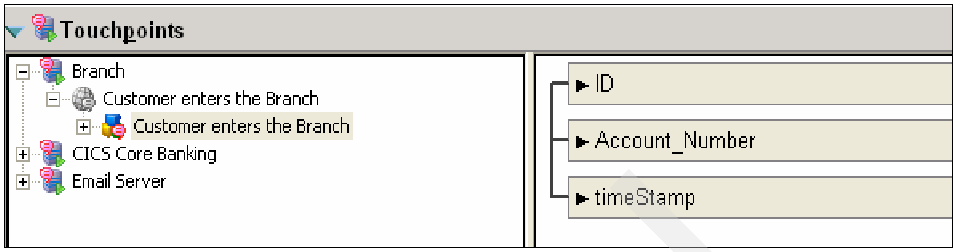


Figure 8-9 Customer Enters the Branch Event Object

Note: The timeStamp field is automatically added by the HTTP connector.

Table 8-2 summarizes the events that the Customer Insight application uses.

Table 8-2 Events by Touchpoints

Touchpoint	Event name	Event object name	Event object fields	Type
CICS Core Banking	Large Cheque Deposit	Large Cheque Deposit_Context	Binding user tag Network UOWID businessevent Capture Spec Name	String Real String String
CICS Core Banking	Large Cheque Deposit	Large Cheque Deposit_Data	Account_Number Amount Transaction_Type Channel	String Real String String
Branch	Customer Enters the Branch	Customer Enters the Branch	ID Account_Number timeStamp	String String DateTime

Note: Five events are generated by CICS to represent a Large Transaction. These events all have the same format. For the sake of simplicity, only the Large Cheque Deposit event is shown in Table 8-2.

Actions

The application developer identifies *actions* that are to be fired at Touchpoint. The events together with the actions are used by the Touchpoints to interact with each other.

The application developer defines the actions that the Customer Insight application needs.

Two actions are defined in our scenario:

- Inform Branch Manager is used to notify the branch manager of the occurrence of the customer entering the branch.
- Ack_Large_Transaction is used to acknowledge the receipt of a large transaction event. The same action is defined for each of the five Large Transaction events that come from CICS.

The action objects must be manually defined using the Insert Action Object pane.

Figure 8-10 represents the Inform Branch Manager object.

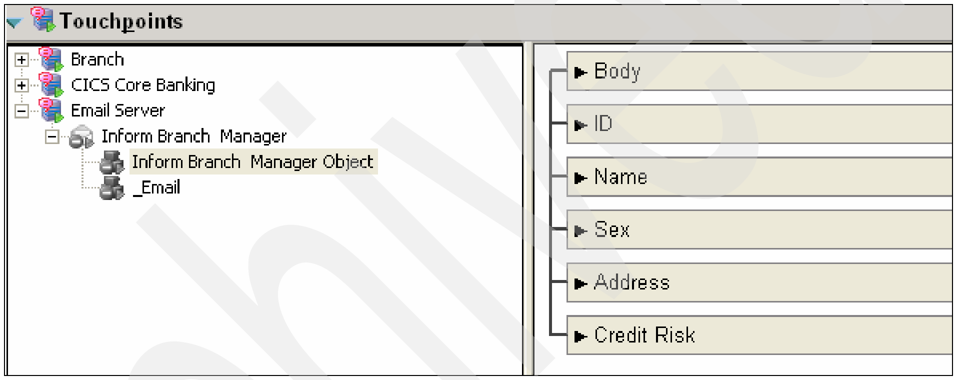
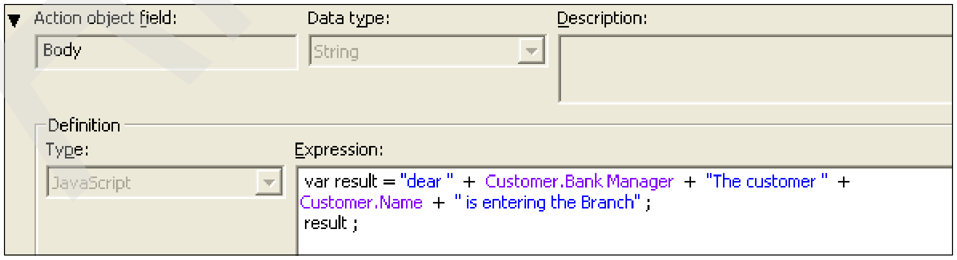


Figure 8-10 Inform Branch Manager object

The content of all of the Object fields are filled in from the Intermediate Object at run time. We explain *Intermediate Objects* in the next section.

The body field is populated using a javascript that picks up information from Intermediate Objects at run time, as shown in Figure 8-11.



Action object field: Body

Data type: String

Description:

Definition

Type: JavaScript

Expression:

```
var result = "dear " + Customer.Bank Manager + "The customer " +
Customer.Name + " is entering the Branch" ;
result ;
```

Figure 8-11 Javascript example

Table 8-3 summarizes the actions that the Customer Insight application uses.

Table 8-3 Events by Touchpoints

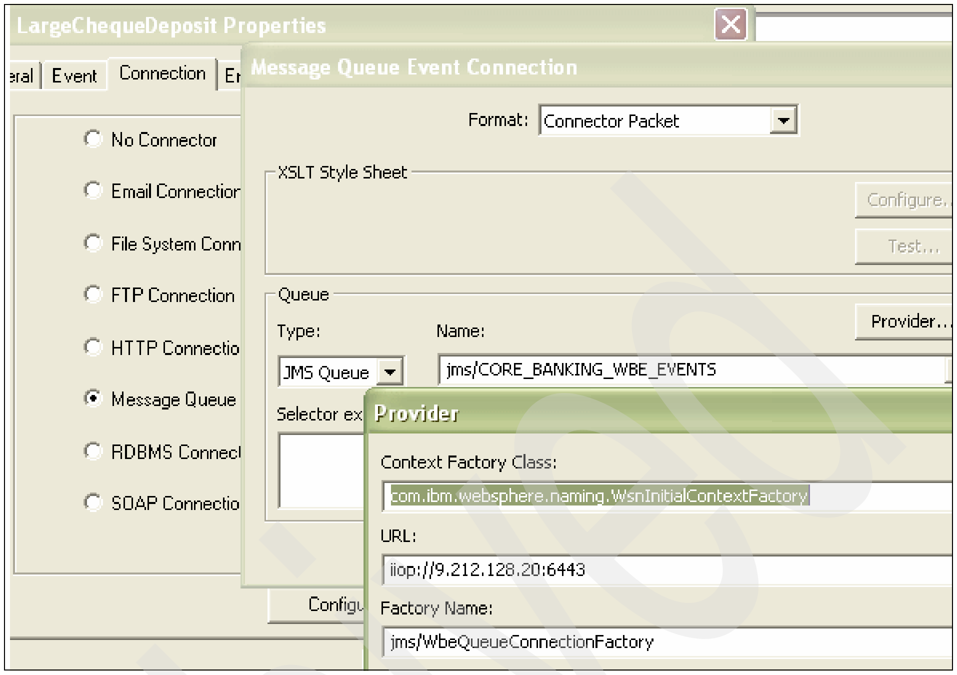
Touchpoint	Action name	Action object name	Action object fields	Type
Email Server	Inform Branch Manager	Inform Branch Manager Object	Body ID Name Sex Address Risk	String Integer String String String Integer
CICS Core Banking	Ack_LargeTransaction	Customer	ID Name Sex Address Risk Bank Manager Account_Number Amount	Integer String String String Integer String Integer Integer

Connector

WebSphere Business Events support several technology connectors that the Touchpoints use to pick up events or to fire actions. The Customer Insight application uses three separate technology connectors:

- Message Queue Connection for event coming from CICS
- Email Connection to notify the branch manager
- HTTP Connection to receive event from the OpenSIM application

Figure 8-12 on page 157 is an example of the definition that we used to manage events that come from CICS.



Format: Connector Packet

Type: JMS Queue

Name: jms/CORE_BANKING_WBE_EVENTS

Context Factory Class: com.ibm.websphere.naming.WsnInitialContextFactory

URL: iiop://9.212.128.20:6443

Factory Name: jms/WbeQueueConnectionFactory

Figure 8-12 Message Queue Connection example

Intermediate Objects

Intermediate Objects are representations of business objects, such as Customer or Account. The primary purpose of Intermediate Objects is to hold the data that will be evaluated by Interaction Sets at run time.

In WebSphere Business Events, the application developer can decouple specific event and action object definitions by mapping both event and action objects to a purely conceptual, reusable, Intermediate Object layer.

In the Customer Insight application, all of the events and actions are contextually related by the customer identification. The main Intermediate Object that the application uses is the Customer. Figure 8-13 on page 158 shows the characteristics of this object.

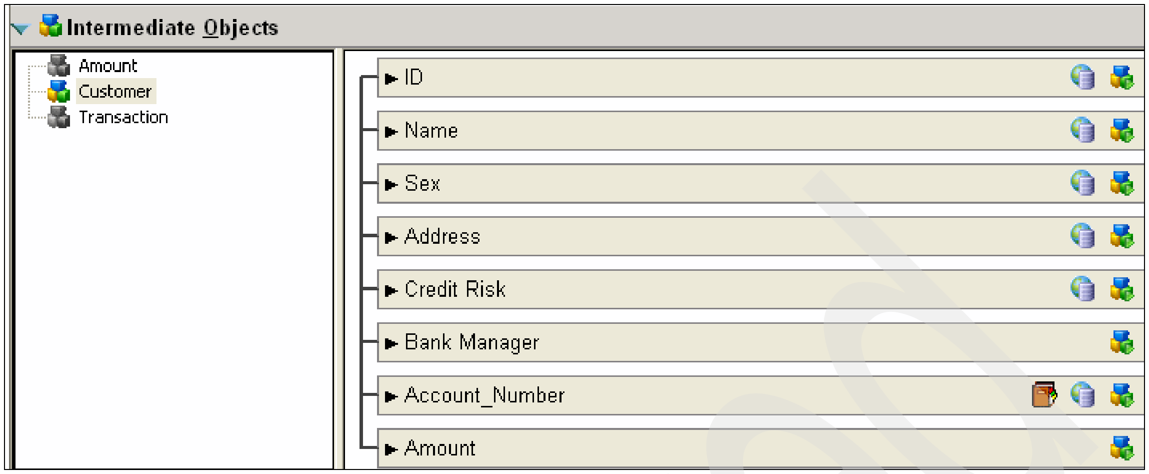


Figure 8-13 Customer Intermediate Object

Each field in an Intermediate Object can be populated from a variety of sources:

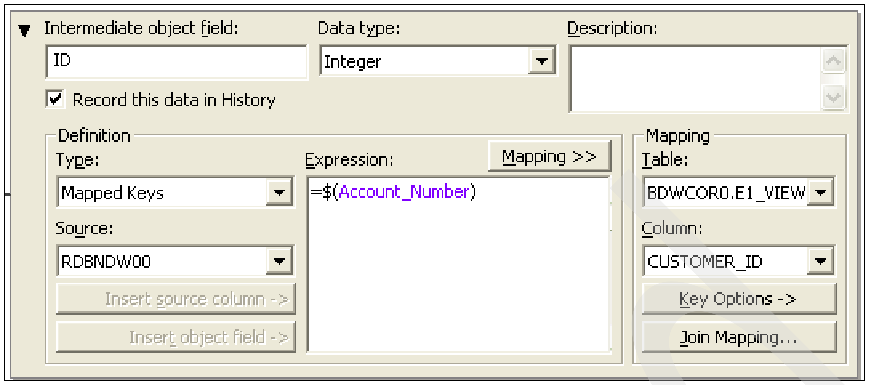
- Event Objects in Events contain fields whose values are passed to Intermediate Objects through field constructors.
- Data Sources can be used to retrieve information not supplied by events.

