

IBM WebSphere ReSTFacade for IBM WebSphere Adapters

Document Version 1.0

| | |
|---|-----------|
| NOTICES | 5 |
| PREFACE | 8 |
| TARGET AUDIENCE | 8 |
| CONTENTS OF THIS SUPPORTPAC | 8 |
| FEEDBACK ON THIS SUPPORTPAC..... | 8 |
| INTRODUCTION TO REST AND RESTFUL SERVICES | 9 |
| INTRODUCTION TO IBM WEBSPPHERE RESTFACADE | 9 |
| PREREQUISITES | 10 |
| INSTALLATION | 10 |
| INSTALLING RESTFACADE AND ITS COMPONENTS | 10 |
| OVERVIEW OF RESTFACADE | 11 |
| RESTFACADE DESIGNER..... | 12 |
| RESTFACADE WEB USER INTERFACE | 12 |
| IBM WEBSPPHERE RESTFACADE WEB APPLICATION..... | 13 |
| RESTFACADE ADMINISTRATIVE CONSOLE | 13 |
| GETTING STARTED | 13 |
| REST ENABLING YOUR EIS OBJECT | 13 |
| MIGRATING FROM DEVELOPMENT TO PRODUCTION | 14 |
| RESTFACADE DEPLOYMENT MODELS | 14 |
| REQUEST PROCESSING | 15 |
| MAPPING HTTP METHODS WITH ADAPTER OPERATIONS..... | 16 |
| REALIZING THE BENEFITS OF RESTFACADE | 16 |
| WORKING WITH URI PATTERNS SUPPORTED BY RESTFACADE..... | 18 |
| INVOKING IN NON-MANAGED AND MANAGED MODES | 21 |
| CREATING A NEW WEB APPLICATION WITH RESTFACADE DESIGNER..... | 21 |
| CREATING ACCESS USING RESTFACADE CONTROLLER WAR..... | 22 |
| WORKING WITH RESTFACADE ADMINISTRATIVE CONSOLE | 22 |
| WORKING WITH RESTFACADE DESIGNER..... | 22 |
| WORKING WITH WEBSPPHERE APPLICATION SERVER | 33 |
| SETTING UP LOGGING, TRACING AND EXCEPTION MESSAGES | 34 |
| ABOUT THE APPENDIX SECTION | 35 |
| APPENDIX A | 36 |
| DEFINING URL PATTERNS FOR SAP DATA SOURCE MANIPULATIONS | 36 |
| TESTING A RESTFUL COMPONENT WITH JCO AND DLL FILES..... | 38 |
| APPENDIX B | 39 |
| USING RESTFACADE IN THE DEVELOPMENT/TESTING MODE..... | 39 |
| APPENDIX C | 45 |
| EXPORTING CONFIGURATIONS FROM DEVELOPMENT/TESTING MODE | 45 |
| APPENDIX D | 47 |
| IMPORTING THE DEVELOPMENT MODE CONFIGURATIONS TO THE PRODUCTION MODE..... | 47 |
| APPENDIX E | 51 |

| | |
|--|-----------|
| WORKING AROUND CURRENTLY IDENTIFIED ReSTFACADE LIMITATIONS | 51 |
| PRODUCT FEEDBACK AND SUGGESTIONS | 53 |

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication.

IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites.

The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation Department

2Z4A/SOM1 294 Route 100

Somers, NY 10589-0100 U.S.A.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtains the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning:

Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US 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 complete and current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Screens are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.*

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

This product includes software developed by the Eclipse Project (<http://www.eclipse.org>).

Preface

Target Audience

This SupportPac is designed for users who:

- Implement and design Web or mash-up applications and want to expose Enterprise Information System content as a ReSTful resource through IBM WebSphere Adapters.
- Work with an HTTP client to expose adapters during both discovery and runtime in a ReSTful manner.
- Deploy existing adapter binaries, to invoke operations in a ReSTful manner.

To make optimum use of this SupportPac, users should have a general awareness of WebSphere Application Server CE and Rational Application Developer.

Contents of this SupportPac

This SupportPac documentation includes:

1. Overview of IBM WebSphere ReSTFacade and its components.
2. Installation and configuration details of ReSTFacade components.
3. Navigation through ReSTFacade.
4. Appendix section on ReST enablement samples, and current ReSTFacade limitations.
5. Product Feedback and Suggestions sheet.

Feedback on this SupportPac

We welcome your constructive feedback on this SupportPac document. Please refer the [Product Feedback and Suggestions](#) section in this document, to fill your comments and suggestions.

You can direct any comments of this nature to raj कुमार@in.ibm.com.

Introduction to ReST and ReSTful services

Representational State Transfer (ReST) is a software architecture style for web based services. Web-based services that follow ReST principles are addressed as ReSTful services. The ReST architecture consists of clients and servers. The client sends requests to the server and responses are returned which involve transfer of “representations” of ‘resources’.

Resources are the objects residing in a server you want to identify, and which is uniquely identifiable. When a client makes a request for a resource, it is not the resource which itself is returned. Rather, a representation of the resource is what is returned. For example a representation can be in a format like HTML or XML or JSON. Resources can be manipulated through these representations.

ReST can be described in the context of HTTP. Clients and servers exchange representations of resources by sending HTTP requests and resources. Each resource is uniquely identified by a HTTP URI. Resources can be created, updated, deleted and retrieved using these HTTP methods:

- POST
- PUT
- DELETE
- GET

Introduction to IBM WebSphere ReSTFacade

ReSTFacade provides a ReSTful access, using WebSphere Adapters for establishing connectivity to an Enterprise Information System.

The objects in an Enterprise Information System like BAPIs in SAP or tables in JDBC are classic examples of resources. Browsing and discovery options of these resources within an Enterprise Information System, through WebSphere Adapters are enabled ReSTfully. These resources can also be manipulated in a ReSTful manner.

The following Adapter operations are used to manipulate the resources in an Enterprise Information System:

- CREATE
- UPDATE
- DELETE
- RETRIEVE

These operations or methods as they are called are mapped to HTTP methods of POST, PUT, DELETE and GET.

Thus ReSTFacade supports both ReSTful browsing, discovery and invocation of resources in an Enterprise Information System.

Prerequisites

- 1) Rational Application Developer v 7.5.5.1
- 2) WebSphere Application Server v 7.0.0.9
- 3) WebSphere Application Server Community Edition v 2.1.1.3 or later
(Link - <http://www-01.ibm.com/software/webservers/appserv/community>)
- 4) WASCE 2.1 WTP Server Adapter Eclipse plug-in
(Link - <http://public.dhe.ibm.com/software/websphere/wasce/updates>)

Note: You need WAS CE and the corresponding Server Adapter plug-in to run ReSTFacade in development mode. The server adapter Eclipse plug-in should have installed on top of Rational Application Developer.

Installation

Installing ReSTFacade and its components

Before you install ReSTFacade and its components, make sure that you have access to the components/software mentioned in the ‘Prerequisites’ section.

You should also have access to the required RAR, JCO and DLL files if your adapter specifications mandate the requirement of these files. For more information on working with third party dependency files, refer the WebSphere Adapter documentation, specific to that adapter.

To install the SupportPac:

1. Download the binaries (xa0a.zip) from the SupportPac web site and unzip it into a temporary directory.
2. Uncompress using the InfoZip Unzip tool. This will create two directories (“development” and “production”) under the temporary directory.
4. Copy the entire content inside 'development' directory (com.ibm.rwad_1.0.0 folder) and paste it inside the plug-ins folder of your Rational Application Developer installation directory. (Example C:\IBM\SDP\plugins). This completes your development environment set up.
5. For the production environment, the directory “production” has the required WAR file which needs to be deployed to your WebSphere Application Server connection where the adapter is deployed at node level.
6. You can now delete xa0a.zip.

Note: Refer the Eclipse plug-in installation instructions to how to install an Eclipse plug-in. Refer the WebSphere Application Server documentation, to how to install a WAR file in WebSphere Application Server.

After you have successfully installed the ReSTFacade plug-in, you should be able to view this as an Eclipse Run menu option.

Overview of ReSTFacade

IBM ReSTFacade enables you to quickly ReST enable your Enterprise Information System with the help of WebSphere Adapters, and make these ReST enabled resources available for other web or mashup applications. For example, as a WebSphere Adapter for SAP Software, you can access and execute a BAPI transaction over HTTP in a ReSTful manner.

You can also consume the URIs exposed by the ReSTFacade component, in another mashup or web application.

You can access ReSTFacade either through the custom built IBM WebSphere ReSTFacade Web User Interface, or through any other custom HTTP client tool, such as Mozilla Firefox Add-On: Poster, if you are an expert user.

IBM WebSphere ReSTFacade for WebSphere Adapters

IBM WebSphere ReSTFacade enables you to quickly ReST enable your Enterprise Information Systems with the help of WebSphere Adapters, and make them available for other web or mashup applications.

For example, a WebSphere Adapters user may find a need to access and execute the SAP systems' BAPIs over HTTP in a ReSTful manner. User may also want to consume the URIs exposed by the 'ReSTFacade for WebSphere Adapters' component in another mashup or web application.

ReSTFacade can be accessed either via custom built UR IBM WebSphere ReSTFacade Web User Interface or via any custom HTTP client tools (example FireFox AddOn - POSTER) if you are an expert user.

1. **IBM WebSphere ReSTFacade Web User Interface**
An intuitive web application that guides the first time users thru several stages of ReST enablement of EIS resources. It supports creation of datasource alias, browsing EIS objects (ex: BAPI in case of SAP) and finally generation of ReSTful HTTP URIs for those selected objects.
2. **IBM WebSphere ReSTFacade Web Application**
Core web application which helps users ReST enable their EIS resources. Since this being complete ReSTful application you need custom HTTP tools/clients to access this. As an alternative approach you can use the custom built UI IBM WebSphere ReSTFacade Web User Interface from here

IBM WebSphere ReSTFacade Administrative Console

A web application provides features like export and import of configuration. In real life scenarios it is very critical when an application needs to be moved from test to production. ReSTFacade Administrative Console helps an user to export all the configurations he/she has created in test environment to production environment there by saving time for the user.

Getting started

Three steps to ReSTenable EIS objects

Below are the typical steps to ReSTenable your EIS objects (say BAPI in case of SAP). Note that these can be done using IBM WebSphere ReSTFacade Web User Interface or by using custom HTTP client tool pointing to the right URL of core ReSTFacade application

- Create datasource instance(s) by supplying valid connection properties to the target EIS system
- Browse the interested objects. You can also enter filter patterns to optimize your search while browsing the objects
- Select the object to ReSTenable and invoke using the URL

Three steps to migrate from 'Development' to 'Production' environment

- Follow above section 'Three steps to ReSTenable EIS objects' to expose your EIS objects as ReSTful resources.
- Create a datasource instance in your production environment with the same name as that of your development environment
- Use the IBM WebSphere ReSTFacade Administrative Console to export the generated artifacts and import them into your production

You can use the Rational Application Developer environment for development or testing purposes with both the discovery and runtime operating in non-managed mode. You can also deploy the ReSTFacade WAR file (downloaded from the SupportPac site) in WebSphere Application Server, for the production environment, with the adapter running

in the node level (managed mode). For more information on adapter node level deployment, refer the IBM WebSphere Adapter documentation/user-guides.

***Note:** The current release plan does not propose to ship Adapter binaries, to not contravene existing licensing terms. The initial mode of delivery will be through a SupportPac on Rational Application Developer.*

The ReSTFacade architecture comprises the following components:

1. IBM WebSphere ReSTFacade Designer
2. IBM WebSphere ReSTFacade Web User Interface
3. IBM WebSphere ReSTFacade Web Application
4. IBM WebSphere ReSTFacade Administrative Console

ReSTFacade Designer

ReSTFacade Designer component, an Eclipse plug-in has to be installed on top of Rational Application Developer. This serves as a user entry point for interacting with ReSTFacade during development time. The wizard based designer component captures adapter binary and any third party dependency files, if required. Additionally, the tool helps you to create a deployable application on the target server (WebSphere Application Server Community Edition).

