Put new capabilities of business activity monitoring (BAM) to work, Part 3: Improved Unit Test Environment in IBM WebSphere Business Monitor Development Toolkit V6.1

Simpler, faster development and testing in an iterative development cycle

Skill Level: Intermediate

Wilfred C. Jamison (wjamison@us.ibm.com)  
Technical Development Manager  
IBM Corporation

Joel H. Maner (jmaner@us.ibm.com)  
Staff Software Engineer  
IBM Corporation

Nabeel W. Abdallah (nabeel@us.ibm.com)  
Advisory Software Engineer  
IBM Corporation

Bret E. Harrison (harrisob@us.ibm.com)  
Advisory Software Engineer  
IBM Corporation

26 Feb 2008

In the first two articles in this series, you learned about the business user experience in Web 2.0 dashboards, iterative development, simplified installation and administration, and improvements to the IBM® WebSphere® Business Monitor V6.1 Installer. This article, Part 3 in the series, demonstrates how much easier iterative testing and developing is within the Eclipse environments, IBM WebSphere Integration Developer V6.1, and IBM Rational® Application Developer V7.0.0.5.
Introduction

IBM WebSphere Business Monitor V6.1 has exciting new features and capabilities that were described in Part 1, "What's new in WebSphere Business Monitor 6.1" (developerWorks, Dec 2007). Part 2 focused on installation improvements. Future articles will focus on specific components of the product.

After you've installed the product, one of your first tasks is to develop monitor models and test them. For this you need the WebSphere Business Monitor Development Toolkit. This article shows you the changes and enhancements to the toolkit, which includes:

- Monitor Model Editor (MME), which is used for authoring monitor models.
- Unit Test Environment (UTE), for publishing and testing monitor models on a monitor test server.

The MME is a big component of the product and warrants a separate article to discuss the enhancements. This article focuses more on what happens after the monitor model has been written and shows an end-to-end perspective of the user experience.

A sample business scenario describes the end-to-end story and walks you through the steps, from authoring the model to testing the monitor models. Along the way, this article highlights improvements in version 6.1. The scenario in this article is used throughout the rest of this series.

Mortgage lending business scenario

The sample scenario is about mortgage lending, one of the major businesses in the financial industry. This sample is also shipped with WebSphere Business Monitor V6.1 and is the same business process model used in the Banking Monitor Model sample for IBM WebSphere Business Monitor V6.0.2 SupportPac.

The majority of house buyers apply for mortgage loans to pay for their houses. Applying for a loan and getting that loan approved is a complex process that most of us probably don't see. The mortgage lending process focuses on the financial business side, which involves evaluation of the loan, approval, financing, and so on. Your interest here is in monitoring how well a mortgage lending business is doing in terms of metrics and indicators, such as number of denied applications, average length of loan application processing, and number of times a loan application has been returned.

Figure 1 shows a high-level view of the mortgage lending business process. The
model, which has been simplified for discussion in this series, shows a sequence of tasks performed for processing mortgage applications. The first task is a major process, because all applications have to be validated for completeness and correctness before proceeding to the subsequent tasks. Setting up loan applications is completely automated and applies complicated rules.

Loan applications that pass through get processed and validated in the second task, which includes assessing the loan applications. Typically, an underwriter is tasked to make the risk assessment or credit worthiness of the applicant. When a loan is approved, it goes through the closing task, when all the legal paper work is completed with the borrower. After closing, a financial institution funds the loan. A post-closing is performed, and the funds are finally disbursed (shipped) to the borrower's account. Human tasks are involved in a majority of the entire process.

**Figure 1. High-level business process model for mortgage lending scenario**

Let's go down one level of detail in the first task. The automated loan setup subprocess in Figure 2 shows what's being done or checked in each step. Annotations indicate failure metrics that are detected. Occurrences of failures are of interest for your monitor model. You'll create a monitoring context for the high-level process and a child-monitoring context for the automated loan setup subprocess.