In our scenario, the CICS events are not conveying the customer identification, which is the information that we need to set a context. Other information that relates to the customer are also missing from the event.

In our application, the Data Warehouse is queried to get the missing information using the Account_Number as key.

WebSphere Business Events provides several methods of retrieving data from a datasource. The Customer Insight application uses the *map key method* that allows Intermediate Object fields to be specified as mapped key fields or mapped fields. At run time, mapped key fields and mapped fields are used to construct the appropriate SQL statement.

Figure 8-14 on page 159 shows how the Customer_ID field is defined as a mapped field. The field description specifies the datasource from where the information is taken (RDBNDW00), the name of the table (BDWCOR0.E1_VIEW), and the column in the table that is used to complete the field (CUSTOMER_ID). The Account_Number is used as an equal key to retrieve the information, as shown in the Expression part of the definition in Figure 8-14 on page 159.


Intermediate object field: ID

Data type: Integer

Description:

☑ Record this data in History

Definition

Type: Mapped Keys

Source: RDBNDW00

Expression: =$(Account_Number)

Mapping

Table: BDWCOR0.E1_VIEW

Column: CUSTOMER_ID

Figure 8-14 Mapped field definition example

8.3.2 Business analyst role

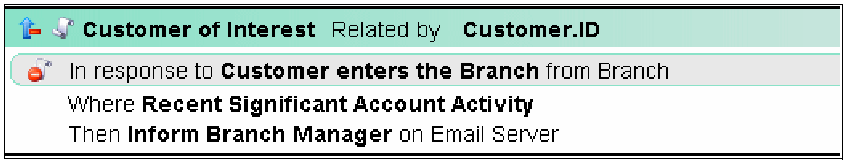
The business analyst uses the WBE Design tool to express the business event logic. He uses the data model that the application developer provides to define the complex event logic.

For our scenario, the complex event logic is comprised of the following tasks:

- Creating an interaction set for every CICS event whose main role is to acknowledge the receipt of the event. There is no filter defined, so as soon as an event arrives, the corresponding action is fired. A single action is associated with the interaction sets, and it is called Ack_LargeTransaction.
- Defining a filter called Recent Significant Account Activity that returns true when at least five large transactions are processed for the customer during the month.
- Creating an interaction set called Customer of Interest that is associated with the Branch event that uses the filter that we previously defined. This interaction set correlates the receipt of the Branch event with the fact that the customer entering the branch is having a recent significant account activity as acknowledged by the filter, to fire the defined action. The action that is associated with the interaction set is the Inform Branch Manager action, which is tailored to send an e-mail to the branch manager.

All of the interaction sets are related using the Customer.ID as the contextID.

Figure 8-15 on page 160 shows the Customer of Interest interaction set.


Customer of Interest Related by Customer.ID

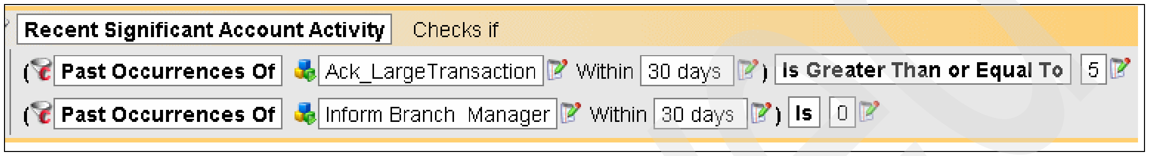
In response to **Customer enters the Branch** from Branch

Where **Recent Significant Account Activity**

Then **Inform Branch Manager** on Email Server

Figure 8-15 Customer of Interest Interaction Set

Figure 8-16 shows the filter Recent Significant Activity that is associated with the Interaction Set to implement the business event logic of the Customer Insight application.


Recent Significant Account Activity Checks if

(**Past Occurrences Of** Ack_LargeTransaction Within 30 days) **Is Greater Than or Equal To** 5

(**Past Occurrences Of** Inform Branch Manager Within 30 days) **Is** 0

Figure 8-16 Recent Significant Account Activity

Figure 8-15 and Figure 8-16 illustrate how WebSphere Business Events provide graphical, non programming user interfaces that allow the business analyst to express the business event logic.

Note: If you are using WebSphere Business Events V7.0, then the WBE Design tool runs under Business Space and provides a different user interface.

8.3.3 Systems programmer role

The systems programmer understands the overall infrastructure that is needed to interconnect the various Touchpoints. Interconnection is needed between the CICS core banking system and the WebSphere Business Events system. WebSphere MQ is used as the underlying transport for the CICS events. The systems programmer defines the JMS objects (destination and connection factories) that are used to receive the event using the Messages Queue connection.

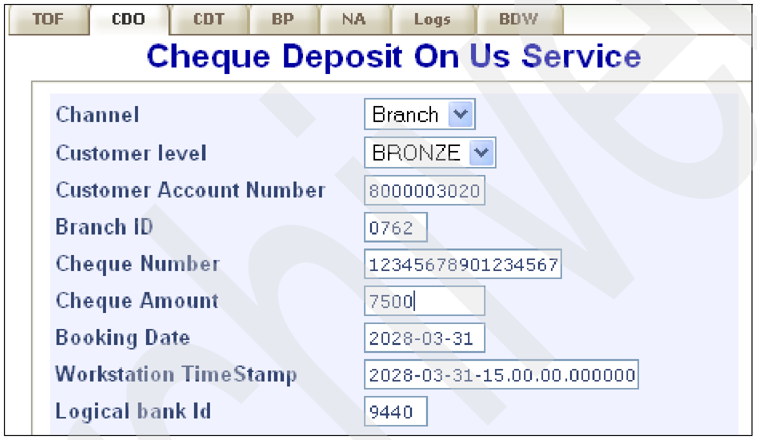
The systems programmer is also responsible for the definition of the datasource that the Customer Insight application uses. He sets up the correct JDBC connectivity and properties.

8.4 Testing the scenario

In this section, we show how we tested our business events scenario. First, we show the large transaction events occurring in the CICS core banking system. Next, we show how we simulate the customer arriving at the bank branch, and we finish by showing the event processing of WebSphere Business Events.

8.4.1 Large transactions in the smarter banking showcase

Banking transactions are simulated using our Web application (see 7.4.4, “Workload simulation” on page 131), for example, Figure 8-17 shows a large cheque deposit for the customer with account number 8000003020.



Cheque Deposit On Us Service

Channel	Branch
Customer level	BRONZE
Customer Account Number	8000003020
Branch ID	0762
Cheque Number	12345678901234567
Cheque Amount	7500
Booking Date	2028-03-31
Workstation TimeStamp	2028-03-31-15.00.00.000000
Logical bank Id	9440

Figure 8-17 Large cheque deposit

Rational Performance Tester (RPT) simulates a core banking workload by simulating a full set of banking transactions for a range of accounts. In Figure 8-18 on page 162, we captured the response times for a small set of transactions.

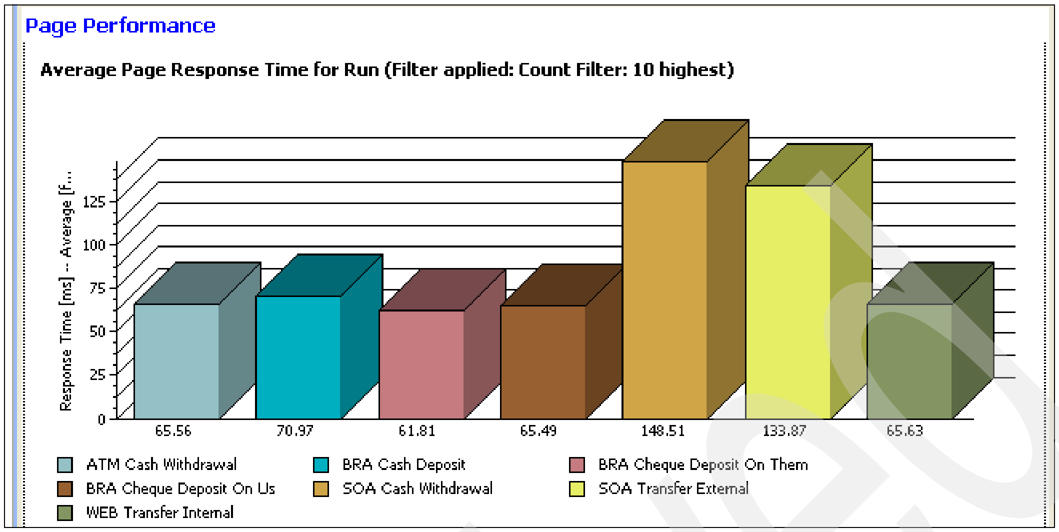
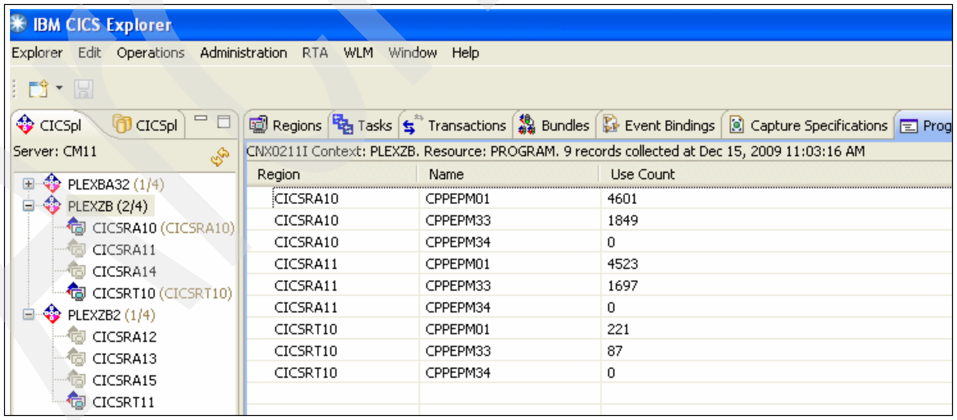


Figure 8-18 Simulated core banking workload with Rational Performance Tester

In 8.2, “Enabling business events in the core banking system” on page 142, we show how the application analyst uses the event binding editor of CICS Explorer to enable events for large transactions. The CICS Explorer is also used to view the core banking workload, for example, transaction and program invocations.

