WebSphere® IBM Branch Transformation Toolkit for WebSphere Studio

IBM

Version 5.2

Product Overview

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 13.

This edition applies to Version 5, Release 2, Modification 0, of *IBM Branch Transformation Toolkit for WebSphere Studio* (5724-H82) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. You can send to the following address:

IBM China Software Development Lab
Branch Transformation Toolkit Product
Diamond Building, ZhongGuanCun Software Park, Dongbeiwang West Road No.8,
ShangDi, Haidian District, Beijing 100094 P. R. China

Include the title and order number of this book, and the page number or topic related to your comment.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Branch Transformation Toolkit Overview

This document provides a high-level introduction to the IBM® Branch Transformation Toolkit for WebSphere® Studio (Branch Transformation Toolkit) product, a component-based toolkit for developing enterprise applications. This document describes the benefits of using the toolkit and gives a brief description of the architecture, each of the provided components, and the development model.

The audience for this document is business and sales professionals, project managers, and anyone else who is interested in a high-level introduction to the Branch Transformation Toolkit. Solution architects and anyone requiring more understanding of the architecture of this product should refer to the Solution Architecture.

New in this release

This section describes the new features of version 5.2 of the Branch Transformation Toolkit (BTT). It also provides you with references to other resources that can help you learn more about BTT.

This version of the Branch Transformation Toolkit has the following new and enhanced features:

- Struts Extension enhancement
- BPEL Editor plugins
- Business Process flexibility enhancement
- Common Hierarchy Area (CHA) with performance improvement
- Enhanced Formatting service
- Web Service JSR 109 standard compliance
- Startup beans for Branch Transformation Toolkit component initialization
- Enhanced Branch Transformation Toolkit exception handling and propagation
- Simplified Branch Transformation Toolkit service architecture
- Supports ″Operation Steps″ (including self-definition)
- Branch Transformation Toolkit HTML client components
- Supports J2EE Connector Architecture (JCA) 1.5 SNA connectors
- End-to-end session management
- Dynamic TPName support at runtime
- Merging tool

Struts Extension enhancement

In BTT 5.2, the function of the struts extension is enhanced:

- **Supports direct mapping from HTTP request data to screen flow context**

  In BTT 5.1, the data submitted by the Http Request is mapped to the Struts FormBean, and then the BTT Struts will map the data from the FormBean to the BTT Flow Context when the FormBean's Field name is the same with the Field name of BTT Flow Context.

The disadvantage of the 5.1 mapping method is that the developer has to define the data not only for the FormBean, but also for the Context, and the data name must be the same.

However, the Struts FormBean itself supports both nested object mapping (such as *beanA.beanB.attribute*), and the BTT Context. If you use the mapping between the Struts FormBean and the Flow Context, the nested object is not supported.

In BTT 5.2, if you specify the configuration in the web.xml file as in BTT 5.1, the BTT Struts Extension will follow the same execution sequences as in BTT 5.1: It will map the data from the FormBean to the Flow Context. But if you want to ignore FormBean, you can define the settings in the web.xml file, and the BTT Struts Extension will retrieve the data from the HTT Request directly, and then set it into Flow Context.

- **Provides configurable RequestProcessor and TokenProcess names**
  - BTT 5.2 provides a configurable RequestProcessor instead of a hard-coded name so that you can replace the included implementation of BTT RequestProcessor with your own implementation.
  - BTT 5.2 provides modifiable TokenProcess name and implementation of ″TokenProcessorInterface extension″ instead of a hard-coded name so that you can replace the default implementation of BTT Token Processor with your own implementation.

BPEL Editor plug-ins

BPEL Editor plug-ins add extension to BPEL Editor in WebSphere Integration Developer and provide a node type named ″BTTSnippet″. Compared with earlier versions of BTT BPEL tools, the new plug-ins are more friendly to users with less limitation on BPEL.

Business Process flexibility enhancement

In BTT 5.2, Business Process has the following enhancements:

- A new node type ″BTT Snippet″ has been added to allow you to access BTT Components quickly in BPEL.
- BTT BPEL programming model has been changed from strong-type to both strong-type and shared context mode support.

Common Hierarchy Area (CHA) with performance improvement

BTT 5.2 has improved the performance of remote context performance.

Enhanced Formatting service

Because of the performance issue in BTT 5.1, the Formatter service has been redesigned to be combined to CHA in BTT 5.2. The server part has been removed but the facade is kept the same as in BTT 5.1.