**Figure 2. Automated loan setup process**
Inbound and outbound events are incorporated next. Figure 3 shows the event flow to and from the automated loan setup monitoring context. Inbound events are used to trigger validation of situations that occur during the loan application process. After validation processing for a situation is complete, an error metric is updated and an outbound event is issued.

**Figure 3. Event flow for automated loan setup process**
Iterative development process

In this section, learn about *iterative development*, the foundation of the UTE. WebSphere Business Monitor model development is a repetition of a series of tasks, as shown in Figure 4. The development life cycle begins with the creation of a business monitoring project and ends when the requirements for the monitor model are satisfied and deemed ready for deployment in a production or production-like environment.

The whole UTE user experience is driven by the iterative development process. Quality is measured by the ease and robustness of executing each step in the cycle as well as the capabilities to support certain user scenarios. (You'll see some limitations of the UTE covered later in this article.) The development process commences when a user starts creating a new monitor model using the MME. (Improvements in the MME, and changes in the monitor programming model, will be covered in a subsequent article.) First you create a new business monitoring project. Creating and editing a monitor model is called *authoring a model*. Users decide when their model is complete, or partially complete, and ready for verification.

**Figure 4. Iterative development process**

First, you must convert a monitor model to a Java™ 2 Platform, Enterprise Edition
(J2EE) application, because the Business Monitor Server component is essentially a J2EE engine. After saving the monitor model, the user generates the J2EE Monitor projects for it in the code generation process. To verify the correctness of the monitor model, the generated J2EE application must be deployed and published in a WebSphere Business Monitor test server. The intention is to send events to the WebSphere Business Monitor test server, while the published model consumes and processes these events. There should be a way to verify whether the events are being processed correctly, and the business dashboard serves this purpose perfectly.

The user configures the dashboard by creating views associated with the monitor model in question. The chosen views are customizable based on a particular purpose. After the dashboard is set up, the user can start executing the test plan. A test plan typically includes different test scenarios that exercise different parts of the monitor model logic. A test scenario has a well-defined purpose and must be repeatable, so it typically starts with a known state of the monitor database. Then a set of events with preset values are emitted in a specific order. With this sequence of events, the user has knowledge of what values to expect in the dashboard based purely on the monitor model logic. By viewing the dashboard and running the test scenarios, the user can verify whether the monitor model is working properly.

Going back to modify the monitor model using the MME is the entry to the next iteration of the development process. The user either corrects problems found after executing his or her test scenarios or adds more to the logic of the monitor model (incremental development). In either case, the user saves the updated monitor model and regenerates the J2EE application. This time, however, a republish of the generated application to the WebSphere Business Monitor test server is automatically executed—a big improvement since the previous version of the product. (Previously, the user had to first remove the published application from the server, then publish the new application.) Automating these procedures definitely increases user efficiency and productivity.

After the application has been published, the user may optionally reconfigure the dashboard views before testing. A simple case is when the user discovers a better way of viewing results. Another is when new test scenarios, which require different views on the dashboard, are added to the test suite. The user then tests the updated monitor model by executing the test scenarios (sending the test events shown in Figure 4).

The user may then go back to the MME to update the monitor model and start a new iteration. The development cycle repeats until the user is satisfied with the monitor model. The efficiency and ease of executing the different stages in this cycle is of utmost importance; it defines the whole end-to-end user experience and influences the development schedule.

The following sections provide more details on how the current WebSphere
Business Monitor Development Toolkit implements the iterative development process.

Configure the UTE

You can install the WebSphere Business Monitor Development Toolkit V6.1 using the IBM Installation Manager on either WebSphere Integration Developer V6.1 or Rational Application Developer V7.0.0.5. This is an improvement from the previous release, especially for pure business activity monitoring (BAM) users. The monitored applications aren't specifically based on WebSphere Integration Developer. The configuration of the WebSphere Business Monitor Development Toolkit isn't much different in both environments.

