



**Composing CICS applications
into business services to support
On Demand Business solutions.**

Contents

2 Executive summary

3 The need for flexibility in IT

4 An opportunity— and a challenge

4 The route to reuse

6 CICS Transaction Server and composing Web services

7 Other important CICS Transaction Server, Version 3.1 features

8 Using the J2EE Connector Architecture to connect to CICS

9 SOA basics

10 The standard SOA architecture

11 Composing Web services from CICS applications

16 Building an application with the CICS Web services assistant

16 Optimizing CICS applications for modern programming models

19 Summary

19 For more information

Executive summary

Simple economics and competition put tremendous pressure on IT departments to extract maximum value from their organizations' existing applications. The quickest route to exploiting this value is through reuse – composing IBM CICS® applications into business services, so they can be used again and again – in new business processes.

For CICS users, never before has the potential for their CICS systems to add business value been so great. The convergence of CICS innovation and robust service oriented architecture (SOA) capabilities can help you build a flexible information infrastructure – while preserving the CICS qualities of performance and reliability that your enterprise depends on.

Using readily available tools and technologies, you can productively transform your existing CICS transactions into versatile program units that you can combine in different ways to rapidly build new business processes. You can derive compelling benefits from this approach in a number of ways:

- *Gain extra value from your existing assets by putting them to use in new programming without having to write new code.*
- *Combine (compose) these versatile program units into business services that lend themselves to rapid change.*
- *Perform all of these tasks with readily available tools and minimal risk.*

This white paper describes how SOA and CICS technologies have converged to create the opportunity to build new business processes – and how you can take advantage of this opportunity. Because these SOA technologies are all based on open standards, your CICS applications have the ability to work with virtually any other application that conforms to the same standards. And you don't need specific CICS skills. You can build new processes using integration programmers, for example, freeing your prized CICS skills to focus on where they're most needed.

Making the move from conventional CICS operations to flexible reuse can be a lot easier than you might think. This white paper is designed to show you how you can apply reuse capabilities to tap into the full potential of your CICS applications.

The need for flexibility in IT

Business managers have often had to ruefully delay their plans because their IT systems had to be laboriously reworked to meet the latest business initiative. The demands for a more flexible approach for aligning IT capabilities with business strategy are well-known. Perhaps the most important driver is simple economics. As the marketplace globalizes, new markets, new work forces and new competitors force companies to look for ways to adapt more quickly. And cycle time continues to shrink between business-process changes. Whereas in the past, companies might have made significant changes yearly, today you might find the same level of changes being made on a monthly, or even weekly or daily basis.

Although business leaders have been focused mostly on cost-containment in recent years, growth is back at the top of today's CEO's agenda – and that growth demands the flexibility to be more nimble than competitors. This new focus on growth doesn't diminish the need to keep costs in check. On the contrary, businesses are looking for ways to make better use of the investments that they already have.

Flexibility in both business model and IT architecture is crucial to become what IBM calls an *On Demand Business*. An On Demand Business is an enterprise whose business processes – integrated end to end across the company and with key partners, suppliers and customers – can respond with speed to any customer demand, market opportunity or external threat.

Two different perspectives can deliver the required flexibility:

- *Business processes that can be readily adapted to suit changes in the business model*
- *Application components that can be readily reassembled to deliver the capabilities required*

The term *composition* describes the ability to build new or different processes from proven subprocesses, and to aggregate different components into new applications. Composable applications and processes are the key to moving toward an On Demand Business model.

An opportunity—and a challenge

In many ways, CICS users are in an enviable position. You have a rich profusion of proven applications that you depend on to run your business reliably and securely. You regard your inventory of highly valuable critical business applications as a major business asset. But it is likely that your CICS applications are locked into processes that prevent you from redeploying them in a flexible fashion. And that's because of common industry challenges that inhibit the flexibility that promotes growth.