ReSTFacade Web User Interface

The intuitive Web interface guides the first time user through several stages of ReST enablement of EIS resources. Using ReSTFacade Web User Interface you can create a data source alias, browse EIS objects (for example, a BAPI transaction), and generate ReSTful HTTP URIs for the objects you have selected.

For a sample on how to use ReSTFacade Web User Interface, refer Appendix B of this document.

Note: The ReSTFacade User Interface component uses the core ReSTFacade component under the cover to interact with an EIS. All the calls from ReSTFacade are routed through the core ReSTFacade component.



IBM WebSphere ReSTFacade Web Application

This core web application helps you to ReST enable your EIS resources, using the custom built IBM WebSphere ReSTFacade Web User Interface. As this is a full fledged ReSTful application, you can also deploy other custom HTTP tools/clients to access services.

ReSTFacade Administrative Console

In real life scenarios, it is very critical when you want to develop and test an application in a test environment, and export the test application to a production environment.

ReSTFacade Administrative Console provides features like export and import of configuration. You can rapidly export all the configurations you have created in the test/development environment, to the production environment.

You can also use ReSTFacade Administrative Console to configure properties needed for ReSTFacade to communicate with the server where adapter is deployed at node level. These are mandatory properties for the ReSTFacade to work in production which is managed mode where adapter is deployed at node level on WebSphere Application Server.

Getting Started

This section describes how to rest enable objects and migrate from a development environment to the production stage.

ReST enabling your EIS object

The following 3 step procedure allows you to ReST enable your EIS object, such as a BAPI transaction in SAP Software.

Note: You can use IBM WebSphere ReSTFacade Web User Interface, or deploy a custom HTTP client tool, to direct you to the correct URL of a core ReSTFacade application.

1. Create a data source instance, after you supply valid connection properties to your target EIS system.
2. Browse the objects you want to discover.
You can also enter filter patterns to optimize your search while you browse your objects.
3. Select the object you want to ReST enable, and invoke your request with the URL.

Migrating from development to production

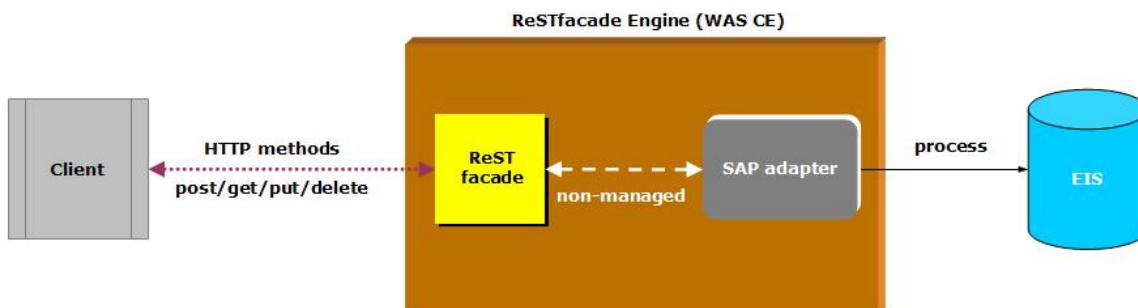
The following 3 step procedure allows you to migrate from your development environment to the production stage.

1. Follow the steps described in the ReST enabling your EIS object section to expose your EIS objects as ReSTful resources.
2. Create a data source instance in your production environment with the same name as that of your development environment.
3. Deploy IBM ReSTFacade Administrative Console to export the generated artifacts, and import them into your production environment.

ReSTFacade deployment models

You can deploy ReSTFacade in two modes. When you run the ReSTFacade Designer plug-in, the designer internally deploys ReSTFacade and the adapter binaries, selected in the non-managed mode for better performance.

The following diagram explains the deployment process.



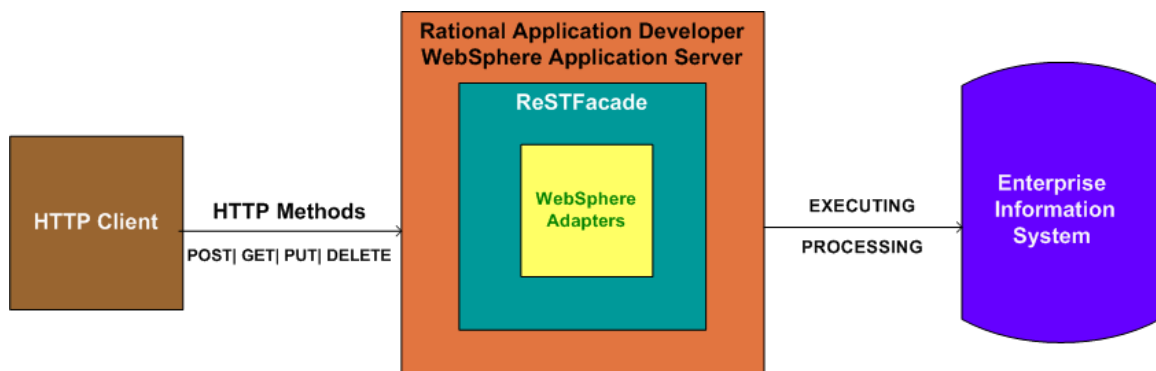
ReSTFacade also supports the managed mode, where an adapter is already deployed at the node level in WebSphere Application Server. However, to deploy ReSTFacade in the managed mode, you need to use the WAR (Web Application Archive) file downloaded from the SupportPac site, as explained in the Installation section of this document.

Request processing

The following steps are notable in processing a request, using WebSphere Adapters in a ReSTful manner:

1. Employ ReSTFacade Designer to extract adapter binary and other third party dependencies.
2. Create a new ReSTFacade application using the wizard, to act as the communication end point.
3. Initiate a new server instance in managed or non managed modes.
4. Route the incoming HTTP request through a server to an HTTP method on ReSTFacade. Ensure that all arguments based on the query string values (URI path segments) are cleared for the request.
5. Deploy the Web application for discovery and runtime invocations.

The following diagram shows how an HTTP client communicates with an Enterprise Information System, using ReSTFacade.



Mapping HTTP methods with Adapter operations

| HTTP Method | CRUD Operation | Action |
|--------------------|-----------------------|---|
| POST | CREATE | Create a new entry, such as a new employee record |
| GET | RETRIEVE | Fetch an entry, such as an employee record. |
| PUT | UPDATE | Update an existing entry, with new data, such as updating an employee record. |
| DELETE | DELETE | Delete an existing entry, such as an employee record. |

Realizing the benefits of ReSTFacade

ReSTFacade for WebSphere Adapters is more useful when you want to quickly build a Web or mash-up application, and want to access the EIS resources in your application in a ReSTful manner. It greatly increases your ROI (return on investment). You have to otherwise implement a custom Web application, around an existing connectivity solution, to get data from EIS.

Consider the current environment.

WebSphere Adapters deployed on WebSphere Process Server or WebSphere Application Server, currently support different communication data formats, through a Service Data Object (SDO) or a JavaBean. Doubtless, these technologies are efficient. However, there is further scope to improve the data retrieval and syndication model approaches.

Though these current technologies offer more flexibility and reusability, they are more connectivity-driven than data-driven. The adapters deployed in this model can be invoked only through Service Component Architecture (SCA) and Common Client Interface (CCI).

ReSTFacade brings in another usability model for WebSphere Adapters. It provides a uniform interface that separates the HTTP client and the access to different EIS systems.

An HTTP client can interact with any number of servers along the way, and not necessarily with only one primary end point. The client remains unaware of the server path, and concentrates only on sending requests. Therefore, more complex situations can enlist the services of intermediary servers to enforce load balancing, security, and improved cache operations.

HTTP clients can cache a response, which means that information will be up-to-date, and current. Old data will not be represented, even for repeated requests. The “clean” interaction between the HTTP client and a server makes it a simpler, where all “parts” of a machinery work efficiently and independently.

There is no metadata stored during the communication between an HTTP client and a server, except for the core request details in any particular state. Servers are more easy to manage and reliable in most network situations.

The following features distinguish ReSTFacade for integration with WebSphere Adapters during discovery and run time, from similar architecture like SOAP and XML-RPC.

- Implemented over the HTTP protocol which is universally available, easy to handle, easy to code and is transparent.
- Identifies EIS data as a ReST "resource" and make the data easily consumable for mash-ups and syndication.
- Fits with the Web 2.0 model, to bring users to a common meeting point, to create, access, and combine all types of information in more interesting and exciting ways.
- Integrates easily with diverse technologies, without the need for a separate client program.
- Requires minimum XML support, to generate human readable results.
- Needs no developer tool kits, as there is nothing to build or customize, and the whole concept works around resources and representations.
- Represents each business object by a unique URI and users can narrow down their search operations, instead of having to view and search a whole Web location.
- Supports the "? define" command at every URI level, to increase the purpose of each URI action.
- A single ReSTFacade instance can synchronously cater to multiple adapter instances.

Working with URI Patterns supported by ReSTFacade

Here are a few examples of URI patterns that are supported.

Note: For the non-managed mode on WebSphere Application Server CE, use the following URIs directly. For the managed mode on WebSphere Application Server, replace “rwa” in the URI with the web application you named, while deploying the WAR.

GET

<http://HOSTNAME:PORT/rwa/rwaservlet>

```
<?xml version="1.0" encoding="UTF-8" ?>
<rwa:collection xmlns:rwa="http://www.ibm.com/xmlns/prod/rwa">
<rwa:resource id="SAP" isCollection="true" ref="http://HOSTNAME:PORT/rwa/rwaservlet/SAP"/>
</rwa:collection>
```

GET

<http://HOSTNAME:PORT/rwa/rwaservlet/SAP?define>

Returns a XSD format. This XSD document represents all the connection properties necessary to create a connection to an SAP system.

POST

<http://HOSTNAME:PORT/rwa/rwaservlet/SAP>

The payload of the HTTP request is a XML document which must be validated with the XSD format, returned by using the SAP ?define request.

The response will return the name of the newly created alias.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rwa:response xmlns:rwa="http://www.ibm.com/xmlns/prod/rwa">
<rwa:resource id="SAP1" ref="http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1"/>
</rwa:response>
```

POST

<http://HOSTNAME:PORT/rwa/rwaservlet/SAP/customName>

The payload of the HTTP request is an XML document which must be validated with the XSD format, returned by using the SAP ?define request.

You can specify your own name for the alias (customName).

The response will return the name of the alias of the specified user.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rwa:response xmlns:rwa="http://www.ibm.com/xmlns/prod/rwa">
<rwa:resource
ref="http://HOSTNAME:PORT/rwa/rwaservlet/SAP/customName"/>
id="customName"
</rwa:response>
```

GET

<http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1?define>

Returns all the connection properties representing this alias.

