Understanding IBM Rational Automation Framework

Key concepts and history

David Brauneis
Lewis Shiner

November 13, 2012

This article introduces the basic concepts behind IBM Rational Automation Framework and documents the history of its releases.

Overview

You use the IBM® Rational® Automation Framework software to automate the following common enterprise tasks:

• Application and application-artifact deployment
• Configuration management
• Installation and patching of IBM® WebSphere® and other IBM software

Rational Automation Framework uses data abstraction to store configuration information in a version-agnostic format. Parallel execution and data abstraction give it a degree of accuracy, consistency, and speed far greater than you can achieve by scripting alone.

As of version 3.0.0.4, Rational Automation Framework contains 100 automation libraries and more than 1200 actions. You can assemble these building blocks into many different automation solutions to fit a wide variety of business needs. Rational Automation Framework is built on an enterprise-level automation engine, IBM® Rational® Build Forge®, and supports numerous WebSphere and other products on multiple platforms, as shown in Table 1. For the latest information, see the list of supported automation targets cited in Related topics.

Table 1. Supported automation targets

<table>
<thead>
<tr>
<th></th>
<th>AIX</th>
<th>HP-UX</th>
<th>Linux and zLinux</th>
<th>Solaris</th>
<th>Windows</th>
<th>z/OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM HTTP Server</td>
<td>Install Configure</td>
<td>Install Configure</td>
<td>Install</td>
<td>Install</td>
<td>Install Configure</td>
<td>Configure Deploy</td>
</tr>
<tr>
<td>Version 6.0, 6.1,</td>
<td>Deploy</td>
<td>Deploy</td>
<td>Deploy</td>
<td>Deploy</td>
<td>Deploy</td>
<td>Deploy</td>
</tr>
<tr>
<td>7.0, 8.0, 8.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WebSphere Application Server</td>
<td>Install Configure</td>
<td>Install Configure</td>
<td>Install</td>
<td>Install</td>
<td>Install Configure</td>
<td>Configure Deploy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Deploy</td>
<td>Deploy</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rational Automation Framework also provides support for interim fixes, fix packs, and feature packs that you can apply to installed WebSphere and other products. Further, it has a growing IBM Business Partner ecosystem to provide support for additional middleware targets through extensions built on the Rational Automation Framework backbone. These partner extensions provide the same interface and the same quality of service to customers:

- MidVision Limited, a leading IBM partner, provides extensions that automate the following environments:
  - WebSphere MQ
  - WebSphere Message Broker
  - Oracle WebLogic Application Server
  - Red Hat JBoss Application Server.
- Ascendant Technologies, also an IBM partner, provides extensions for the following environments:
  - WebSphere Commerce
  - WebSphere DataPower SOA Appliances
  - WebSphere BPM V7.5 and later versions

**Environment generation**

A configuration environment is a logical container on the framework server, where Rational Automation Framework configuration information is stored. The environment contains one or more configurations, usually representations of WebSphere cells.

You can use the Environment Generation wizard to create these configuration definitions in either of two ways:
• The new cell version of the wizard interactively defines the topology of a new WebSphere product cell. The wizard creates an automation plan that installs the WebSphere product and creates and federates the associated profiles. You also have the option to install IBM HTTP Server, plug-ins, and other features.

• The existing cell version of the wizard uses a minimal set of information to discover the topology of an existing WebSphere cell and build a representation of that cell on the framework server. The wizard also creates an automation plan to import and abstract nearly all of the configuration information in the existing cell.

**Modes of operation**

Technically, an action is a target element within an Apache Ant build file. Functionally, it is a unit of work that makes up a part of a larger automation process. You can use actions to call scripts, executable programs, or other actions.

Rational Automation Framework supports several modes of operation. Changing the mode of an action changes its behavior. Because the actions are designed to handle a specific portion of the configuration, an action can be used to pinpoint the precise scope at which you want to operate. Table 2 shows the modes of operation that are supported.