It's worth emphasizing again that a business can be only as flexible as the IT systems that support it. And, in a typical CICS user environment, the totality of these IT systems can be very sophisticated. This level of sophistication – or complexity in this case – does not lend itself well to supporting a flexible business.

What leads to this kind of situation? When there are no common standards or best practices to guide business processes, they tend to grow out of control. The same thing happens when architectural policies and application purchases are focused on solving specific, one-time issues without considering long-term needs. The result can be a tangled set of hard-wired application connections that are very difficult, expensive and time-consuming to unravel when it's time to implement changes. So, paradoxically, the continued fine tuning of core enterprise application assets over the years that has made them so valuable, also complicates and constrains flexible reuse.

The route to reuse

A first priority when considering how to gain flexibility is to preserve the CICS qualities of service that you've become accustomed to and depend on. When incorporating your valuable CICS assets into On Demand Business solutions, you want to balance reduced cost, risk and delivery time against the performance, reliability and security that characterizes CICS systems.

Web services provide an answer. Web services are self-contained, modular, dynamic applications that can be described, published, located or invoked over a network to create products, processes and supply chains. They can be local, distributed or Web-based, and they're built on open standards such as TCP/IP, HTTP, Java™ technology, HTML and XML. To put it simply, Web services are building blocks that you can connect in almost any way that you like to build the processes and capabilities that you need to run your business smoothly.

Another consideration might be to exploit Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA). JCA is supported by the IBM CICS Transaction Gateway and offers a more tightly coupled connectivity method that can coexist and complement a Web services approach.

The crucial component in Web services is that they use open standards. Effectively, open standards shield you from the need to know anything about the underlying technologies. A piece of program code can be wrapped as a Web service, and then accessed by developers to rapidly build open-standards-based applications independent of the CICS business logic program that they will interact with. In other words, you can connect functions and applications without having to know how each is implemented. The applications know how to “talk” to each other behind the scenes, because information is provided in a standard format, using standard protocols.

For example, Web services could link key business functions within the applications that a business shares with its customers, suppliers and business partners to interact with high security and reliability, independent of platform, environment or application language. This ability to plug in a Web service anywhere makes the architecture loosely coupled, and hence, very flexible.

Implementing a JCA adapter is an open-standards approach that defines a standard programming method for implementing connectivity. It requires an understanding of the interface formats exposed by the target system – in this case, a CICS application communications area (COMMAREA) – balanced by the advantages of more-specific transactional and security coupling.

This flexibility is the reason that Web services are so valuable for processes. Businesses are focusing on how to drive efficiencies and clarity by better defining their business processes – especially in Web-based services. To take advantage of Web services, you need an SOA, which is an implementation-independent approach for providing business services that are standards based, modular and inherently adaptable. It isn't a rigid architecture or a set of specific products – rather, it's a set of architectural principles and standards that help you to take advantage of open-standards-based services. An SOA identifies and then loosely couples the various Web-based functions and services required to enable the processes.

If you could transform your CICS applications into Web services, you could extend their value considerably. Not only could you have the ability to reuse their functions wherever you wanted, but you could transfer the same qualities of service, and save the effort and risk of developing new code.

CICS Transaction Server and composing Web services

Perhaps the most significant enhancement in IBM CICS Transaction Server for z/OS, Version 3.1 is its new role as a Web services end point. You can now integrate CICS technology-based applications with an SOA, enabling them to be exposed as Web services – with all the benefits described previously in this white paper.

Another major change is that CICS Transaction Server, Version 3.1 can now act as both a Web services provider – playing the classic role of a host for business logic where services are evoked on CICS Transaction Server – and as a Web services requester, so IT personnel can extend and deeply integrate CICS technology-based business logic into the fabric of their overall architecture. As a result, CICS applications can call Web services wherever they're hosted – even on platforms such as J2EE or Microsoft® .NET.