URI Patterns for EIS Browsing and resource discovery

GET

<http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1>

Returns list of resources at the next level in the EIS

```
<?xml version="1.0" encoding="UTF-8" ?>
<rwa:collection xmlns:rwa="http://www.ibm.com/xmlns/prod/rwa">
<rwa:resource id="RFC" isCollection="true" ref="http
http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1/BAPI"/>
</rwa:collection>
```

GET

<http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1/BAPI>

Returns list of resources at the next level in the EIS

```
<?xml version="1.0" encoding="UTF-8" ?>
<rwa:collection xmlns:rwa="http://www.ibm.com/xmlns/prod/rwa">
<rwa:resource id="RFC" isCollection="true" ref="http
http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1/BAPI/RFC"/>
</rwa:collection>
```

GET

http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1/BAPI/RFC?id=BAPI_CUSTOMER_GETDETAIL
*
-

```
<?xml version="1.0" encoding="UTF-8" ?>
<rwa:collection xmlns:rwa="http://www.ibm.com/xmlns/prod/rwa">
<rwa:resource id="BAPI_CUSTOMER_GETDETAIL" isCollection="false"
ref="http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1/BAPI/RFC/BAPI_CUSTOMER_GETDE
TAIL"/>
<rwa:resource id="BAPI_CUSTOMER_GETDETAIL1" isCollection="false"
ref="http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1/BAPI/RFC/BAPI_CUSTOMER_GETDE
TAIL1"/>
<rwa:resource id="BAPI_CUSTOMER_GETDETAIL2" isCollection="false"
ref="http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1/BAPI/RFC/BAPI_CUSTOMER_GETDE
TAIL2"/>
</rwa:collection>
```

GET

http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1/BAPI/RFC/BAPI_CUSTOMER_GETDETAIL?define

Returns the XSD document representing this resource (BAPI_CUSTOMER_GETDETAIL) in this Enterprise Information System. You can discover this resource in this Enterprise Information System.

GET

<http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1/BAPI/RFC/SapPeAddress.xsd>

Returns the corresponding XSD format. This can be used to see the child objects of the top level object. For example BAPI_CUSTOMER_GETDETAIL would return SapBapiCustomerGetDetail.xsd. This references SapPeAddress.xsd which can be obtained using this call.

The pre-requisite is that the resource (BAPI_CUSTOMER_GETDETAIL) should have been discovered previously using the ?define request

URI Patterns for Invocation

POST

http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1/BAPI/RFC/BAPI_CUSTOMER_GETDETAIL

The payload will be a XML representing a representation of the resource you are manipulating. This should be a valid XML format, conforming to the XSD of the resource obtained during discovery using the ?define request.

GET

http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1/BAPI/RFC/BAPI_CUSTOMER_GETDETAIL?CustomerToBeRequired=000000001&SalesOrganization=0001

The response retrieves the specified record.

Query parameters are the name value pairs wherein the name represents the name of an attribute of a top level object and the value the value you want to set to it for requesting the object.

Note: Currently the GET action is supported for simple attributes of a top level object. You cannot perform this action for child objects.

Invoking in non-managed and managed modes

You can perform runtime invocation in either of these modes:

1. Non-Managed mode

This mode of operation is supported when working with ReSTFacade using Rational Application Developer. In this case, the adapter runs in a non-managed mode in the in-built WebSphere Application Server (Community Edition). The adapter works without any intervention from the J2EE application server.

2. Managed mode

In this mode, you must deploy the adapter on WebSphere Application Server at the node level. (Standalone). ReSTFacade will interact with the adapter, while the J2EE application server will provide support for connection management, security, and so on.

When you create a data source, the framework automatically creates and configures a J2C Connection Factory, used to obtain connections to an Enterprise Information System.

Important: You must ensure that you have the SOAP connector running in the background. ReSTFacade uses the Admin Client SOAP connector, to manage administrative functions within the architecture.

Creating a new Web application with ReSTFacade Designer

ReSTFacade Designer wizard runs on Rational Application Developer, as the user entry point, to interact with ReSTFacade. The wizard takes the adapter RAR and the necessary third party dependencies from the configuration, and creates a new ReSTFacade Web application.

ReSTFacade Designer uses an Eclipse plug-in to perform these actions.

When the ReSTFacade Designer wizard completes this activity, a new WebSphere Application Server CE instance, called ReSTFacade engine is created and started. The ReSTFacade Web application will be deployed on it, ready to be used for discovery and invocations.

You can install the ReSTFacade Designer plug-in, by copying the plug-in binaries downloaded from the SupportPac site to the plug-ins folders of Rational Application Developer.

Creating access using ReSTFacade Controller WAR

You can deploy this Web application to provide access to your Enterprise Information System, for browsing, discovery and invocation of resources.

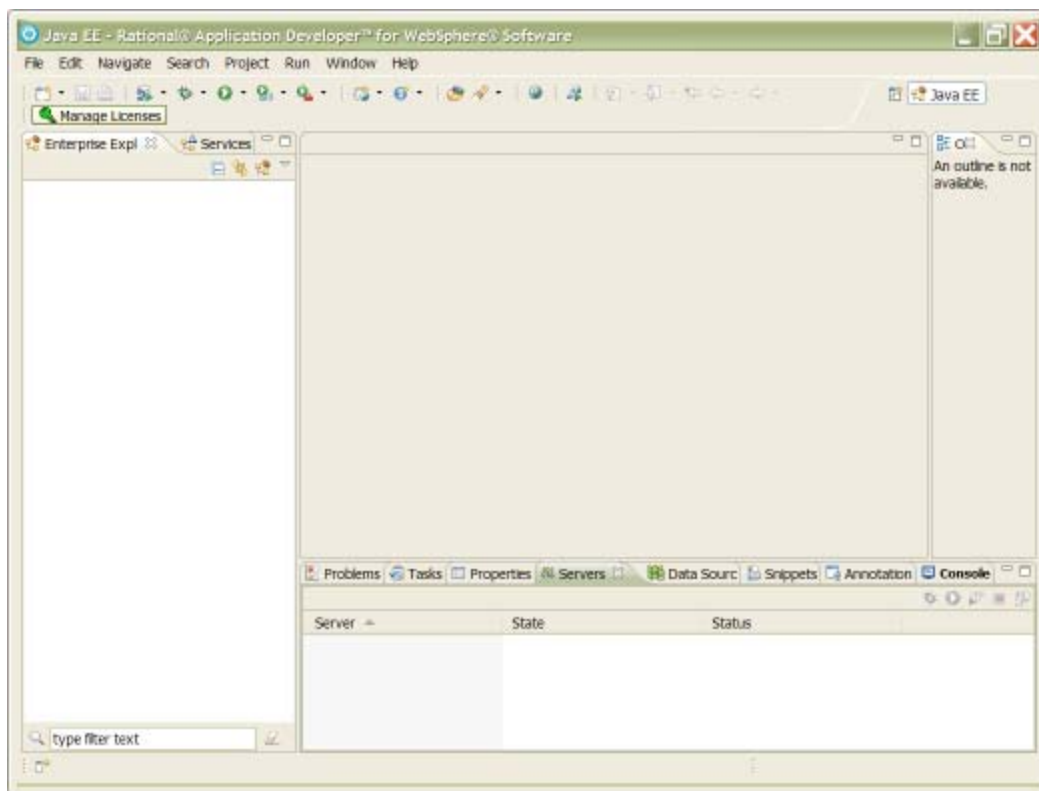
Working with ReSTFacade Administrative Console

The administrative console provides you the option to import or export data source configurations and artifacts. You can then configure the ReSTFacade level properties, from this console.

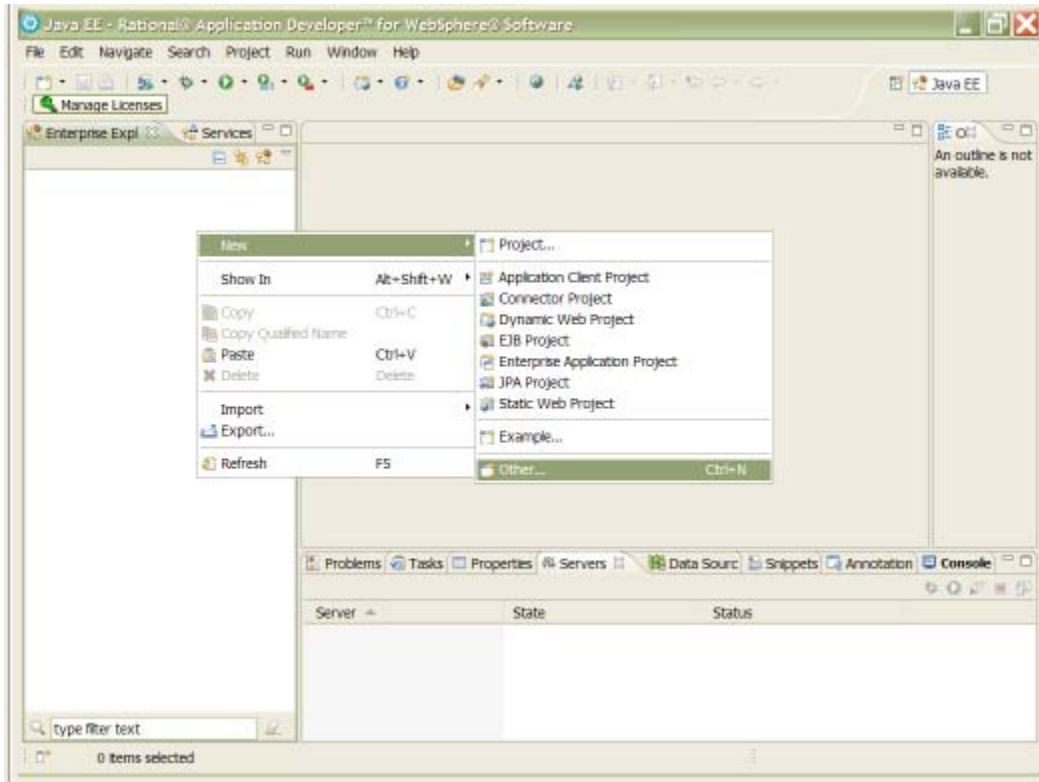
Working with ReSTFacade Designer

The following steps describe how you can import an adapter, and deploy it in WebSphere Application Server.

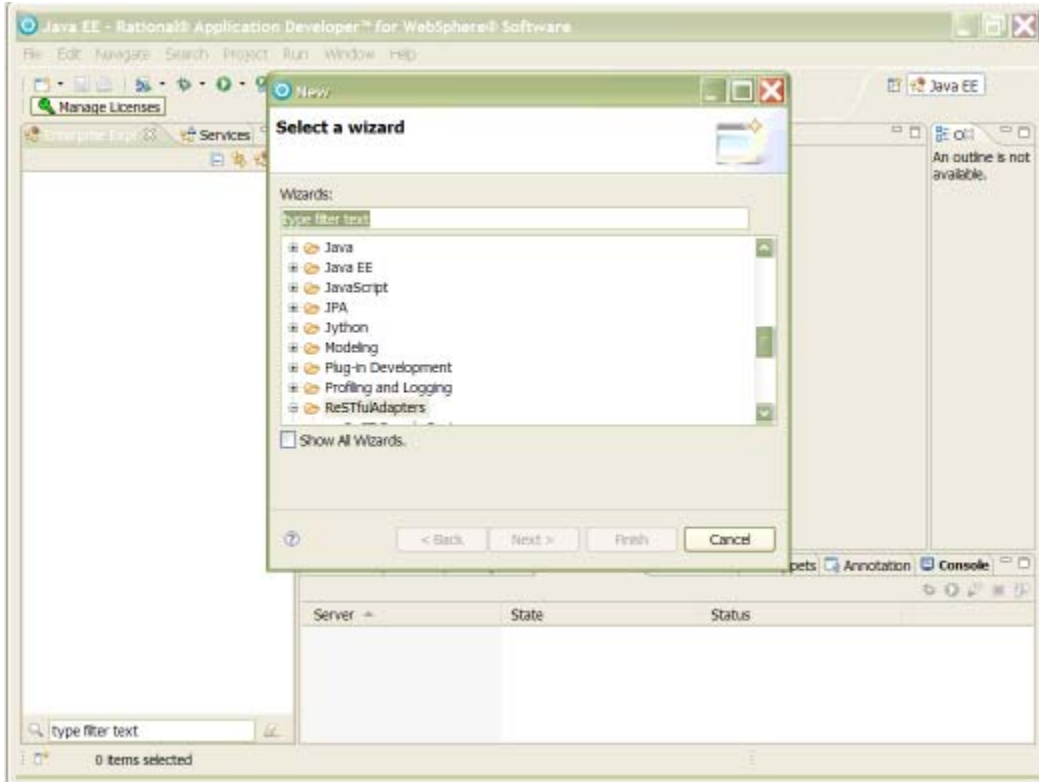
1. Launch the Rational Application Developer workspace.



