IBM Branch Transformation Toolkit
for WebSphere Studio

Overview

Version 5.1

Note!
Before using this information and the product it supports, be sure to read the general information under "Notices" on page 17.

Eighth Edition (July 2005)

This edition applies to Version 5, Release 1, Modification 0, of *IBM Branch Transformation Toolkit for WebSphere Studio* (5648-D89) 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
Information Development
10F Shui On Plaza, 333 Middle Huai Hai Road
Shanghai 200021, 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.

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 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 WebSphere Studio Application Developer. The Graphical Builder and its companion CHA Editor, Formatter Editor, Struts Tools BTT Extension, and Business Process BTT Wizard plug-ins enable users to define application screen flows, core business processes (using Integration Edition'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 Studio Application Developer Integration Edition, 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.

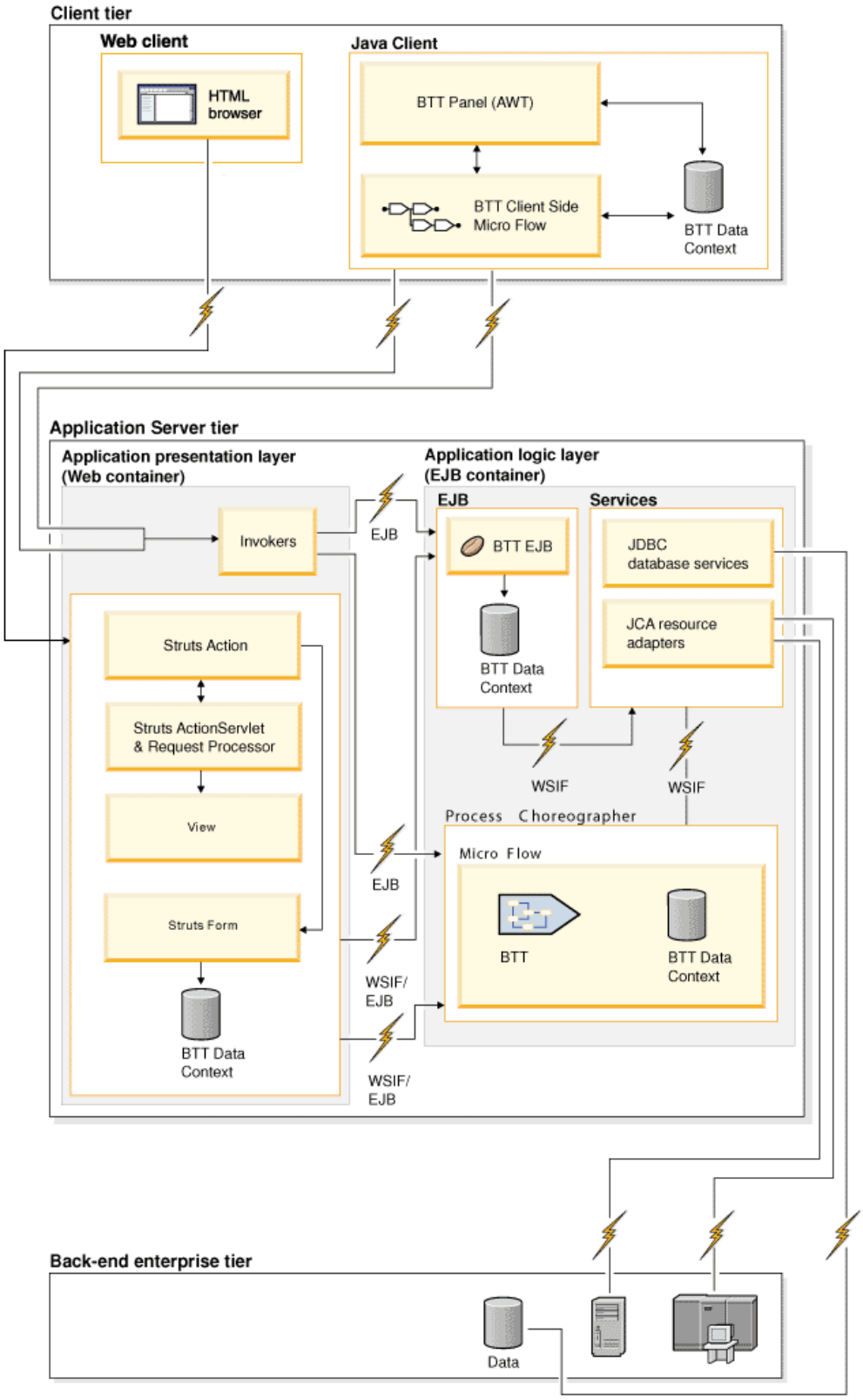
Platform portability and system scalability
Thanks to the portability of Java between operating systems and hardware platforms, toolkit applications can be ported from one operating system to another with minimal impact. You can easily change your branch platform (for example, from Windows® to Linux®), 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.