Web Service JSR 109 standard compliance

JSR-109 facilitates the building of interoperable Web services in the Java™ 2 Platform, Enterprise Edition (J2EE) environment. It standardizes the deployment of Web services in a J2EE container and provides a universal and simple way to enable stateless EJB and SERVLET to the Web Service. The BTT 5.2 component has been enhanced to comply with the JSR109 standard and can be easily wrapped upon customers' request.

Startup beans for Branch Transformation Toolkit component initialization

The Branch Transformation Toolkit uses the startup beans to do the initialization for some of its components, such as the CHA, Formatter, and services. Startup beans can also do clean up when the application stops.

Startup beans provides the following advantages:
- Startup beans run in security contexts.
- Startup beans use JNDI to lookup and use other resources because they run within the namespace of WebSphere Application Server.

Enhanced Branch Transformation Toolkit exception handling and propagation

In BTT 5.1, many exceptions were caught by BTT components, but the BTT components would not throw them out. In addition, if an exception is caught and re-thrown, the original exception trace will be lost. If you use these exceptions in a web service environment, there might be problems because BTT 5.1 exception objects cannot support class serialization.

BTT 5.2 provides the following features for dealing with exceptions and exception tracing:
- For Web Services Support, two new constructions are added to BTT related exception classes, and 21 files have been modified. However these changes have no impact on the existing code.
- All caught exception are re-thrown out. And if a try-catch is not necessary, you can remove all useless try-catch statements, and let the exception be thrown out by the method itself.

Simplified Branch Transformation Toolkit service architecture

The service architecture of BTT 5.1 is very complex and causes both performance and efficiency problems. In addition, it has web service compatibility problems when used in conjunction with the WAS6 series.

The service architecture of BTT 5.2 has been simplified to provide greater compatibility. It's compatible with the BTT 4.3 service architecture and provides samples of how to deploy the service in the same way as EJB or WebService.

In order to be compatible with BTT 5.1, the BTT 5.1 service infrastructure jar files are still included. You can still use your BTT 5.1 style service at this time.

Note: You need to run your applications in J2EE 1.3 compatible mode to support the 5.1 WSIF framework.

Supports ″Operation Steps″ (including self-definition)

The BPEL of BTT 5.2 is leveraged on providing the Server Operation with Step functions, which had already been replaced by the IBM WebSphere Process Server (BPEL) in BTT 5.1. To make migration easier, BTT 5.2 provides the Server Operation with a Step Adaptor mechanism, in case you do not want to use the BPEL engine.

The Self-Definition with Operation/OpStep is still supported in BTT 5.2. Using a self-defined operation is preferable when the application is running and cannot be stopped and you need to add a new operation or change an existing one. It is also useful when the application cannot immediately download the full set of toolkit definition files, or the generic definition files become unwieldy due to their large size.

Branch Transformation Toolkit HTML client components

The Branch Transformation Toolkit provides a set of features and mechanisms that support an HTML-based graphical user interface (GUI) that is presented in a Web browser using an HTTP connection. An HTML-based GUI can be particularly appropriate for implementing home-banking applications, but it can also be used in commercial environments including teller, call center, and branch sales environments.

Supports J2EE Connector Architecture (JCA) 1.5 SNA connectors

BTT 5.2 supports JCA 1.5.

Bi-directional connectivity between a J2EE application and an enterprise information system (EIS) is an essential requirement for enterprise integration. In JCA1.0, only outbound communication is supported. It means that, using the CCI API of JCA1.0, the application first sends a message and then waits for the reply message synchronously. J2EE1.4 contains significant changes to two existing features that will extend the capabilities of J2EE applications. JCA1.5 and the EJB2.1 Message Driven Bean combine to handle inbound unsolicited messages from the EIS asynchronously.

Besides inbound communication support, JCA1.5 also provides other new features to make the inbound/outbound communication between J2EE applications and the EIS more reliable, more securely and more efficient. The new system contracts specified in the JCA1.5 specification are: Lifecycle Management Contract, Work Management Contract, Transaction Inflow Contract, Message Inflow Contract.

End-to-end session management

In BTT 5.1, the httpSession Object is managed by both Websphere and BTT to provide the BTT Session Management.