Default configuration

Figure 5, part a, shows the default server created for Business Monitor Server (also called Business Monitor test server) after installing the toolkit on Rational Application Developer 7.0.0.5. Part b of the figure shows the WebSphere Business Monitor test servers created for WebSphere Integration Developer V6.1, where both IBM WebSphere Enterprise Service Bus V6.1 and IBM WebSphere Process Server V6.1 are installed. Notice how each WebSphere Business Monitor test server is qualified by its underlying WebSphere run time (WebSphere Application Server, WebSphere Enterprise Service Bus, or WebSphere Process Server).

Figure 5. Default servers after installing WebSphere Business Monitor Development Toolkit
The default WebSphere Business Monitor test server configuration on WebSphere Integration Developer is shown in Figure 6 (see a larger version of Figure 6). View this by double-clicking the WebSphere Business Monitor test server in the Server view. In this release, the WebSphere Business Monitor test server is secure by default on WebSphere Integration Developer but not on Rational Application Developer. Everything else is the same for both environments as far as default values are concerned. You can modify the settings and save your changes, though it's encouraged to keep these settings at least for the default WebSphere Business Monitor test servers. The next section explains how you can create your own WebSphere Business Monitor test server.

**Figure 6. Default Business Monitor Server configuration**

Create a new WebSphere Business Monitor test server

At times you might want to create your own WebSphere Business Monitor test server after installing the WebSphere Business Monitor Development Toolkit. It can be useful when you want to use a different run time than is installed on your local system, especially because multiple Business Monitor Servers can now be installed on the same machine. Or, if you want to create a remote WebSphere Business Monitor test server, you need to specify the host name of the target Business Monitor Server.

Figure 7 shows the steps to create a WebSphere Business Monitor test server in this release. The same steps apply for both WebSphere Integration Developer and Rational Application Developer. The Define a New Server window pops up, allowing
you to choose.

Part of the design philosophy is to make the user experience consistent within WebSphere Integration Developer or Rational Application Developer when creating a new server, regardless of which type of test server is being created. The goal is essentially achieved, which is a significant improvement from the previous release.

As shown in Figure 7 (see a larger version of Figure 7):

1. Start by right-clicking the Server view.

2. From the menu that displays, choose New > Server. A window lets you choose the server type based on vendors. You choose IBM if you want to create a new WebSphere Business Monitor test server, and choose among the three supported run times (WebSphere Application Server V6.1, WebSphere Process Server V6.1, WebSphere Enterprise Server Bus) for Business Monitor Server.

3. You can also specify the host name of the server that contains the run time. A remote server is possible. You choose the appropriate run time for your selected server type. If you want to specify a different run time installation that's unknown to WebSphere Integration Developer or Rational Application Developer, click Installed Runtimes, and browse through the file system to point to the home directory of your target run time.

4. Click Next.

Figure 7. Create new WebSphere Business Monitor test server
5. The next window, which lets you specify settings for the chosen run time, detects the profiles that are created in that installation. Make sure you choose the right profile. Choose between RMI and SOAP connections, and specify the right port to communicate with the target server. For better performance, you should click **Run server with resources within the workspace**.

6. If the target server is secure, make sure to reflect that in this window and provide the correct security credentials. Be sure to specify the correct name of the target server; the default is server1.

7. Click **Next**.

8. At this point the server is already created, and you have the option to publish a J2EE application if there is one. When you click **Finish** you should see the newly created WebSphere Business Monitor test server in the Server view.

**Business Monitoring perspective preferences**

Another configuration item is the Preferences page for the Business Monitoring perspective, as shown in Figure 8:

1. Click **Windows** at the top menu of your WebSphere Integration Developer or Rational Application Developer environment, then select **Preferences**.