Architecture

The Branch Transformation Toolkit is based on the J2EE standard. The development environment of toolkit applications is integrated with WebSphere Studio Application Developer, which is based on the Eclipse platform; and the runtime environment of toolkit applications is based on WebSphere Application Server or WebSphere Business Integration Server Foundation.

The architecture of the Branch Transformation Toolkit application solution is based on a logical three-tier model: back-end enterprise tier, application server tier, and client tier. The following figure depicts these tiers:



The client tier is responsible for presenting an interface to the user in a client device. The Branch Transformation Toolkit supports the following types of clients:

- Java clients running as applets in a browser or as Java applications
- HTML browser clients

For Java clients, the toolkit provides a set of graphical user interface (GUI) JavaBeans for building the client user interface and a mechanism for navigating the views of the application. For the HTML browser clients, the toolkit Struts Extension framework on the application presentation layer of the application server tier manages the user interface and view navigation.

The application server tier has two parts: the application presentation layer and the application logic layer. The application presentation layer is responsible for creating the request for the invocation of business logic hosted within the application logic layer. For HTML browser clients, the presentation layer is also responsible for providing the view navigation. The application presentation layer converts an action made by the user in the user interface into a WSIF message or EJB method invocation that the presentation layer then sends to the application logic layer. The application logic layer is responsible for performing the business process that fulfils the presentation layer request. A business process consists of a set of activities and may involve accessing and manipulating enterprise data and performing financial service procedures in the back-end enterprise tier. The application logic layer can use two mechanisms for performing the business process. If the application server tier is running on WebSphere Business Integration Server Foundation, the application logic layer can use the Process Choreographer and work area features. If the application server tier is running on WebSphere Application Server, the application logic layer can use a Single Action EJB.

The back-end enterprise tier consists of enterprise level databases and legacy systems that provide existing business logic and services. The application logic layer communicates with these databases and systems through JCA connectors, database services, and formatters. The connectors, database services, and formatters form an interface so that toolkit applications do not impose any changes directly on these databases and systems.

The design and portability of the product (a result of being written in Java) allows the application tier 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.

Sample deployment configurations

The following samples show how the Branch Transformation Toolkit supports some of the channels that a financial institution can use to deliver applications.

Bank teller

A typical configuration for bank tellers is a grouping of client workstations with physically attached financial devices and each one having an HTML browser or a Java client desktop. The client side of the teller application can be a Java applet downloaded on demand from an HTTP server. The client side, which mainly deals with presentation and the management of local financial devices, has access to the server side in the application presentation layer through the HTTP or SSL protocol. The server side provides a common set of services to the teller client workstations and provides access to the business processes provided by the application logic layer. Note that application can be deployed in a server physically located in the branch or in a regional or central data center without any changes to the coding of the application.

Internet banking

Internet banking users access financial services through a Web (HTML) browser running on a device connected to the Internet. The user interface design is normally based on HTML and other browser-supported technologies such as JSPs and XML. The HTTP protocol connects the client side to the server side, which is in the application presentation layer. The server side of the application processes requests from the Web browsers and invokes a business process in the application logic layer to access the data from enterprise servers. The server side also generates the views that the client displays. The application is usually located at a central site behind a firewall.