Figure 8-19 shows how the systems programmer uses the CICS Explorer to view the frequency of invocations for the core banking CICS programs.

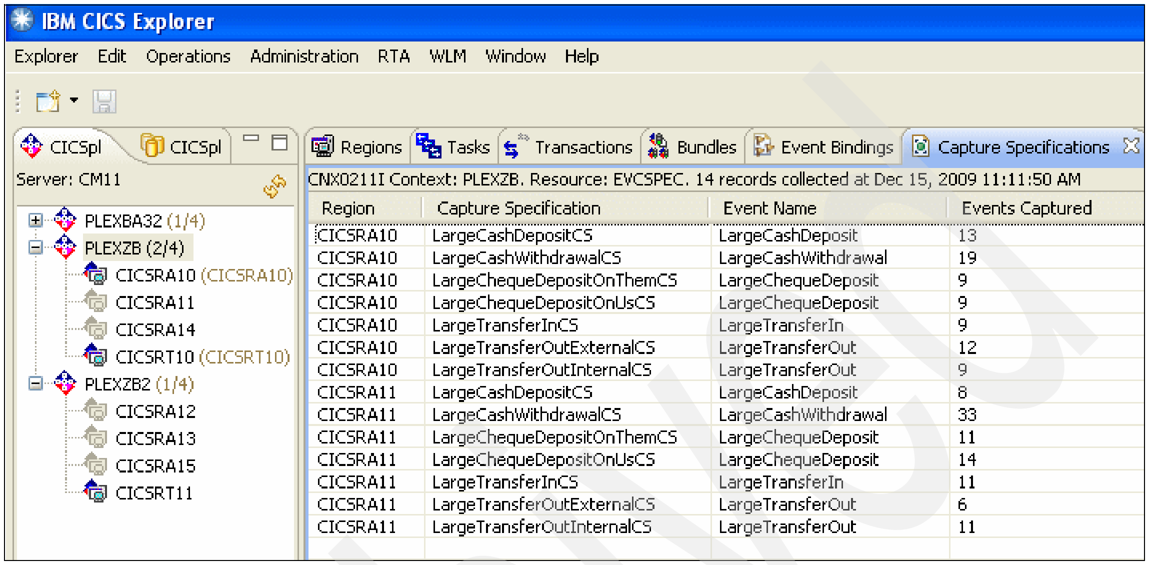


CNX0211I Context: PLEXZB. Resource: PROGRAM. 9 records collected at Dec 15, 2009 11:03:16 AM

Region	Name	Use Count
CICSRA10	CPPEPM01	4601
CICSRA10	CPPEPM33	1849
CICSRA10	CPPEPM34	0
CICSRA11	CPPEPM01	4523
CICSRA11	CPPEPM33	1697
CICSRA11	CPPEPM34	0
CICSRT10	CPPEPM01	221
CICSRT10	CPPEPM33	87
CICSRT10	CPPEPM34	0

Figure 8-19 CICS Explorer Programs view

Figure 8-20 shows how the systems programmer uses the CICS Explorer to view the occurrence of large transaction events, which is a small subset of the total number of transactions.



CNX0211I Context: PLEXZB. Resource: EVCSPEC. 14 records collected at Dec 15, 2009 11:11:50 AM

Region	Capture Specification	Event Name	Events Captured
CICSRA10	LargeCashDepositCS	LargeCashDeposit	13
CICSRA10	LargeCashWithdrawalCS	LargeCashWithdrawal	19
CICSRA10	LargeChequeDepositOnThemCS	LargeChequeDeposit	9
CICSRA10	LargeChequeDepositOnUsCS	LargeChequeDeposit	9
CICSRA10	LargeTransferInCS	LargeTransferIn	9
CICSRA10	LargeTransferOutExternalCS	LargeTransferOut	12
CICSRA10	LargeTransferOutInternalCS	LargeTransferOut	9
CICSRA11	LargeCashDepositCS	LargeCashDeposit	8
CICSRA11	LargeCashWithdrawalCS	LargeCashWithdrawal	33
CICSRA11	LargeChequeDepositOnThemCS	LargeChequeDeposit	11
CICSRA11	LargeChequeDepositOnUsCS	LargeChequeDeposit	14
CICSRA11	LargeTransferInCS	LargeTransferIn	11
CICSRA11	LargeTransferOutExternalCS	LargeTransferOut	6
CICSRA11	LargeTransferOutInternalCS	LargeTransferOut	11

Figure 8-20 CICS Explorer Capture Specification view

Figure 8-20 shows the number of large transaction events that occurred across two of the CICS AORs. These events are emitted using the WMQ EP adapter so that they can be processed by WebSphere Business Events.

Note: For the initial testing of an event binding we recommend using the CICS temporary Storage Queue (TSQ) adapter. Events are emitted to a TSQ and can be inspected using the CICS supplied transaction CEBR. After everything is working as expected, the CICS bundle that contains the event binding can be redeployed with the WMQ Queue EP adapter specified.

8.4.2 Customer enters branch

We simulate the event that a customer enters the bank branch by using the virtual world of OpenSimulator. In reality, such an event can be triggered by a RFID embedded in the customer’s credit card.

OpenSimulator, often referred to as OpenSim, is an open source server platform for hosting virtual worlds. It is most recognized for compatibility with the Second Life client, but it is can also host alternative worlds with differing feature sets with multiple protocols.

In the smarter banking showcase environment, OpenSim runs in a Linux partition of a System z10.

Figure 8-21 shows the banking customer avatar entering the virtual bank branch.

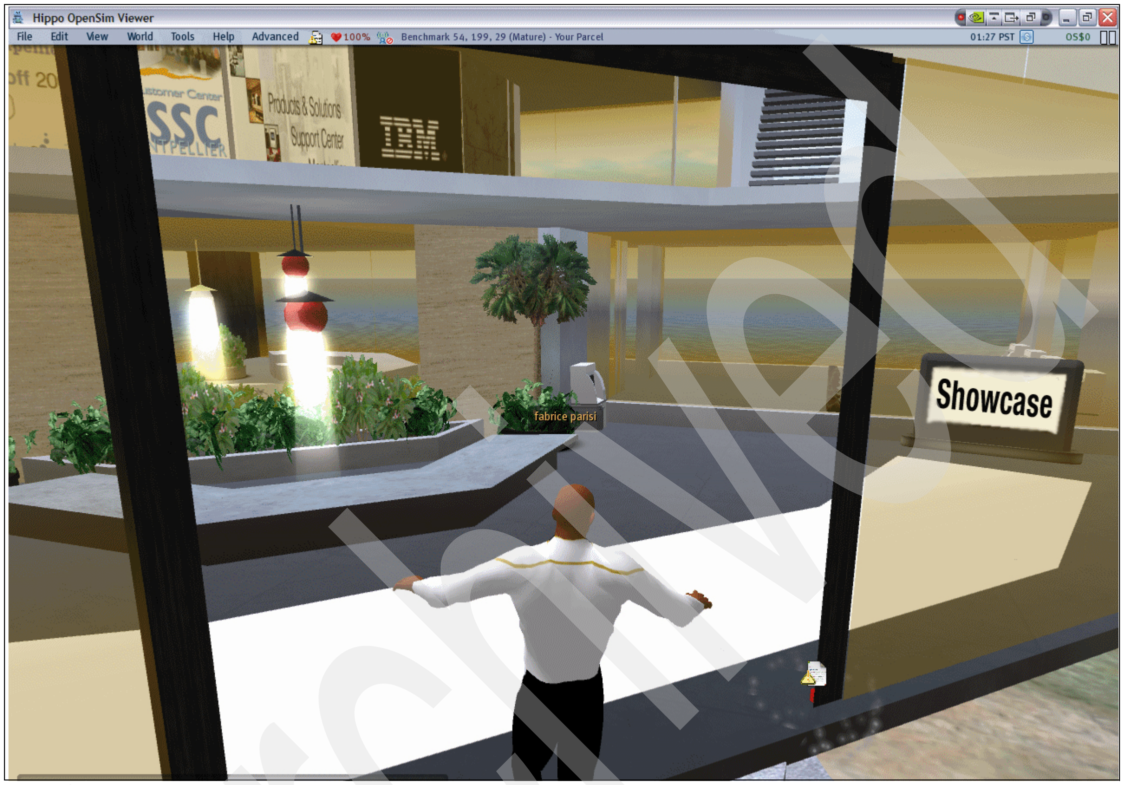


Figure 8-21 Customer enters branch

OpenSim is configured to send an HTTP request when any of the bank's customers enter the virtual branch.

Note: In reality this capability can be enabled only for a certain set of customers, for example, gold card customers or private banking customers.

Example 8-2 on page 165 shows the URL that we used to send the branch event to WebSphere Business Events.

Example 8-2 HTTP request sent from branch to WebSphere Business Events

```
http://9.212.128.20:6447/wbe/servlet/EventConnectorServlet?_event=Custo
mer+enters+the+Branch&_object=Customer+enters+the+Branch&ID=10128142691
0&Account_Number=8000003020
```

8.4.3 Processing events in WebSphere Business Events

Business events can be viewed using Business Space powered by WebSphere. The business analyst creates and configures a Dashboard that contains charts that display real time activity within WebSphere Business Events.

Figure 8-22 is an example of a pie chart that represents the large transactions activity at the CICS Core Banking Touchpoint.