BTT 5.2 provides two ways of Session Management. Besides the BTT 5.1 method, the end-to-end management is also supported. In the end-to-end management mode, BTT session management APIs are wrapper APIs which call httpSession APIs directly. No information is stored in BTT 5.2, which means that APIs of query information using httpSession ID or TID APIs are not available in this condition. You need to specify HttpSession as parameter to let the wrapper go.

Dynamic TPName support at runtime

The TP Name FIX is used to dynamically change the LU62 TP name of the pool connection if required. In general, one WebSphere Application Server pool only supports one TP name. In BTT 5.2, one WAS pool can support multiple LU62 TP for JCA LU62 Connectors.

Merging tool

In order to support parallel developments, BTT 5.2 provides a merging tool to generate a single set of dse configuration files from several separate dse configuration files sets.

Introduction

Financial institutions are diversifying their offerings and adapting their products and services to ensure that they are able to respond to future market challenges and support changing business operations in an increasingly competitive environment.

The traditional teller is becoming obsolete. Many existing branch information systems are based on old technologies, such as financial-specific controllers or basic PC systems. These systems are no longer adequate or appropriate for meeting the challenges of the new environment, which include competitive factors such as the following:

- Reduced margin, especially in traditional products
- Increased competition
- Multiple channel environments
- Better informed customers who are sensitive to price and service quality
- Faster product introduction and reduced product life cycles

Financial institution services are mainly supported by applications whose core logic and data reside on host or enterprise systems, which are based on online transaction processing products such as CICS® or IMS™. For a bank teller application, access to these services (for example, to conduct a withdrawal transaction) requires delivery channels and a transaction posting engine that can handle the many tasks involved with transaction processing. The delivery channel and transaction posting engine must be able to manage the user interface, gather operation data, build host messages, process host responses, log transaction information into an electronic journal, access financial devices, and all other activities involved with processing the transaction. The IBM Branch Transformation Toolkit for WebSphere Studio is the transaction posting engine used by many financial institutions and other organizations for accessing back-end systems for banking delivery channels such as the traditional branch, call center, banking kiosk, Internet banking, and mobile access.

The Branch Transformation Toolkit provides a set of facilities to help with each of the processes and concepts mentioned above, modeling the real-life components of a teller system as objects and presenting them to development teams in a very familiar way. It accomplishes the following:

- Implements a simple but effective architecture that ties all components together in a loosely coupled fashion and makes them highly independent of each other.
- Uses normal object-oriented techniques that enable you to adapt to specific customer requirements; but is also highly parametric, which is a ″must″ requirement for teller systems.
- Abstracts the commonalties of local branch operations for financial transactions in a way that is easy to understand, develop, and maintain.
- Provides a way to deliver financial transactions as reusable and easily maintainable ″model″ objects.

- Provides an architecture and a class library that facilitate the structuring and development of teller applications by promoting reuse and providing the services required for the transaction processes.

In summary, the IBM Branch Transformation Toolkit for WebSphere Studio product is a pragmatic infrastructure designed and built so that existing mission-critical systems can evolve rather than be replaced. Its architecture provides an environment for high development productivity and great flexibility to meet the challenges of the new pace of change in both technology and the banking industry.

Rationale behind the toolkit

The IBM Branch Transformation Toolkit for WebSphere Studio product is a component-based framework for developing enterprise e-business applications. It offers software components that package a coherent set of functions. Each component package explicitly specifies the interface for the services it provides and also for the services it requires from other components. Component implementation details are encapsulated and kept separate from the interface specifications. The components can be independently developed, delivered, and installed in a way that allows you to build larger components and complete solutions.

Component-based application development is more cost-efficient and competitive than traditional methods. These benefits are realized through reduced requirements for software development skills and reduced development time. The value of these benefits continues to increase as the market demands increasingly sophisticated software applications at the same time that competitive pressures demand reduced time to market.

The Branch Transformation Toolkit is well suited for building Web-based financial services applications such as bank branch systems as well as building solutions for a wide variety of retail delivery channels, including Internet banking, call centers, stand-alone kiosks, automated teller machines (ATMs), and mobile access terminals such as wireless access protocol (WAP) capable cellular phones. The toolkit's multichannel support and dynamic component composition provide the foundation used to simultaneously meet the requirements of each of these retail delivery channels.