In most cases, the client views are HTML pages (as for Web browsers); however, an HTTP protocol can use other technologies and presentation options such as XML messages if the client device supports them.

Kiosks and ATMs

Toolkit applications can also appear in kiosks or ATMs that run Internet technologies such as a Web browser and Java. The solutions are similar and equally successful. The client is usually a Java application or applet that manages both the presentation logic and the support for the financial devices present in the terminal. For example, a kiosk might have a magnetic stripe reader (MSR), chip card reader, receipt printer, passbook printer, bar code reader, or touch screen display.

The kiosk or ATM terminals may be connected to the toolkit server using the HTTP or SSL protocols. They may also be located inside branches and connect and handle like branch workstations. Terminals can also connected directly to the server through public or private data lines.

Components

The Branch Transformation Toolkit splits into components in the application presentation layer and components in the application logic layer.

Note that some Branch Transformation Toolkit components can only run on WebSphere Business Integration Server Foundation, while some toolkit components can run on both WebSphere Business Integration Server Foundation and WebSphere Application Server.

Components that can only run on WebSphere Business Integration Server Foundation include:

- Toolkit startup beans, which uses the startup bean feature of WebSphere Business Integration Server Foundation
- Business Process component, which leverages the process choreographer feature of WebSphere Business Integration Server Foundation

The following features are only available when your application runs on WebSphere Business Integration Server Foundation:

- Work area
- Activity session

These features can be used by the session management component of the Branch Transformation Toolkit.

Java client

Branch Transformation Toolkit components that run on Java clients interact with each other to provide the Java client navigation function, pass client requests to the application presentation layer, and enable Java clients to present a response to the request.

The following components run on Java clients:

Contexts
A context is the container for the data elements needed by a business entity such as a user or branch. Contexts have a hierarchy to enable these business entities to share common data. For example, in a branch each teller would have a context that contains data about the teller but they would all share the branch context, which would contain data about the branch. The branch context is the parent and each teller context is a child in the relationship.

Formatters
A formatter transforms a string into data in a context or data in a context to a String. This enables an application to move data into and out of the context hierarchy and to create messages to send to a host, financial device, or service in a format understood by the message's destination. The toolkit provides an extensive set of the most commonly needed formatters for financial service applications including EBCDIC, date, numeric, packed, binary, and other formatters.

Data elements
A data element is a field that contains a value or a collection of other data elements. Certain data elements are type-aware. The typed data elements represent business objects such as Date, ProductNumber, and Money. Each typed data element has an associated property descriptor, which provides information about the data such as its type, its validator, and its set of converters.

Flows A flow is a particular route through a business process or presentation sequence in the application presentation layer. A flow processor handles a specific flow and it is typically with many branches and compound and complex conditions on those branches. Within a flow, there is a sequence of states. These states can have actions. An action is a task that the flow processor performs such as display a view or invoke a business process in the application logic layer.

Externalizers
An externalizer is an object factory that uses definitions in an external file to instantiate a specific toolkit entity. The toolkit provides externalizers for contexts, data elements, formatters, services, and flow processors. The definition files are ASCII files using XML syntax. This makes configuring and customizing these defined objects (or implementing new ones) possible with something as simple as a basic text editor although the Development Workbench provides an easier and more controlled environment for this editing.

Events
An event is how components within the application presentation layer communicate with each other. A notifier is the sender of an event. A handler, as the receiver of that event, is responsible for consuming the

event or propagating it to other handlers. An Event Manager acts as the event controller between notifiers and handlers to manage both local and remote events.

Exceptions
A toolkit exception enhances the standard Java exception mechanism to facilitate applications accessing information about the exception.

Visual beans
The visual beans facilitate the development of a GUI for Java clients by adding features and properties to normal visual beans to support financial applications such as date fields, numeric fields, or account data entry fields. The navigation controller provides a way for the Java clients to have a multiple view GUI.

