



IBM Software Group

WebSphere® Business Process Management V6.2

WebSphere Integration Developer V6.2 Development enhancements overview



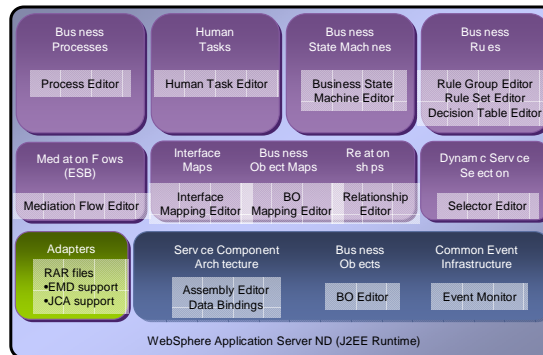
@business on demand.

© 2008 IBM Corporation
Updated December 17, 2008

This presentation provides an overview of what is new in version 6.2 of WebSphere Integration Developer.

V6.2 overview

- What this presentation covers
 - Summary of V6.2 development environment enhancements only
 - Focuses on the edges surrounding server functionality only
 - Example
 - WebSphere Enterprise Service Bus runtime and tool enhancements covered together in WebSphere Enterprise Service Bus presentations



This presentation covers a summary of V6.2 development environment enhancements only. Meaning, it will focus on the edges surrounding server functionality only. For example, WebSphere Enterprise Service Bus runtime and tool enhancements are covered together in WebSphere Enterprise Service Bus presentations.

Agenda

- Enhanced Java™ integration
 - Content assist for SDO API
 - Service discovery for EJB import
- Richer sticky note functionality
- Business calendar editor enhancements
- Test client enhancements
- Business object enhancements
- XSLT mapping enhancements
- Simplified authoring experience
- Platform enablement
- Service version handling
- Server log view
- Integration solution



Licensed Materials - Property of IBM Corp. © Copyright by IBM Corporation and other(s) 2000, 2008. IBM, and WebSphere are trademarks of IBM Corporation. Built on Eclipse is a trademark of Eclipse Foundation, Inc. Java and all Java-based marks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., and all terms are trademarks or registered trademarks in the United States, other countries, or both.

3

WebSphere Integration Developer V6.2 development enhancements overview

© 2008 IBM Corporation

The agenda for this presentation consists of wide array of WebSphere Integration Developer V6.2 enhancements. You will start with enhancements to existing functionality and move directly into new features introduced in version 6.2.

Enhancements start with content assist for coding Service Data Object, or SDO, APIs when getting and setting business object elements adds to current content assist to make coding SDO APIs easier. Service discovery for EJB import makes it easier to add WSDL and Java type EJB imports to a module. Sticky Notes received richer functionality with the addition of task tag highlighting and hyperlinks. The business calendar editor adds several UI changes and an overview tab to make it easier to use. The Test Client adds the ability to import and edit SOAP input for Web service exports. Extended ability for business objects has been added in order to promote private business objects and create abstract business objects. XSLT mapping or XML Maps now has options to work with large data structures easier and adds a new lookup transform to interact with static relationships. The enhancements section finishes with a few simplified authoring experience enhancements and the upgraded platform enablement.

New features for version 6.2 start with service version handling, a deployment concept to version SCA modules and libraries. Then the server log view is introduced. It is a new way to view console and problem determination information, mainly cross-component trace information. You will finish the presentation with learning about Integration Solution, a new feature to organize and display how modules, mediation modules, libraries, and other dependent projects connect.

