WebSphere® IBM WebSphere Multichannel Bank Transformation Toolkit

Version 6.1

Release notes

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 7.

This edition applies to Version 6, Release 1, 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.

Contents

IBM WebSphere Multichannel Bank Transformation Toolkit 6.1 release notes

This document contains information about version 6.1 of the IBM® WebSphere® Multichannel Bank Transformation Toolkit (Bank Transformation Toolkit). It describes the new components and functions at an architectural level.

New in this release

This section describes the new features of version 6.1 of the Bank Transformation Toolkit (BTT). It also contains references to other resources that you can use to learn more about BTT.

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

- Rich Client support and desktop integration
- Web 2.0 support
- JSF extension
- Tools
- Business components
- Inversion of Control Container
- Formatter
- Invoker
- SOA integration
- Multichannel samples

Rich Client support and desktop integration

BTT Rich Client support and desktop integration capability extends the Eclipse Rich Client Platform and Lotus® Expeditor, which is used by banking customers to rapidly build teller or desktop systems. It provides the following add-on features:

- Activity framework to specify operations
- Automatically generated navigation tree
- XML based perspective layout management support
- Application integration support (Swing, SWT, ActiveX, Web based applications)
- Property broker service to transfer data for different applications
- Global context for data sharing
- SWT widgets theme customization
- Very flexible architecture that is easy to extend
- Minimum coding effort required during development

Web 2.0 support

BTT version 6.1 provides support for Web 2.0 as one of the BTT channels. You can implement presentation tier using Web 2.0 technologies and build server-side applications using the BTT framework.

The Web 2.0 Channel uses multichannel support to send and receive XML and JSON formatted messages. In addition, Web 2.0 provides the request handler and presentation handler required for multichannel support.

JSF extension

BTT version 6.1 provides support to JavaServer Faces (JSF) as one of the BTT channels. You can implement a presentation tier using JSF technology and build server-side applications using the BTT framework.

Tools

BTT version 6.1 provides the following new development, runtime and support tools:

- Transaction Editor:

  The Transaction Editor is a development tool that provides an easier way for XML based end-to-end programming model.
- Validation Tool:

  The Validation Tool is a development tool with a graphical interface that you can use to validate your definition files .
- SWT Visual Beans Editor:

  The SWT Visual Beans Tool is a development tool that you can use to develop SWT based transaction panels.
- Formatter Simulator:

  The Formatter Simulator is a development tool that simulates the Formatter unformat process (translating binary message, XML and JSON into structured data).
- Runtime Monitor:

  The Runtime Monitor is a runtime tool that you can use to monitor the status of BTT on the server side.
- APAR Tool:

  The APAR Tool is a support tool that you can use to detect the build level of your BTT product version and obtain the detailed APAR numbers.

Business components

Business components are especially tailored for bank teller applications. Business components in version 6.1 include the following:

- Cash Drawer Management, which simulates the cash drawers in a retail bank. This helps teller or supervisor to track the total amounts, counts, and cash values in the whole branch. The main functions of this component are creation, maintenance, assignment, and deletion of the cash drawer and its components
- Foreign Exchange, which is responsible for exchanging one type of currency for another. The default implementation of the Foreign Exchange is a design and implementation template, which you can refer to and implement according to your actual business requirements
- Counter, which keeps track of numeric values

More business components will be introduced in future releases.

Inversion of Control Container

The Inversion of Control design pattern is a widely accepted design pattern. In BTT version 6.1, the Externalizer component is reconstructed as BTT ElementFactory. The BTT ElementFactory is loosely coupled with other BTT components, and can fully support the design pattern of Inversion of Control. BTT ElementFactory is widely used in BTT version 6.1, and you can use it directly in your own components.