The toolkit is built on Java, an open industry standard and the object-oriented programming language of choice. Because the WebSphere software platform for e-business adheres to open industry and Internet standards, your investment is well protected. These standards include TCP/IP, HTML, HTTP, J2EE (Java, Java Server Pages, JCA, JDBC, EJB, and so on), Struts, BPEL, and Web Services. The toolkit components promote highly productive application development by supporting code reuse and the use of parameterization techniques to define business operations and their related objects. The toolkit preserves investment in existing enterprise systems by providing specially designed components that can communicate with these systems.

The Branch Transformation Toolkit is used to build applications with a multichannel architecture that extends the reach of a financial institution's information system services to all of its delivery channels. Financial institution services are most often supported by applications whose core logic and data reside on large-scale host systems. These enterprise systems run online transaction processing (OLTP) products such as CICS or IMS. Financial service delivery channels, such as a bank teller application or an Internet banking application, must access transaction functions on these systems (for example, to transfer funds between accounts). The toolkit uses JCA connectors to integrate delivery channels with large-scale OLTP system. It includes components designed to handle all aspects of transaction processing for every channel: managing the user interface, providing navigation dialogs, gathering operation data, building host messages, processing host responses, logging transaction information, accessing financial devices, and more.

The Branch Transformation Toolkit is highly customizable and its application is not limited to the financial services industry. Consider the toolkit a potential solution to your transaction processing requirements no matter what your industry.

The toolkit runtime provides National Language Support (NLS) for the following languages in group 1: Brazil Portuguese, French, Japanese, Korean, Simplified Chinese, Traditional Chinese, and Spanish. The toolkit also provides NLS for the following languages in group 2: Arabic. It also provides Bi-directional Languages Support (BIDI). The toolkit externalizes any end user text or messages from runtime components in resource bundles.

The Branch Transformation Toolkit provides pre-tested, user-configurable application components that can be quickly assembled into a complete financial services application. The development of these applications is based on the Branch Transformation Toolkit Graphical Builder plug-in for Rational® Application Developer and WebSphere Integration Developer. The Graphical Builder and its companion CHA Editor, Formatter Editor, Struts Tools BTT Extension, and BTT Business Process Editor enable users to define application screen flows, core business processes (using WebSphere Integration Developer's Process Editor), and their associated data structures. The toolkit entity definitions are managed with wizards designed to simplify this task. This approach to application development is a key benefit of the toolkit, as it minimizes the need for raw code development and promotes code reuse.

Benefits of using the toolkit

The Branch Transformation Toolkit is based on J2EE and other mature Internet technologies. This ensures that toolkit-built applications can be deployed with confidence as integral parts of robust production systems. The toolkit's design hides technical complexity from solution designers, which allows them to focus on business function rather than on the underlying technical details.

These features create benefits in the areas of project completion time, intermediate- and long-term cost-effectiveness, and readiness for future changes, improvements, and evolution.

Reduced risk

The Branch Transformation Toolkit is a fast and competitive way to solve your application needs, but being fast and competitive does not mean that you are left exposed to risk. Here are some of the ways that the toolkit reduces risk:

Proven product
The toolkit is a mature product developed by an industry leader in software applications for the financial services sector.

Systems work together
The extensive use of open computing industry standards (including Internet standards) protects against incompatibilities between systems.

Protection against obsolescence
The inherent flexibility and updateable nature of toolkit-based applications protects these applications from becoming obsolete.

Fast response to the business environment
The application development environment allows quick changes to applications in response to changing business conditions.

Build it right the first time
The application development environment supports teamwork, and this in turn promotes dialog and sharing of ideas; fewer details will be overlooked.

Preserve stable IT infrastructures
The toolkit provides JCA LU0 and JCA LU62 Connectors for you to connect your toolkit applications to existing systems that have been providing reliable services.

Faster time to market

The development approach that the toolkit promotes is designed to shorten development cycles and flatten the learning curve for the project team. The objective of this approach is to effectively save development effort, improve consistency, and reduce the time to market for all delivery channels. Following are some of the ways that the toolkit reduces time to market:

Shortened development cycles
The toolkit provides an environment that supports rapid application development by exploiting the benefits of component reuse. It does this by promoting the extensive use of object-oriented techniques and a high degree of application object parameterization.

Ready-to-use components
The toolkit provides a set of pre-built infrastructure components with well-defined interfaces. The components are ready to be incorporated into delivery channel applications; a project team needs only to learn how to use them, not how to build them.

Parametric application definition
The toolkit reduces the effort required to add new function to a toolkit-based application by providing the richness tooling plug-ins to create the definitions for the function.