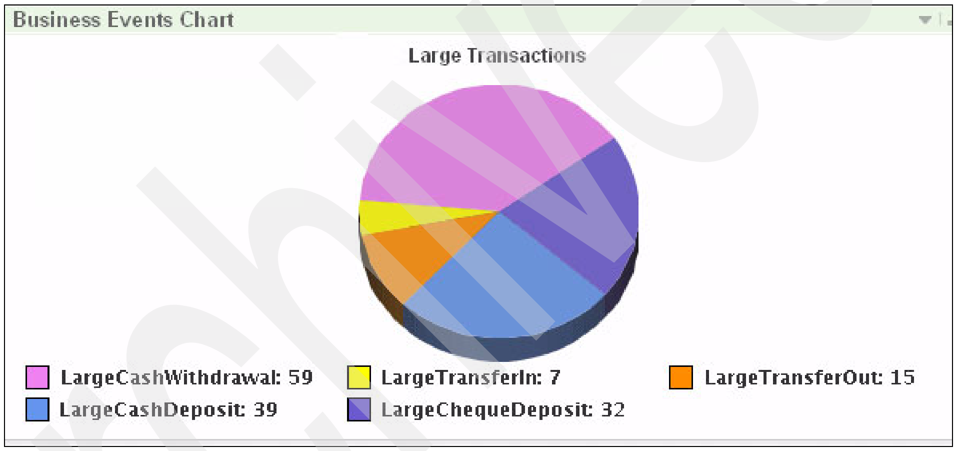


Figure 8-22 Business Event Chart

Figure 8-23 on page 166 is another Dashboard that represents the number of times that the Inform Manager action is fired. This action is initiated when a specific customer enters a branch if that customer has performed five or more large transactions within the previous 30 days. This Dashboard can help the business analyst validate the overall scenario.

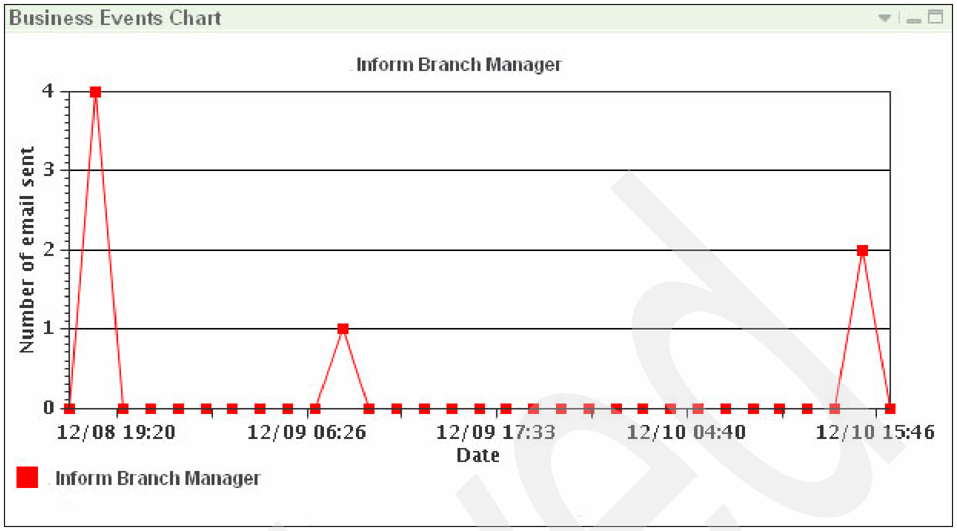


Figure 8-23 Inform Branch Manager chart

8.5 Summary

In this chapter, we showed how CICS event processing support can be used to provide end-to-end visibility of a bank customer's transactions. When correlated with other events, for example, the physical presence of the customer in the branch, real-time actions can be automated, enabling more effective sales and service.

9

Enabling Web 2.0

In this chapter, we show how we enabled the smarter banking showcase for Web 2.0.

First, we explain our business scenario, which exposes account information and payment terminal location information (point-of-sale devices and ATMs) as Atom feeds. We then show how these two information sources can be mashed up in an Android mobile application to give the location of where each transaction occurred.

In our example, the mobile device is GPS enabled. The application uses GPS history data together with transaction locations to alert the user if they were not near a transaction when it happened, which can potentially identify fraudulent activity on the account. The customer can then take appropriate action to respond to fraudulent activity within minutes rather than days or weeks, which greatly reduces the risk of credit card fraud.

We describe the setup that is required to emit the feeds from CICS and discuss how the feeds can be consumed and used by Web 2.0 applications, such as our showcase mobile application.

9.1 Introduction to Web 2.0

Web 2.0 is not a technology; instead, it is more of an idea, a philosophy, or even an architecture that describes how users interact with Web applications. Web 2.0 is a term that helps to describe the shift away from old world Web 1.0 applications where every user of an application consumes and interacts with the same information presented as the author intended.

In a Web 2.0 application, each user can customize their experience to their specific needs. The user can select what information they want to be displayed and how they want to display it. Furthermore, the user is not restricted to one information source but can pull information from various locations and mash them together however they want.

See Chapter 6, “Web 2.0 support” on page 87 for a comparison of Web 1.0 and Web 2.0 solutions.

9.2 Web 2.0 scenario

As we discussed in the business scenarios in 3.2, “Core banking transformation” on page 35, customer differentiation is at the heart of the core banking transformation.

It is becoming more and more common for customers to have multiple accounts in separate banks, and it is not easy to manage these accounts. Checking multiple online banking Web sites can be time consuming, and it is easy for customers to miss something important, such as a bounced check or a direct debit because they are only likely to check the site a few times a month. In addition, budgeting across accounts from different banks can be very difficult because there is no way to easily extract and integrate the information from the various Internet banking Web sites.

In 3.2.3, “Accounts in multiple banks” on page 36, we discussed the trend towards customers having multiple accounts in multiple banks, and the customer’s request to see all their bank accounts in one consolidated view. For the showcase, the solution lies in expanding the bank’s online presence to include information sources that can supply account transaction data to a customer in an easily consumable format, in this case Atom feeds.

In our scenario, we configure two Atom feeds:

- An Atom feed of transaction data from an account
- An Atom feed that contains location data of the payment terminals

The intention is that a customer can use Web 2.0 applications to consume this data in various ways, depending on their needs. Some examples are:

- Mobile application, which uses the feed of terminal locations to find the nearest ATM to the customers current location.
- Budgeting application, which uses the feed of account transactions from a number of the customer's accounts to aid budgeting across accounts.
- GPS enabled mobile application, which mashes both feeds to determine the location of a transaction and correlates these locations with GPS history data to flag when the customer was not near a transaction at the time it took place, which indicates potential fraudulent activity.

These are just a few examples, but the advantage of Web 2.0 is that after these feeds are published any third-party application provider can find new and innovative ways to utilize the data. Not only does this provide existing customers with added value, but this improved customer experience differentiates the showcase from its competitors in the marketplace, improving its customer retention and allowing it to attract a greater number of new customers.

In the next sections, we show how the new CICS TS 4.1 Atom feed support, which described in 6.3.1, "Atom feeds" on page 102, can be used to emit data from the showcase's core banking system ready for Web 2.0 consumption. We also look at how it is possible to create a GPS-enabled mobile application from the previous examples using the Android operating system for mobile devices.

Atom feeds: Atom feeds are enabled in the showcase in a non-invasive way, that is without changes to the core banking system.

9.2.1 Roles

The workflow for this scenario requires the participation of these people:

- The application analyst programmer

  The application analyst programmer determines how to configure our feed. He understands:

  - Where the data for the feeds is located
  - How this data must map to the Atom feed

We provide an overview of the actions of the application analyst in 9.3.1, “Application analyst programmer role” on page 170.

- The systems programmer

  The systems programmer sets up the infrastructure in support of the Atom feeds. He defines and installs RDO resources into CICS using the information that the application analyst supplies.

  We provide an overview of the actions of the systems programmer in 9.3.2, “Systems programmer role” on page 176.

- The Web 2.0 application developer

  The Web 2.0 application developer develops the GPS-enabled mobile application, which consumes the feeds. He might be employed by the bank or working for a third-party application developer. He understands:

  – How to extract data of interest from the feed
  – How to mash data from the two feeds
  – How to program a mobile application using the Android SDK

  We provide an overview of the actions of the Web 2.0 application developer in 9.4.1, “Web 2.0 application developer role” on page 180.

9.3 Web 2.0 enablement of core banking systems

In this section, we summarize how we enabled Atom feeds in the showcase CICS core banking system.

9.3.1 Application analyst programmer role

In our scenario, it is the role of the bank’s application analyst programmer to identify where the data for our feeds resides in our CICS core banking system. After the data is identified he must supply all of the necessary feed configuration information to tell CICS how to map the data source to the feed.

The data for the two feeds is located in the following places:

- For the terminal feed

  There is a VSAM KSDS file called DEVLOC that contains data about each point-of-sale (POS) merchant terminal and ATM on the showcase network. Each file record contains the following data:

  – Terminal ID, for example, ATM0001 or BOOTS01
  – Shortname of the device, which includes the merchant name for POS
  – Address

– Town/City
– Country
– Latitude
– Longitude

- For the account feed

  There is an existing DB2 table that is a part of the showcase Banking Data Warehouse (BDW), which holds near real time data about account transactions. The following columns of interest are available on the table:

  – Terminal ID matches the Terminal ID in the VSAM file in the previous bullet
  – Account Number
  – Amount
  – Date of Transaction
  – Time of Transaction
  – Retail Description which describes the type of transaction

Because CICS TS V4.1 provides support for Atom feeds from VSAM files without additional programming, the terminal feed is the simpler case, which we discuss first.

> **Important:** The next section describes how data that is stored in traditional formats can be served as Atom feeds with little or no programming using CICS TS V4.1.

Terminal Location Feed: Atom feed from VSAM file

Having identified the DEVLOC file, the application analyst first must generate a wsbind file that describes the mapping between the file's record structure and an XML format. To do this they must find the copybook that represents the record structure and use the DFHLS2SC utility program that is provided with the CICS XML assistant to generate a wsbind file from it. Example 9-1 shows the COBOL copybook for the record structure.

Example 9-1 DEVLOCCB copybook for DEVLOC file record structure

```
        10 TERMID                      PIC X(16).
        10 SHORTNAME                   PIC X(16).
        10 ADDRESS                     PIC X(30).
        10 CITY                        PIC X(20).
        10 COUNTRY                     PIC X(20).
        10 LONGITUDE                   PIC X(12).
        10 LATITUDE                    PIC X(12).
```

DFHLS2SC generates a wsbind file from the copybook that describes the mapping between the COBOL fields to the XML structure that are shown in

Example 9-2. This structure then makes up the content of the terminal location feed entries.