Desktop
The Desktop is a fully configurable desktop for Java clients. It contains most of the features commonly required of this type of user interface. It includes many common UI features and can by dynamically personalized to the current user.

Operations
An operation is what Java clients use to launch business processes in the application logic layer. An invoker maps the client operation to the business process.

Generic Pool
The Generic Pool service enables multiple client operations to share certain objects (classes and services), which makes the objects reusable. This reuse reduces the average time to execute these operations and also reduces the garbage collection work.

Trace Facility
The Trace Facility provides a way to see what is happening with an application while it is running. The information it logs can be used to solve problems during development and during runtime.

Financial device services
The Branch Transformation Toolkit provides services to access the most commonly used financial devices in financial service applications, including financial printers, check readers, magnetic stripe readers, chip card devices, and passbook printers. Financial device services allow applications to access devices that are compliant with WOSA/XFS and IBM LANDP® protocols. The toolkit also supplies 100% Java access to specific financial devices following the current specifications of the J/XFS Forum.

JXFS Service
The JXFS Service enables applications to access devices that use J/eXtensions for Financial Services (J/XFS). The JXFS Service uses the typical interface between applications and J/XFS devices: Device Control (DC) and Device Manager (DM).

LANDP MSR/E Device
The LANDP MSR/E Device service enables applications to access a magnetic stripe reader and encoder (MSR/E).

WOSA Device
The WOSA Device service enables applications to use WOSA/XFS to access financial devices. These devices include financial printers, identification cards, and teller assist units

Application server components

The Branch Transformation Toolkit components that run on application servers can be divided into two categories: components running in the Web container of the application server and components running in the EJB container of the application server. The Web container and the EJB container share some toolkit components.

Shared components across containers

The following Branch Transformation Toolkit components are shared across the Web container and EJB container of the application server:

Data elements
A data element is a field that contains a value or a collection of other data elements. Certain data elements are type-aware. The typed data elements represent business objects such as Date, ProductNumber, and Money. Each typed data element has an associated property descriptor, which provides information about the data such as its type, its validator, and its set of converters.

CHA The Common Hierarchical Area holds data within a context hierarchy for the Business Process Component when it performs a business process. This is a distributed component that allows the data to exist anywhere. It also enables non-toolkit applications to store general global session data. The application uses the CHA API to mover data into and out of the CHA.

CHA Formatter Service
The CHA Formatter Service handles formatting and unformatting of strings and data items for applications and services. It converts a specific data item into a string representation of the data item and parses a string into a specific data item.

Events
An event is how components within the application presentation layer communicate with each other. A notifier is the sender of an event. A handler, as the receiver of that event, is responsible for consuming the event or propagating it to other handlers. An Event Manager acts as the event controller between notifiers and handlers to manage both local and remote events.

Externalizers
An externalizer is an object factory that uses definitions in an external file to instantiate a specific toolkit entity. The toolkit provides externalizers for contexts, data elements, formatters, services, and flow processors. The definition files are ASCII files using XML syntax. This makes configuring and customizing these defined objects (or implementing new ones) possible with something as simple as a basic text editor although the Development Workbench provides an easier and more controlled environment for this editing.

Exceptions
A toolkit exception enhances the standard Java exception mechanism to facilitate applications accessing information about the exception.

Trace Facility
The Trace Facility provides a way to see what is happening with an application while it is running. The information it logs can be used to solve problems during development and during runtime.

Web container components

The following Branch Transformation Toolkit components run in the Web container of the application server:

Sessions

A session is a conversation between a user (browser), client, or server that contains one or more sets of requests and responses. Sessions enable these entities to share data within the conversation yet distinguish the data from data in other conversations.

Invoker

An invoker is the interface to an EJB in the application logic layer. A request handler (part of the multichannel architecture) or the toolkit Struts Extension uses a specific invoker to start the business process performed by the EJB associated with the invoker.