This dual provider-requester role means that CICS Transaction Server can now be a full participant in new business processes. With CICS applications wrapped and exposed as services, applications can become interoperable, helping to enable virtual enterprises to link heterogeneous systems as required. A newly merged company, for example, can integrate disparate IT systems and business processes from its constituent organizations. Or a travel application can be exposed as a service and made available for use by various devices in a service-oriented environment. One of the most important features of CICS Transaction Server, Version 3.1 is that it supports the industry-leading Secure Sockets Layer (SSL) security protocol. This support enables fine-tuned control of network security, and provides a faster, more comprehensive security solution by exploiting advanced IBM z/OS® security features.

The CICS service-flow feature is a business-service integration adapter for all CICS applications. It delivers the tooling and run time to build CICS business services so that you can create new services more quickly and easily by composing a sequence of CICS application interactions. The high-level, business-function interfaces to the services allow you to integrate them in many modern solution styles, including SOA. The CICS service-flow feature includes the CICS service-flow run time, along with ten licenses to IBM WebSphere® Developer for zSeries for composing CICS business services. The licenses enable you to install and run WebSphere Developer for zSeries functions, including the service-flow modeler, XML Services for the Enterprise (XSE) and XML tools, and Web services capabilities.

Using the CICS service flow feature can help you unlock critical IT assets and either invoke them with a JCA adapter or reuse them as part of your SOA – a far more efficient process than creating new applications.

Other important CICS Transaction Server, Version 3.1 features

The focus of CICS Transaction Server, Version 3.1 is to deliver a set of capabilities that provide value by enabling business flexibility through IT simplification. These capabilities are represented in the following categories:

- *Application transformation enables you to enhance existing applications and build new applications using contemporary programming languages, constructs and tools.*
- *Enterprise management enables you to effectively manage large runtime configurations using modern user interfaces, so that you can meet demanding service-level objectives.*
- *CICS integration (the primary subject of this white paper) enables you to reuse CICS applications, within a flexible IT infrastructure, using standard application programming interfaces (APIs) and protocols.*

These categories also align with the building blocks of SOA – development, infrastructure and management.

Using the J2EE Connector Architecture to connect to CICS

A J2EE application, or an enterprise service bus (ESB) based on J2EE technology, such as IBM WebSphere ESB or IBM WebSphere Message Broker, can also connect to CICS Transaction Server using the JCA. The JCA defines a standard architecture for connecting the J2EE platform to heterogeneous enterprise information systems (EISs), such as CICS Transaction Server. The underlying infrastructure of the JCA automatically manages the connection pooling, transactional scope and security qualities of the composite applications, so that these capabilities do not have to be individually coded into each application – enabling application developers to concentrate on the development of business logic rather than on quality-of-service provisioning.

CICS Transaction Gateway software provides an implementation of the JCA that has been proven over many years to deliver high-performing, security-rich and scalable access to CICS applications. Implementing CICS Transaction Gateway software requires minimal changes to CICS systems and usually no changes to other existing CICS applications, enabling rapid exploitation of existing enterprise applications.

JCA, along with other J2EE standard services such as Java Message Service (JMS) and Java Database Connectivity (JDBC), is a tightly coupled connectivity method. Tightly coupled direct connections and loosely coupled Web services can coexist to fully exploit the agility of an On Demand Business environment, and together they enable you to integrate all of your CICS assets in an enterprise-class SOA.