Note: The feed contains many entries. Each entry in the feed maps to one record in the VSAM file.

Example 9-2 DEVLOCCB XML structure

```
<DEVLOCCB
xmlns=”http://www.ibm.com/xmlns/prod/cics/atom/bindfile/devloccb”>
   <termid>...</termid>
   <shortname>...</shortname>
   <address>...</address>
   <city>...</city>
   <country>...</country>
   <longitude>...</longitude>
   <latitude>...</latitude>
</DEVLOCCB>
```

The next step is to define the ATOMSERVICE config file. Example 9-3 shows the config file.

Example 9-3 ATOMSERVICE config file for terminal location feed

```
<?xml version="1.0"?>
<cics:atomservice type="feed1"

xmlns:cics="http://www.ibm.com/xmlns/prod/cics/atom/atomservice"
         xmlns:atom="http://www.w3.org/2005/Atom">
   <cics:feed>
      <cics:resource name="DEVLOC3" type="file2">
         <cics:fieldnames atomid="termid7" title="shortname8"/>
         <cics:bind root="DEVLOCCB6"/>
      </cics:resource>
      <cics:selector format="hexadecimal"/>
      <cics:authority
         name="http://www.ibm.com/cics/Showcase" date="2009-12-01"/>
   </cics:feed>
   <atom:feed>
      <atom:title type="text">Showcase Terminal Feed9</atom:title>
      <atom:subtitle>
         A feed of location data for Smarter Banking Showcase
Terminals10
      </atom:subtitle>
       <atom:link rel="self" href="/Showcase/terminals/feed4" />
      <atom:entry>
         <atom:link rel="self" href="/Showcase/terminals5" />
         <atom:content type="application/xml"
               cics:resource="DEVLOC3" cics:type="file2" />
      </atom:entry>
   </atom:feed>
</cics:atomservice>
```

See “Atom” on page 92 for general information about the composition of Atom feed documents. The ATOMSERVICE config file for the terminal location feed specifies the following bold, numbered items in Example 9-3 on page 172:

1. This is a feed rather than a collection (feeds are read only whereas collections can be updated).
2. This feed is for a file resource.
3. This feed is for a file named DEVLOC.
4. The URI for this feed is HOST/Showcase/terminals/feed.
5. The URI for entries of this feed is HOST/Showcase/terminals/SELECTOR (SELECTOR is a unique identifier for each entry).
6. The root tag of the wsbind file is <DEVLOCCB>.
7. Map the <termid> field, from the XML Example 9-2 on page 172 generated from the COBOL structure in Example 9-1 on page 171 to the <id> field in the Atom entry.
8. Map the <shortname> field, from the XML Example 9-2 on page 172 generated from the COBOL structure in Example 9-1 on page 171, to the <title> field in the Atom entry.
9. The title of the feed is Showcase Terminal Feed.
10. The subtitle of the feed.

This completes the steps for the application analyst programmer for the terminal location feed. The config and wsbind files are passed to the systems programmer who installs the necessary CICS resources that enable the feed (9.3.2, “Systems programmer role” on page 176).

The feed of terminals from the showcase does not include every terminal in the world but the mobile application can consume multiple terminal feeds from a number of sources to maintain a more complete list. The name and location of terminals (ATMs or Point of sale devices) that are not part of a Banks own network typically arrives with the Interchange (VISA, Mastercard) standard message. We have assumed that in the future this includes GPS coordinates or that this information could be added. So we can imagine a more or less static feed that lists the terminals that form part of the Banks network and then a second feed that is more transient listing the (not on-us) terminal details coming in from the Interchanges.

Account Transaction Feed: Atom feed from a DB2 table

CICS does not have direct support for serving feeds from DB2 tables; therefore, for the account transaction feed, the application analyst programmer must write a program that utilizes the ATOMSERVICE's PROGRAM interface. The program is called a *service routine*.

A service routine's job, in the case of a feed, is to return data for the various fields of an entry with a given selector. A selector is an ID that can uniquely identify each entry within the feed. Because our feed is from a database table we use the primary key as the selector. It must also return a selector for the next and previous entries, if they exist. If the service routine is called and a selector is not given, it should return data for the first entry in the feed (the most recent).

CICS uses the service routine when responding to a feed request. It is used to browse a data source (in our case a DB2 table) one entry at a time with each call to the service routine providing data for the current entry and a selector for the next entry. The next selector is used in the next call to the service routine and so on and so forth. This process continues until either CICS retrieves enough entries or no next selector is returned, which indicates the end of the feed. CICS handles all of the feed creation and management, so all we are concerned about is accessing our data.

The service routine SBATMACC uses the Date, Time, and Terminal ID to uniquely identify a transaction in our DB2 table, which are concatenated to form the unique selector for each entry, for example, 2009-12-1613.24.35ATM0001.

It is also important to note that the next selector that SBATMACC returns is not necessarily the next row in the DB2 table; instead, it is the next transaction with the same account number as the current entry (so that we only emit transactions for one account).

Note: A good way to supply the account number in the feed request is by using the query string. Query strings can easily be read by SBATMACC using the new QUERYPARM API in CICS TS V4.1.

SBATMACC emits the transaction data from the DB2 table row to the following entry fields:

- Retail Description to <title>
- Account number (masked for security) to <summary>

- Date and time reformatted to a RFC3339 time stamp to <published> and <updated>
- Amount and Terminal ID to an XML structure inside the entries content tag

After the service routine SBATMACC is written, the application analyst produces a config file for the account transaction feed. Example 9-4 shows the config file.

Example 9-4 ATOMSERVICE config file for the account transaction feed

```
<?xml version="1.0"?>
<cics:atomservice type="feed[1]"

xmlns:cics="http://www.ibm.com/xmlns/prod/cics/atom/atomservice"
        xmlns:atom="http://www.w3.org/2005/Atom">
   <cics:feed>
      <cics:resource name="SBATMACC[3]" type="program[2]">
         <cics:bind root="SBTRANSD[6]"/>
      </cics:resource>
      <cics:authority
         name="http://www.ibm.com/cics/Showcase" date="2009-12-01"/>
   </cics:feed>
   <atom:feed>
      <atom:title type="text"> Account Feed[7]</atom:title>
      <atom:subtitle>
         A feed of transactions for a Smarter Banking Showcase account[8]
      </atom:subtitle>
       <atom:link rel="self" href="/Showcase/account/feed[4]" />
      <atom:entry>
         <atom:link rel="self" href="/Showcase/account[5]" />
         <atom:content type="application/xml"
               cics:resource="SBATMACC[3]" cics:type="program[2]" />
      </atom:entry>
   </atom:feed>
</cics:atomservice>
```

The ATOMSERVICE config file for the account transaction feed specifies the following bold, numbered items in Example 9-4:

1. This is a feed rather than a collection (feeds are read only, whereas collections can be updated).
2. This feed is for a program resource.
3. This feed is for a program named SBATMACC.
4. The URI for this feed is HOST/Showcase/account/feed.

5. The URI for entries of this feed is HOST/Showcase/account/SELECTOR (SELECTOR is a unique identifier).
6. The root tag of the XML in the entry content is <SBTRANSD>.
7. The title of the feed is Account Feed.
8. The subtitle of the feed.

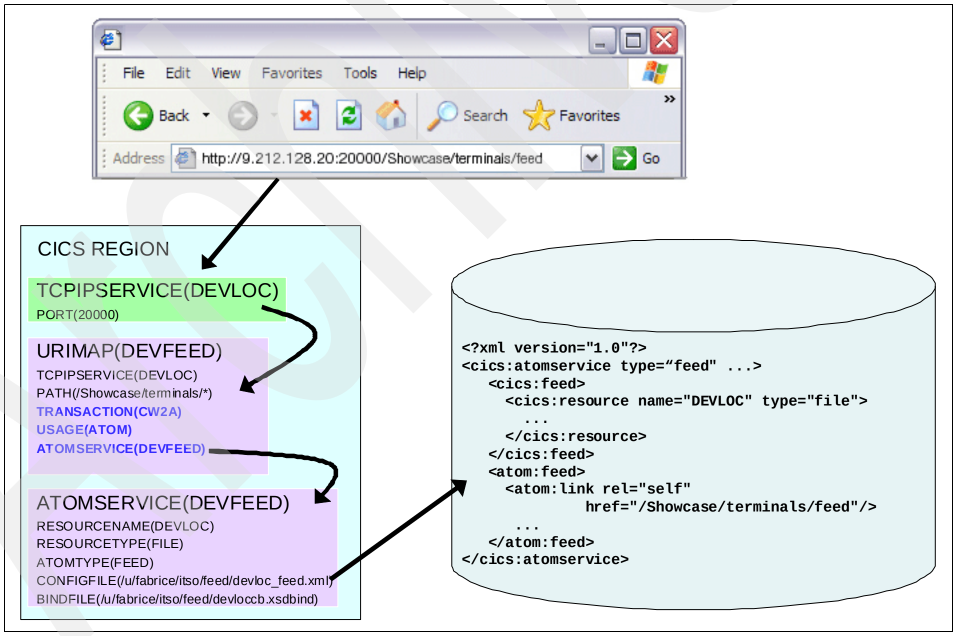
This completes the steps for the application analyst programmer for the account transaction feed. The config and wsbind files are passed to the systems programmer who installs the necessary CICS resources, which enable the feed.

9.3.2 Systems programmer role

The systems programmer sets up of the necessary resources to enable the terminal location and account transaction Atom feeds.

Terminal location feed setup

Figure 9-1 shows the resources that are defined for the terminal location feed.


File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://9.212.128.20:20000/Showcase/terminals/feed
Go
CICS REGION

```
TCPIPSERVICE(DEVLOC)
PORT(20000)

URIMAP(DEVFEED)
TCPIPSERVICE(DEVLOC)
PATH(/Showcase/terminals/*)
TRANSACTION(CW2A)
USAGE(ATOM)
ATOMSERVICE(DEVFEED)

ATOMSERVICE(DEVFEED)
RESOURCENAME(DEVLOC)
RESOURCETYPE(FILE)
ATOMTYPE(FEED)
CONFIGFILE(/u/fabrice/itso/feed/devloc_feed.xml)
BINDFILE(/u/fabrice/itso/feed/devloccb.xsdbind)
```

```
<?xml version="1.0"?>
<cics:atomservice type=“feed" ...>
   <cics:feed>
     <cics:resource name="DEVLOC" type="file">
       ...
     </cics:resource>
   </cics:feed>
   <atom:feed>
     <atom:link rel="self"
              href="/Showcase/terminals/feed"/>
     ...
   </atom:feed>
</cics:atomservice>
```

Figure 9-1 Resources configured for terminal location ATOM feed

TCPIPSERVICE resource

The TCPIPSERVICE defines the port on which CICS can receive HTTP requests for the ATOM service. This location is where we can specify security requirements, such as SSL, that protect from unauthorized access.