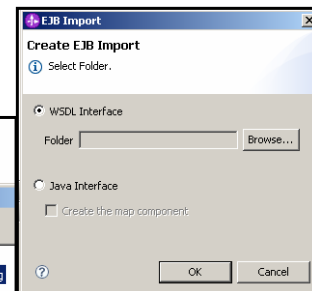
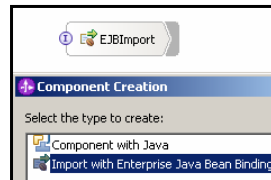
Enhanced Java integration

- Content assist for SDO API
 - ▶ Getting and setting business object elements
 - Adds to current content assist options
 - Speeds coding
 - ▶ Validation added
 - Wrong type
 - Wrong element
- Service discovery for EJB import
 - ▶ Easier to add w-type and j-type EJB imports to module
 - ▶ Drag-and-drop to assembly diagram
 - ▶ External service wizard

```
DataObject testBO = factory.createByElement
String boString = "b" + "oi";
commonj.sdo.DataObject bo1 = factory.create

bo1.setString("stringField", "test");
bo1.getString("stringField");
bo1.getInt("intField");
bo1.setInt("intField", 10);
commonj.sdo.-----appropriate fields of bo1-----
bo2Field.get intField:int
bo2Field.set
```

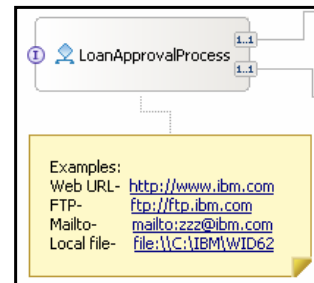
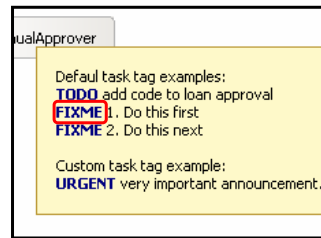
```
bo1.setString("stringField", "test");
bo1.getString("stringField");
The field intFields does not exist in the Business Object {http://m1}bo1.
bo1.setInt("intFields", 77);
```



There are two main enhancements for Java integration for WebSphere Integration Developer version 6.2. The first is content assist for SDO API. This is in addition to the current content assist options, primarily focused on helping integration developers speed coding for getting and setting business object elements. Validation has been added for the case that a wrong type or wrong element is attempted to be used. The second Java integration enhancement is for service discovery for EJB imports. Two ways to automatically populate the assembly diagram with an EJB Import has been added. The first, the ability to drag an EJB Java file onto the assembly diagram will automatically prompt you for the type of EJB import you want to create. The second way is to use the external service wizard. A new Java option has been added to the wizard that will allow you to select the appropriate Java file and prompt you for the type, either WSDL or Java, of the EJB import you want to be generated onto the assembly diagram.

Richer sticky note functionality

- Task tag highlighting
 - ▶ Java > compiler > task tags
 - ▶ Default - TODO
 - ▶ Add custom
- Hyperlinks added
 - ▶ Web URL
 - ▶ Ftp
 - ▶ Mailto
 - ▶ Local file system



Sticky notes, used in four editors of WebSphere Integration Developer, has received richer functionality in version 6.2. Task tag highlighting has been added to provide automatic highlighting when typing a predefined task tag in a sticky note. Custom task tags can be added by navigating to windows, preferences, Java, compiler, task tags in the top menu bar. The TODO task tag is a default. Another useful addition to sticky notes are the ability to have four different types of hyperlinks; Web URL, FTP, Mailto, and to your local file system. The Web URL and FTP hyperlinks bring up your default browser clients, where the mailto hyperlink brings up a new e-mail in your default mail client.

Business calendar editor

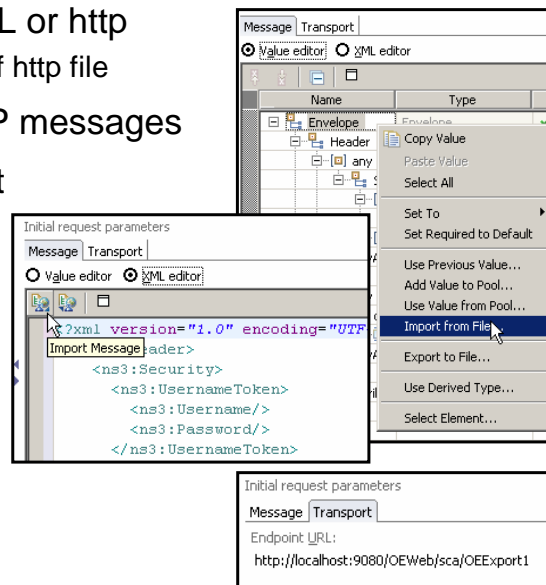
- New default
 - ▶ Positive time must be defined
- Two panes for intervals and exceptions
- Properties > overview
 - ▶ See calendar view