2. In the window that appears, click **Business Monitoring**.
In the Business Monitoring pane, you see three options:

- **Force code generation when validation errors exist** lets you override the validator if it's selected. Code generation continues even when the validator has detected some potential problems. Be careful when enabling this option; make sure that it's safe to proceed with the code generation even when validation errors are ignored. By default this option is not selected.

- **Prepare for deployment** lets you enable Enterprise JavaBeans (EJB) deployment immediately after code generation is executed. When this option is selected, you don't have to manually execute the Prepare for deployment option after the code has been generated. Prepare for deployment is under the J2EE perspective, and by default this option is selected.

- **Automatically republish project to Server** lets you enable automatic republish. When a J2EE project has already been published and now you've regenerated a new J2EE project for the same model, the currently published J2EE project is removed automatically, and the new J2EE project is published. By default this option is selected. **Note:** You have to select the second option to select the third option.

**Authoring monitor models**

At this point you’re ready to use the WebSphere Business Monitor Development
Toolkit and see how iterative development is accomplished. WebSphere Business Monitor Development Toolkit is intended for IT developers with a good understanding of the monitor programming model.

As mentioned, the initial step in the development cycle is authoring monitor models. You've created the monitor model for the business scenario, mortgage lending. Part of the resulting monitor model is shown in Figure 9 (see a larger version of Figure 9). A snippet of the WebSphere Business Monitor details is shown in part a. Part b shows a snapshot of both the key performance indicator (KPI) and dimensional models. One intention of the model is to monitor the automated loan setup subprocess, so you created a child monitoring context for the overall mortgage lending process. You can download the project interchange of this monitor model to see the details of the monitor model logic.

**Figure 9. Part of monitor model**
It's outside the scope of this article to discuss how to use the MME to author monitor models; see the Resources section for more information.

Now that a monitor model is in place, you can proceed to the next step.

Generate a J2EE application

To generate the J2EE application for the mortgage lending monitor model:

1. Look for the monitor model in the Project Explorer window. In this case, right-click **MortgageLendingBAM.mm** to see the menu, as shown in Figure 10 (see a larger version of Figure 10).

2. Select **Generate Monitor J2EE Projects**, which opens a window that lets you specify the target project names for the generated code. There are three projects generated:
   - **Model Logic Project Name** contains the implementation for the monitor model logic specified in the MME.
   - **Moderator Project Name** contains the implementation for communicating with the Business Monitor Server run time.
   - **J2EE Application Project Name** contains both the Moderator and Model Logic projects.

You can use the default project names that are provided. If these projects already exist and you want to overwrite them, make sure the **overwrite existing projects** option is selected. In this example, the default project names are MortgageLendingBAMModelLogic, MortgageLendingBAMModerator, and MortgageLendingBAMApplication.

**Figure 10. Generating code for mortgage lending monitor model**
3. Click **Finish**, which kicks off the code generation immediately. A progress bar appears that also prints out the tasks that the code generator is executing.

If the second option on the Preferences page is selected, the code generation is followed immediately by the deployment of the generated EJB. If not selected, the code generation terminates successfully with the three created projects.

4. Manually deploy the EJBs using the **Prepare for Deployment** option in the J2EE perspective.

Deployment time was a limitation of version 6.0.2, so automatic deployment is another big improvement in this release. It saves time in terms of changing perspective and several clicks.

Assuming that the generated J2EE application project hasn't been published before, the next step is to publish the J2EE application on the WebSphere Business Monitor test server. If the application is currently published, the next step is no longer necessary.
Publish the generated J2EE application

Publishing, which has improved significantly in version 6.1, is now consistent with the way you publish regular J2EE applications using WebSphere Integration Developer or Rational Application Developer (see Figure 11 for an example; a larger version of Figure 11 is also available).

