

Technical Preview



IBM Rational Asset Analyzer Batch Population into Rational Asset Manager 7.1

Technical Preview



IBM Rational Asset Analyzer Batch Population into Rational Asset Manager 7.1

Contents

Chapter 1. Introduction	1	Comprehensive example	8
Chapter 2. Prerequisites	3	Synchronizing with deleted Rational Asset Analyzer artifacts	9
Chapter 3. Rational Asset Analyzer control file construction	5	Deleting all Rational Asset Analyzer artifacts from Rational Asset Manager.	9
Establishing the connection to Rational Asset Manager.	5	Chapter 4. Deploying and maintaining the integration	11
Uploading all artifacts of a given type (without relationships)	6	Command Line Interface	11
Uploading a specific artifact by attribute (without relationships)	6	Preparing for integration	11
Adding selected relationships to artifacts	6	Performing the integration	11
Adding container artifacts	7	Synchronizing the integration	12
Uploading specific artifact attributes	7	Removing Rational Asset Analyzer environment from Rational Asset Manager	12

Chapter 1. Introduction

The IBM® Rational® Asset Analyzer provides an easy way to populate Rational Asset Analyzer artifacts into IBM Rational Asset Manager 7.1 as assets. This allows users of Rational Asset Manager to consolidate data collected from Rational Asset Analyzer conveniently with other important data stored in Rational Asset Manager. Rational Asset Analyzer accomplishes this by providing an intuitive REST Interface that clients can deploy, as well as an interface to integrate specified artifacts from the REST instance by creating a user-defined XML control file. For more information on the Rational Asset Analyzer REST Interface, refer to the *IBM Rational Asset Analyzer External Interfaces Guide*.

The integration of Rational Asset Analyzer 5.5 to Rational Asset Manager 7.1 is provided as a Java™ application in the **RaaRamIntegration.zip** file. You can extract the **RaaRamIntegration** folder to any selected directory.

Chapter 2. Prerequisites

Before attempting to upload Rational Asset Analyzer artifacts into Rational Asset Manager, you must complete the following installation and verification requirements:

1. Ensure that Rational Asset Manager 7.1 is installed and functions properly.
2. Install Rational Asset Analyzer and verify that it functions properly.
3. Deploy the Rational Asset Analyzer REST Interface Ear file and ensure the XHTML and XML instances are accessible. This file can be obtained from the Rational Asset Analyzer Installation directory at *{Rational Asset Analyzer installation root}/Install/dmhRestEar.ear*

Attention: Deploy the ear file into the same web server instance as Rational Asset Analyzer. For detailed information about the REST Interface, refer to the *IBM Rational Asset Analyzer External Interfaces Guide*.

4. Gather proper authentication information for the following tools:
 - Rational Asset Manager
 - Web server application hosting Rational Asset Manager and Rational Asset Analyzer REST Interface
 - Rational Asset Analyzer
5. Ensure Java Runtime Environment 1.5 or later is installed on the machine used for running the integration

Chapter 3. Rational Asset Analyzer control file construction

The first step for batch Rational Asset Analyzer population into Rational Asset Manager is to configure an XML control file that defines the Rational Asset Analyzer artifacts that should be uploaded into Rational Asset Manager as assets. To upload Rational Asset Analyzer artifacts into Rational Asset Manager from the REST Interface, you must configure the XML control file to obtain an integration that fits your requirements. This must be done because a Rational Asset Analyzer instance typically contains a large number of artifacts, and in most cases a user does not need access to all of these artifacts.

Establishing the connection to Rational Asset Manager

To establish a connection to Rational Asset Manager, you must enter the proper information into an XML formatted control file that is used to exchange data between Rational Asset Analyzer and Rational Asset Manager. This information should appear at the top of each control file, described in later sections. The required information includes the following:

- The root URL to the XML version of the Rational Asset Analyzer REST instance
- A valid Rational Asset Analyzer username and password
- The Rational Asset Manager **Web Services Path** URL, found in the Rational Asset Manager configuration page (This is not the Rational Asset Manager home page)
- A Rational Asset Manager username and password with administrative privileges

Each of the sample control files shipped with the class=Spelle>RaaRamIntegration package in <RaaRamIntegrationInstall>\ControlFiles contains an example of the required format:

```
<?xml version="1.0"?>
<RamIntegration>

  <Raa href="Root URL To REST XML"
    user="RAA Username" password="RAA Password" />

  <Ram href="RAM Web Services Path URL"
    user="RAM Username" password="RAM Password" />

  <LOG>true</LOG> <!-- toggle logging messages -->

  <!-- USER DEFINED INTEGRATION PARAMETERS -->

</RamIntegration>
```

Note:

For the remainder of this document, we refer only to the commented section in the above code: <!-- USER DEFINED INTEGRATION PARAMETERS -->. All user-defined parameters must be nested within the initialization code, as shown in the example above.