### Table 2. Supported modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Purpose</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import</td>
<td>Capture configurations</td>
<td>Actions extract configuration data from a live configuration (usually a WebSphere cell) into the associated configuration definition.</td>
</tr>
<tr>
<td>Execute</td>
<td>Apply stored data to a live configuration</td>
<td>Actions apply configuration data from a framework server configuration definition to a live configuration (usually a WebSphere cell).</td>
</tr>
<tr>
<td>Augment</td>
<td>Add resources to a live configuration</td>
<td>Actions add configuration elements to a live configuration without overwriting.</td>
</tr>
<tr>
<td>Compare</td>
<td>Compare stored data to live data</td>
<td>Actions report on the differences between the data in a framework server configuration definition and a live configuration.</td>
</tr>
<tr>
<td>Promote</td>
<td>Update one configuration definition with data from another</td>
<td>Actions copy requested data from one configuration definition on the framework server (such as a test environment) to another (such as a staging environment).</td>
</tr>
</tbody>
</table>

**Import mode**

Imported configuration data is stored in a version-agnostic and normalized XML format on the framework server. This configuration data can be version-controlled, so that you can apply change management in the same way that you would to source code. Similar to WebSphere Application Server, where many resource types are stored in a single XML file, Rational Automation Framework stores related resources in individual XML files. Resources for some products are stored at an even higher granularity than those for WebSphere Application Server. For example, data sources, Java Database Connectivity (JDBC) providers, and WAS40 data sources are all...
stored in the jdbc.xml file. Additionally, a jdbc.xml file is created at each scope where an import mode command is run.

**Figure 1. Import mode**

```
<table>
<thead>
<tr>
<th>Framework server</th>
<th>Target system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import Refresh data from live cell</td>
<td></td>
</tr>
</tbody>
</table>
```

**Execute mode**

After capturing configuration information in import mode, you can play back the data by running the same configuration action in execute mode. Execute mode first deletes all configuration elements of the given type and then recreates them based on the information stored in the Rational Automation Framework configuration definitions.

**Figure 2. Execute mode**

```
<table>
<thead>
<tr>
<th>Framework server</th>
<th>Target system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute: Run an action</td>
<td></td>
</tr>
</tbody>
</table>
```

**Augment mode**

The augment process is similar to additive and does not update existing configuration elements. Actions check to see if an element to be added exists. If so, a warning message is displayed and the element is skipped. Details of the resources to be augmented are specified in a file named type_augment.xml; for example, jdbc_augment.xml for JDBC-related resources.

**Figure 3. Augment mode**

```
<table>
<thead>
<tr>
<th>Framework server</th>
<th>Target system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Augment: Add data to existing config</td>
<td></td>
</tr>
</tbody>
</table>
```

**Compare mode**

Over time, differences can accumulate between the live configuration and the configuration definition. This situation is called “configuration drift.” Although most companies’ development and operations staff diligently attempt to keep their staging and production environments in sync, it’s inevitable that a change is introduced that goes unnoticed. A common source of such rogue changes is an attempt to debug an issue in production. When a critical application is down at 2:00 a.m., it is possible that not all changes made while attempting to bring the application back online are documented, or reverted if that is not required. In the short term, the changes in these applications might seem harmless, but after you deploy a new application, it begins to behave
differently in your production environment than it does in your staging environment. Alternatively, a difference might occur between data centers in different time zones. A change in the "heat of battle" might seem innocuous at first, but when it begins to cause problems, you are left to find the needle in a haystack.

Compare mode logs the differences between the configuration definition on the framework server and the corresponding configuration in the live cell. Using the Rational Automation Framework automation engine, you can schedule these comparisons during off-peak hours and run them regularly. If a change is caught early, it is much easier to make an informed decision about whether the change should be accepted, rather than trying to recall why a change was made six months after the fact.

When a difference is flagged, you have two choices:

- Update the configuration definition with the new value by running the corresponding action in import mode.
- Reapply the configuration definition by running the corresponding action in execute mode. This process recreates the configuration according to the values saved on the framework server.

**Figure 4. Compare mode**

![Diagram of Compare mode](image)

**Promote mode**

As an application and its configuration changes move along the development lifecycle, great care must be taken to move the related resources from one stage to the next. For example, the data sources required by a new application must move from the shared test environment into the staging environment when the application moves. Although it is desirable to keep almost all of the data the same (such as the database and Java Naming and Directory Interface, or JNDI, names), there are other values that should change from one environment to another. For example, it is unlikely that you would want the data sources in the test environment to point to your production database server.

Rational Automation Framework Promote mode was designed to handle this specific challenge. Promote mode acts as a pull operation and moves the data from one configuration definition into another. To use a typical example, you might pull configuration data from the shared test environment into staging. However, bidirectional promotion is supported. This is useful if you find a problem in your production configuration and want to promote that same configuration to your test server to debug the issue.

You can also change scope during promotion. For example, if you work in a stand-alone, single server environment and create data sources at the server scope, then during promotion into a
Network Deployment clustered shared test environment, you can move the data sources to the appropriate cluster scope.