Note: A TCPIPSERVICE that is already defined and is being used for support of other HTTP requests can be reused for Atom feeds.

URIMAP resource

The URIMAP maps the URI that is specified in the HTTP request to the ATOMSERVICE that is to be used to service the request. The URIMAP specifies USAGE(ATOM) and the name of the ATOMSERVICE DEVFEED. The PATH /Showcase/terminals* matches the URIs in the ATOMSERVICE config file.

ATOMSERVICE resource

The ATOMSERVICE defines the feed that CICS delivers in response to the HTTP request. It specifies the CICS resource type (FILE) that provides the data for the feed, the location of the Atom configuration file that defines the feed document (devloc_feed.xml), and the XML binding that describes the layout of the resource (devloccb.xsdbind).

Example 9-5 shows an example entry from the terminal location feed.

Example 9-5 Terminal location feed entry

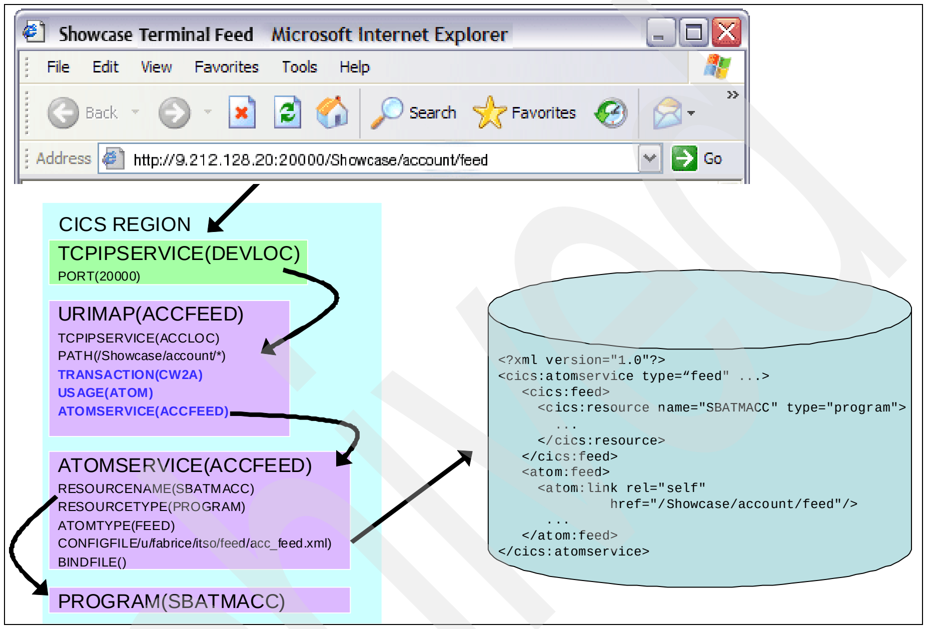
```
<entry>
   <link rel="self"
href="http://9.212.128.20:20000/Showcase/terminals/c1e3d4e3f0f0f5404040
404040404040"/>
   <id>ATMT005</id>
   <title>ATMT005 BATH</title>
   <updated>2009-12-16T05:07:14+00:00</updated>
   <published>2009-12-16T05:07:14+00:00</published>
   <content type="application/xml">
      <DEVLOCCB

xmlns="http://www.ibm.com/xmlns/prod/cics/atom/bindfile/devloccb">
         <address>Argyle Street, BA2 4BA</address>
         <city>Bath, Somerset</city>
         <country>England</country>
         <longitude>-2.361546</longitude>
         <latitude>+51.381853</latitude>
      </DEVLOCCB>
   </content>
```

```
</entry>
```

Account transactions feed setup

Figure 9-2 shows the resources that are defined for the account transaction feed.


Showcase Terminal Feed Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://9.212.128.20:20000/Showcase/account/feed Go
CICS REGION

```
TCPIPSERVICE(DEVLOC)
PORT(20000)

URIMAP(ACCFEED)
TCPIPSERVICE(ACCLOC)
PATH(/Showcase/account/*)
TRANSACTION(CW2A)
USAGE(ATOM)
ATOMSERVICE(ACCFEED)

ATOMSERVICE(ACCFEED)
RESOURCENAME(SBATMACC)
RESOURCETYPE(PROGRAM)
ATOMTYPE(FEED)
CONFIGFILE/u/fabrice/itso/feed/acc_feed.xml)
BINDFILE()

PROGRAM(SBATMACC)
```

```
<?xml version="1.0"?>
<cics:atomservice type="feed" ...>
   <cics:feed>
     <cics:resource name="SBATMACC" type="program">
       ...
     </cics:resource>
   </cics:feed>
   <atom:feed>
     <atom:link rel="self"
              href="/Showcase/account/feed"/>
      ...
   </atom:feed>
</cics:atomservice>
```

Figure 9-2 Resources configured for account transaction ATOM feed

TCPIPSERVICE resource

We used the same TCPIPSERVICE that we created for the terminal location feed.

URIMAP resource

The URIMAP maps the URI specified in the HTTP request to the ATOMSERVICE that is to be used to service the request. The URIMAP specifies USAGE(ATOM) and the name of the ATOMSERVICE ACCFEED. The PATH /Showcase/account* matches the URIs in the ATOMSERVICE config file.

ATOMSERVICE resource

The ATOMSERVICE defines the feed that CICS delivers in response to the HTTP request. It specifies the CICS resource type (PROGRAM) that provides the data for the feed and the location of the Atom configuration file that defines the feed document (acc_feed.xml).

PROGRAM resource

CICS calls the SBATMACC service routine to provide information from the DB2 database records in response to a feed request.

Example 9-6 shows an example entry from the account transaction feed.

Example 9-6 Account transaction feed entry

```
<entry>
   <link rel="self"
href="http://9.212.128.20:20000/Showcase/account/2009-12-0815.34.57SAIN
S001"/>
   <id>4988********85692009-12-0815:34:57SAINS001</id>
   <title>Retail sale in non-specialised stores with food, beverages or
tobacco</title>
   <summary>4988********8569</summary>
   <updated>2009-12-08T15:34:57Z</updated>
   <published>2009-12-08T15:34:57Z</published>
   <content type='application/xml'>
      <SBTRANSD

xmlns='http://www.ibm.com/xmlns/prod/cics/atom/Showcase/transdetails'>
         <amount>00000000007945</amount>
         <termid>SAINS001</termid>
      </SBTRANSD>
   </content>
</entry>
```

Security

Obviously security is a concern when exposing account information in any format, so we want to make sure that these new feeds are secure.

Because it is unlikely that each customer has a CICS user ID, it is impractical to rely on basic authentication at the CICS level to protect the feeds. It is more likely that users will log on to a portal server that retrieves the feeds from CICS on their behalf. Using a portal server in this way enables us to set up the following security model:

- Setup a secure connection between the portal server and CICS, for example, by using SSL client authentication or a virtual private network (VPN).
- Authorize user access in the portal server so that users can only obtain account data for their own accounts.

There is also some scope here to cache feeds in the portal server meaning that we can limit the number of requests into CICS to meet our performance needs.

9.4 Creating the GPS-enabled mobile application

In this section, we explain the basic principles behind the Android mobile application that the Web 2.0 application programmer creates to consume the showcase Atom feeds. This is not a tutorial about how to create an Android application because that is beyond the scope of this book. What we do cover are the general principles behind the application and how it makes use of the Web 2.0 information sources that are supplied using the Atom Publishing Protocol from the showcase.

Figure 9-3 shows the Android mobile phone onto which we deploy our application.

Figure 9-3 Android mobile phone

9.4.1 Web 2.0 application developer role

The task of the Web 2.0 application developer is to create the Web 2.0 application, which consumes the terminal location and account transaction Atom feeds. In this section, we provide an outline of the application.

Consuming a feed

One of the main tasks of the application is to read a feed and store its contents. To do this the application simply gets the feed XML, parses it, and extracts the data for each entry to store in its database. If an entry has the same <id> as one that it already stored, it updates that entry rather than storing a new one because <id> is unique to each entry. The entry also takes note if the feed supplied a next link, which tells us there are more entries to get.

It is rare to find a feed that returns its entire list of entries in one go, and CICS is no different. Feeds are returned in small pages that have links to the next and previous pages. Our application continues to follow the next link and stores the entries until no next link is returned or until we see an entry where the <updated> tag has a time stamp that is before our last update. Because the Atom protocol specifies that entries are returned in their order of update, with the most recent first, we can be sure at this point that there are no more updates that we are not aware of. In this way, the application usually only has to get the first page of the feed each time it polls for updates, which makes the whole process much more efficient.

Mashing the feeds with GPS data

Our application consumes two types of feed: the Account Transaction feed and the Terminal Location feed. It mashes the data in these feeds by using the Terminal ID from each transaction to lookup location data from the corresponding terminal. This data can then be used to determine the location at which the transaction took place.

The feed of terminals from the showcase do not include every terminal in the world, but the mobile application can consume multiple terminal feeds from a number of sources to maintain a more complete list. The name and location of terminals (ATMs or point-of-sale devices) that are not part of a Banks own network typically arrives with the Interchange (VISA, Mastercard, and so on) standard message. We assumed that in the future this includes GPS coordinates, or that this information can be added. So we can imagine a more or less static feed that lists the terminals that form part of the Banks network, and then a second feed that is a more transient listing coming from the Interchange messages.

When we know where a transaction took place, we can compare that location to the location of the mobile device at the time and date of the transaction by using the GPS history data. Because most people now carry their phones wherever they go, it is probably fairly safe to say that if the two points are sufficiently far apart, the application flags possible fraudulent behavior to the customer so that they can take action immediately. The user also has the option to easily remove flags from transactions if they recognize them as legitimate.

After the application is finished, it can be deployed to the phone. Figure 9-4 shows a shortcut to the application on the Android home screen.