The business calendar editor has been enhanced from its introduction in version 6.1.2. When creating a new business calendar, a time interval, or some positive time, is set by default. Positive time must be defined or you will start with an error. In addition, there is two separate panes for intervals, positive time, and exceptions, and negative time. To get a good view of the net time of a business calendar, an overview tab has been added to the business calendar properties. This enhancement provides you a calendar view of the business calendar where green is intervals, red is exceptions, and blue is the net business calendar “on” time.

Testing Web services export

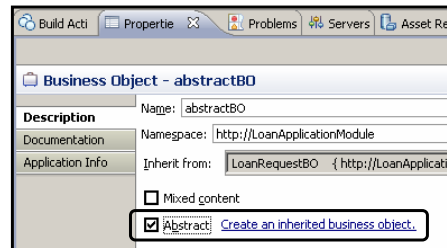
- Able to import from file, XML or http
 - ▶ SOAP message stripped out of http file
- Able to edit raw input SOAP messages
- Full XML editor in test client
 - ▶ Formatting
 - ▶ Syntax validation
- Previously
 - ▶ Could only import XML
 - ▶ Not full XML editor
- Transport tab
 - ▶ Edit WSDL endpoint URL, host name dynamically



The Test Client adds the ability to import and edit SOAP input for Web service exports. Previously, you were able to import XML only. Now you can import from an XML or http file. The http file can be captured from TCP/IP Monitor or from an attach. The SOAP message is stripped out of the http file to be used by the test client as input for the Web service export. Each of the two editors available, the value editor and the XML editor has their own way to import messages. Right click in the value editor to access the “Import from File” option. Use the icon in the XML editor’s menu bar to import messages. In addition, the XML editor inside the test client is a full XML editor normally found in WebSphere Integration Developer. This means you are able to edit raw input SOAP messages and have the formatting and syntax validation found in the XML editor. The transport tab allows you to edit WSDL endpoint URLs and host names dynamically.

Business object enhancements

- Problem: Cannot use private business objects with business object or XML maps
 - ▶ Private business objects = Local elements in business object with anonymous complex type definitions
- Solution: two new enhancements
 - ▶ Promote local element to within same file
 - Move element under the schema and create an element reference
 - ▶ Copy the element to another file
 - Leaves the original untouched
 - Use case: cannot modify original XSD
- Create abstract business objects
 - ▶ Allows for a business object instance to inherit from an abstract business object
 - ▶ Same concept as Java