SOA basics

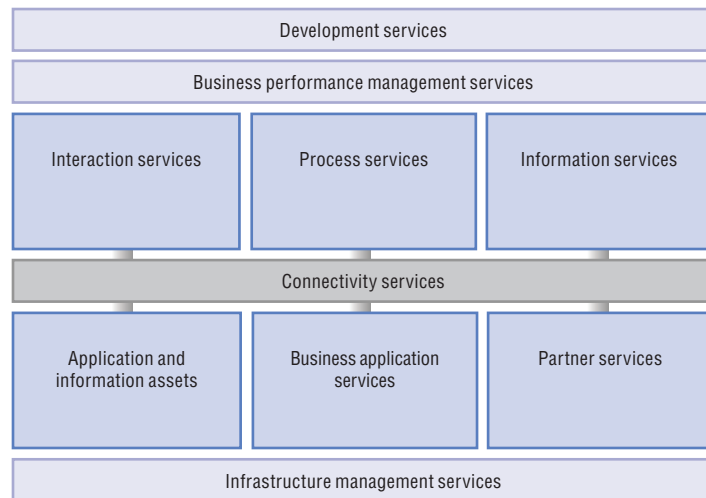
SOA has become the accepted term for the approach of using business services to build a standardized integration infrastructure. *SOA* helps move an enterprise toward becoming an On Demand Business by providing a high degree of flexibility.

An ESB is a pattern for *SOA* implementation that provides access between services in a consistent style, regardless of their technology implementations. As well as transport and event services, an ESB also provides the mediation services that enable loose coupling between interacting services in the system. An ESB plays a key role in the service orientation of the architecture, and you could implement it today to help you meet the quality-of-service requirements of any integration solution.

The concept of composing applications and processes – so fundamental to programming models – is made possible by using standardized communications transports and structured data formats across an ESB, in a style that helps ensure that platform-specific implementation details are encapsulated behind well-defined service interfaces. Web Services Definition Language is the technology that is recommended for describing the business services in the *SOA*, and SOAP is one protocol that can be used to implement exchanges between business services.

The standard SOA architecture

The IBM business integration reference architecture (shown in the following figure) is a technical framework for enterprise transformation that enables software to be delivered as reusable, shareable services. Essentially, it's a reference architecture for SOA implementations. It provides you with the ability to bridge disparate systems spread across your entire enterprise. And because its components are modular, you can start small and grow your implementation to cover your evolving integration needs, both internally and externally.



The IBM business integration reference architecture

The IBM business integration reference architecture shows the key areas of integration capability that are required for comprehensive, enterprise-wide strategies and solutions. Tools are an essential component of any comprehensive integration architecture. The IBM business integration reference architecture includes development tools that can be used to implement custom artifacts that exploit infrastructure capabilities, and business performance management tools that can be used to monitor and manage the runtime implementations at both the IT and business-process levels.

Business analysts who evaluate business-process requirements need modeling tools that enable business processes to be charted and simulated. *Software architects* need tool perspectives that enable them to model data, functional flows, system interactions and so on. *Integration specialists* require capabilities that enable them to configure specific interconnections in the integration solution. *Programmers* need tools that enable them to develop new business logic with little concern for the underlying platform. All of these classes of tools are available within the CICS portfolio of tools, or in associated areas. They are described in more detail in the following section.

Composing Web services from CICS applications

IBM has a complementary set of CICS and WebSphere tools that make it easier to build and deploy new composite applications from new and existing programs.

Three different roles can be involved with composing new services:

- Integration developers *design and develop new processes from reusable components. They have to understand the inputs and outputs from existing components and the requirements for code snippets to create the flow. They work with a component developer to identify existing components and to specify requirements for new components.*
- Component developers *create individual software modules that are intended to be integrated into and reused across multiple applications. They identify whether existing applications can be used to create the necessary component, and work with application developers to identify existing components and applications, and to specify new applications.*
- Application developers *deliver new functional components, coding applications to specifications provided by an integration developer (or solution architect).*

It is possible for one person to handle more than one of these roles. And the individuals in these roles can also work in concert with architects and business analysts. Regardless of who is involved, IBM clients have indicated that they think about SOA in terms of a life cycle. They start in the *model* phase by gathering business requirements and designing their business processes. After processes are optimized, they *assemble* them by combining new and existing services to form these business processes. They then *deploy* these assets into a highly secure and integrated services environment. After the business processes are deployed, IBM clients *manage* and monitor these business processes from both an IT and a business perspective. Information gathered during the manage phase is fed back into the life cycle to enable continuous process improvement. Underpinning all of these life-cycle stages is *governance* that provides guidance and oversight for the SOA project.