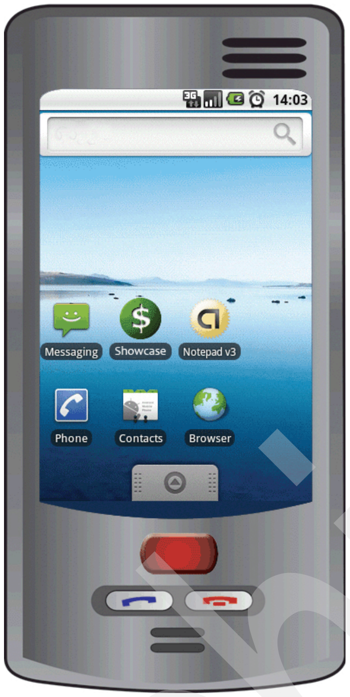


Figure 9-4 Showcase application deployed to phone

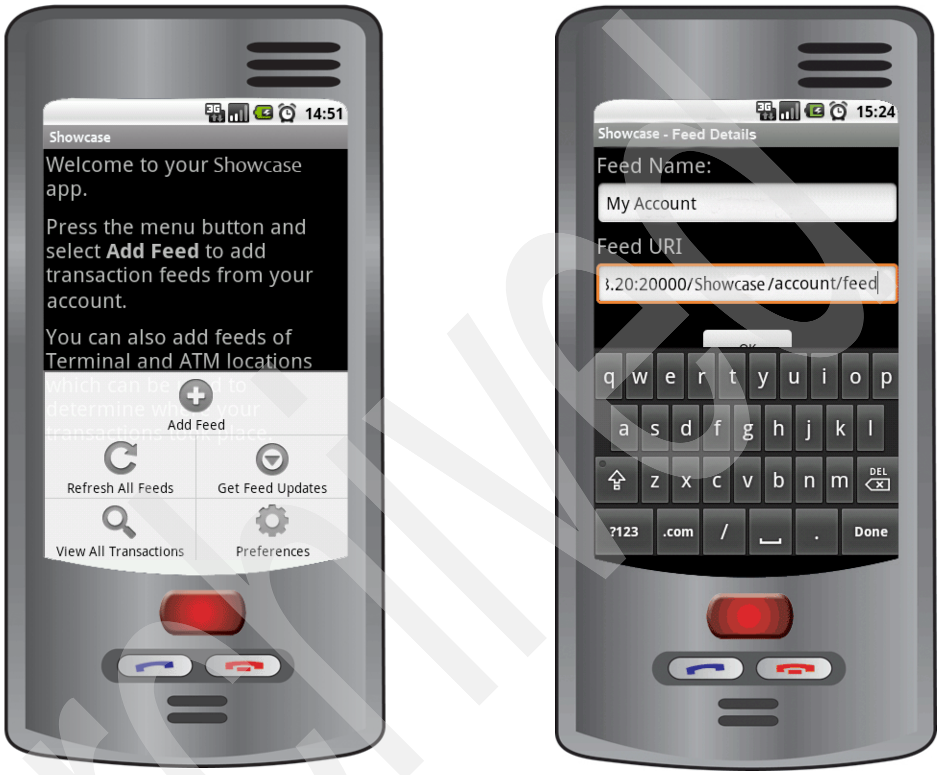
9.5 Testing the scenario

In this section, we show how we tested the Web 2.0 scenario. First, we show that the mobile application can consume the feeds, that data can be viewed correctly, and that location data from the terminal feed is being correctly mashed with the transaction data. Then, we show that account updates are received correctly though the polling mechanism. We then test that transactions over a threshold amount, set by the user, are flagged as large transactions. Finally, we test that transactions that occur in odd locations are flagged as suspicious transactions.

9.5.1 Viewing account data

Account data can be viewed through the Web 2.0 application using the new Atom feeds as a data source.

When running the application for the first time it is necessary to input the details of our two feeds. Figure 9-5 shows the initial application Welcome screen and the add new feed screen. On the add new feed screen, we see the URI that we specified in the account transaction feed config file in Example 9-4 on page 175.


14:51

Showcase

Welcome to your Showcase app.

Press the menu button and select **Add Feed** to add transaction feeds from your account.

You can also add feeds of Terminal and ATM locations which can be used to determine where your transactions took place

15:24

Showcase - Feed Details

Feed Name:

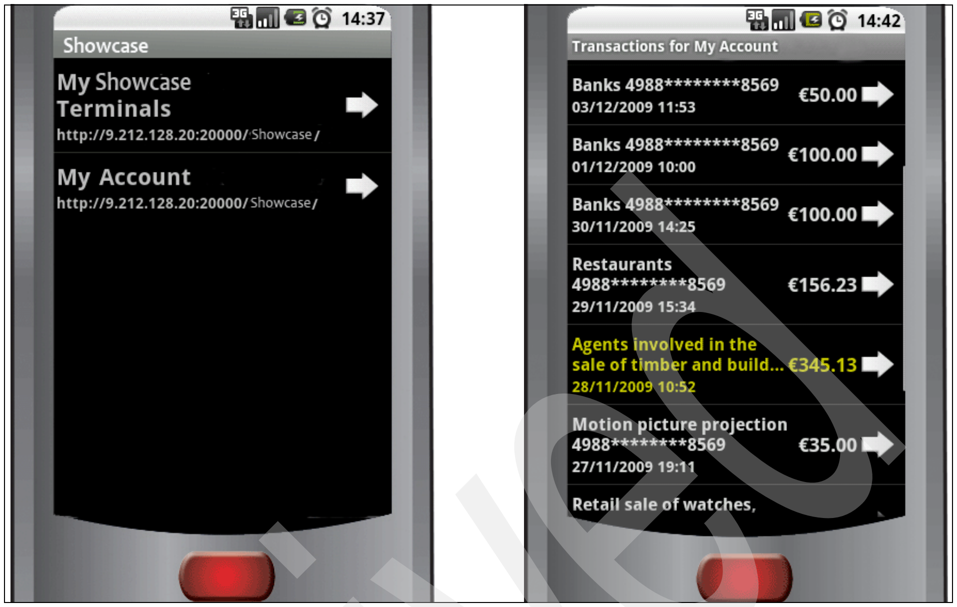
My Account

Feed URI

3.20:20000/Showcase/account/feed

Figure 9-5 Initial welcome screen (left) and the add new feed screen (right)

After the feeds are added, we no longer see the Welcome screen; instead, we see a list of all of the feeds that we added. If we touch our account feed, we can see the list of transactions for the account from the CICS DB2 table (see Figure 9-6 on page 184). We can also see the list of terminal locations by touching the terminal's feed. Figure 9-7 on page 184 shows the list of terminals from the CICS VSAM KSDS file and details for one particular terminal.



Showcase

My Showcase Terminals
http://9.212.128.20:20000/Showcase/

My Account
http://9.212.128.20:20000/Showcase/

Transactions for My Account

Banks 4988********8569 €50.00
03/12/2009 11:53

Banks 4988********8569 €100.00
01/12/2009 10:00

Banks 4988********8569 €100.00
30/11/2009 14:25

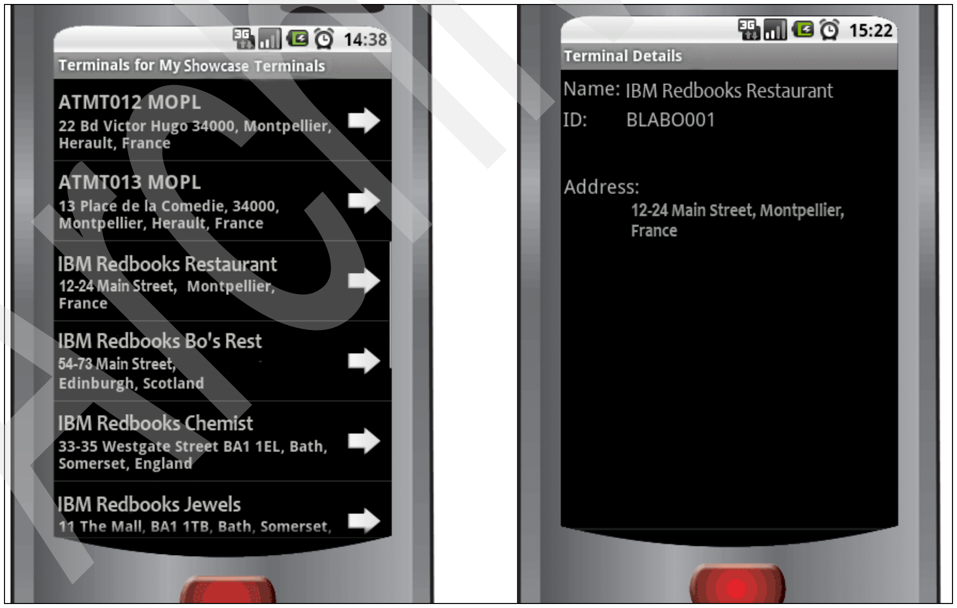
Restaurants
4988********8569 €156.23
29/11/2009 15:34

Agents involved in the
sale of timber and build... €345.13
28/11/2009 10:52

Motion picture projection
4988********8569 €35.00
27/11/2009 19:11

Retail sale of watches,

Figure 9-6 List of feeds (left) and list of transactions, from a CICS DB2 table (right)



