Oracle® Cloud Adapter

User Guide for Oracle Cloud Adapter for Salesforce.com

12c Release (12.1.3.0.0)

October 31, 2014

ORACLE

Oracle Fusion Middleware User Guide for Oracle Cloud Adapter for Salesforce.com, 12c (12.1.3.0.0) Release

Copyright © 2014 Oracle and/or its affiliates. All rights reserved.

Primary Author: Vishal Bhardwaj

Contributors: Bo Stern, Robert May, Ravindran Sankaran, Simeon Greene, Brian Volpi, Shalindra Singh, Mohammad Usman Ul Haque, Monika Ahuja, Priyanka Gupta, Himanshu Grover, Amanpreet Wraich, Raman Dhawan, Marian Jones, and Vikas Anand.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Content

Content ..... **iii**

Preface ..... **viii**

Audience viii
Documentation Accessibility viii
Related Documents viii
Conventions ix

Part I Introduction to Oracle Cloud Adapter for Salesforce.com **1**

1.1 Architecture 1
1.1.1 Architecture of Oracle Cloud Adapter for Salesforce.com 2
1.2 Lifecycle 3

Part II Oracle Cloud Adapter for Salesforce.com ..... **i**

1 Oracle Cloud Adapter for Salesforce.com *1-1*
1.1 Overview 1-1
1.1.1 Functional Overview 1-1
1.1.2 Design Overview 1-2
1.2 Supported Versions and Platforms 1-3
1.2.1 Salesforce.com 1-3
1.2.1.1 Supported Versions 1-3
1.2.1.2 Supported WSDLs 1-4
1.2.1.3 Supported API 1-4
1.2.1.4 Support for Outbound 1-5
1.2.2 Oracle SOA Versions 1-5

2 Getting Started *2-1*
2.1 Oracle Cloud Adapter for Salesforce.com Connection Parameters 2-1
2.1.1 User Name 2-1
2.1.2 Password 2-1
2.1.3 CSF Key 2-2
2.1.4 Downloaded Enterprise WSDL 2-2
2.2 IP Address Registration and Restrictions 2-2
2.3 Salesforce.com Permissions 2-5
2.4 Importing Salesforce.com Certificate into Client/Server 2-6
2.4.1 Generating Salesforce.com Certificate 2-6
2.4.2 Importing Salesforce.com Certificate using Keytool 2-10
2.5 Enabling Oracle Cloud Adapter for Salesforce.com in Design-Time and Run-Time 2-12
2.5.1 What are Design-Time and Run-Time? 2-12

2.6 Import the WSDL File into Your Development Platform....2-14

3 Oracle Cloud Adapter for Salesforce.com - Supported Features....3-1
3.1 SOAP API....3-1
3.2 Supported SOAP API Operations....3-2
3.2.1 CORE Operations....3-2
3.2.1.1 convertLead....3-2
3.2.1.2 getDeleted....3-3
3.2.1.3 getUpdated....3-4
3.2.1.4 merge....3-5
3.2.1.5 undelete....3-5
3.2.1.6 upsert....3-6
3.2.2 CRUD Operations....3-7
3.2.2.1 create....3-7
3.2.2.2 retrieve....3-8
3.2.2.3 update....3-9
3.2.2.4 delete....3-9
3.2.3 MISC Operations....3-10
3.2.3.1 getUserInfo....3-11
3.2.3.2 process....3-12
3.2.4 SOSL and SOQL Operations....3-13
3.2.5 query....3-14
3.2.6 queryAll....3-15
3.2.7 search....3-15
3.2.8 queryMore....3-16
3.3 Salesforce.com SOAP Headers....3-17
3.3.1 Request Headers....3-18
3.3.1.1 AllOrNoneHeader....3-18
3.3.1.2 AllowFieldTruncationHeader....3-19
3.3.1.3 AssignmentRuleHeader....3-19
3.3.1.4 EmailHeader....3-20
3.3.1.5 DebuggingHeader....3-20
3.3.1.6 MruHeader....3-21
3.3.1.7 PackageVersionHeader....3-21
3.3.1.8 QueryOptions....3-21
3.3.2 Response Headers....3-22
3.3.2.1 DebuggingInfo....3-22
3.3.2.2 LimitInfoHeader....3-22
3.4 Session Management....3-22
3.4.1 Design Phase....3-22
3.4.2 Execution Phase....3-23
3.5 Handling Polymorphic Behavior of Salesforce.com Schema....3-24
3.6 Clear Cache....3-26
3.7 Offline Mode....3-26
3.8 Behaviors Expected When Using Offline Mode....3-28
3.9 Security Management....3-28
3.10 Test Functionality....3-28
3.10.1 Design-Time Test Functionality....3-29
3.11 Suppress Response....3-29
3.12 Fault Handling....3-30
3.13 Salesforce.com Limit and Restriction Handling....3-31
3.14 Support for Bind Parameters....3-34