BTT ElementFactory supports the following features:

- Different ways to instantiate Object: default constructor, constructor, static factory, instance factory, and factory element
- Different ways to inject dependency: setter method, constructor parameters, and self-defined method
- Macro definitions
- Lazy initialization
- Lifecycle management
- Factory extensions: ElementProcessor, FactoryElement, and Lifecycle processor

Formatter

BTT version 6.1 introduces a new set of formatters. Although the formatting algorithm is similar to what is in previous versions of BTT, the BTT version 6.1 formatters have extended abilities. The new formatters are loosely coupled with BTT DataElement and Context and support the following functions:

- Translating BTT DataElement and Context to and from binary messages, XML and JSON
- Translating Java™ value object to and from binary message, XML, and JSON
- Translating java.util.HashMap to and from binary message, XML, and JSON.

Invoker

Invoker is an XML-based invocation framework. It removes the technical barriers inherent in technologies such as J2EE, JMS, EJB, DII and WebService.

BTT Invoker supports the following technical invocations in both synchronized and asynchronized modes:

- POJO
- EJB
- Web Service Proxy
- Web Service DII
- JMS

SOA integration

For backend integration, BTT enables channel applications to support Service Oriented Architecture (SOA). BTT 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 SOA requirements.

The toolkit supports Web Services JSR 109 standard and Web service invocation from the toolkit's business layer. This can be useful when a complex transaction supports backend Web services. Alternatively, BTT business logic can be treated as a service and reused by the other application systems.

Multichannel samples

BTT version 6.1 provides the following multichannel samples:

- HTML sample
- Rich Client sample
- Web 2.0 sample
- JSF sample

Product fix history

This section lists the closed APARs by version 6.1 of the Bank Transformation Toolkit.

Table 1. List of closed APARs against version 5.2

APAR	Description
JR26373	NameNotFoundException is thrown from Context. The JMS function of BTTEvent is turned on by default. So, if JMS queue is not configured in WebSphere Application Server, exceptions are thrown.
JR26375	The *getElementAt()* API is missing from OperationStep.
JR26378	Several GUI problems are restricting the use of the application. When using the DSEGuiBeans for the Java™ client, frequent screen freezes and unexpected behaviour with text fields and combo boxes are encountered.
JR26627	ava.lang.ClassCastException is thrown when using org.apache.struts.ActionMapping and com.ibm.btt.struts.BTTActionMapping together in BTT 5.2.0.

JR26283	SYNC_GET_LOCAL_LU_NAME and SYNC_GET_PARTNER_LU_NAME verbs added to get the LU name from an LU62 connection:
	• Verb SYNC_GET_LOCAL_LU_NAME, used to get local LU name for the LU62 conversation
	• Verb SYNC_GET_PARTNER_LU_NAME, used to get remote LU name for the LU62 conversation.
	For the two new verbs, BTT will call SNA CPI-C APIs: Extract_Local_LU_Name and Extract_Partner_LU_Name by JNI call.
	Note: The two APIs only apply to the Communication Server running on AIX® and Linux® systems.
	The following is the sample code for the two verbs:
	`Lu62InteractionSpec ixnSpec = new Lu62InteractionSpec();`
	`Lu62Record outgoingData = new Lu62Record();`
	`Lu62Record returnData = new Lu62Record();`
	`ixnSpec.setInteractionVerb(ixnSpec.SYNC_GET_LOCAL_LU_NAME);`
	`connection.execute(ixnSpec, outgoingData, returnData);`
	`System.out.println("Local LU name :"+returnData.getData());`

JR26917	Java client side does not support failover in BTT 5.2.0.
JR27142	String leaks in the CHA EJB cache.

Table 1. List of closed APARs against version 5.2 (continued)