2. Select File > New > Other.

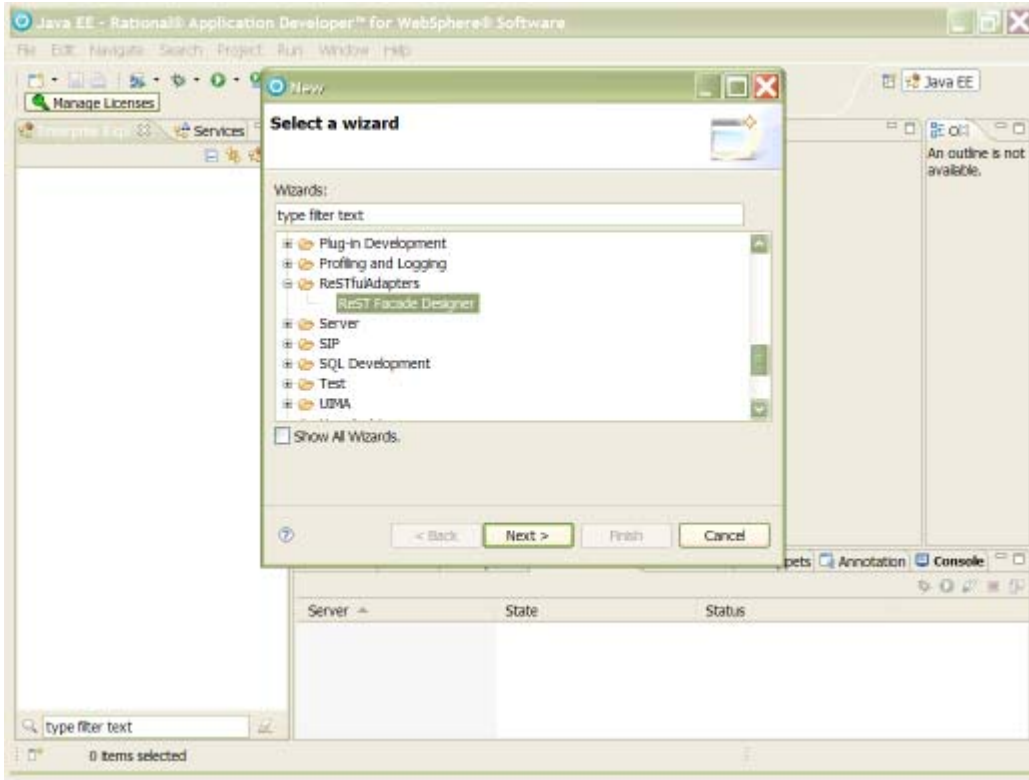


3. Select and expand the ReSTful Adapters node in the Select a wizard screen.



4. Click the ReSTFacade Designer node.
5. Click Next.

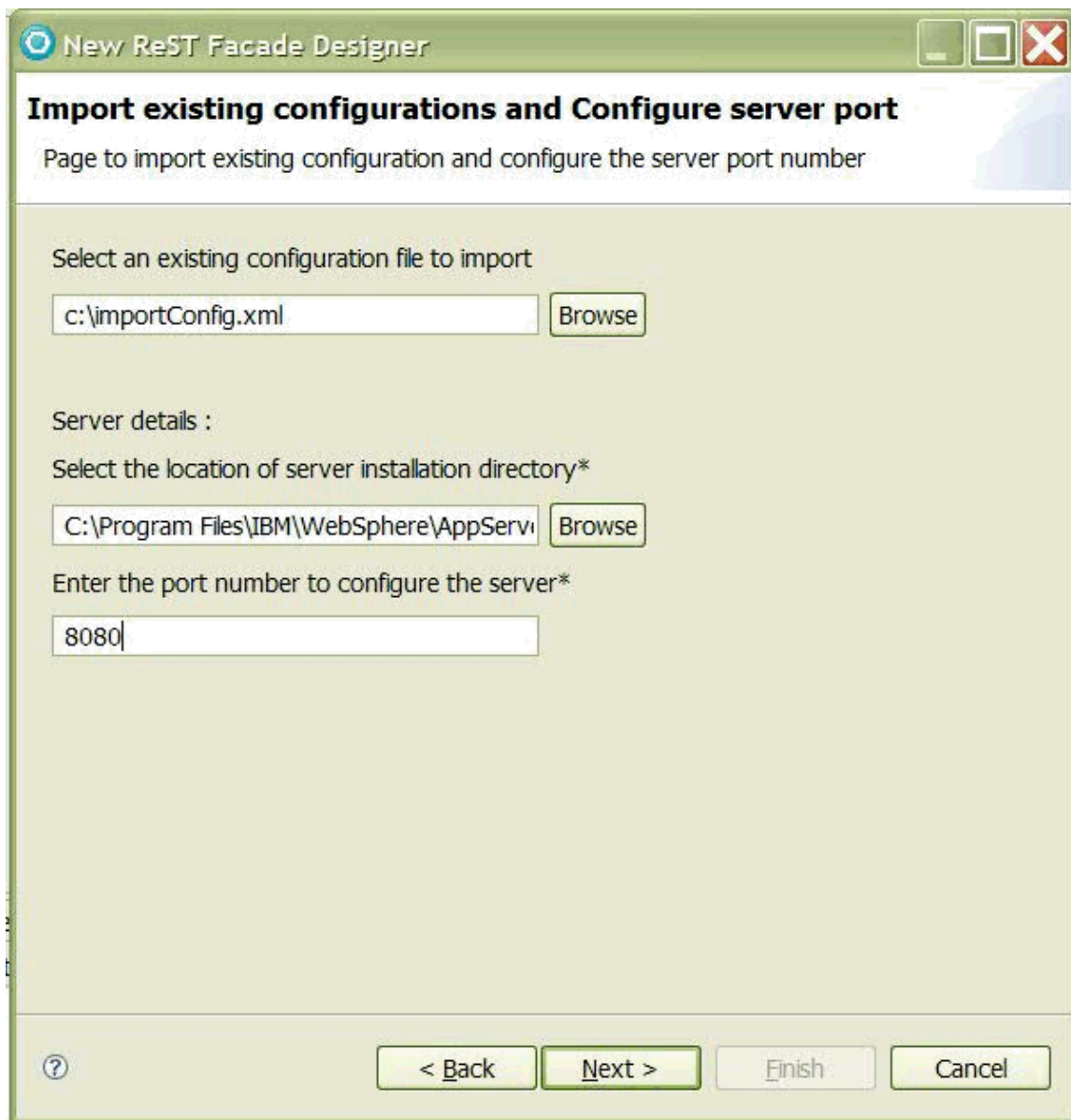
The ReSTFacade Designer wizard opens.



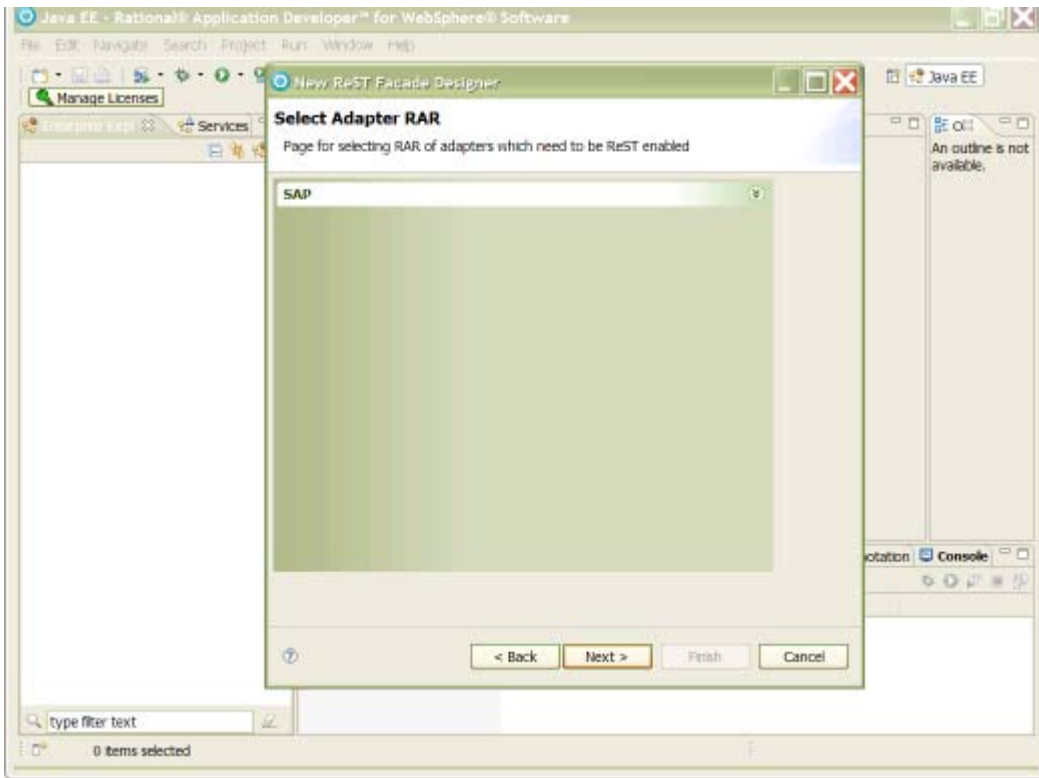
The Import existing configurations and Configure server port screen appears. You can either import an existing configuration file, or configure your server and port number.

If you are running the wizard for the first time, you need to enter the path of WAS CE server installation directory and the port you want to launch the web application.

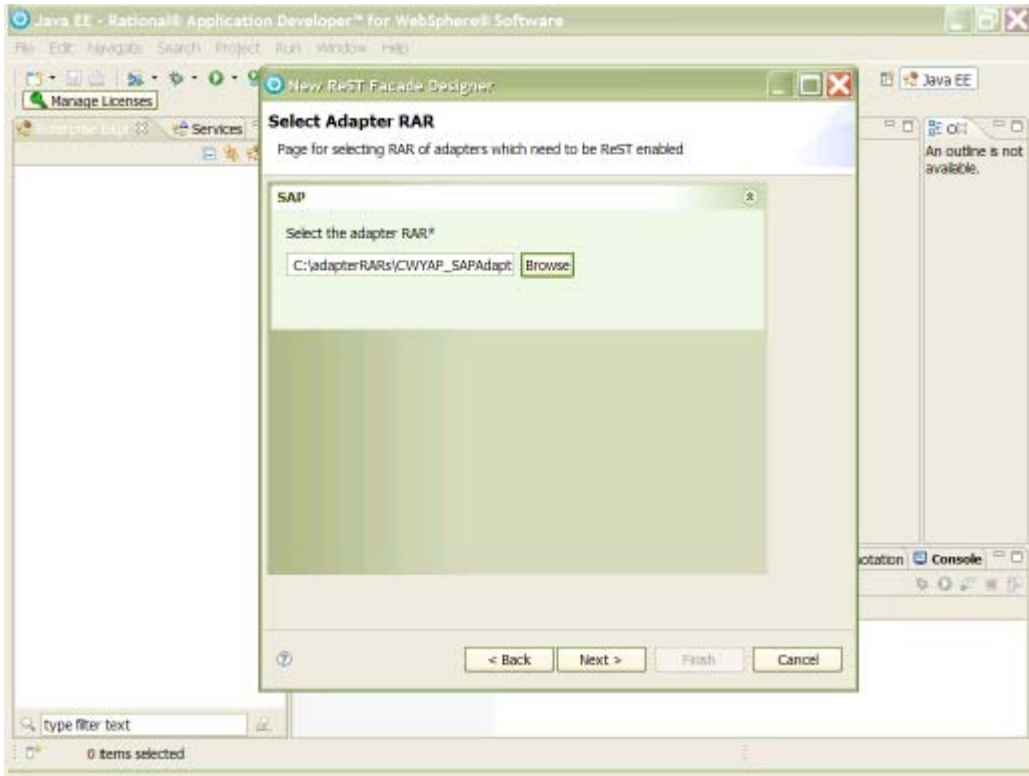
Importing an existing configuration file will allow you to import values from an earlier wizard run.