4 Design-Time: Using Oracle Cloud Adapter for Salesforce.com Configuration Wizard 4-1
4.1 Oracle Cloud Adapter for Salesforce.com Plug-in 4-1
4.2 Oracle Cloud Adapter for Salesforce.com Walkthrough 4-1
4.2.1 Welcome Page 4-2
4.2.2 Salesforce Cloud Server Connection Page 4-3
4.2.2.1 WSDL Location 4-4
4.2.2.2 Authentication Key 4-5
4.2.2.3 Offline Configuration 4-6
4.2.2.4 Clear Cache 4-7
4.2.2.5 Test Connection Functionality 4-7
4.2.3 Salesforce.com Cloud Operation Configuration Page 4-7
4.2.3.1 Operation Category 4-8
4.2.3.2 SFDC Operation 4-9
4.2.3.3 API Version 4-9
4.2.3.4 WSDL Operation 4-9
4.2.3.5 Suppress Response 4-9
4.2.3.6 Business Objects: 4-9
4.2.3.7 SOQL and SOSL Page 4-10
4.2.4 Header and Properties Page 4-13
4.2.5 Finish Page 4-14
4.3 Design-time Artifact Generation 4-15
4.3.1 JCA File 4-15
4.3.2 Integration WSDL 4-16

5 Integration with Different Service Components (BPEL/Mediator) in Oracle SOA Suite 5-1
5.1 Overview 5-1
5.2 Configuring the CSF Key on Enterprise Manager Console 5-2
5.3 Designing a Composite for Service Integration 5-2
5.3.1 Define Composite for BPEL and Mediator 5-2
5.4 Configure Oracle Cloud Adapter for Salesforce.com 5-8
5.5 Integration with BPEL 5-19
5.6 Deploy the Composite 5-27
5.7 Test the Composite 5-28
5.7.1 Test the Outbound Process 5-28

6 Configuring Outbound Processing Using Oracle Service Bus 6-1
6.1 Overview of Application Adapter Integration with Oracle Service Bus 6-1
6.2 Creating Outbound Processes Using Oracle Service Bus 6-1
6.2.1 Creating OSB Projects Using OSB Console 6-2
6.2.2 Creating OSB Projects Using JDeveloper 6-8
6.3 Testing OSB project from Service Bus Console 6-24

7 Configuring the Outbound Processing Using BPM 7-1
7.1 Overview 7-1
7.2 Configuring the CSF Key on Enterprise Manager Console 7-1
7.3 Designing a Composite for Service Integration 7-2
7.3.1 Define Composite for BPM 7-2
7.4 Configure Oracle Cloud Adapter for Salesforce.com 7-8
7.5 Integration with BPM 7-19
7.6 Deploy the Composite 7-46
7.7 Test the Composite 7-47
7.7.1 Test the Outbound Process 7-47