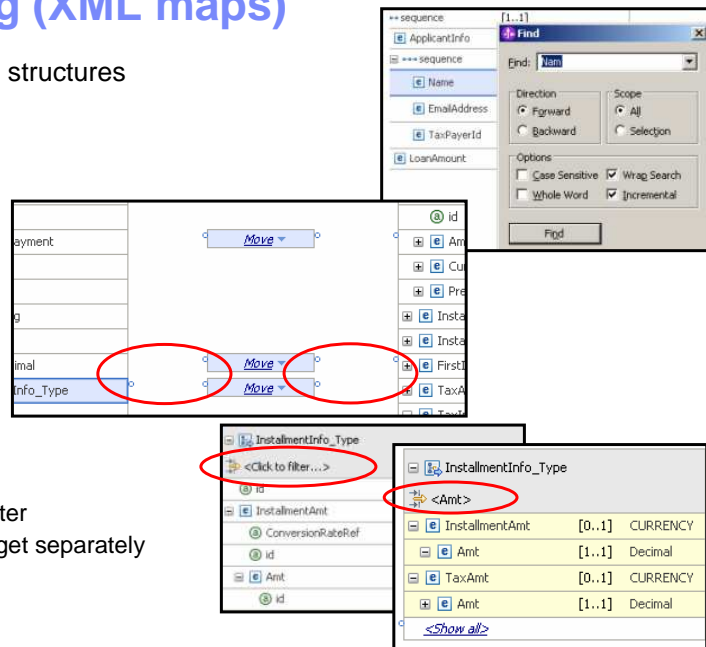


8

There have been several business object enhancements. The first enhancement deals with the issue of not being able to use private business objects with business object or XML maps. Private business objects are local elements in business object with anonymous complex type definitions. WebSphere Integration Developer provides two new enhancements to solve this issue. The first option is to promote local elements to within the same file. This moves the element under the schema and creates an element reference. The other option is to copy the element to another file. This option should be chosen if you want to leave the original XSD untouched. The other enhancement to business objects is the ability to create abstract business objects. This is the same abstract concept as Java and allows for a business object instance to inherit from an abstract business object.

XSLT mapping (XML maps)

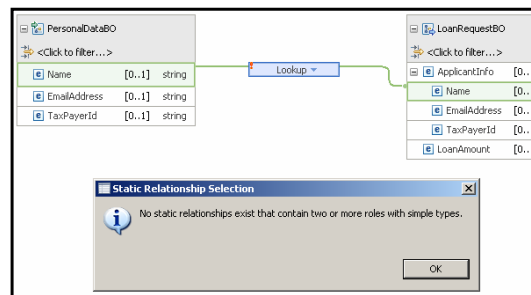
- Work with large data structures easier
- Find support
 - Ctrl-F
- Expand/collapse support
 - Hide complex types with child elements with transformations
 - Transforms stay
 - Become hyperlinks
 - Hyperlink expands to show wiring
 - Dots are anchors
- Filter support
 - String comparison filter
 - Filter source and target separately
 - Hierarchy kept



There are also several enhancements to XSLT mapping, or XML Maps. The first set of enhancements is summed up by allowing you to work with large data structures easier. Find support has been added to easily find source and target elements in large XML maps. Expand and collapse support has been added to hide complex types with child elements that have transformations. Though the source or target complex types can be collapsed the transforms stay and become hyperlinks to expand the complex types to show the map wiring again. The dots are anchors that represent where mapping wires will normally be if expanded. The last of these large data structure enhancements is the filter support to focus on certain elements in the source and target separately. The filter is a string comparison filter that finds elements matching your input. Once entries are found the hierarchy is kept to show you where the matching elements are located in the data structure.

XSLT mapping (XML maps – continued)

- New lookup transform
- To look up a static relationship from an XML map
 - ▶ Relationship definition must be a static relationship
 - ▶ Input and output roles must be simple XSD types



The 'Transform - Lookup' dialog box has a 'General' tab. The title is 'Relationship to call in the WebSphere Process server or WebSphere Enterprise Service Bus relationship service:'. It contains the following fields:

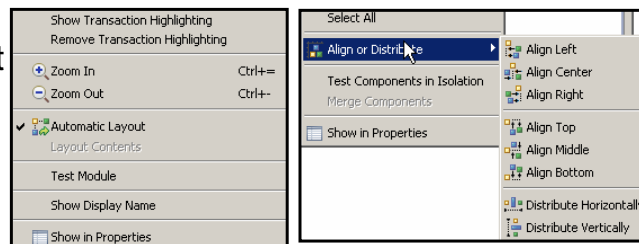
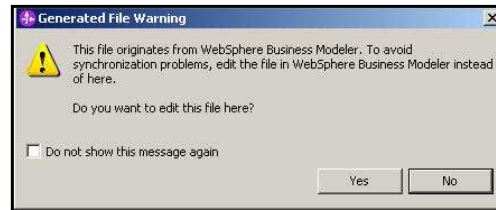
- Static relationship lookup definition: [text field] [Browse...] [Edit...] [New...]
- Input Role Name: [dropdown menu]
- Output Role Name: [dropdown menu]

A new lookup transform has been added to XML maps to allow you to lookup a static relationship in the XML map. The transform converts simple data and uses a Java file created during the build to provide a bridge at runtime, between the XML map and the relationship service. For this transform to work, the relationship definition must be a static relationship and both the input and output roles must be simple XSD types.

