

WebSphere® IBM WebSphere Multichannel Bank Transformation Toolkit
Version 7.0

Product Overview



Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 15.

This edition applies to Version 7, Release 0, Modification 0, of *IBM WebSphere Multichannel Bank Transformation Toolkit* (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
Bank Transformation Toolkit Product
Diamond Building, ZhongGuanCun Software Park, Dongbeiwang West Road No.8,
ShangDi, Haidian District, Beijing 100193 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.

© Copyright IBM Corporation 1998, 2010.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Bank Transformation Toolkit overview. . . 1

Introduction	1
Rationale behind the toolkit	2
Benefits of using the toolkit	4
Reduced risk	4
Faster time to market	4
Cost-effective application development	5
Reduced application operating costs	5
Extendable and adaptable applications.	6
Architecture	7
Client.	8
Channel application server	8
Enterprise server	9

Sample deployment configuration	10
Components	11
Core components	11
Presentation components	11
Service components	12
Business components	12
Development tools	13
Runtime tools	13
Support tool	13

Notices 15

Trademarks	17
----------------------	----

Bank Transformation Toolkit overview

This document provides a high-level introduction to the IBM® WebSphere® Multichannel Bank Transformation Toolkit (Bank 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 Bank Transformation Toolkit. Solution architects and anyone requiring more understanding of the architecture of this product should refer to the Solution Architecture.

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 channel application is becoming obsolete. Many existing channel application systems such as teller, internet banking 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. 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 WebSphere Multichannel Bank Transformation Toolkit 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 Bank 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 channel application 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 channel application 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 WebSphere Multichannel Bank Transformation Toolkit 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.

To help you understand BTT value proposition better, the following diagram is the Positioning of Selected IBM Products:

Positioning of Selected IBM Products

	Lotus Expeditor <small>Lotus software</small>	WebSphere Portal <small>WebSphere software</small>	WebSphere MBTT <small>WebSphere software</small>	WebSphere ESB <small>WebSphere software</small>	WebSphere Process Server (Includes ESB capabilities) <small>WebSphere software</small>	FileNet <small>Information Management software</small>	WebSphere Business Services Fabric (Includes WPS capabilities) <small>WebSphere software</small>	WebSphere Message Broker <small>WebSphere software</small>
General Purpose RCP* Extensions	★							
Banking specific RCP* Extensions			★					
Web Portlet Container		★						
Web At-the-Glass Integration		★						
Screenflow			★					
Channel Application Microflow			★	★				
Service Composition					★			
Services Orchestration					★			
Macroflow (BPEL & BSM)						★		
Content and Human Centric Process Control							★	
Dynamic Selection of Web Services				★				★
WS Routing and Transformation								

★ = Recommended SWG Product * RCP = Eclipse Rich Client Platform

Rationale behind the toolkit

The IBM WebSphere Multichannel Bank Transformation Toolkit product is a Java EE component-based framework that greatly accelerates multi-channel application development for banks, providing an integrated development environment on top of Rational Application Developer.

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 Bank Transformation Toolkit is well suited for building web based or Rich Client-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, JSP, JCA, JDBC, EJB, and so on), Web 2.0, Rich Client, 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 Bank 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. 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 Bank 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 is.

The toolkit runtime provides National Language Support (NLS) for the following languages in group 1: Brazil Portuguese, French, Hebrew, 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 Bank Transformation Toolkit provides pre-tested, user-configurable application components that can be quickly assembled into a complete financial services application. The Bank Transformation Toolkit provides a number of tools that support the development of applications. All the tools are plug-ins of Rational® Application Developer (RAD)△

Benefits of using the toolkit

The Bank Transformation Toolkit is based on Java 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 Bank 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. It also provides support for JMS, MQ, and Web service for modern IT infrastructures.

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 Bank 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.

Reduced development efforts by a rich set of development tools

Bank Transformation Toolkit provides a rich set of development tools that can help developers build BTT transaction rapidly. These tools also helps to shorten the development cycle.

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 Rational Application Developer 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.