Model

One of the key attributes of an SOA is the ability to reuse existing program assets. This process starts with discovering which program assets you already have that you can reuse in new applications. IBM WebSphere Studio Asset Analyzer helps you discover these hidden assets by determining which programs are good candidates for reuse in Web applications based on the number and type of changes required.

After you understand what business operations you already have coded and available, you can use IBM WebSphere Business Modeler software to create end-to-end process models that represent key business processes throughout the enterprise, some of which are represented by the assets discovered by WebSphere Studio Asset Analyzer. WebSphere Business Modeler helps you understand and transform your business through business modeling, simulation, analysis and collaboration capabilities. You can use the models you build with WebSphere Business Modeler to direct your assemble and deploy activities to create composite applications.

Other tools can also help you to reuse your existing applications. Many enterprises use CICS Transaction Server to run their core business, and they have large amounts of reliable and trusted CICS code. IBM CICS Interdependency Analyzer helps you understand, preserve, evolve and integrate existing CICS applications with new application models that can help you achieve your On Demand Business goals. And IBM Asset Transformation Workbench helps you identify the business rules coded within your core applications, and helps in restructuring large applications into more-manageable segments, helping to remove dead code.

At the end of the modeling phase, you should have a clear idea of the assets that you have available, and where they can be used in the new business processes that you have modeled. The next phase is to create services and assemble them into deployable composite applications.

Assemble

WebSphere Developer for zSeries, Version 6 includes capabilities that help make traditional mainframe development, Web development and composite-application development faster and more efficient. If you use CICS, IBM IMS™ or WebSphere transactional environments, WebSphere Developer for zSeries simplifies the development of new Web user interfaces (WUIs), traditional terminal interfaces and back-end business logic. You can wrap your CICS and IMS transactions as Web services, and because so much business is transacted using CICS applications, you can model and reuse core assets from CICS systems using the WebSphere Developer for zSeries service-flow modeler. This feature models and renders a wide variety of CICS transactions, such as 3270 and COMMAREA transactions, into callable Web services.

IBM WebSphere Integration Developer, Version 6 is a new tool designed to enable rapid assembly and testing of business solutions. It helps enable you to construct a range of processes with one programming model based on Business Process Execution Language (BPEL). Because WebSphere Integration Developer requires relatively minimal skills, and provides easy-to-use authoring tools, it can help increase developer productivity. When you're happy with your composite-application model, you can deploy it to IBM WebSphere Process Server in the next stage.

WebSphere Integration Developer combines well with other development tools from IBM, including IBM Rational® Application Developer and WebSphere Business Modeler, in the model phase of the SOA life cycle.

Criminal justice service: Building an enterprise SOA using CICS systems

Thousands of organizations run CICS mainframe technology. And although they wouldn't think of replacing it, they would readily consider how they can use this valuable CICS functionality across other enterprise systems, just as a North American government agency did.

This agency realized that its underlying CICS system supported its functional needs, but the user interface terminals were inappropriate for today's needs. To update the technology used to access its criminal justice system, it decided to redesign the interaction model using a portal-based Internet interface. The new solution provides easier access and more efficient interaction to a much broader audience, which helps reduce training costs and helps support a growing number of users. And the agency can continue to take advantage of the proven functionality of its existing CICS system, and is spared the time and expense of rewriting this logic on another platform.

Deploy

WebSphere Process Server, Version 6 is the runtime engine that controls the implementation of all the Web services created in the previous steps, and that are combined using the WebSphere Integration Developer tool. Using WebSphere Process Server is the culmination of discovering assets and business rules, bringing them into a model of a business process, and then plugging individual implementable Web service components into the model. WebSphere Process Server deploys the composite applications, and controls the implementation of the process, choreographing the individual programs that make up the process into an automated flow. The runtime process is very flexible. For example, you can replace a human task (like an approval) with a business rule without changing anything else in the entire solution. This is an extremely powerful capability that can fully deliver on the promise of SOA to provide greater flexibility and responsiveness to your business.