Figure 11. Publishing the generated J2EE application

1. Before you can publish to a server, make sure that it's started. Right-click the server to pull up the menu, and select Add and Remove Projects. A new window displays, as shown in Figure 11. The window is divided into two sections: On the left are J2EE projects that aren't yet published to a server; on the right are J2EE projects that are currently published to this
given server. In this case, choose the
MortgageLendingBAMApplication J2EE project on the left side, then
click Add. That project is moved to the right side.

2. Click Finish, and the J2EE application is published to the given
WebSphere Business Monitor test server. The Console view appears,
showing the progress while the application is being published. When the
publication is finished, MortgageLendingBAMApplication appears in the
Server view under the selected WebSphere Business Monitor test server.
Expand MortgageLendingBAMApplication to see the other two
projects, MortgageLendingBAMModelLogic and
MortgageLendingBAMModerator.

As you can see, deployment and publishing in WebSphere Business Monitor V6.1
has improved significantly, reducing these tasks to essentially two steps.

Test and verify

Now you need to test and verify the published WebSphere Business Monitor model.
Before you can do this, make sure that the MortgageLendingBAMApplication is in a
Started state. Presumably, you've created your test suite at this point (a written set
of test scenarios).

For the first iteration, you need to configure the business dashboard to create some
views to be used for verification:

1. Right-click the Monitor test server, and select WebSphere Business
Monitor Dashboard.

2. If you're using WebSphere Integration Developer V6.1, enter a user name
and password. The Getting Started home page appears.

3. Click the Dashboard tab, then click New within that tab, as shown in
Figure 12.

4. You're prompted for a name for this dashboard. Call it Test.

5. In your Test dashboard, you need to create views. For simplicity, create a
simple Instances view where you can see all instances of events that you
pass to the WebSphere Business Monitor test server and that are
consumed by the monitor model. Click Add to Dashboard, and you're
prompted with the type of item or view to be added on this dashboard.

6. Select the Instances view, then click OK. You now have an Instances
view, as shown in Figure 12.
7. The view needs to be configured for the specific monitor model that's being verified. Click **Personalize**. A window appears where you can select the metric names whose values you want to appear in the Instances view, as shown in Figure 13 (see a [larger version of Figure 13](#)).

**Figure 13. Configuring an Instances view**
8. Click the right arrow to move them to the Selected side. The refresh rate is the time interval (in seconds) in which the view is updated to reflect the current state of the database. When all this is done, you end up with the view containing the metric names you've chosen with no instances.

Configuring the business dashboard for testing purposes is no different from configuring it for monitoring purposes in production. You can use your final dashboard that you plan to create for your solution as your verification dashboard. Perhaps the only difference is that for testing you want a simple set of views that you can create quickly to reveal potential problems. Some of the main types of problems to catch include:

- Are the events getting through the WebSphere Business Monitor test server at all?
• Are the events being filtered correctly?
• Are the events being correlated to each other correctly?
• Are the events being processed correctly?

The last question is rather general. You need to know if triggers are triggered when they’re supposed to be, metrics are getting the right values and expressions are evaluated correctly, counters and stopwatches are being updated properly, and so on. However, if you know exactly how the model logic should behave, given the current state of the database and the set of events you’re throwing in, you should be able to figure out if something isn’t right by looking at the dashboard you configured.

The business dashboard is a major piece of the product that has been improved significantly. Version 6.0.2 had a very limited set of views to use. With this release, all views in production are available in the UTE as well. Stay tuned for a future article highlighting the improvements in the business dashboard.

Now that the business dashboard is configured, testing can begin by using the Integrated Test Client (ITC). The ITC is a convenient new tool that can be used to send sample events to the WebSphere Business Monitor test server. This feature definitely expedites the testing process because the tool is already integrated, and you can create your sample events on the fly using an easy graphical interface.

The ITC is invoked directly from the monitor model; an instance of ITC is associated with a specific monitor model.

1. Right-click the monitor model in the Project Explorer.

2. From the pull-down menu, click Launch Integrated Test Client. An ITC instance for the MortgageLendingBAM model is shown in Figure 14 (see a larger version of Figure 14).