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 and invokers 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 Bank 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, JPA, WebServices, 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

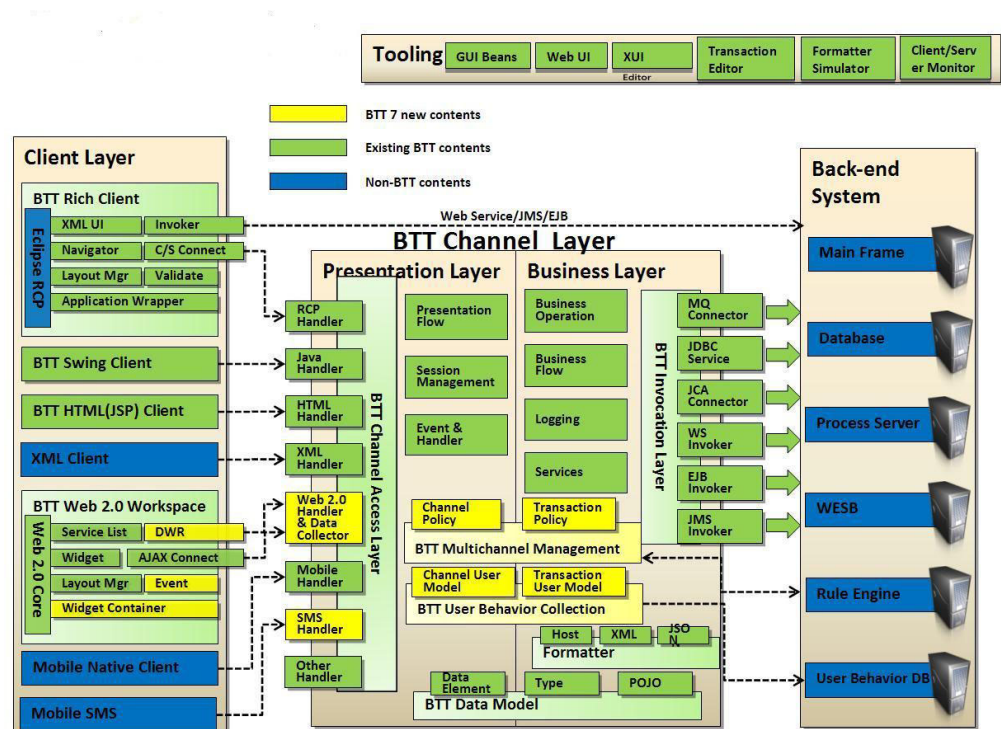
The toolkit enables customer applications to support Service Oriented Architecture (SOA). The transaction flow can now be implemented using the Process Choreographer. When a complex transaction involves backend web services, the toolkit supports Web Services JAX-RPC and JAX-WS standard and it allows 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.

Architecture

This section introduces the parts of Bank Transformation Toolkit product:

- Enterprise server
- Middle-tier application server
- Client
- Sample deployment configuration

The following figure is the overview of BTT architecture.