Simplified authoring and platform enablement

- Simplified authoring experience
 - ▶ Notify when editing modeler generated artifacts
 - ▶ Automatic layout
 - ▶ Align or distribute

- Platform enablement
 - ▶ Rational® Application Developer V7.5.1
 - ▶ Eclipse V3.4

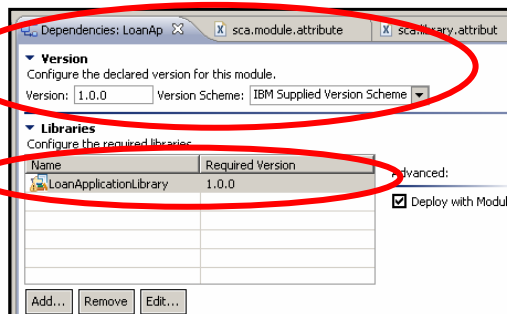


There are also other enhancements to simplify the authoring experience. The first is a warning dialog that opens notifying you when attempting to edit WebSphere Business Modeler generated artifacts. A change here will need the compare/merge feature to get the edit back into Modeler. For the assembly diagram and the mediation flow editor, an enhanced automatic layout has been introduced that automatically adjusts components when dragging and dropping onto the editors and cleans up complex editor components by uncrossing wires and logically placing components. To further organize the editors, selecting several components in the editors allows you to align or distribute components as you see fit.

For platform enablement, WebSphere Integration Developer moves up to being based off of Eclipse 3.4 and Rational Application Developer 7.5.1.

Service version handling

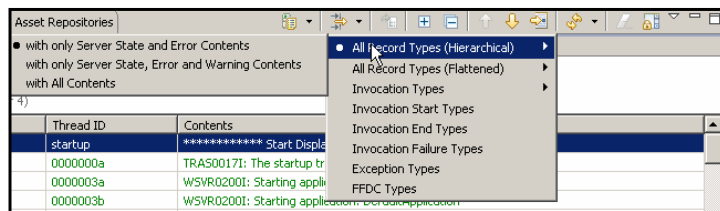
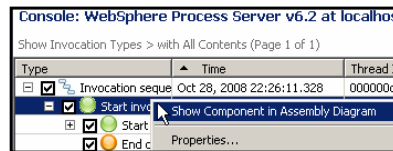
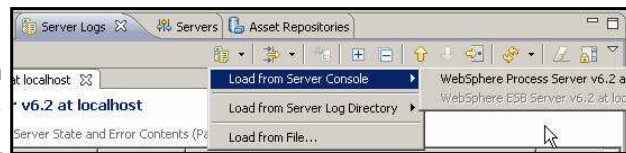
- Deployment concept to version SCA modules and libraries
- Provides first class way to evolve SCA applications
- Version and version scheme set in dependencies editor
 - Test server is not version aware
- SCA bindings only
- Metadata in held in files
 - sca.module.attributes
 - sca.library.attributes
- ServiceDeploy used to create version aware .ear
 - name changes to make versioning possible
 - MyModule
 - MyModule_v1_0_0App.ear
- Administration console
 - View version information
 - Cannot edit
 - Version set once deployed
- Dynamic routing scenario – WebSphere Services Registry and Repository and endpoint lookup mediation
- Unique cell ID added to name for clustered scenario



For the new functionality, service version handling is being introduced in version 6.2. It is a deployment concept to version SCA modules and libraries and provides first class way to evolve SCA applications. WebSphere Integration Developer is used to turn on version handling and to set different versions. Versions and version schemes for the module and libraries are set in dependencies editor, in addition to imports. The test server in WebSphere Integration Developer is not version aware because it is a deployment concept and will cause problems in team environments. Service version handling is currently limited to SCA bindings only. To see the version metadata, open the sca.module.attributes file for modules and sca.library.attributes file for libraries. ServiceDeploy is currently the only way to create a version aware .ear file. Service deploy senses the version and version scheme to make the appropriate name changes to make a version aware application. For example, if you have a module in WebSphere Integration Developer named MyModule, then the version aware .ear file would be MyModule_v1_0_0App.ear. Once a version aware .ear file is deployed to a WebSphere Process Server, you can view the version information in the administration console, but cannot edit. Once deployed, the version for the application is set and should not be edited. However, there is a dynamic routing scenario involving WebSphere Services Registry and Repository and the endpoint lookup mediation used in WebSphere Enterprise Service Bus. For a clustered environment, a unique cell ID is added to the version name so a version can be deployed to a cluster.