To handle known environmental differences, Rational Automation Framework passes promoted data through a filter. The filter understands two types of expressions:

- The first is a simple text-based replacement that uses matching specified in promote.properties files at both the origin and destination scopes. The files must contain the same keys. In the stand-alone server to Network Deployment cluster promotion example above, the keys in each file are:
  - Stand-alone configuration at the server scope: myKey=ABC
  - Network Deployment configuration at the cluster scope: myKey=XYZ
  When the promote mode action is run, it finds the matching keys and changes all occurrences of ABC to XYZ. This approach is powerful, but it does not enable fine-grained control.

- The other type of filter is based on XPath expressions. In much the same way that you use SQL for relational databases, you use XPath to navigate to various places in an XML document. For XPath keys, you specify the key and value in the promote.properties file for the destination scope only. Two keys are created for each substitution:
  - One with the .xpath suffix that contains the XPath expression
  - One with a .value suffix that represents the new value

Example:

```
myExpression.xpath=/DataSource[@jndiName="myDS"]/diagnoseConnectionUsage
myExpression.value=true
```

**Figure 5. Promote mode**

![Promote mode diagram]

**Early history**

Rational Automation Framework began as an Ascendant Technology consulting services asset named the WebSphere Installation and Configuration Automation (WICA) framework. This application was used to automate the installation, configuration, and deployment of large WebSphere environments.

Development began around 2005 at Ascendant Technology, an IBM Business Partner, in the context of customer engagements. The initial design included an extensible framework based on Apache Ant, and that overall design remains in place today. IBM subsequently acquired the Ascendant Technology WICA asset, which became the starting point for the initial release of

Release history

Initial release: Version 7.1.1, May 2009

Rational Automation Framework for WebSphere Version 7.1.1 supported automated application deployment, configuration management, and installation and patching for the following WebSphere products:

- IBM HTTP Server 6.0 on Windows, Linux, z/Linux, and UNIX
- IBM HTTP Server 6.1 on Windows, Linux, z/Linux, and UNIX
- IBM HTTP Server 7.0 on Windows, Linux, z/Linux, and UNIX
- WebSphere Application Server 6.0 on Windows, Linux, z/Linux, and UNIX
- WebSphere Application Server 6.1 on Windows, Linux, z/Linux, and UNIX
- WebSphere Application Server 7.0 on Windows, Linux, z/Linux, and UNIX
- WebSphere Portal 6.0 on Windows, Linux, z/Linux, and UNIX
- WebSphere Portal 6.1 on Windows, Linux, z/Linux, and UNIX

This release supported the import, execute, and compare modes of operation.

Version 7.1.1.1, August 2009

This fix pack included defect fixes and some minor enhancements.

The fix pack added enhancements for the automated deployment of applications and application artifacts, including support for deploying loose files that are related to the application (rules, configuration, or properties files).

Support was provided for the latest fix pack versions of WebSphere products.

Version 7.1.1.2, November 2009

This fix pack included defect fixes and some minor enhancements.

The fix pack added improvements for deploying applications, including a finer-grained method of specifying deployment options. Additionally, support was added for the following WebSphere products:

- IBM HTTP Server 6.0 on z/OS
- IBM HTTP Server 6.1 on z/OS
- IBM HTTP Server 7.0 on z/OS
- WebSphere Application Server 6.0 on z/OS
- WebSphere Application Server 6.1 on z/OS
• WebSphere Application Server 7.0 on z/OS

Support was provided for the latest fix pack versions of WebSphere family products.

**Version 7.1.1.3, January 2010**

This fix pack included defect fixes and some minor enhancements.

The fix pack added enhancements to allow Windows centralized servers to manage heterogeneous WebSphere environments, including those running on Windows, Linux, z/Linux, UNIX, and z/OS.

Support was provided for the latest fix pack versions of WebSphere products.

**Version 7.1.1.4, April 2010**

This fix pack included defect fixes and some enhancements.

The fix pack included integration with WebSphere CloudBurst Appliance. This change made it possible to combine the quick provisioning and deployment of WebSphere patterns by WebSphere CloudBurst Appliance with the customization of the variable portions of the cell by Rational Automation Framework for WebSphere (including actions for resource configuration, application deployment, and so forth). Additionally, the fix pack added support for the following WebSphere products:

- WebSphere Portal 6.0 on z/OS
- WebSphere Portal 6.1 on z/OS

Support was also provided for the latest fix pack versions of WebSphere family products.

This release included several usability improvements:

- Augment mode was added so that customers could add resources to a configuration. This mode is especially valuable when dealing with cloud-based or standardized WebSphere images.
- Promote mode was added to help manage configuration along the software lifecycle, ensuring consistency from environment to environment.
- The existing cell option for the Environment Generation wizard was added to bring existing WebSphere configuration under control more easily.
- Super-composite actions were added so that customers could collect all the configurations in a scope for a WebSphere cell into a single call.
- The usability of command line options was improved by the addition of long arguments and the ability to pass optional parameters.