6. Click **Browse** to select the WebSphere Application Server CE location.
7. Click **Next**.

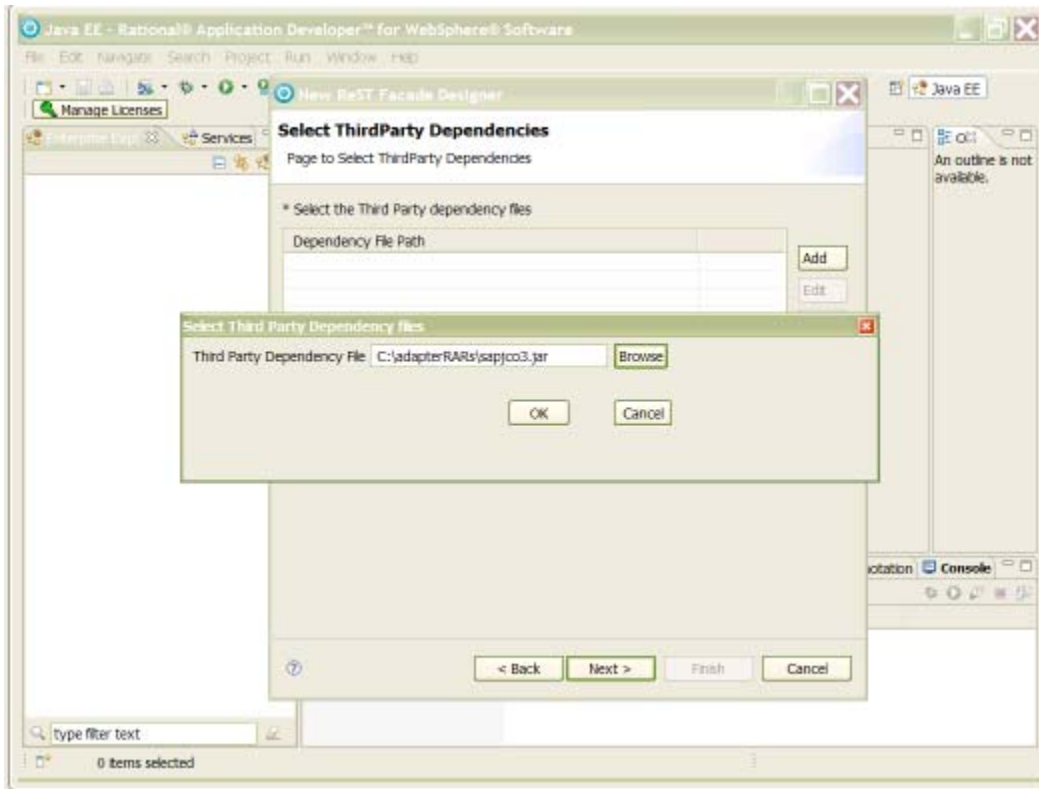


The Select Adapter RAR screen appears.

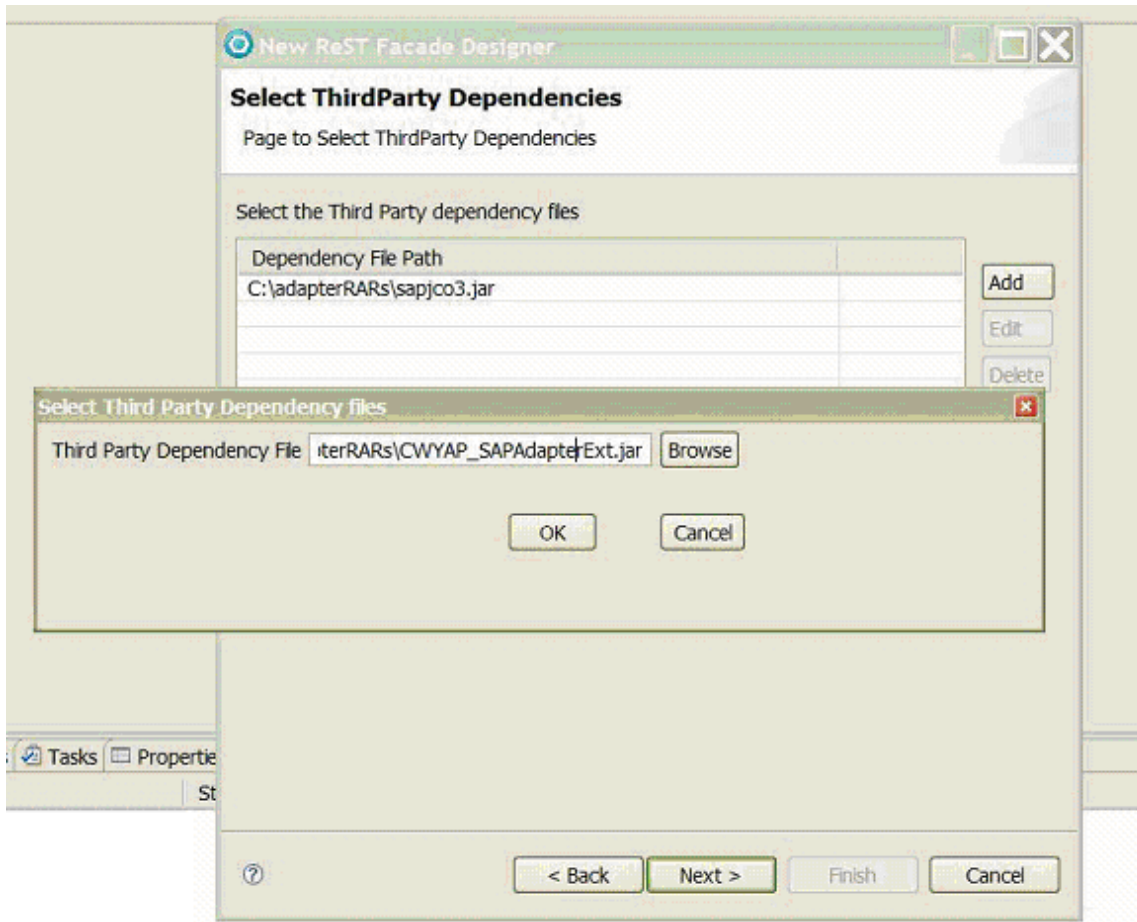


8. Click **Browse** and select the SAP Adapter RAR file.
9. Click **Next**.

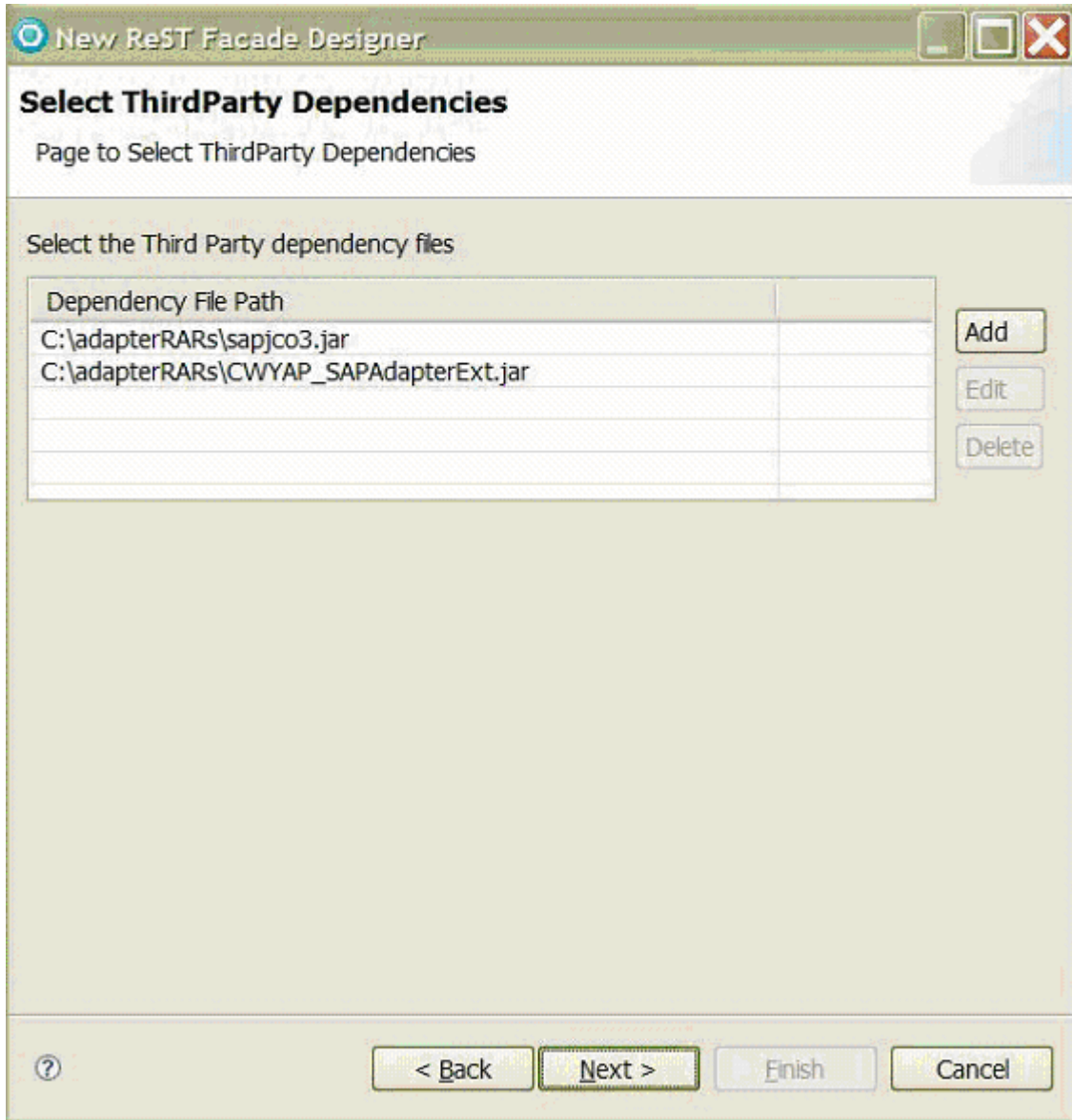
The Select ThirdParty Dependencies screen appears. You can add here any third party dependencies required by the adapter.



10. Click **OK**.



11. Click **Add**, to add a dependency file to the list.



12. Click **Finish**.

The Summary and Export Configurations screen appears, displaying the ReSTFacade wizard operation summary, with the port name, the selected Adapter RAR and the list of selected third party dependencies.

You can also export a WAR file to deploy in the managed mode, and to export the configuration information on the current wizard run.

13. Click **Finish** to complete the wizard run.

You can optionally export the entire wizard run in an xml format and import it for future runs using the export option provided at the last page of the wizard.