Server log view

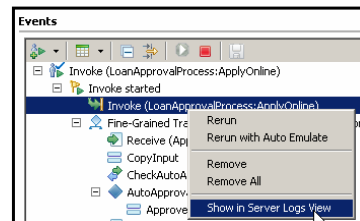
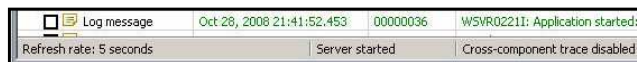
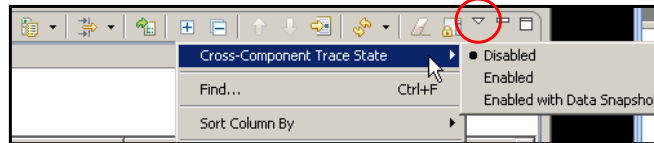
- Replaces console view
 - ▶ Easier to read/filter/search
 - ▶ View log in structured way
 - ▶ Console view still available
- Able to load multiple logs
- Able to open assembly diagram from log content
- More filtering options



A new view introduced in version 6.2 is the server log view. This view replaces the console view and is used to view server console output. This view is easier to read, filter, and search than the console view, though you can still access the console view by clicking on window > show view > console view. There are several things you can do in the server log view that you cannot do in the console view. First, you are able to load multiple logs and traces into the server log view to get a bigger picture of an error. In addition, you are able to open the assembly diagram from log content. The most important difference between the console view and the server log view are the filtering options. The server log view allows you to focus on certain items where the console view provides a list that you have to visually search for items of interest.

Server log view

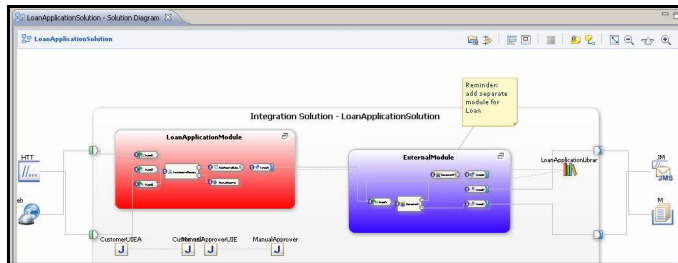
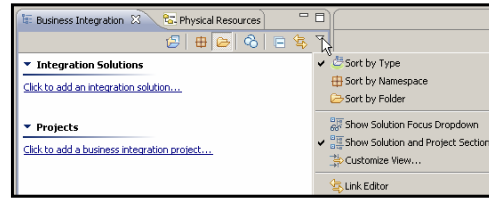
- Cross-component trace (XCT) now in server log view
- Manage XCT settings from development environment
 - ▶ No administration console to turn XCT on or off
 - ▶ Runtime change only
 - Restart will disable
- See status on bottom bar
- Round trip to/from test client
- More filters added
 - ▶ Example: filter XCT records by log message content
- Find support
 - ▶ Find errors, invokes faster



The server log view also replaces the cross-component trace view. The hierarchical view of server log information is the same, but a great enhancement for version 6.2 is the ability to manage cross-component trace, or XCT, settings from development environment. Previously, you would have to open the administration console to turn cross-component trace on or off. However, managing the cross-component trace settings from the server logs view only makes a runtime change, instead of a runtime configuration change. This means that restarting the test server will disable cross-component trace. You are also able to see status on bottom bar of the server logs view: refresh rate of the server logs view (default is five seconds), whether the server is started or stopped, and if cross-component trace is disabled, enabled, or enabled with data snapshot. In version 6.1.2, you can load a cross-component trace into the test client to make an editable, re-runnable scenario. Now in version 6.2, you can round trip to and from the test client with the option in the test client to show certain log entries in the server logs view. More filters for cross-component trace has been added in version 6.2 since it's inception in version 6.1.2. It is worthy to point out the find support in the server logs view for cross-component traces since the find function not only searches the log entries shown in the view, but also the underlying data accessible when double clicking any row in the server logs view. This find support allows you to access errors or certain invokes faster.