Client/Server Messaging API

The Java Client/Server connectivity component enables the Java client and application presentation layer to communicate through a specific communication channel. The component contains a request handler, a Bean Invoker Factory, and a presentation handler. The request handler passes the request to the Bean Invoker Factory, which then instantiates an invoker to call a Single Action EJB or a business process in the application logic layer. The presentation handler handles the response from the application logic layer to render the result appropriately for the Java client.

Note: This component only works for Java clients.

Struts Extensions

The Struts Extensions component 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. The toolkit Struts Extensions component is based on the Apache Struts Web Application Framework.

Note: This component only works for HTML clients.

JSPs and JSP tags

JSPs and JSP tags enable the application presentation layer to dynamically generate HTML pages for HTML clients. They separate the generation of dynamic content from its presentation.

Note: This component only works for HTML clients.

EJB container components

The following Branch Transformation Toolkit components run in the EJB container of the application server:

Business Process Component

This component enables applications to perform business processes using the Process Choreographer in WebSphere Application Server Enterprise Edition. Applications can invoke the business process using a request handler feature of the multichannel architecture and an EJB interface or using a flow processor through the EJB or WSIF interface. The request handler and the flow processor both reside in the application presentation layer.

Single Action EJBs
This component enables applications to perform business process using stateful session EJBs. The Invoker component in the presentation layer is the interface to the single action EJBs.

Startup beans
A startup bean is a session EJB that loads and runs before an application starts. The Branch Transformation Toolkit uses the startup beans to do the initialization for some of its components, such as the CHA, CHA Formatter service, and services.

Generic Pool
The Generic Pool service enables multiple client operations to share certain objects (classes and services), which makes the objects reusable. This reuse reduces the average time to execute these operations and also reduces the garbage collection work.

Services
The following services enable the business process to connect to the back-end enterprise tier.

Communication services
These are JCA-based services that applications can use to access data and services in the back-end enterprise tier. The toolkit provides the **SNA JCA LU60 resource adapter** and the **SNA JCA LU62 resource adapter**. Both of the adapters conform to the J2EE JCA architecture and implement the Common Client Interface (CCI). This interface isolates the application from differences between communication protocols.

Database services
These provide JDBC connectivity to databases. The database services are enabled as an EJB or a Web service by using the Web Service Invocation Framework (WSIF). The **Database Table Mapping** service enables any application to access a database through a common application interface. The service converts messages into SQL statements to perform the requested database operation for the application. The **Electronic Journal** service enables a financial institution to store the services and processes used or performed by an entity such as branch, user, or terminal in a set of database tables. The Electronic Journal service uses the Java Database Connectivity (JDBC) standard to store its data.

Tools

The Branch Transformation Toolkit provides a number of tools that support the development of applications. All the tools are plug-ins of WebSphere Studio Application Developer or WebSphere Studio Application Developer Integration Edition. Note that some functions of the Graphical Builder are only available when you are using the Integration Edition of WebSphere Studio Application Developer.

The Graphical Builder provides a set of tools to define entities required by applications, and distribute the runtime files. The Graphical Builder provides a development environment throughout the development cycle of toolkit applications. It also acts as a portal from where you can start other tools that the toolkit provides.

The CHA Editor and Formatter Editor provide user-friendly interfaces for creating or maintaining the definitions needed by the applications in the application logic layer.

The Business Process BTT Wizard provides a graphical user interface (GUI) to help you extend your business processes for taking advantage of toolkit specific entities. Note that this tool is only available when you are using the Integration Edition of WebSphere Studio Application Developer.

The Struts Tools BTT Extension provides a GUI to help you extend your Struts configuration files for taking advantage of toolkit specific entities.

Apart from all the development tools listed above, the toolkit also provides a toolkit migration tool to help you migrate your toolkit applications developed with version 4.3 of the toolkit to the new version 5.1 architecture.

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
2/F Deshi Building, No.9 Shangdi Dong Rd,
Beijing 100085, 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	OS/390
AIX	z/OS
CICS	LANDP
WebSphere	Tivoli
DB2	DB2 Universal Database
Informix	IMS
MQSeries	RS/6000
zSeries	

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.