This flexibility and responsiveness are possible because of the ESB. WebSphere Message Broker is IBM's advanced ESB, and has the capability to handle connectivity to virtually any application, regardless of whether it is based on open standards. This connectivity means that you can choreograph practically any of your applications through WebSphere Process Server, which can be particularly beneficial if you have a large base of trusted existing applications, for example, running in an IBM @server® zSeries® environment. The power of WebSphere Message Broker is that it can mediate and transform messages in flight to meet the needs of the receiving applications, offloading the mediation processing to a zSeries Application Assist Processor (zAAP), with an immediate benefit in performance and throughput.

Criminal justice service: Building an enterprise SOA using CICS systems (continued)

SOAP for CICS technology was used to access the functionality hosted on these critical CICS systems. SOAP interfaces were defined in Web Services Description Language (WSDL) using tools provided with CICS Transaction Server, and these WSDL definitions were used to create a Java technology-based SOAP proxy that can be called within the agency's Internet portal solution.

CICS compliance with SOA made it straightforward to reuse existing components and avoid risky development of new functionality. The result is that legal professionals have quicker, easier access to criminal justice information through a single, integrated view of information from both the CICS system and other sources.

Manage

IBM WebSphere Business Monitor, Version 6 measures the actual performance time of every process and process step. This capability provides accurate information about how the process is run by and for its users, and the knowledge you need to design a particular Web site, increase the capacity or any other changes that might benefit the majority of your users. Furthermore, because the information is dynamic, anything of interest – like a group of users having really long response times – can be investigated on the spot by clicking and focusing on a particular set of data. The answers are immediately available and presented in graphical form as well as in columns and text.

The real power of WebSphere Business Monitor is that you can load your real-time data back into it for analysis. The only way to optimize your processes is to run simulations and what-if scenarios using your real business data, which is what WebSphere Business Modeler does, completing the loop of continual improvement by enabling you to reuse existing assets.

SOA foundation cycle: Summary

The SOA life cycle shows a beneficial loop of continual improvement based on reuse of existing, trusted assets. IBM, through its CICS and WebSphere product portfolios, provides a complete set of tools that enable you to build dynamic new flexible processes that capitalize on your existing CICS applications, reusing them in new and profitable ways. All aspects of the development cycle, from identifying what existing assets you can use, through deploying them in new and flexible processes, to monitoring the efficiency of your new processes, are covered with easy-to-use and effective tools.

Building an application with the CICS Web services assistant

To simplify the transformation of an existing CICS COMMAREA application into a simple Web service with a single operation, CICS Transaction Server, Version 3.1 provides an application-development feature called the *CICS Web services assistant*. This feature supports COBOL, C/C++ and PL/I, helping to ensure that traditional program languages can participate and deliver immediate value in the Web services context.

The CICS Web services assistant provides the capability to map between a Web services interface described in WSDL and the language structure interface – that is, the COMMAREA interface implemented by a CICS application. This capability can be applied to derive WSDL from other languages that represent an existing language structure or to derive a new language structure from a WSDL service description.

The appropriate information is generated by the CICS Web services assistant to map the data between two CICS Web services, using a data-mapping service supplied by CICS Transaction Server to perform the translation of payload data between the pipeline and the CICS application. To help best practices, a new sample application illustrates how to code and implement a Web service application. This capability helps ensure that you can generate immediate value from the new capability.

Optimizing CICS applications for modern programming models

So far, this white paper has discussed how to connect to CICS Transaction Server to access applications flexibly. Now, it is going to describe how those applications can be optimized for the mode of access that modern programming models require.