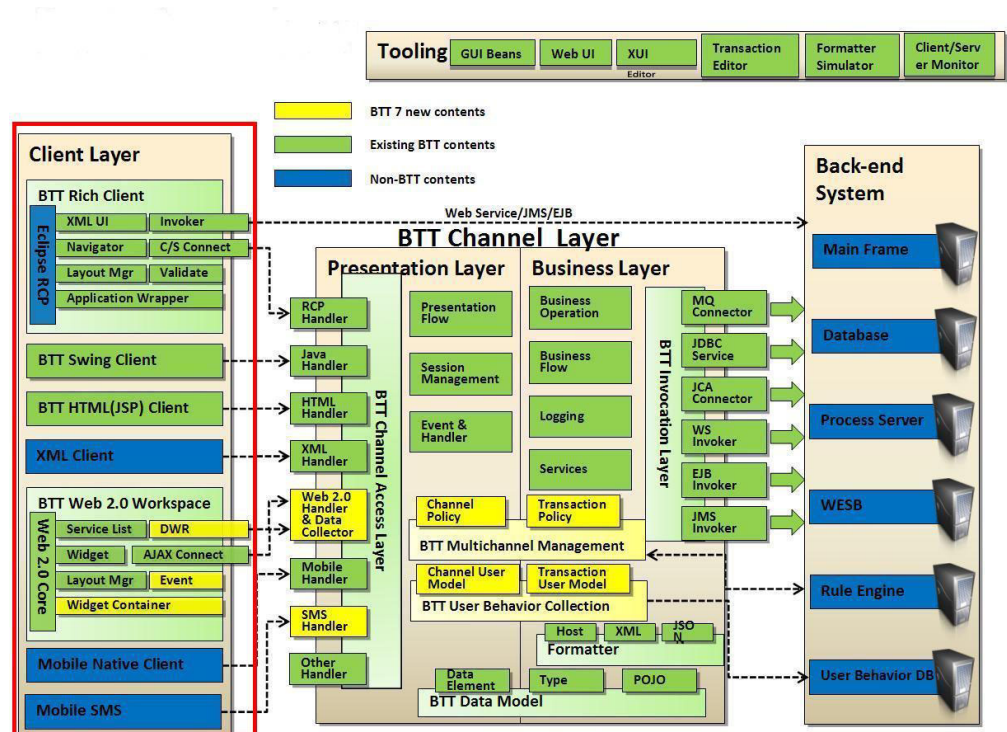
Client

A client in the three-tier architecture contains little logic. The logic it does have is usually presentation logic or logic required locally to do such things as accessing financial devices or validating entered data. The code to execute the client logic is downloaded on an on-demand basis, and therefore does not reside on the client, but on a Web server. The Bank Transformation Toolkit supports any kind of physical client device that uses the following technologies:

- Java Desktop Client (BTT Swing and RCP)
- HTML Client (Web 1.0 Client)
- Web 2.0 Client
- Mobile Client

The toolkit provides implementations for current client technologies but these concrete implementations anticipate that significant differences might be found when realizing solutions. The toolkit is not limited to these technologies because its design is generic and can be extended to support other technologies.

The following diagram shows position of clients in the architecture of BTT:

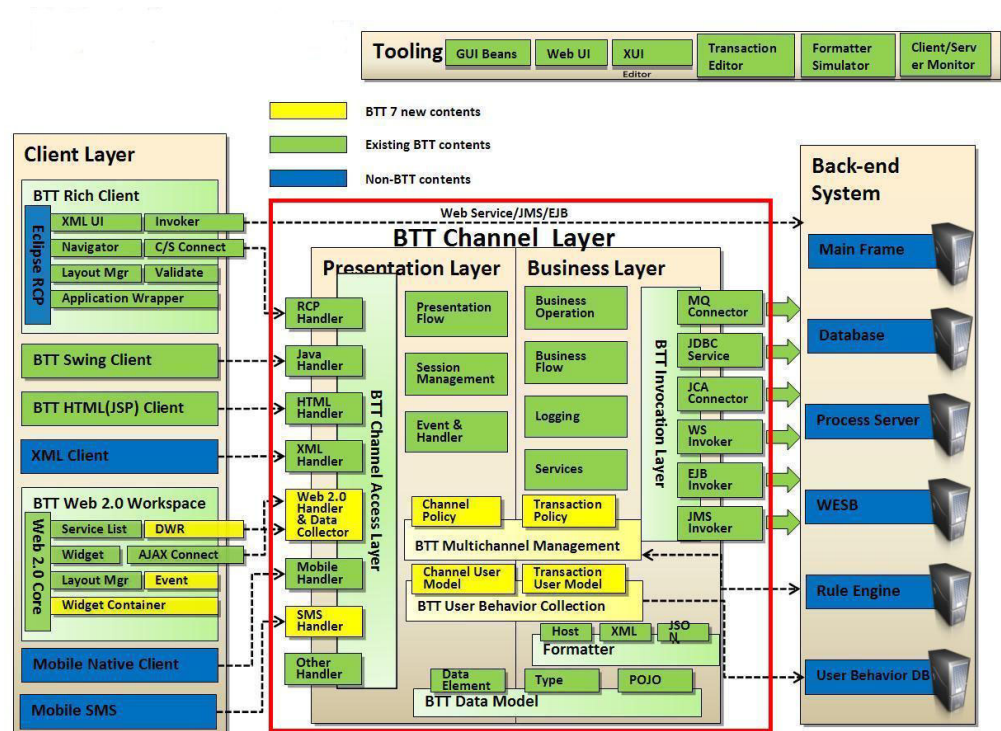


Channel application server

The channel application server hardware (also known as the Front End Server) delivers the toolkit-based application to the clients using a TCP/IP connection. IBM WebSphere Application Server runs on the channel application server for this purpose and processes requests from clients once the application is running. Handling client requests involves managing dialogs (and the user interface for light client types), launching business operations that interact with back-end transactional systems or SOA based systems, processing local transactions, and sending responses to the client. This architectural layer contains logic that is intended to be common across delivery channels, as well as technical functions such as electronic journaling, parameter tables, and store-and-forward.

The design and portability of the product allow channel application servers to exist at either the branch level (one server per branch), the regional level (one server per a group of branches), or even at a centralized level (a single server for the entire financial institution). These options provide the flexibility to achieve the right balance between the number of servers and network bandwidth, with no changes to application logic.

The following diagram shows the channel application server.



Enterprise server

The enterprise server, or the back-end server, contains the existing core business logic of the financial institution that is accessed by the toolkit application. A toolkit application does not require changes to such a system or changes to its messaging interface. This is possible because the toolkit includes a rich set of back-end system connector components and message formatters. BTT JCA SNA and Invoker component are provided for SNA (LUA interfaces), JMS, EJB, WebServices, and any other customer extensions.

On the other hand, if you have already built up the SOA based back-end system such as ESB, the toolkit enables your applications to support Service Oriented Architecture (SOA) integration.

Bank Transformation Toolkit interfaces with WebSphere Process Server (which contains WebSphere ESB) for business process automation and enterprise application integration. WebSphere Message Broker and WebSphere Business Services Fabric can be added depending on the SOA requirements. But typically, they do not interface with BTT directly.

The entire banking SOA reference architecture includes the following flow and control concepts:

- Channel Interaction Orchestration
 - Screenflow: A lightweight Web/Rich client tier control mechanism (usually a finite-state-machine) that guides the user from screen to screen. States and flows are encoded in XML.
 - Channel Application Microflow: A lightweight Web/rich client tier control mechanism that provides a structured way to organize channel application operations such as screen flows, logging, reusable channel specific logic, invocation of business processes, and invoking back-end services. Tooling is specific to and integrated with the channel application platform. Flows are visually designed and encoded in XML.
- Business Process Automation
 - Macroflow: Long-running process or process involving human tasks to be performed by multiple people. Encoded in BPEL as a linear process or Business State Machines.
- Enterprise Application Integration
 - Service Composition: The creation of a coarse-grained service from a number of finer-grained services and simple flow logic. Usually created using SCA components.
 - Service Orchestration: Invocation of multiple services in the context of a microflow or macroflow execution. A flow or state machine can be used as the control construct to create a composite service from element services.
 - Routing and Transformation: Routing of a service request to a service provider at runtime according to pre-determined rules and the transformation of the service name, number and type of parameters, and data structures as needed so as to insulate service consumers from service providers.
 - Dynamic Service Selection: Determination of how to resolve a service binding at runtime.

Sample deployment configuration

Bank teller

A bank teller application topology consists of a number of client workstations with financial devices attached. The workstation downloads the client application on request from a Web server. The client applications, which mainly deal with presentation and local financial device handling, have access to the branch server (that is, the solution application server) using the HTTP or SSL protocols.

The solution application server provides common services such as electronic journaling and parameter tables to the client workstations, as well as access to the transactional logic of the back-end enterprise servers. A toolkit server application can also be deployed on the physical server for a regional or central data center without changes to the application.