Uploading all artifacts of a given type (without relationships)

The following XML shows how to upload all assets of the **Application** artifact type from Rational Asset Analyzer without uploading any related assets:

```
<SubmitToRam>
  <AssetTypes>
    <Application/>
  </AssetTypes>
</SubmitToRam>
```

Uploading a specific artifact by attribute (without relationships)

The following example shows how to upload a chosen artifact from the Rational Asset Analyzer REST Interface, based on one of its attributes.

```
<SubmitToRam>
<AssetTypes>
<Program>

  <conditions>
    <attributes>
      <Name>QAD</Name> <!-- The regular expression to match -->
    </attributes>
  </conditions>

</Program>
</AssetTypes>
</SubmitToRam>
```

This XML code is defining the integration to submit to Rational Asset Manager, all artifacts from Rational Asset Manager that meet the criteria that they are of the **Program** asset type that meets the **condition** that their **Name attribute** contains the string **QAD**.

For a list of valid artifact types, navigate to the home page of the Rational Asset Analyzer REST Interface being utilized for the integration. All supported artifact types are listed under **types**.

Note: <Name> takes a regular expression as input, therefore entering QAD uploads all program artifacts that contain QAD in the Name attribute. To match a string exactly, you must enclose the regular expression within ^\$. For example, ^QAD\$ uploads only attributes that match **QAD** exactly.

Adding selected relationships to artifacts

The following example shows how to add selected relationships into Rational Asset Manager as they are uploaded. The following control file sample uploads all program artifacts that contain **QAD** in the **Name** attribute, and uploads them along with all related artifacts of the **FilesNeededToBuild** and **DataElements** type.

```
<SubmitToRam>
<AssetTypes>
<Program>

  <conditions>
    <attributes>
      <Name>QAD</Name> <!-- The regular expression to match -->
    </attributes>
  </conditions>

  <asset-relationships>
    <FilesNeededToBuild/>
```

```

    <DataElements/>
  </asset-relationships>

</Program>
</AssetTypes>
</SubmitToRam>

```

Note: You can also define a custom inverse relationship type for the relationships by adding the relationship types. For example:

```

<asset-relationships>
  <FilesNeededToBuild reverseName="Custom Inverse Relationship Name"/>
</asset-relationships>

```

Adding container artifacts

The following example shows how to add container artifacts to Rational Asset Analyzer artifacts that are uploaded to Rational Asset Manager as assets. The following XML description uploads all applications with **RAA** in the **Name** attribute, and adds the **File** and **Db2Table** relationships as container artifacts for the asset. Container relationships are different from asset relationships in that container relationships only contain a link to the artifact in question in Rational Asset Analyzer, and these artifacts are not submitted to Rational Asset Manager as assets.

```

<SubmitToRam>
<AssetTypes>
<Application>

  <conditions>
    <attributes>
      <Name>RAA</Name>
    </attributes>
  </conditions>

  <containment-relationships>
    <File/>
    <Db2Table/>
  </containment-relationships>

</Application>
</AssetTypes>
</SubmitToRam>

```

Uploading specific artifact attributes

By default, every available attribute for a Rational Asset Analyzer artifact is included as data in the **RAA Attributes** field on the artifact's corresponding Rational Asset Manager general details page. To upload only a specified set of attributes to the **RAA Attributes** field, you must define the desired attributes in the control file. The following example shows how the XML control file must be modified to upload only the **ContainerName** and **FileName** attributes to the Rational Asset Manager general details page for an artifact of all program types uploaded as assets to Rational Asset Manager.

Note: You can also define the Rational Asset Manager display name for the artifacts using the XML tag `ramName`.

```

<RamPropagationConfig>
  <attributes-propagate-as-ram-asset-attributes>
    <Program>
      <ContainerName ramName="RAA File Location" />

```

```

        <FileName      ramName="RAA File Name"      />
    </Program>
</attributes-propagate-as-ram-asset-attributes>
</RamPropagationConfig>

```

Comprehensive example

The following example provides a more complex structure to help give you an understanding of how you can construct your own control file to obtain a Rational Asset Analyzer environment within Rational Asset Manager that best fits your requirements. Explanations can be found within the comments of the example code.

```

<RamPropagationConfig>
<!-- upload only the ContainerName attribute and display it in RAM as "RAA File Location" -->
  <attributes-propagate-as-ram-asset-attributes>
    <Program>
      <ContainerName ramName="RAA File Location" />
    </Program>
  </attributes-propagate-as-ram-asset-attributes>
</RamPropagationConfig>

<SubmitToRam>
  <AssetTypes>

  <!-- Upload SAMPLE_APPLICATION to RAM -->
  <Application>
    <conditions>
      <attributes>
        <Name>SAMPLE_APPLICATION</Name>
      </attributes>
    </conditions>

    <!-- Upload Program relationships for the SAMPLE_APPLICATION -->
    <asset-relationships>
      <Program/>
    </asset-relationships>

    <!-- Upload File and Db2Table relationships as containment artifacts -->
    <containment-relationships>
      <File/>
      <Db2Table/>
    </containment-relationships>
  </Application>

  <!-- Upload Programs with QAD in the Name attribute to RAM -->
  <Program>
    <conditions>
      <attributes>
        <Name>QAD</Name>
      </attributes>
    </conditions>

    <!-- Upload FilesNeededToBuild relationships for QAD Programs -->
    <asset-relationships>
      <FilesNeededToBuild/>
    </asset-relationships>
  </Program>

</AssetTypes>
</SubmitToRam>