Integration solution

- Diagram to display how modules, mediation modules, libraries, and other dependent projects connect
- Business integration view changes
- Development time concept only
 - ▶ Not runtime concept
 - ▶ Eases team environment
- Acts like a viewer
 - ▶ Solution.graphml
 - ▶ Can open editors
- Transaction highlighting



Another new feature in WebSphere Integration Developer is the integration solution. It is a new feature to organize and display how modules, mediation modules, libraries, and other dependent projects connect. Once you open WebSphere Integration Developer version 6.2, you will notice the business integration view is now split between projects, the normal view from previous versions, and integration solutions. The integration solutions section helps users organize multiple modules, mediation modules, libraries, and other dependent projects into one scope. Now this is currently a development time concept only. Meaning, the servers do not know what an integration solution is. The servers still will view an integration solution as a set of .ear files. The integration solution has connections to team environments, like CVS or Clearcase, where a developer can check out all the projects associated with an integration solution in one action. The second and most important part of the integration solution is the diagram. It allows you to visualize how modules, mediation modules, libraries, and other dependent projects connect. Think of the integration solution diagram more like a viewer instead of an editor; though it can open editors, can display transaction highlighting, updates automatically to changes in projects encapsulated in the integration solution, and has several other editor like functions.

Summary of V6.2 enhancements

- Content assist for SDO API for getting and setting business object elements
 - Adds to current content assist to make coding SDO APIs easier
- Service discovery for EJB import
 - Easier to add w-type and j-type EJB imports to module
- Richer sticky note functionality
 - Task tag highlighting and hyperlinks
- Business calendar editor
 - Easier to use with several UI changes and Overview tab
- Test client enhancements
 - Ability to import and edit SOAP input for Web service exports
- Business object enhancements
 - Promote local elements/private business objects and create abstract business objects
- XSLT mapping
 - Work with large data structures easier and new lookup transform
- Simplified authoring experience in main editors
- Platform enablement upgraded to Rational Application Developer V7.5.1 and Eclipse V3.4
- Service version handling
 - Deployment concept to version SCA modules and libraries
- Server log view and cross component trace
 - New way to view console and problem determination information
- Integration solution
 - Organizes and displays how modules, mediation modules, libraries, and other dependent projects connect

In summary, you started with enhancements to existing functionality and moved directly into new features introduced in version 6.2. Enhancements started with content assist for coding Service Data Object, or SDO, APIs when getting and setting business object elements, making coding SDO APIs easier. Service discovery for EJB import makes it easier to add WSDL and Java type EJB imports to a module. Sticky Notes received richer functionality with the addition of task tag highlighting and hyperlinks. The business calendar editor adds several UI changes and an overview tab to make it easier to use. The Test Client adds the ability to import and edit SOAP input for Web service exports. Business objects received extended ability to promote private business objects and create abstract business objects. XSLT mapping or XML Maps now has options to work with large data structures easier and adds a new lookup transform to interact with static relationships. Then enhancements section finished with a few simplified authoring experience enhancements and the upgraded platform enablement. New features for version 6.2 started with service version handling, a deployment concept to version SCA modules and libraries that is set in WebSphere Integration Developer. Then the server log view was introduced. It is the new view for console and cross-component trace information. You finished the presentation with learning about integration solution, a new feature to organize and display how modules, mediation modules, libraries, and other dependent projects connect.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WBPMv62_IEA_WIDWhatsNew.ppt

This module is also available in PDF format at: ..\\WBPMv62_IEA_WIDWhatsNew.pdf



You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

Rational WebSphere

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 other IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

EJB, Java, and all Java-based trademarks and logos are trademarks 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.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. 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 in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

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

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.