**Version 7.1.2, October 2010**

This modification included defect fixes and some enhancements.
Version 7.1.2 improved performance and scalability for agentless media transfer, agentless initial data transfer, and core automation functions such as scheduling and purging. Additional improvements were made to the logging facility and the installation experience.

The modification added support for the following WebSphere products:

- WebSphere Portal 6.1.5 on Windows, Linux, z/Linux, and UNIX
- WebSphere Portal 6.1.5 on z/OS
- WebSphere Process Server 6.2 on Windows, Linux, z/Linux, and UNIX
- WebSphere Enterprise Service Bus 6.2 on Windows, Linux, z/Linux, and UNIX
- WebSphere Service Registry and Repository 6.2 on Windows, Linux, z/Linux, and UNIX
- WebSphere Service Registry and Repository 6.3 on Windows, Linux, z/Linux, and UNIX
- WebSphere Virtual Enterprise 6.1 on Windows, Linux, z/Linux, and UNIX
- WebSphere Application Server Feature Packs on Windows, Linux, z/Linux, and UNIX:
  - WebSphere Application Server 6.0: Web 2.0
  - WebSphere Application Server 6.1: Web 2.0, EJB3, Web Services
  - WebSphere Application Server 7.0: Web 2.0, CEA, XML, SCA, and OSGi and JPA

Support was provided for the latest fix pack versions of WebSphere products.

Product extensibility was improved in the following ways:

- The process of writing custom actions that use Java and Jython code was simplified.
- Custom environment templates were added.
- An extender's guide was added to the information center.

**Version 7.1.2.1, January 2011**

This fix pack included defect fixes and some enhancements.

The fix pack added support for JDK 1.6 for both the framework server and target systems.

With this fix pack, a resource adapter archive (.rar) file must be deployed at some scope within WebSphere Application Server before the framework server can manage the configuration of resource adapters.

Support was provided for the latest fix pack versions of WebSphere products.

**Version 7.1.2.2, April 2011**

This fix pack included defect fixes and some enhancements.

The fix pack included support for Red Hat Enterprise Linux 6 Server as a platform for the framework server.

IBM Business Partner MidVision developed and made available from IBM Passport Advantage the following new extensions to Rational Automation Framework for WebSphere:
• MidVision Extension to IBM Rational Automation Framework for WebSphere for WebSphere Connectivity. This extension supports the following middleware targets for automation:
  • IBM WebSphere MQ
  • IBM WebSphere Message Broker
• MidVision Extension to IBM Rational Automation Framework for WebSphere for Oracle WebLogic Application Server. This extension supports Oracle WebLogic Application Server as a target for automation.
• MidVision Extension to IBM Rational Automation Framework for WebSphere for Red Hat JBoss Application Server. This extension supports Red Hat JBoss Application Server as a target for automation.

Support was provided for the latest fix pack versions of WebSphere products.

**Version 3.0, September 2011**

This service release included a name change to Rational Automation Framework, as well as defect fixes and enhancements.

With version 3.0, Rational Automation Framework became an independent product with no purchase dependency on Rational Build Forge.

The release included a new Eclipse client for the product supporting configuration, automation plans, and cross-cell comparison.

Enhanced modes and actions were enabled for the following automation targets:

• WebSphere Virtual Enterprise
• WebSphere Process Server
• WebSphere Enterprise Service Bus

Import mode was supported for deployment actions.

Support was provided for the latest fix pack versions of WebSphere products.

**Version 3.0.0.1, December 2011**

This fix pack included defect fixes and some enhancements.

Support was added for WebSphere Application Server Version 8 configuration and deployment actions.

Eight new actions were added for the service integration bus, replacing the single was_common_configure_sibus action with more specific actions.

The existing cell Environment Generation wizard was modified to produce three automation plans instead of one:

• RAFW_env_cell_import
• RAFW_env_cell_compare
• RAFW_env_cell_execute

The Java SDK, Build Forge Automation Engine, Apache Tomcat, and PHP versions were updated to provide enhanced security.

Augment mode was changed to allow the update of existing resources.

Promote mode now used token-based replacement.

Support was provided for the latest fix pack versions of WebSphere products.

**Version 3.0.0.2, April 2012**

This fix pack included defect fixes and some enhancements.