Application transformation

In a traditional scenario, CICS applications were designed to be accessed from a 3270 terminal. The applications contain presentation logic that exploits interfaces provided by CICS Transaction Server to generate and interpret the screen content and to manage the control and interaction with the 3270 terminal.

The design point for the CICS application presentation logic was to deliver a subsecond response to the user, because user productivity was a key measure of effectiveness for the business system. To achieve this productivity improvement, the interaction with the business user on the terminal was optimized to exchange small amounts of information in quick succession – a fine-grained interaction pattern. Accordingly, to complete a business operation – like order fulfilment or bill payment – the user might have had to interact with many different application programs.

Although this works well when all the business operations are hosted in the same mainframe environment, typical IT infrastructures have grown to contain a rich set of capabilities in different environments for new processing needs. For example, a Web portal might be hosted on a J2EE application server such IBM WebSphere Application Server. Typically, solutions of this nature exchange data by way of XML data structures that are lengthier and more expensive to process than data streams optimized for a device like a 3270 terminal. This use of XML data structures affects the nature of the integration between a modern enterprise solution and an existing enterprise system, because it becomes much more efficient to minimize the number of interactions between the systems, suggesting a coarse-grained interaction pattern across the system boundary complemented by highly optimized processing of the request payload within the bounds of each system.

CICS supports the transports common in today's enterprise solutions, so the most efficient way of running CICS programs in new environments is to transform the behavior of the applications to coarse grained – while retaining the qualities of service that your enterprise depends on.

Composition

CICS Transaction Server has introduced adapter technology to allow existing applications to remain unchanged while being wrapped as Web services that can then be accessed by different means. The Link 3270 bridge is one example: it allows you to implement an adapter for a terminal application by writing new application components that emulate the terminal user input and actions.

You need tools to help transform and reuse your CICS applications in the most efficient way, so that you can encapsulate the existing application function while preserving the characteristics of the existing implementation, as well as helping to ensure the optimal coarse-grained approach. The ideal approach is independent of communication transport methods to help ensure that your implementation can be reused in the widest-possible way.

One way of aggregating CICS applications into a modern interface is to use IBM WebSphere Host Access Transformation Services (HATS). HATS enables you to provide programmed navigation through multiple CICS screens. For example, you can combine data from multiple host screens, as well as multiple data sources, into a single HATS screen. HATS delivers all you need to quickly and easily extend your existing applications to business partners, customers and employees. HATS makes 3270 and 5250 applications available as HTML through the most popular Web browsers. Just as important, you can improve the workflow and navigation of your host applications without any modification to source code.

Another approach is to use the WebSphere Developer for zSeries service-flow modeler to help make the most of your investment in existing enterprise information systems and the CICS run time, while maintaining quality of service and enabling the move towards SOA. You can use the service-flow modeler to model a new business service, or flow, by defining an interface and outlining implementation steps. It can help you capture existing interfaces to steps in the flow, map the data between elements in the flow and publish a business flow as a Web service.

Summary

Your existing CICS applications represent a rich seam of untapped potential for reuse. You can transform them straightforwardly into Web services and compose them into business flows to export all the qualities of service that you depend on into new environments. And the available tools can help you harvest the benefits without special skills, and with low risk.

For more information

To learn more about the new capabilities of CICS Transaction Server and CICS tools, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/cics

To learn more about the WebSphere tools that are described in this white paper, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/websphere



© IBM Corporation 2006

IBM United Kingdom Limited
Hursley Park
Winchester
Hampshire
SO21 2JN
United Kingdom

Produced in the United States of America
02-06
All Rights Reserved

CICS, @server, IBM, the IBM logo, IMS, the On Demand Business logo, Rational, WebSphere, z/OS and zSeries are trademarks of International Business Machines Corporation in the United States, other countries or both.

Microsoft is a trademark of Microsoft Corporation in the United States, other countries or both.

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

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