New ReST Facade Designer

Summary and Export Configurations

Page to display summary and export configurations

ReST facade Wizard Operation Summary

Port Configured

8080

Adapter RARs selected

SAP-C:\adapterRARs\CWYAP_SAPAdapter.rar

Third Party Dependencies selected

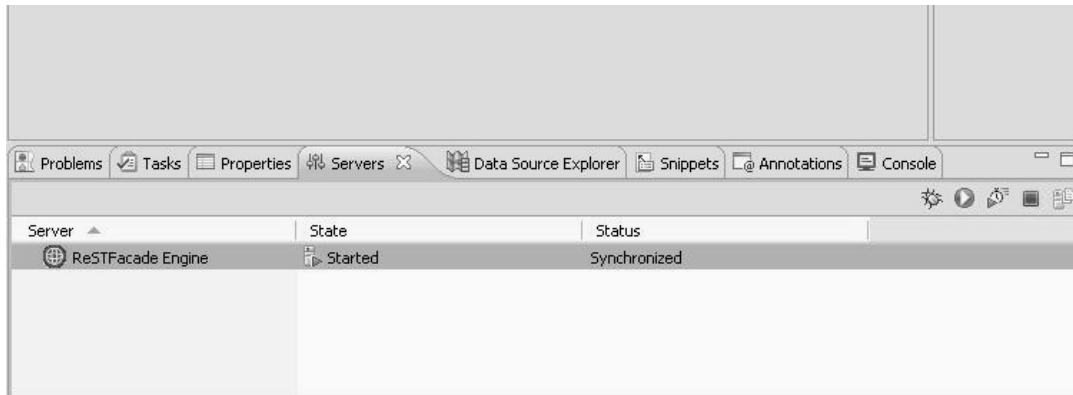
C:\adapterRARs\sapjco3.jar
C:\adapterRARs\CWYAP_SAPAdapterExt.jar

Advanced

Select the location to save the exported configuration

C:\exportConfig.xml

A new WebSphere Application Server CE instance, named ReSTFacade Engine appears in the server panel.



A ReSTFacade web application will be created and deployed on the server, and a new browser window appears, to start your discovery process.

You can browse, discover and invoke objects within an Enterprise Information System, by using the URI patterns defined for ReSTFacade. For more information, refer the section on URI patterns in this document.

In this environment, the adapter runs in the non-managed mode, and typically used for development and testing purposes.

Working with WebSphere Application Server

When you move to a production environment, using WebSphere Application Server, perform the following steps:

Note: In this environment, the adapter runs in the managed mode.

1. Install the Resource Adapter at node level (Standalone).
2. Download the ReSTFacade WAR file from the SupportPac site.
3. Install the WAR file as a web application in WebSphere Application Server.
4. Open ReSTFacade Administrator Console, and specify the following mandatory information:
 - HOSTNAME- the host name where WebSphere Application Server is running.
 - SOAPPOR- the port of the SOAP connector.
Make sure this is running since the Admin Client SOAP Connector is needed to perform administrative functions like the automatic creation of the J2C Connection Factory.
 - USERNAME- the user name for WebSphere Application Server.

- PASSWORD- the password for WebSphere Application Server.
 - ADAPTER NAME-the name of the installed adapter to use at the node level.
 - Name Server Bootstrap Port - Application server name server bootstart port
5. Creating a new data source configuration in WebSphere Application Server CE/ importing a pre-existing data source configuration from the Rational Application Developer environment will result in a J2C Connection Factory being created and configured for each data source instance.

Setting up logging, tracing and exception messages

Information about the runtime state of the ReSTful adapter is critical, to track operations. When you implement the `LogUtils` class, you can filter information by its levels of importance, and generate trace, log and exception messages.

ReSTFacade allows you to set up and generate Logging, Tracing and Exception messages.

- Log messages convey timely information on warnings about potential problems, errors that have occurred and suggested fixes for those errors, and information that is necessary or helpful to understanding how the adapter operates.
- Trace messages convey information that is intended for support teams and developers. Such information includes stack dumps for exceptions and operation logic for debugging purposes.
- Exception messages are describe a problem, its likely cause, the probable solution or workaround, and how to avoid a problem in future.

Note: *You may not be able to view the log/trace messages as the request is through an HTTP layer. To enable the view of log/trace messages, you have to open another port/URL through another HTTP request.*

About the Appendix Section

1. Appendix A describes the patterns of URIs needed to ReST enable BAPI transactions for SAP.
2. Appendix B describes how to use ReSTFacade User Interface in the development mode, to ReST enable BAPI transactions for SAP.
3. Appendix C describes how to use ReSTFacade Administrative Console to export all your configurations, made during development time (Appendix B).
4. Appendix D describes how to import the configurations made during development time, into the production server.
5. Appendix E lists the currently identified ReSTFacade limitations, and the means to working around these limitations.

Appendix A

Defining URL patterns for SAP data source manipulations

Here are a few sample URL patterns for an Enterprise Information system.

GET

<http://HOSTNAME:PORT/rwa/rwaservlet>

```
<?xml version="1.0" encoding="UTF-8" ?>
<rwa:collection xmlns:rwa="http://www.ibm.com/xmlns/prod/rwa">
<rwa:resource id="SAP" isCollection="true"
ref="http://HOSTNAME:PORT/rwa/rwaservlet/SAP"/>
</rwa:collection>
```

GET

<http://HOSTNAME:PORT/rwa/rwaservlet/SAP?define>

Returns an XSD format. This XSD document represents all the connection properties necessary to create a connection to an SAP system.

POST

<http://HOSTNAME:PORT/rwa/rwaservlet/SAP>

The payload of the HTTP request is a XML document which must be validated with the XSD format, returned by using the SAP ?define request.

The response will return the name of the newly created alias.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rwa:response xmlns:rwa="http://www.ibm.com/xmlns/prod/rwa">
<rwa:resource id="SAP1" ref="
http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1"/>
</rwa:response>
```

POST

<http://HOSTNAME:PORT/rwa/rwaservlet/SAP/customName>

The payload of the HTTP request is an XML document which must be validated with the XSD format, returned by using the SAP ?define request.

You can specify your own name for the alias. (Customer name)

The response will return the name of the alias of the specified user.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rwa:response xmlns:rwa="http://www.ibm.com/xmlns/prod/rwa">
<rwa:resource id="SAP1"
ref="http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1"/>
</rwa:response>
```

GET

<http://HOSTNAME:PORT/rwa/rwaservlet/SAP/alias?define>

Returns all the connection properties representing this alias.

URL Patterns for EIS Browsing and resource discovery

GET

<http://HOSTNAME:PORT/rwa/rwaservlet/SAP/alias>

Returns list of resources at the next level in the Enterprise Information System.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rwa:collection xmlns:rwa="http://www.ibm.com/xmlns/prod/rwa">
<rwa:resource id="RFC" isCollection="true" ref="http
http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1/BAPI"/>
</rwa:collection>
```

GET

<http://HOSTNAME:PORT/rwa/rwaservlet/SAP/alias/BAPI>

Returns list of resources at the next level in the Enterprise Information System.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rwa:collection xmlns:rwa="http://www.ibm.com/xmlns/prod/rwa">
<rwa:resource id="RFC" isCollection="true" ref="http
http://HOSTNAME:PORT/rwa/rwaservlet/SAP/SAP1/BAPI/RFC"/>
</rwa:collection>
```

GET

http://HOSTNAME:PORT/rwa/rwaservlet/SAP/alias/BAPI/RFC?id=BAPI_CUSTOMER_GETDET*

```
<?xml version="1.0" encoding="UTF-8" ?>
<rwa:collection xmlns:rwa="http://www.ibm.com/xmlns/prod/rwa">
<rwa:resource id="BAPI_CUSTOMER_GETDETAIL" isCollection="false"
ref="http://HOSTNAME:PORT/controller3/rwaservlet/SAP/SAP1/BAPI/RF
C/BAPI_CUSTOMER_GETDETAIL"/>
<rwa:resource id="BAPI_CUSTOMER_GETDETAIL1" isCollection="false"
ref="http://HOSTNAME:PORT/controller3/rwaservlet/SAP/SAP1/BAPI/RF
C/BAPI_CUSTOMER_GETDETAIL1"/>
<rwa:resource id="BAPI_CUSTOMER_GETDETAIL2" isCollection="false"
ref="http://HOSTNAME:PORT/controller3/rwaservlet/SAP/SAP1/BAPI/RF
C/BAPI_CUSTOMER_GETDETAIL2"/>
</rwa:collection>
```

GET

http://HOSTNAME:PORT/rwa/rwaservlet/SAP/alias/BAPI/RFC/BAPI_CUSTOMER_GETDETAIL?define

Returns the XSD document representing this resource (BAPI_CUSTOMER_GETDETAIL) in this Enterprise Information System.

GET

<http://HOSTNAME:PORT/rwa/rwaservlet/SAP/alias/BAPI/RFC/SapPeAddress.xsd>

Returns the corresponding XSD file. This can be used to see the child objects of the top level object. For example BAPI_CUSTOMER_GETDETAIL would return SapBapiCustomerGetDetail.xsd. This references SapPeAddress.xsd which can be obtained using this call.

The pre-requisite is that the resource (BAPI_CUSTOMER_GETDETAIL) should have been discovered previously using the ?define request.

URL Patterns for Invocation

POST

http://HOSTNAME:PORT/rwa/rwaservlet/SAP/alias/BAPI/RFC/BAPI_CUSTOMER_GETDETAIL

The payload will be a XML representing a representation of the resource you are manipulating. This should be a valid XML format, conforming to the XSD of the resource obtained during discovery using the ?define request.

Testing a ReSTful component with JCo and DLL files

(For SAP users)

WebSphere Adapter for SAP uses the SAP Java Connector (SAP JCo) API, to connect to SAP applications. SAP JCo requires a DLL file, to run in the ReSTFacade environment.

Copy the files to the server.

1. In the Rational Application Developer testing environment, copy the files to the `${WAS_INSTALL_ROOT}/runtimes/bi_v62/lib/ext` directory.
2. In a production environment, copy the files to the `${WAS_INSTALL_ROOT}/lib/ext` directory of WebSphere Application Server.
3. Add a `LIBPATH` variable for the `SAPJCO` folder.
4. Add a `CLASSPATH` variable that points to the `/SAPJCO/sapjco3.jar` directory
5. Restart the server and deploy the application again.
6. The `sapjco3.jar` and associated files are now part of your runtime environment.

Appendix B

Using ReSTFacade in the development/testing mode

You can use developer tools from standard browsers like posters, to make HTTP requests, and interact with Web services or resources. You can construct each element in an URI, piece by piece, to work the ReSTful way.

On the other hand, you can also use the ReSTFacade User Interface supplied with this offering, to browse and select a business object from an Enterprise Information System and ReST enable it.

Here are the advantages of using the ReSTFacade User Interface:

1. You do not have to search for, and test various posters and similar tools to make HTTP requests.
2. You can use the easy-to-follow user interface to build a request response, and avoid the element of human error, like when you have to perform URI constructs.
3. You can customize this user interface to respond to any technology, like SAP, JDBC, and so on.
4. You can then select a list of business objects (like BAPIs or tables) through HTTP in a ReSTful manner, as opposed to laboriously creating manual HTTP requests.

The following section explains how to use the ReSTFacade User Interface.

***Note:** This example demonstrates how the user interface has been customized for SAP, to select and work with data source instances. You can however, customize the user interface to work with any of the WebSphere Adapter technologies.*

Using the Create instance option

1. Click **Create instance**.

The Enter connection properties screen appears.

Enter connection properties

Note: If the adapter requires any third party libraries (ex: jar) or native libraries (ex: dll) refer the official WebSphere Adapter documentation to configure them.

Datasource Alias:

Application Server Host*:

Username*:

Password*:

System Number*:

Client*:

Language:

Codepage:

Gateway Host:

Gateway Service:

Ignore BAPI Return:

Load Balancing:

Partner Charset:

ABAPDebug:

2. Fill all the mandatory information, such as:

- Datasource Alias
- Application Server Host
- UserName
- Password
- System Number
- Client

The other fields are optional. For more information on these fields, refer the WebSphere Adapter for SAP Software User Guide.