Flattened learning curve
The toolkit productivity tools hide the underlying technical details of the toolkit. This reduces the amount of time and effort needed by a project team to learn the toolkit features and how to use them to deliver a solution. The development model creates a clear separation of roles that allows project team members to focus on their specific tasks.

Cost-effective application development

The Branch Transformation Toolkit application development product can provide cost savings at the earliest stages of development planning and all the way to deployment. Here are some of the ways that the toolkit reduces development costs:

Less reliance on high-level programming skills
From back-end connectors to user interface building blocks, the toolkit provides components that are easy to understand and use. This increases the size of the developer pool and reduces training costs.

Write once and deploy on several platforms
A toolkit-based application is portable across several platforms. Instead of

a costly ″from-the-ground-up″ development effort for each target platform, you define the application just once and then manage the deployment to any of the supported platforms.

Faster application development with fewer developers

The quicker an application can be developed the lower the cost. The toolkit's pre-built components reduce the person-hours needed to complete an application.

Improved development team communication

The WebSphere Studio provides team development environment. A common repository for the development products keeps the entire team synchronized and up-to-date. This avoids costly duplication of effort and rework.

Enhanced development using a graphical user interface

With the Process Editor in WebSphere Process Server, developers can visually choreograph business processes for various applications. They do not have to spend time working with different interfaces and low-level APIs. Drag-and-drop tools allow them to define the sequence and flow of information between different business logic activities. Individual business logic activities and even entire workflows become building blocks that can be reused in developing other applications. Further gains in productivity are possible because runtime support for these new J2EE workflow capabilities is fully integrated in the application server to deliver a single administration and deployment environment.

Reduced application operating costs

After an application is deployed, the costs of operating the application become an important measure of success. The toolkit offers cost savings that take effect at and continue beyond deployment. Following are some of the ways that the toolkit reduces operating costs:

Preservation of back-end systems

Deployment of a toolkit-based application does not require changes in existing business logic or transactions run in back-end systems. The toolkit uses JCA connectors to connect existing back-end systems and the application located on a middle-tier server.

Reduced maintenance and operational costs

The use of the network computing architecture, which is based on Internet technologies, results in immediate cost savings on client administration, code distribution, and server management. In addition, toolkit solutions minimize the code distribution that is required for incremental changes.

Operational portability

If operational conditions require that the application be moved to another platform, this can be quickly performed since the application is platform-independent.

Ease of maintenance

During operation, it is common to discover that application changes are needed. The environment and the distributed nature of the application support easy, quick, and universal application updates no matter how many application delivery channels and users are affected.

Adjustments to suit available system resources

Technology and systems are subject to change; toolkit-based applications can quickly be adapted to take advantage of more system resources or compensated for a reduction in resources.

Reduced workstation requirements

The distributed architecture of toolkit-based applications reduces the resources needed to deliver the application to the user. User workstations need to do little more than support the application presentation and any directly connected peripherals. Adding workstations is extremely cost-efficient since the server-based application can be distributed to any number of client workstations.

Common functionality across channels

An application can be designed to provide a common set of functions across multiple delivery channels. This consistent approach to service delivery promotes user satisfaction and reduces the training time needed if the user moves between channels.

Extendable and adaptable applications

Information technology is fast changing and so is the financial services market. An application developed for the needs of today can rapidly become obsolete. To protect you from this, every Branch Transformation Toolkit solution has features that allow it to be easily extended and adapted. Changes can be made to match the evolution of IT systems, business expansion, business diversification, and other predictable or unforeseen shifts. Following are some of the ways that the toolkit allows you to compete instead of becoming obsolete:

Multichannel enabled

The multi-tier architecture enables component reuse among delivery channels. Transaction requests from every channel are handled by the enterprise systems in the same way as any other channel. This promotes uniform, consistent, and rapid deployment of financial services through all your delivery channels. New channels may be added as needed or as they become available.

Easy integration with existing (and upcoming) systems

The toolkit integrates easily with present and future systems because it is based on open Internet standards such as HTML, SSL, HTTP, XML, TCP/IP, JavaBeans™, Enterprise JavaBeans, and JDBC. It includes JCA compliant SNA LU0 and LU62 resource adapters to facilitate its ability to develop applications that interact with other systems.