Terminals for My Showcase Terminals

ATMT012 MOPL
22 Bd Victor Hugo 34000, Montpellier, Herault, France

ATMT013 MOPL
13 Place de la Comedie, 34000, Montpellier, Herault, France

IBM Redbooks Restaurant
12-24 Main Street, Montpellier, France

IBM Redbooks Bo's Rest
54-73 Main Street, Edinburgh, Scotland

IBM Redbooks Chemist
33-35 Westgate Street BA1 1EL, Bath, Somerset, England

IBM Redbooks Jewels
11 The Mall, BA1 1TB, Bath, Somerset,

Terminal Details

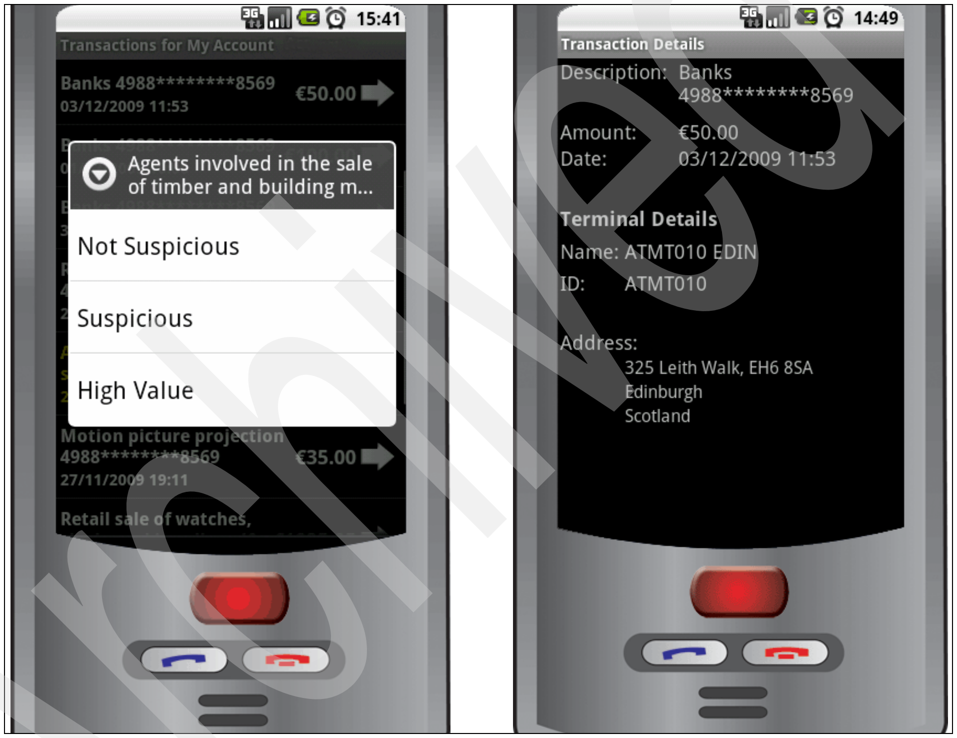
Name: IBM Redbooks Restaurant
ID: BLABO001

Address:
12-24 Main Street, Montpellier, France

Figure 9-7 List of terminals, from a VSAM file (left) and single terminal record (right)

In Figure 9-6 on page 184 there is a transaction highlighted in yellow because it is over 200.00, and this is what the user set as their threshold for flagging large transactions (there is a preferences menu for this). This flag can be easily removed by touching and holding the transaction to bring up a context menu, which is demonstrated on the left of Figure 9-8.

To get more details about a transaction the user simply touches the transaction of interest, which we show on the right of Figure 9-8 where we can also see how the transaction data was mashed with the terminal feed to provide location data for this transaction.


15:41
Transactions for My Account
Banks 4988********8569 €50.00
03/12/2009 11:53
Agents involved in the sale of timber and building m...
Not Suspicious
Suspicious
High Value
Motion picture projection 4988********8569 €35.00
27/11/2009 19:11
Retail sale of watches,
14:49
Transaction Details

Description: Banks 4988********8569
Amount: €50.00
Date: 03/12/2009 11:53

Terminal Details
Name: ATMT010 EDIN
ID: ATMT010

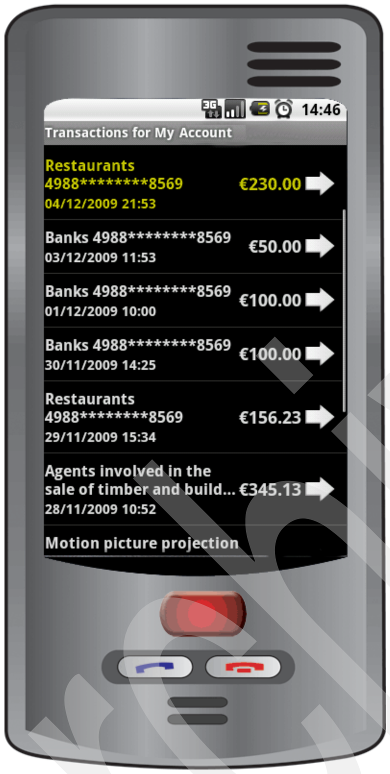
Address:
325 Leith Walk, EH6 8SA
Edinburgh
Scotland

Figure 9-8 Change flag menu (left) and details of a transaction location (right)

9.5.2 Large transactions

In this section, we show that the application can flag unusually large transactions to bring them to the customer's attention. In this section, we also demonstrate that the application can receive account updates through its polling mechanism.

When a customer performs a new large POS transaction, a row is added to the DB2 database in the Banking Data Warehouse for his account. The next time that the Web 2.0 application polls the account feed it will receive this new entry and, upon seeing that the amount is over the 200.00 euros threshold, the application flags it as a high-value transaction (yellow text) to bring it to the attention of the customer. Figure 9-9 shows the addition of the high-value transaction to the top of the transaction list.



Transactions for My Account

Restaurants 4988********8569 €230.00
04/12/2009 21:53

Banks 4988********8569 €50.00
03/12/2009 11:53

Banks 4988********8569 €100.00
01/12/2009 10:00

Banks 4988********8569 €100.00
30/11/2009 14:25

Restaurants 4988********8569 €156.23
29/11/2009 15:34

Agents involved in the sale of timber and build... €345.13
28/11/2009 10:52

Motion picture projection

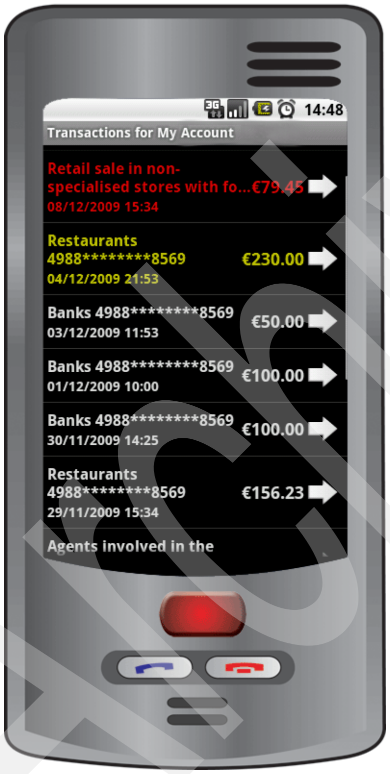
Figure 9-9 High-value transaction from the DB2 database

The threshold value is set in the preferences menu. A flagged large transaction can be reset by touching and holding the transaction data to bring up the context menu, as shown in Figure 9-8 on page 185.

9.5.3 Suspicious transaction location

When a transaction occurs at a location that does not match the phone's historical GPS location for the time of the transaction, the transaction is flagged as suspicious to bring it to the customer's attention.

For this example, we perform a new POS transaction for this account at a terminal in Winchester when the phone is in Edinburgh. The transaction results in a new row in the BDW, which the application receives as a new entry the next time it polls the feed. The application performs a lookup of the transaction's Terminal ID against its list of known terminals, acquired from the terminal feed. When it finds a match, it can use the terminal's location as the location of the transaction. Using its GPS history data, it compares the transaction's location to the phone's location at the time of the transaction. If the distance between the two locations is more than a couple of kilometers, it flags the transaction as suspicious (red text) to bring it to the customer's attention. Figure 9-10 shows the addition of the suspicious transaction to the top of the transaction list.



Retail sale in non-specialised stores with fo... €79.45
08/12/2009 15:34

Restaurants 4988********8569 €230.00
04/12/2009 21:53

Banks 4988********8569 €50.00
03/12/2009 11:53

Banks 4988********8569 €100.00
01/12/2009 10:00

Banks 4988********8569 €100.00
30/11/2009 14:25

Restaurants 4988********8569 €156.23
29/11/2009 15:34

Agents involved in the

Figure 9-10 Suspicious transaction from the DB2 database

9.6 Summary

In this chapter, we showed how CICS support for Web 2.0 can be used to provide a richer user experience for the showcase customers by providing real-time feeds of account transactions and ATM locations to mobile devices.

Related publications

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

IBM Redbooks

For information about ordering these publications, see "How to get Redbooks" on page 190. Note that some of the documents that are referenced here might be available in softcopy only:

- Implementing Event Processing with CICS, SG24-7792

Other publications

These publications are also relevant as further information sources:

- *Achieving business agility with BPM and SOA together* (WSW14078-USEN-00), IBM, 2009. Authors: Claus Torp Jensen, Rob High and Steve Mills.
- *Global Financial Stability Report (GFSR)." International Monetary Fund,* 2009.
- *IBM / CFA Institute Survey 2009,* IBM Institute for Business Value.
- *Fit, focused and ready to fight,* IBM Institute for Business Value*, 2009.*
- *Undressing in public: Harnessing the power of Web 2.0 to rebuild trust in banking,* IBM, *2009.*
- *Why to choose CICS Transaction Server for new IT projects,* IBM, *2008.* Authors: Andrew Bates and Timothy Sipples.
- *Service-oriented architecture: Revolutionizing today's banking systems, IBM Institute of Business Value,* 2009. Authors: Jay DiMare and Richard S. Ma.
- *Design an SOA solution using a reference architecture,* IBM developerWorks. Authors: Ali Arsanjani, Liang-Jie Zhang, Abdul Allam, Michael Ellis, et al..
- *SOA Reference Architecture, Draft Technical Standard,* The Open Group, 2009. Authors: The Open Group.

- *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry*. Author: Martin Campbell-Kelly.
- *Modern Information Management: CICS Transaction Server 4.1 helps organizations compete, comply and control*, IBM Systems Magazine. Author: Nick Garrod.

Online resources

These Web sites are also relevant as further information sources:

- For further information about CICS Transaction Server:

  http://ibm.com/cics/tserver/v41/

- For further information about WebSphere Business Events:

  http://www.ibm.com/software/integration/wbe/

- For further information about the Android operating system:

  http://developer.android.com
  http://www.helloandroid.com
  http://www.anddev.org

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 publications, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

Smarter Banking with CICS Transaction Server

(0.2”spine)
0.17”<->0.473”
90<->249 pages

Smarter Banking with CICS Transaction Server

Unlock business data in the core banking system

Discover CICS TS V4.1 support for business events and Web 2.0

Learn about the IBM Smarter Banking Showcase

It goes without saying that 2009 was a year of unprecedented change in global banking. The challenges that financial institutions are facing require them to cut costs but also to regain trust and improve the service that they provide to an increasingly sophisticated and demanding set of customers.

In the past, siloed and rigid IT systems often inhibited banks in their attempts to re-engineer their business processes. The IBM smarter banking initiative highlights how more intelligent software can be used to significantly improve the end-to-end integration of banking processes.

In this IBM Redbooks publication, we aim to show how software technologies, such as SOA, Web 2.0 and event driven architectures, can be used to implement smarter banking solutions. Our focus is on CICS Transaction Server, which is at the heart of most bank's core banking implementations.

SG24-7815-00 ISBN 0738434124