APAR	Description
JR27127	In general, there are two ways to manage conversations when sending continuous messages. One is to establish a new conversation for every message. The other is to create a single conversation and reuse it for sending different messages. The second method results in better performance. This APAR provides support for the first conversation mode: the host closes the conversation used by each unsolicited message.
JR27170	This APAR provides two parameters to configure the length of a TID/thread name in trace.
JR27198	In a performance environment, there might be thousands of threads pending, which can cause an out of memory crash in WebSphere Application Server. Each thread is waitOn Semaphore in the `receive()` method. When the connection is terminated, the waiting thread cannot be notified on Semaphore. As a result, it cannot exit. This APAR changes the event notification method to avoid creating too many threads.
JR27406	BTT does not support Opera Browser.
JR27597	BTT provides an extension for WTS to implement the `BTTRequestProcessorUtil.getProcessId (HttpServletRequest request)` method. However, when processing a transaction request, `createProcessorId()` is invoked unexpectedly, which causes the BTT module to replace the original processor ID with a new one. As a result, a new context is created. This causes a memory leak.
JR27322	Referenced objects are reserved after the conversation/ Lu62ManageConnection is terminated. These objects should be deleted, because this causes a memory leak.
JR27685	BTT LU62 JCA parameter "establishConversationRetries" does not work if its value is greater than 0.
JR27954	When BTT JCA deallocates LU conversation, a conversation object must be printed in the trace. However, before printing, the conversation object is set to null. After BTT JCA deallocates an LU conversation, the object in the conversation is accessed. This APAR fixed these two problems.
JR28052	Every time BTT JCA creates a conversation, it loads a library, which is time-consuming.
JR28061	When implementing an invoker class which extends com.ibm.btt.cs.invoker.base.BeanInvokerImpl, the reference to the CHA Instance Id from the HttpSession will be removed and the HttpSession will be invalidated. So when the BTT client application calls the logoff operation (BTTServerOperation) on the server, the IllegalStateException from http session will be thrown.
JR28183	Request WAS pool to destroy BTT JCA connection immediately.
JR28225	JCA Performance issue: Exception is thrown when server stops.
JR27898	Missing header message when LU62 Message has multiple frames.
JR28208	Request to add parameter in BTT LU0 JCA to control whether to send TERMSELF after timeout.
JR28209	Change Request: a new custom property "sendInitSelf" is provided in order to disable sending init-self. The default value is true. Set it to false to disable sending init-self.
JR28345	JCAERR00047R timeout problem. Convert the `getListenThreadSem()` method to a synchronized method to prevent JCA timeout problem.
JR28687	BTT JCA supports 64-bit AIX systems.
JR28841	LU API SYNC_GET_ACTUAL_LU_NAME does not work in version 5.2.

Table 1. List of closed APARs against version 5.2 (continued)

APAR	Description
JR28704	Unknown corrupt trace message due to I18N problem.
JR28375	The establishSession is slow for the listen thread because `run()` is pending on `sleep(timeBetweenBidRetry)` if `init()` is not executed.
IY43326f_1	Session down is not correctly recovered when Lu0SnaSession is using a LU that belongs to a Communication Server pool.

Table 2. List of closed APARs against version 4.3

APAR	Description
IZ01114	In the JCA clone environment, a new method in the BTT LU0 service for closing an LU0 session established on anther clone. Currently there is a method ccClose() which uses the user session ID, but customers need the same method using the LU0 name, which is passed as a parameter. A new API ccClose(String luName) is added to Lu0SnaSessionService and Lu0SnaSession class. The method will call Communication Server NOF API deactive_lu0_to_3 to terminate the LU0 session for the specified LU name. On the other clone that has already established a session for the LU, it will receive an UN-BIND message from the host and terminate the session.
JR26363	Add a new method setInitSelfUserData in Lu0SnaSessionService to send user data to the host when BTT LU0 Connector send INIT-Self to establish LU0 session.
JR24998	Cannot establish conversation with host using LU6.2.

Hardware and software requirements

For details of the hardware and software required to set up the Bank Transformation Toolkit development and runtime environments, see Hardware and software requirements in the *IBM WebSphere Multichannel Bank Transformation Toolkit: Installation Guide*.

Limitations, known issues, and workarounds

This section lists the limitations and known issues of this release. It also provides information about any fixes or workarounds that exist for resolving these limitations and issues.

The following limitations have been identified:

- The BTT Trace clause must be in the first paragraph in the BTT definition file (btt.xml). This is because BTT components are initialized in sequence specified in this file and the Trace component must be initialized first before all the other components.
- WebSphere Application Server sometimes reports ″servlet temporarily unavailable″ when you update Java Channel application frequently. The workaround is to restart your application or restart WebSphere Application Server.

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.