```

After you have completed defining a custom XML control file, save it and note the location of the file for further use.

Synchronizing with deleted Rational Asset Analyzer artifacts

When an artifact is deleted from the Rational Asset Analyzer, you can use the XML control file to synchronize Rational Asset Manager to reflect these changes. The following example shows an XML control file configuration that can be used to synchronize the current Rational Asset Manager environment with Rational Asset Analyzer.

```
<!-- Synchronize with the current RAA instance.
      Setting delete="true" will delete any RAA assets from RAM
      that are no longer represented in RAA. Setting this tag to
      "false" will not delete the asset, but rather set its state to "Retired."-->
<Synchronize delete="true"/>
```

Deleting all Rational Asset Analyzer artifacts from Rational Asset Manager

To delete all Rational Asset Analyzer artifacts from Rational Asset Manager, you must configure the control file. The following example shows an XML control file that must be used to delete all assets related to Rational Asset Manager to reverse the changes made to the Rational Asset Manager environment, such as deleting the Rational Asset Manager community, relationship types, and so on.

```
<!-- Remove all RAA-related assets from RAM -->
<DeleteEnvironment/>
```

Chapter 4. Deploying and maintaining the integration

After you have defined an XML control file, the integration of Rational Asset Analyzer into Rational Asset Manager can be deployed. Rational Asset Analyzer provides a command line interface that allows you to submit artifacts to Rational Asset Manager based on the XML control file you defined.

You can find several sample XML control files in the **RaaRamIntegration** directory in the **ControlFiles** folder. To use these files, modify **href**, **user**, and **password** values for both **Raa** and **Ram** attributes to correspond with the environment in use.

Command Line Interface

Preparing for integration

Before you integrate Rational Asset Analyzer to Rational Asset Manager, you must complete the following:

1. Obtain the **RaaRamIntegration.zip** file and extract the **RaaRamIntegration** folder to any directory.
2. Navigate to **RaaRamIntegration/ControlFiles** for an example control file and modify all logon information marked as **xxxxxxxxxx** in the chosen control file.
 - Raa href, as described in Rational Asset Analyzer REST Interface documentation
 - Raa user
 - Raa password
 - Ram href, as found under the Administration tab of Rational Asset Manager
 - Ram user
 - Ram password
3. If using **integration-example.xml**, modify the **<SubmitToRam>** section of the control file to submit the desired information to Rational Asset Manager (not required if testing connectivity).

Performing the integration

You can perform the integration in two ways.

Method 1 works from the command line:

1. Open a command line prompt and change the directory to the **RaaRamIntegration** folder which contains **RaaToRam.bat**.
2. Enter **RaaToRam {path to a properly defined control file}** and press enter to run the integration.

Method 2 works from a file browser:

1. Open the **RaaRamIntegration** folder in a file browser and run **RaaToRam.bat**.
2. When prompted, enter the location of a defined XML control file, and press enter to run the integration.

Synchronizing the integration

The synchronization process updates Rational Asset Manager assets to correspond with deleted files from Rational Asset Analyzer. If an artifact from Rational Asset Analyzer is detected as deleted, the corresponding Rational Asset Manager asset either has its state set to **Retired** or is deleted from Rational Asset Manager as defined by the `\RaaRamIntegration\ControlFiles\synchronize.xml` XML control file. You can set this by modifying the following line in the control file:

```
<Synchronize delete="false"/> <!-- set the asset's state to "retired" -->
<Synchronize delete="true"/> <! delete the assets -->
```

If you want to update the contents of the Rational Asset Manager assets that were imported as a result of running an integration, you must re-run the integration using the same XML control file that was initially used to populate the assets into Rational Asset Manager.

Removing Rational Asset Analyzer environment from Rational Asset Manager

The `delete- environment.xml` file is available in the `ControlFiles` directory. When this XML file is configured to match the Rational Asset Analyzer and Rational Asset Manager instances, it removes all remnants from the Rational Asset Analyzer to Rational Asset Manager Integration from the existing Rational Asset Manager instance (except for **RAA** user).

Configure `delete- environment.xml` to the desired instances of Rational Asset Manager and Rational Asset Analyzer and run the integration using this control file.



Printed in USA