Figure 14. Integrated Test Client
A big advantage of using the ITC is that it automatically detects all defined monitoring contexts and all inbound event definitions under each monitoring context for the given monitor model. Thus, you can navigate the pull-down menus to pick the inbound event definition you want to create a sample event for. ITC automatically shows the fields in either the Extended Data Element section of the event (if it’s a version 6.0.2 style event) or the Event Part Details (if it’s a version 6.1 style event). All you need to do is fill out the forms to create a sample event. (See the Information Center in the Resources section for more information about version 6.0.2 and version 6.1 style events.)

A test script is a sequence of commands where a command can be an emission of an event that you created, an indefinite pause, a pause for a certain period of time, or an emission of events that are persisted in an XML file in the file system.

3. Click Emit to start executing the test script, as shown in Figure 14.

The execution results for creation of Common Base Event (CBE) events are sent to the WebSphere Business Monitor test server where your WebSphere Business Monitor model was published. You can run your script repeatedly just by clicking Emit. You can also save it into your file system for future use by the ITC. Figure 15 shows the state of the Instances view after sending two events using the ITC (see a
The UTE also provides other tools that you can use for verification, including the CBE browser that you can invoke from the WebSphere Business Monitor test server pull-up menu. **Note:** Part 4 of this series will describe the capabilities of the ITC and other tools you can use in conjunction with the ITC.

With this tool, you can verify whether the CBE events that you sent were actually received by the Common Event Infrastructure (CEI) server. You can view the details of each CBE in the event database.

**Modify and republish monitor models**

The next iteration starts, after testing and verification, when you go back to edit the monitor model using the MME. For example, you've seen that events are being received properly, which is the only type of verification you're interested in. Next you want to extend the monitor model by adding more metrics and the corresponding mapping expressions for them. Then you save the modified monitor model to republish it.

Unlike in version 6.0.2, you no longer need to stop the currently published monitor model and remove it from the WebSphere Business Monitor test server. In version 6.1, it takes just one step:

1. Right-click the monitor model in the Project Explorer, and select **Generate Monitor J2EE Projects** from the pull-down menu. Figure 16 shows what happens with this action (see a larger version of Figure 16).

**Figure 16. Republishing a WebSphere Business Monitor model**
2. A window appears where you specify the names of the generated J2EE projects. Ensure that the names are the same names you specified in the previous iteration and that you select **overwrite existing projects**.

3. In the upper-left corner of Figure 16, notice that the MortgageLendingBAM project is currently published in the WebSphere Business Monitor test server. As soon as you select **Finish**, the progress bar appears and the MortgageLendingBAM project disappears from the Server view, indicating the project has been removed successfully from the test server. This also means that the monitor tables for this model are dropped from the monitor database. The progress bar then indicates the generation of the projects and eventually the deployment of the EJBs. Finally, it publishes the generated projects in the test server, creates a new set of monitor tables in the monitor database, and indicates that the task has finished. In the lower-left corner of Figure 16, the MortgageLendingBAM project is back in the Server view. This process reflects a major improvement in generating projects.

From here, you can immediately go back to testing and verification. In this example, you might need to reconfigure the dashboard to add the new metrics in your
Instances view, then you can go back to your ITC to send events. Before that, you may need to edit the sample events because you added new metrics. After reviewing the Instances view, you may decide to start a new iteration by going back to the MME.

This section described two iterations of the development cycle.

Limitations

IBM WebSphere Business Monitor Development Toolkit V6.1 has by no means reached its Nirvana. There are still some areas to be improved in the overall end user experience, including:

- Improved code generation time for large monitor models.
- Adoption of background builders, which eliminates the need for a modal dialog for code generation.
- A debugging capability for troubleshooting monitor model logic problems.
- Support for monitor model versioning.
- Option to preserve the current state of the monitor database when moving to the next iteration.
- Ability to edit imported events in the ITC.
- Better progress indicator for the ITC.