3. Click **Submit**.

Once a data source alias is successfully created you can start your discovery/search process to ReST enable interested EIS object. You can start your discovery process by clicking on the data source alias link. Optionally you can also enter filter patterns to filter objects to be returned from EIS.

Filter pattern:

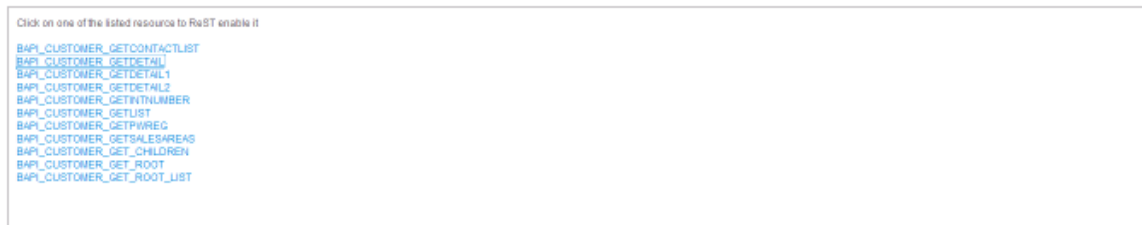
Click on link to browse and select the leaf level EIS resource(s)

[RFC](#)

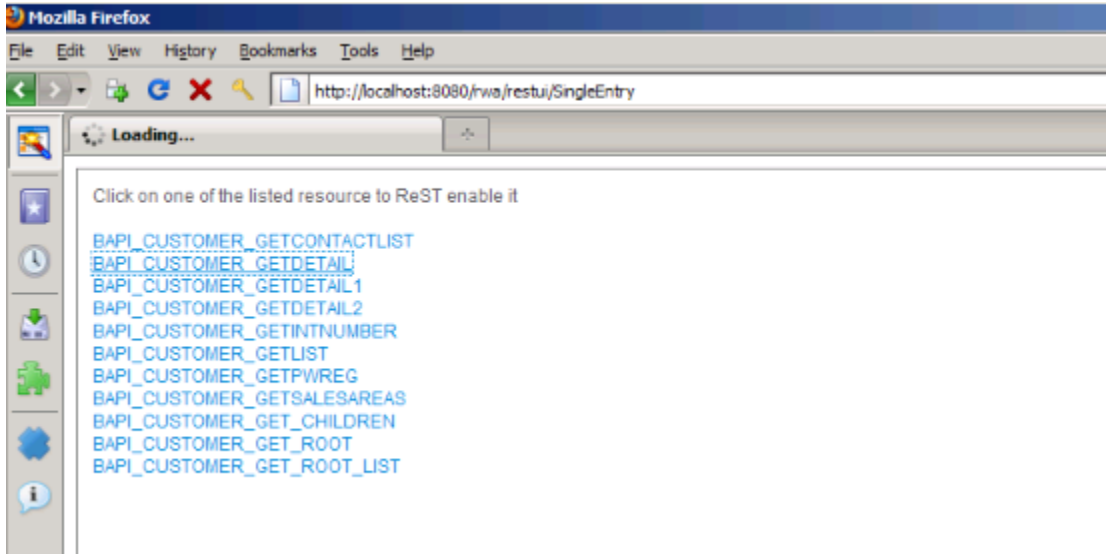
4. Click the link to browse and select a leaf level EIS resource.



5. Select a Filter pattern path, such as BAPI_CUSTOMER_GET* to make an HTTP GET request.
6. Click the RFC link.



A list of BAPI transaction links appears.



7. Click any link in this screen, for example, `BAPI_CUSTOMER_GET_DETAIL`.

| | |
|------------------------|------------|
| CustomerToBeRequired * | xsd:string |
| Division | xsd:string |
| SalesOrganization * | xsd:string |
| DistributionChannel | xsd:string |

How to invoke the BAPI via HTTP GET

You can use either one of the below URIs to access the BAPI via HTTP GET in a ReSTful manner. However you need to replace the values of the parameters with the actual values before invocation.

ReST invocation URL with only required attributes

URL in text form
`http://localhost:8080/rwa/hasen/istSAP/objoereta/istve/BAPI/RF0C/BAPI_CUSTOMER_GETDETAIL?CustomerToBeRequired=rl_it_me&SalesOrganization=rl_it_me`

URL as a link

ReST invocation URL with all attributes (of the top level BAPI)

URL in text form
`http://localhost:8080/rwa/hasen/istSAP/objoereta/istve/BAPI/RF0C/BAPI_CUSTOMER_GETDETAIL?PassBuffer=rl_it_me&CustomerToBeRequired=rl_it_me&Division=rl_it_me&SalesOrganization=rl_it_me&DistributionChannel=rl_it_me`

URL as a link

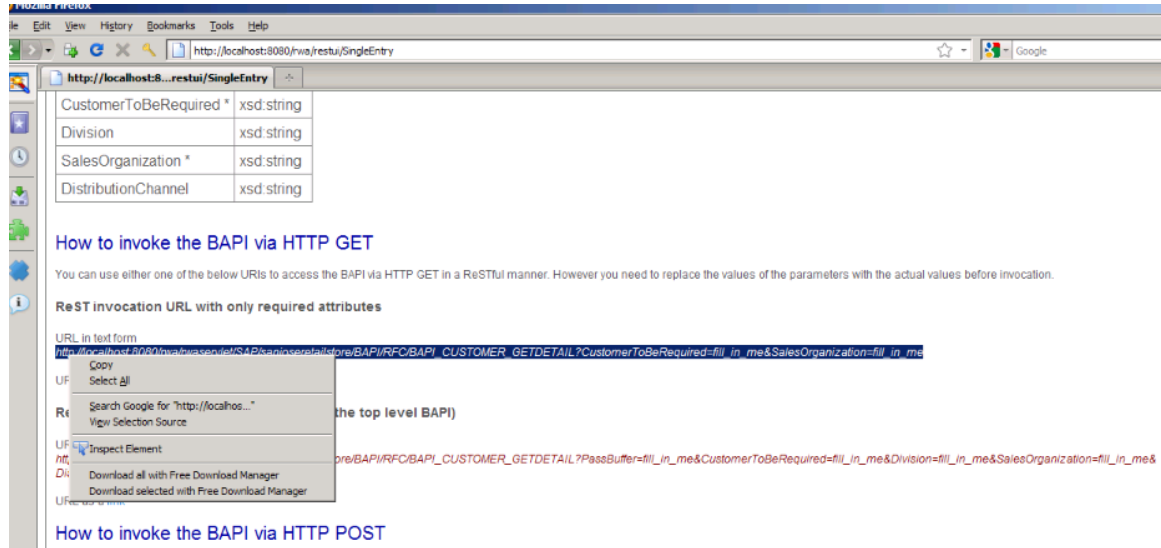
How to invoke the BAPI via HTTP POST

To invoke the BAPI via HTTP POST, you need to create an XML in accordance with the XSD (XML schema) displayed below and do a POST on the URL `http://localhost:8080/rwa/hasen/istSAP/objoereta/istve/BAPI/RF0C/BAPI_CUSTOMER_GETDETAIL`