Platform portability and system scalability

Thanks to the portability of Java between operating systems and hardware platforms, toolkit applications can be ported from one operating system to another with minimal impact. You can easily change your branch platform (for example, from Windows® to Linux®), and you can also change the role played by your IT system components. For example, you may want to move part of the business logic from a branch system running on one operating system to a regional or centralized system based on a different operating system; moving the logic between server levels and operating systems is a simple procedure.

Service Oriented Architecture (SOA) enablement

The toolkit enables customer applications to support Service Oriented Architecture (SOA). The transaction flow can now be implemented using the Process Choreographer. When a complex transaction involves backend web services, the toolkit supports Web Services JSR 109 standard and it allows web service invocations from the toolkit’s own business layer. On the other hand, the BTT business logic can be treated as a service to be reused by the other application systems. Furthermore, the web service interfaces of JCA SNA LU0/LU62 connectors are in readiness for the web service invocation for legacy connectivity.

Supported platforms and technical requirements

This section presents the supported platforms and software required by each of the Branch Transformation Toolkit components. Because the toolkit is built in Java, any additional platform that provides the corresponding Java Virtual Machine is supported by the toolkit architecture.

A new solution with additional platforms may require changes to the toolkit to make it more generic so that the new solution can cope with the current platform as well as the new one. These changes may involve enabling the toolkit interfaces or components to support the new platform, and may be required for both the hardware and software components of the solution. In cases where native interfaces are required, a gap analysis is needed to support the new specific modules not provided by the toolkit. The components that are actually used depends on the specific requirements of each customer.

Components and platforms

In the following table, an X indicates that the service or component can be installed on that particular platform. Note that nothing prevents an application from accessing a service or component installed on another platform.

Table 1. Java client components

Component name		Windows Server 2003	Windows XP	Linux Intel®	Pure Java
Contexts, data elements, typed data elements		X	X	X	X
Formatters		X	X	X	X
Flow processors		X	X	X	X
Events		X	X	X	X
Externalizers		X	X	X	X
Exceptions		X	X	X	X
XML Desktop and Visual Beans		X	X	X	X
Financial devices	WOSA/XFS	X	X		
	J/XFS	X	X	X	X
Generic pool		X	X	X	X

Table 2. Application server components

Component name		Windows Server 2003	Windows XP	AIX®	Solaris	Linux Intel	Pure Java
Data elements and typed data elements		X	X	X	X	X	X
CHA		X	X	X	X	X	X
Java Client/Server Messaging APIs		X	X	X	X	X	X
Struts Extensions		X	X	X	X	X	X
Business Process Component		X	X	X	X	X	X
Single Action EJB		X	X	X	X	X	X
Startup beans		X	X	X	X	X	X
Events		X	X	X	X	X	X
Externalizers		X	X	X	X	X	X
Exceptions		X	X	X	X	X	X
Communications	JCA LU0 or LU62	X	X	X		X	

Table 2. Application server components (continued)

Component name		Windows Server 2003	Windows XP	AIX®	Solaris	Linux Intel	Pure Java
Database services	Database Table Mapping	X	X	X	X	X	X
	Electronic Journal	X	X	X	X	X	X

Table 3. Tools

Component name	Windows Server 2003	Windows XP	AIX	Solaris	Linux Intel	Pure Java
Graphical Builder	X	X			X	X
CHA Editor	X	X			X	X
Format Editor	X	X			X	X
BTT Business Process Editor	X	X			X	X
Struts Tools BTT Extension	X	X			X	X

Components and technical requirements

The following table shows the additional technical prerequisites of the Branch Transformation Toolkit components. For version information, see the Installation Guide.

Table 4. Additional technical prerequisites

Component name		Technical requirements
Financial device services	WOSA/XFS	WOSA/XFS manager and device-specific SPM
	J/XFS	J/XFS manager and specific device service
Database services	Table Access	JDBC driver
	Electronic Journal	JDBC driver
Tools		Rational Application Developer or WebSphere Integration Developer JDK

Notices

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

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

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Lab Director
IBM China Software Development Lab
Diamond Building, ZhongGuanCun Software Park, Dongbeiwang West Road No.8,
ShangDi, Haidian District, Beijing 100094 P. R. China

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

COPYRIGHT LICENSE:

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

Trademarks and service marks

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

IBM
AIX
CICS
WebSphere
DB2
Informix
MQSeries
zSeries
OS/390
z/OS
LANDP
Tivoli
DB2 Universal Database
IMS
RS/6000

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

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

MMX, Pentium, and ProShare are trademarks or registered trademarks of Intel 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.