There's always room for improvement, and the team will keep raising the bar higher for the next release. Our quality is benchmarked by the ease and efficiency in which each step in the development process is carried out.

Summary
The updated UTE provides a simpler, more streamlined and well-integrated development environment that reduces the time required to develop and test your business models in an iterative development cycle. To summarize, areas of improvement, and new features, include:

- Support for multiple runtime environments. The WebSphere Business Monitor Server can now be installed as a test server on three run times: WebSphere Application Server V6.1, WebSphere Enterprise Server Bus V6.1, and WebSphere Integration Developer V6.1. The WebSphere Business Monitor Development Toolkit can now be installed in either Rational Application Developer V7.0.0.5 or WebSphere Integration Developer V6.1. The minimum runtime environment requirement is WebSphere Application Server V6.1.

  Automatic creation and configuration of embedded test servers for supported server types includes WebSphere Application Server, WebSphere Process Server, and WebSphere Enterprise Server Bus using Rational Application Developer or WebSphere Integration Developer.

- Predefined server templates that simplify creation and customization of all supported test server types.

- Optional project publishing settings to automatically deploy and republishing applications using a single-step process.

- ITC for defining and emitting test model events.

- An easy-to-use and production-like business dashboard to verify results when testing your monitor models.

- Context-sensitive menus in applications to test and verify the correctness of your model event processing, such as CBE browser, monitor business dashboards, and Alphablox Admin Console.

Stay tuned for Part 4 of this series for more details on the capabilities of the Integrated Test Client.
## Downloads

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Size</th>
<th>Download method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mortgage lending project sample</td>
<td>mortgage_lending.zip</td>
<td>36KB</td>
<td>HTTP</td>
</tr>
</tbody>
</table>

*Information about download methods*
Resources

Learn

• Check out the other parts of this series:
  • Part 1, "What's new in WebSphere Business Monitor 6.1"
    (developerWorks, Dec 2007)
  • Part 2, "WebSphere Business Monitor 6.1 installation improvements"
    (developerWorks, Jan 2008)

• Visit the IBM WebSphere Business Monitor V6.1 main site.

• Browse the IBM WebSphere Business Monitor SupportPacs InfoCenter to learn
  more about the banking sample.

• Browse the IBM WebSphere Business Monitor Information Center.

• Learn more about WebSphere Application Server V6.1.

• Learn more about WebSphere Portal for Multiplatforms V6.0.

• Browse the technology bookstore for books on these and other technical topics.

Get products and technologies

• Download BA76: WebSphere Business Monitor - Banking Monitor Model, which
  is the original source of the mortgage lending sample used in this series.

About the authors

Wilfred C. Jamison
Dr. Wilfred C. Jamison is currently a manager and team lead for the Code
Generation, Validator, and UTE components of WebSphere Business Monitor. He
has been with IBM for eight years and worked on products including IBM Firewall and
IBM WebSphere Application Server. His expertise includes network security,
performance analysis, concurrent and distributed systems, J2EE, Linux, and others.
He also worked on incubation projects for the On-Demand Software Development
team. He has worked with customers found on the New York Stock Exchange, eBay,
and Toronto Dominion Bank.

Joel H. Maner
Joel Maner works on the development team for the Code Generation, Validator, and
UTE components of WebSphere Business Monitor. He has over eight years of industry experience with WebSphere and J2EE.

---

Nabeel W. Abdallah
Nabeel Abdallah works on the development team for the Code Generation, Validator, and UTE components of WebSphere Business Monitor.

---

Bret E. Harrison
Bret Harrison works on the development team for the Code Generation, Validator, and UTE components of WebSphere Business Monitor.

---

Trademarks

IBM, the IBM logo, Rational, Redbooks, and WebSphere are registered trademarks of IBM in the United States, other countries or both.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.