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.ibm.com/xmlns/prod/websphere/j2ca/sap
/sapbapicustomergetdetail"
xmlns:sapbapicustomergetdetail="http://www.ibm.com/xmlns/prod/websphere
/j2ca/sap/sapbapicustomergetdetail" xmlns:sapasi="http://www.ibm.com
/xmlns/prod/websphere/j2ca/sap/metadata"
xmlns:sapaddress="http://www.ibm.com/xmlns/prod/websphere/j2ca/sap
/sapaddress" xmlns:sapreturn="http://www.ibm.com/xmlns/prod/websphere
/j2ca/sap/sapreturn">
<xsd:import namespace="http://www.ibm.com/xmlns/prod/websphere/j2ca/sap
/sapaddress" schemaLocation="SapAddress.xsd"/>
<xsd:import namespace="http://www.ibm.com/xmlns/prod/websphere/j2ca/sap
/sapreturn" schemaLocation="SapReturn.xsd"/>
<xsd:import namespace="http://www.ibm.com/xmlns/prod/websphere/j2ca/sap
/metadata" schemaLocation="..//SapBusinessObjectMetadata.xsd"/>
<xsd:annotation xml:space="preserve">
</xsd:annotation>
```

A screen with instructions on how to invoke the BAPI through the HTTP GET or POST method appears.

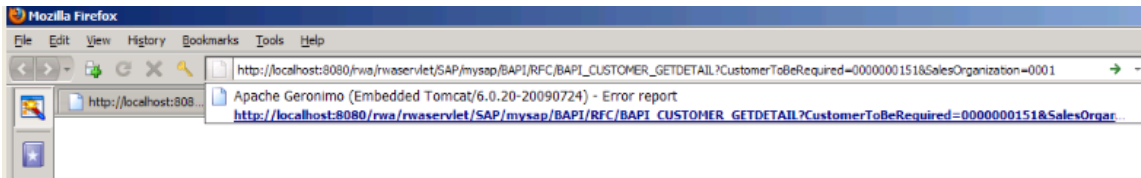
8. Select the URL in the text form link.



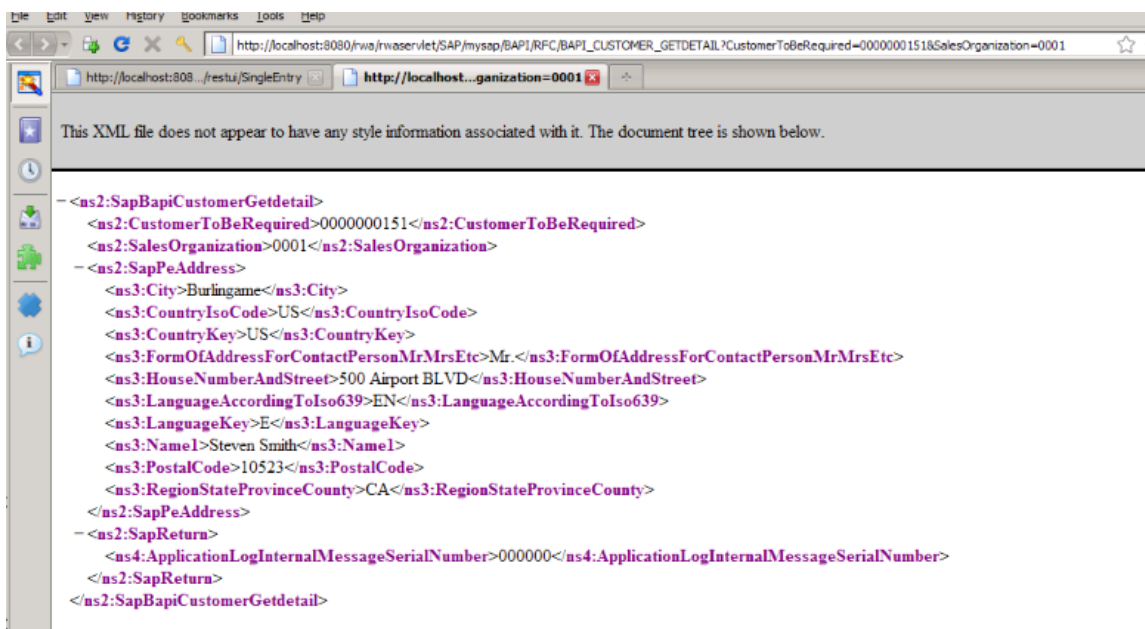
9. Edit the URL link that you copied, to substitute the parameter values with the actual values.

10. Paste the modified link in your default browser address area.

11. Click **Enter**.



The browser fetches the details of your BAPI transaction.



An XML rendition of the request appears in the browser. A very simple example is shown here. However, you can use the same link inside your web application or mashup, to invoke more complex BAPI transactions in a ReSTful manner.

Appendix C

Exporting configurations from development/testing mode

1. From the home page of ReSTFacade, choose IBM WebSphere ReSTFacade Administrative console.



2. Click the Configurations management link.



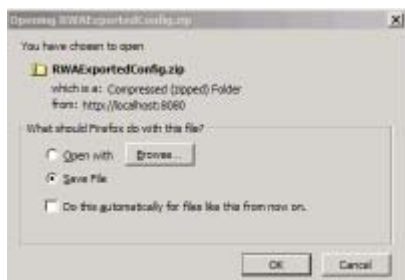
The IBM WebSphere ReSTFacade Administrative Console screen displays the following Export Configuration options.

- Business object repository - to export all the generated artifacts for objects that are already ReST enabled. These artifacts will be exported as zip format that you can import to your production system, or any other environment.
- Configuration settings - to export all connection related properties supplied to create data source configurations. These properties are automatically configured, when you import them into another (production) system.

3. Check one or both the Export Configuration options

Note: You can also check both the options, to export all the generated artifacts for objects that are already ReST enabled, and export all connection related properties to create data source configurations.

4. Click Export.



A screen that allows you to save the RWA ExportedConfig.zip. This is a compressed zip format file.

5. Save this file to import this into your production server later (see Appendix – D).

Appendix D

Importing the development mode configurations to the production mode

Defining application server properties

1. Click the IBM WebSphere ReSTFacade Administrative Console link in the IBM WebSphere ReSTFacade for WebSphere Adapters screen.

IBM WebSphere ReSTFacade for WebSphere Adapters

IBM WebSphere ReSTFacade enables you to quickly ReST enable your Enterprise Information Systems with the help of WebSphere Adapters, and make them available for other web or mashup applications.

For example, a WebSphere Adapters user may find a need to access and execute the SAP systems' BAPIs over HTTP in a ReSTful manner. User may also want to consume the URIs exposed by the ReSTFacade for WebSphere Adapters' component in another mashup or web application.

ReSTFacade can be accessed either via custom built UI IBM WebSphere ReSTFacade Web User Interface or via any custom HTTP client tools (example Firefox AddOn - POSTER) if you are an expert user.

1. **IBM WebSphere ReSTFacade Web User Interface**
An intuitive web application that guides the first time users thru several stages of ReST enablement of EIS resources. It supports creation of datasource alias, browsing EIS objects (ex: BAPI in case of SAP) and finally generation of ReSTful HTTP URIs for those selected objects.
2. **IBM WebSphere ReSTFacade Web Application**
Core web application which helps users ReST enable their EIS resources. Since this being complete ReSTful application you need custom HTTP tools/clients to access this. As an alternative approach you can use the custom built UI IBM WebSphere ReSTFacade Web User Interface from here

IBM WebSphere ReSTFacade Administrative Console

A web application provides features like export and import of configuration. In real life scenarios it is very critical when an application needs to be moved from test to production, ReSTFacade Administrative Console helps an user to export all the configurations he/she has created in test environment to production environment thereby saving time for the user.

Getting started

Three steps to ReSTenable EIS objects

Below are the typical steps to ReSTenable your EIS objects (say BAPI in case of SAP). Note that these can be done using IBM WebSphere ReSTFacade Web User Interface or by using custom HTTP client tool pointing to the right URL of core ReSTFacade application

- Create datasource instance(s) by supplying valid connection properties to the target EIS system
- Browse the interested objects. You can also enter filter patterns to optimize your search while browsing the objects
- Select the object to ReSTenable and invoke using the URL.

Three steps to migrate from 'Development' to 'Production' environment

- Follow above section 'Three steps to ReSTenable EIS objects' to expose your EIS objects as ReSTful resources.
- Create a datasource instance in your production environment with the same name as that of your development environment.
- Use the IBM WebSphere ReSTFacade Administrative Console to export the generated artifacts and import them into your production

The IBM WebSphere ReSTFacade Administrative Console screen appears.

The IBM WebSphere ReSTFacade Administrative console allows you to export the configurations you have generated, and import them to your production environment. This is especially useful when you want to migrate from your development environment to a production environment.

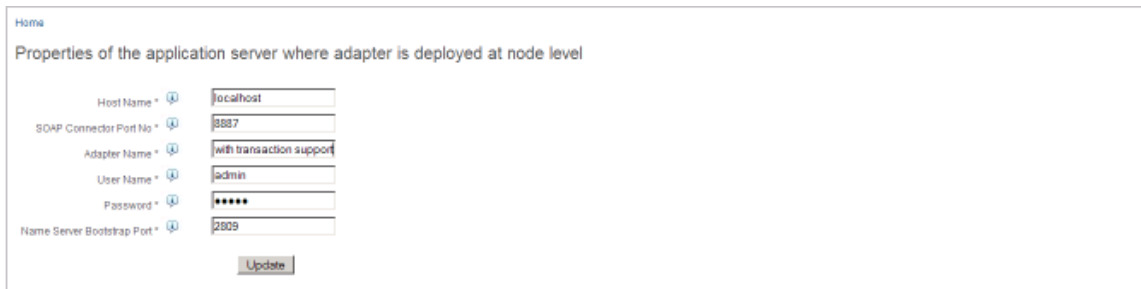
You need to configure a set of properties so that ReSTFacade can successfully communicate with the binary, deployed at the node level on your target application server.

Note: *It is important that you configure the properties required to connect to the application server. Otherwise, it is likely that ReSTFacade will not function properly.*



2. Click the Properties needed for ReSTFacade to connect to application server link.

You can now specify the properties of the application server, where you will deploy an adapter.



3. Enter the required information against the following fields, as shown in the figure:

- Host Name
- SOAP Connector Port No
- Adapter Name
- User Name
- Password
- Name Server Bootstrap Port

4. Click Update.



After you have filled your information correctly, you will get an appropriate notification.

5. Click the Home link.

The IBM WebSphere ReSTFacade Administrative Console screen appears.



6. Click the Configurations management link.

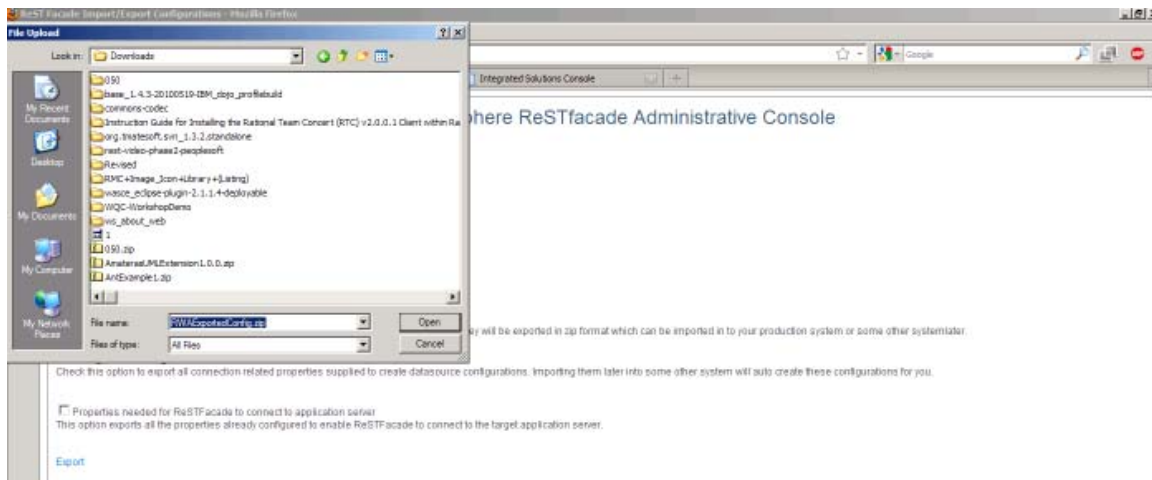
The Import and Export Configurations screen appears.

From this screen, you can browse and select the configuration file you want to import. Additionally, you can also use the following export options:

- Business object repository - to export all the generated artifacts for objects that are already ReST enabled. These artifacts will be exported as zip format that you can import to your production system, or any other environment.
- Configuration settings - to export all connection related properties supplied to create data source configurations. These properties are automatically configured, when you import them into another system.
- Properties needed to ReSTFacade to connect to the application server- to export all the properties you have already configured, so that you enable ReSTFacade to connect to the target application server.



7. Click Import to select the file you want to import.



8. Click Import. A screen will display an acknowledgement for your import action. You can now fetch your import configurations by using ReSTFacade Web User Interface.
9. Click the IBM WebSphere ReSTFacade Web User Interface link from the home page.



From this screen, you can select the list configured instances, to verify the new configurations that are just imported.

Once you have successfully imported the configurations, you can directly invoke the BAPI using the production server URL. The required artifacts are already deployed and ready for execution when you completed the import operation.

To construct the production server URL, you can copy the development server URL and replace it with the target production server host and port information, as well as the context of the ReSTFacade application.

Appendix E

Working around currently identified ReSTFacade limitations

The following defects or inconsistencies will be addressed in the next release:

1. Users can invoke a ReST enabled BAPI even though the URL is pointing to another BAPI. For example if you have a discovery GET_CUSTOMER_BAPI and try to make a HTTP POST with another BAPI URL, it will still work as the current version of ReSTFacade does not maintain a map to cross check the payload.
2. There is no RFC trace route generated in the HTTP default path or bin directory. However, some users have identified the presence of a trace file in the WAS CE/bin. This is a minor issue, and will not impact the current core performance of ReSTFacade.
3. Error messages are returned when users try to create a connection alias with a default format. Aliases that are either automatically generated, or those aliases that do not cross automatically generated patterns, do not return error messages.
4. Currently there is no support for connection to an SAP system that is configured for load balancing, such as using MessageServerHost, SystemID, or LogonGroup properties. Users can connect to an SAP system, by specifying the SAP application server host name in the ApplicationServerHost property.
5. Currently, when the ReSTFacade Designer wizard completes its assigned tasks, a new browser window appears. By default, the browser window will display the pathname "http://localhost:8080/rwa/index.html". If users have configured the server on a port other than 8080, the URL has to be modified to http://localhost:<your_server_port_no>/rwa/index.html.
6. ReSTFacade will not support an EIS resource which has '/' (slash) in its name. In such scenarios, you will find unexpected behavior. As a workaround, you can rename the EIS resource to remove all the slash (es) and try ReST enabling the resource.

7. At times you may encounter issues when ReST enabling BAPI through ReSTFacade Web User Interface. It is not a consistent issue. However when you use POSTER or other HTTP tools, your operation will work fine because the issue is related only to the web UI, and not the core ReSTFacade component.
8. There will be issues, when you “export” the configurations immediately after another “export” from ReSTFacade Administrative Console. For example, you may see unexpected behavior when you click on ‘Export’ immediately after you are done with ‘Export’.

Product Feedback and suggestions

Please spare a few moments of your time, to respond to these few questions.

1. Do you think the ReST based access model was useful? If so, in what sort of scenarios was it useful?
2. Was it easy to install the ReSTFacade for IBM WebSphere Adapters?
YES or NO. If NO, what suggestions do you have to make the installation easier?
3. How easy did you find it to use the ReSTFacade?
Easy? Somewhat easy? Not Easy? Any suggestions?
4. How useful was the documentation provided?
Useful? Somewhat useful? Not Useful? Any suggestions?
5. How useful was the documentation on pre-requisites?
Useful? Somewhat useful? Not Useful? Any suggestions?
6. Is the user interface friendly and intuitive?
YES or NO. If NO, what suggestions do you have, to make the user interface more user friendly?
7. This version currently only exposes the SAP BAPI interface. What other interfaces or EIS connectivity (for example, JDBC, PeopleSoft), do you feel would be more useful in your environment?
8. Would you like this offering to be officially productized?
9. Would you like to add any additional comments?

Thank you for taking the time to give us your observations. Our development team will review your comments, to help us improve your experience and our product. You can direct your response to raj कुमार@in.ibm.com.