The fix pack added support for the following WebSphere products:

• WebSphere Application Server Version 8.0 installation actions
• IBM HTTP Server Version 8.0 installation actions
• IBM WebSphere Portal Version 7.0 installation, configuration, and deployment actions
• IBM WebSphere Virtual Enterprise and Intelligent Management Pack Version 7.0 installation, configuration, and deployment actions

New actions were added to support the following features:

• Federated security
• PMI Request Metrics
• OSGi specifications
• Service integration bus (SIBus) configurations

The ability to use multiple private SSH keys was also added.

The Environment Generation wizard added save and restore capabilities.

Support was provided for the latest fix pack versions of WebSphere products.

**Version 3.0.0.3, June 2012**

This fix pack included defect fixes and some enhancements.

The fix pack added support for the following automation targets:

• WebSphere Application Server 8.5 installation, configuration, and deployment actions
• IBM HTTP Server 8.5 installation and configuration actions
• Stand-alone (non-clustered) WebSphere Application Server proxy servers
• Stand-alone servers in Network Deployment (ND) cells
For customers who also use IBM Workload Deployer (see Related topics), formerly known as WebSphere Cloudburst Appliance, the following capabilities were added:

- A virtual system pattern (VSP) generator in the Environment Generation wizard
- New integration script package parameters to associate a database with a pattern (Database as a Service, or DBaaS)

The following enhancements were also made to the Environment Generation wizard:

- Prompt for IBM Installation Manager version number
- Prompt for the location to install the WebSphere Customization Toolkit for each IBM HTTP Server node

The was_common_configure_all action security actions were removed and replaced by the following actions, where suffix represents a specifier, such as "federated" or "groups":

- was_common_configure_security_suffix
- was_common_configure_security_fileRegistry_suffix

An action command summary was added to the information center.

New versions of Eclipse (V3.6.2 with security patch), JDK (Java 6 SR10-FP1), and RXA (2.3.0.8) were included.

Compare mode output format was streamlined.

Support was provided for the latest fix pack versions of WebSphere products.

**Version 3.0.0.4, August 2012**

This fix pack included defect fixes and some enhancements.

The fix pack added support for the following automation targets:

- WebSphere Portal version 8.0 installation actions

For IBM Workload Deployer customers, information formerly entered as attributes of the script package are now entered in the wizard.

Configuration environments, and their associated cell definitions, can now be placed under source control in Rational Team Concert with built-in support.

Rational Automation Framework server on Linux or UNIX can now be installed by a non-root user.

A change was made to was_common_configure actions that start or stop servers, causing the actions to fail if the specified server fails to start or stop.

The was_common_deploy_update_auto_start action was added to control whether an application is started automatically.
Support was provided for the latest fix pack versions of WebSphere products.

**Version 3.0.0.5, October 2012**

This fix pack included defect fixes and some enhancements.

The fix pack added support for the following automation targets:

- WebSphere Application Server version 8.5, Liberty profile, configuration, installation, and deployment actions
- WebSphere Portal version 8.0, configuration and deployment actions

Integration with IBM Rational Asset Manager was added for asset management. Two scenarios were supported:

- Store configuration definitions as assets
- Install stored applications during VSP deployment in IBM Workload Deployer

Support was provided for the latest fix pack versions of WebSphere products.

**Conclusion**

The key to understanding Rational Automation Framework is automation. Automation begins with actions and ends with a series of automation plans. These plans transfer data back and forth between the configuration definitions on the framework server and the live data on the target systems. This repeatable, accurate, and fast automation infrastructure is inherently suited to your WebSphere software environments and offers enhanced extensibility for those who have already invested in WebSphere scripting.
Related topics

- Browse the Rational Automation Framework overview page, review the features and benefits, explore documentation in the information center, watch these getting started videos on YouTube, and check the Supported products as automation targets list for updates. Also explore these IBM developerWorks resources:
  - Deploying WebSphere software with Rational Automation Framework by Werner Vanzyl (February 2012)
  - The Support Authority: Manage your WebSphere product installations, configurations, and application deployments with Rational Automation Framework for WebSphere by Tim Robertson and Mahesh Rathi, PhD (July 2010)
- For more about automation, read Automating the construction of applications and application environments by Amrhein, Brauneis, Willenborg, and Williamson (developerWorks 2009)
- See the overview page to find out more about IBM Workload Deployer (formerly WebSphere CloudBurst), and read this background article: WebSphere CloudBurst plus Rational Automation Framework for WebSphere by Dustin Amrhein (developerWorks, July 2010).
- Download a free trial version of Rational software.
- Evaluate other IBM software in the way that suits you best.

© Copyright IBM Corporation 2012
Trademark
(www.ibm.com/developerworks/ibm/trademarks/)