Internet banking

In an Internet banking topology, users obtain access to financial services through a Web browser (or other device) connected to the Internet. The user interface is normally HTML with additional technologies such as JavaScript™, DHTML, or XML. In such an environment, the solution application server is able to process requests from Web browsers (or other devices that issue HTTP requests), obtain the proper data from enterprise servers, and generate the appropriate view for the client device to display using HTML pages for Web browsers or XML messages for those devices that support it. The application server is usually located at the central site, and is protected by a firewall.

Kiosks and ATMs

The toolkit can be used in kiosks or ATMs that run Internet technologies such as a Web browser and Java. In this environment, the client usually is a Java application (or applet, SWT). In addition to the presentation logic, the client application manages the financial devices normally present in a kiosk (such as MSR/E, chip card reader, receipt printer, passbook printer, bar code readers, and touch screen displays) using the financial device services that the toolkit provides. The kiosk connects to the application server using the HTTP or SSL protocols. In some cases, kiosks are located in branches, which handle them as branch workstations. Kiosks can also be connected directly to the server through public or private lines.

Mobile terminal and PDA

Users equipped with notebooks running a Web browser can connect to corporate toolkit servers using the SSL protocol. In this scenario, the toolkit server is usually located at the central site and is protected by a firewall. It is also feasible to have mobile users connected to the branches to which they belong. Using the BTT mobile channel, you can connect to BTT server from the eRCP client on mobile phone. BTT also provides SMS channel which helps to build a mobile bank which can be accessed by smart phones and simple SMS-based messaging.

Components

The Bank Transformation Toolkit contains the following components:

Core components

The Core components are the main entities of the IBM WebSphere Multichannel Bank Transformation Toolkit.

The core components include the following:

- Operation
- Flow
- Contexts
- Formatter
- Data element
- Initialization manager
- Element factory
- Exception
- Events
- Externalizer
- Trace

Presentation components

The Bank Transformation Toolkit provides components that facilitate the construction of the client presentation logic. The toolkit includes the Rich Client infrastructure, a set of SWT Visual beans, as well as the support for JavaServer Pages (JSPs).

The following are the provided presentation components:

- Rich Client infrastructure, which offers pre-build components to save your effort on developing teller systems.

- Web 1.0 support, which includes custom JSP tags and utility beans to enable applications to retrieve information from the operation context hierarchy, get resources, and handle errors. If your application requires additional behavior, you can build new tags using the JspContextServices interface.
- Web 2.0 support, which includes entire Web 2.0 component running on browser. The Web 2.0 component is consisted by a set of XML, DHTML, JavaScript and CSS.
- XML UI Engine, which simplifies the UI design by defining the configuration files.

Service components

The Bank Transformation Toolkit provides a set of service objects that enable an application to complete an operation. These services include host communications, journaling, store-and-forward for off-line operations, financial devices for input and output operations, and more.

The service components include the following:

- Communication services:
 - JCA LU0
 - JCA LU62
 - MQ connector
- Database services:
 - Database Table Mapping
 - Electronic Journal
 - Store
- LDAP Access service
- Generic Pool
- Financial Device services:
 - Check Reader Device service
 - JXFS service
 - LANDP MSR/E Device service
 - WOSA Device service

Business components

The business components include the following:

- Branch application components:
 - Foreign Exchange
 - Cash Drawer
 - Electronic Journal and Electronic Journal Viewer
 - Override
 - Access Control
- Supporting components:
 - Remote
 - Asynchronous Message
 - Connection Pool

Development tools

The Bank Transformation Toolkit provides a number of tools that support the development of applications. All the tools are plug-ins of Rational Application Developer (RAD).

The development tools include the following:

- Transaction Editor
- Deployment Descriptor Editor
- XUI Editor
- Formatter Simulator
- Application Wizard
- Migration Tool

Runtime tools

Runtime tools are used to fetch or record information of the BTT runtime.

The Runtime tools include the following:

- Rich Client Monitor
- Web Transaction Monitor
- Server Runtime Monitor
- Trace Facility

Support tool

The Support tool includes the following:

- APAR Viewer

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 100193 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

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. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol ([®] or [™]), these symbols indicate U.S. 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 "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

Microsoft and Windows 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.

Java is a trademark of Sun Microsystems, Inc. in the United States, other countries, or both.

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