8 Configuring the Oracle Cloud Adapter for Salesforce.com on Oracle WebLogic Server 8-51

8.1 Oracle Cloud Adapter for Salesforce.com Run-Time Properties 8-51
8.1.1 Generic Properties 8-51
8.1.2 Properties available in the response 8-52
8.1.3 Retry Properties 8-54
8.1.3.1 Providing the Property Values in the Composite 8-55
8.1.3.2 Providing the Property Values at Enterprise Manager Console 8-56
8.1.4 Precedence of Salesforce.com Property Values 8-57

9 Troubleshooting and Error Messages 9-1
9.1 Troubleshooting and Error Messages 9-1
9.1.1 Oracle SFDC Cloud Adapter Design-Time JDeveloper 9-1
9.1.2 Oracle SFDC Cloud Adapter Run-time 9-2
9.2 API Fault 9-3
9.3 Status Code 9-3
9.4 Known Issues 9-3
9.5 Limitations 9-6

10 Migration Support 10-1
10.1 Complete Backward Compatibility 10-1
10.2 Migrating 11g Application and Projects to 12c 10-1
10.3 Points to Remember 10-7

11 Oracle Cloud Adapter for Salesforce.com Use Cases 11-1

Part I BPEL Use Cases 11-2

11.1. How can I build integration in which the SOA client is not interested in the response? 11-2
11.1.1. Creating the BPEL Process 11-2
11.2. How to use query and queryMore Operations? 11-33
11.2.1. Create BPEL Process 11-34
11.3. Creating Integration using Oracle Cloud Adapter for Salesforce.com without internet connection or when Salesforce.com is down. 11-72
11.4. How to use MDS for importing WSDL into JDeveloper? 11-75
11.5. How to use Debug Header (Response header)? 11-84

Part II BPM Use Cases 11-118

11.6. Define Composite for BPM 11-118
11.7. Configure Oracle Cloud Adapter for Salesforce.com 11-125
11.8. Integration with BPM 11-136
11.9. Deploy the Composite 11-165
11.10. Test the Composite 11-166
11.10.1. Test the Outbound Process 11-167

A Appendix 1

A.1 Generating the Enterprise WSDL 1
A.2 CSF Key in Enterprise Manager 4

Glossary 1

Terms and Acronyms 1

Index 1

Preface

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

Oracle Fusion Middleware User Guide for Oracle Cloud Adapter for Salesforce.com is intended for those who wish to use the Adapter for integrating Applications with Salesforce.com.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program Web site at
`http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit
`http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit
`http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Fusion Middleware 12c Release (12.1.3.0.0) documentation set:

- *Oracle Fusion Middleware Programming Resource Adapters for Oracle WebLogic Server*
- *Oracle Fusion Middleware User's Guide for Technology Adapters*
- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*
- *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*
- *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Part I

Introduction to Oracle Cloud Adapter for Salesforce.com

This chapter provides an introduction to the Oracle Cloud Adapter for Salesforce.com.

With an exponential increase in adoption of Cloud Based Applications across Enterprises, there is also an increased need for simplified, reliable and secure connectivity between these applications with other Cloud and On-Premise Applications. Most Cloud Applications in the market today have their own distinct data model and unique mechanisms for enabling connectivity, authentication, authorization and session management, etc. This disparity adds complexity in the development and maintenance of integrations, and an increased time-to-market as the number of Applications in the Enterprise integration mix grows. Oracle Fusion Middleware has significantly simplified integrations with these Cloud Applications by providing a standard platform for integration that not only enables connectivity, but also lays a strong foundation to address aspects of audits, compliance, security and governance.

The Oracle Cloud Adapters, a key component of the Oracle SOA Suite, builds on the above platform to enable Standards based Connectivity to Cloud based Applications from On-Premise, Legacy and other Cloud Applications, while significantly simplifying the overall life-cycle and user experience. It shields the integration modeler from hand-coding and configuring dedicated logic for handling connectivity, security, and session management etc. individually for each Cloud Application being integrated with. It also safeguards the user from the need for in-depth expertise on the complex functional and technical knowledge of the applications. Thus, with these Adapters addressing all the requisites for managing integration with the applications, the Developers can focus on building the business logic for the integration and business processes.

This chapter contains the following sections:

- Section 1.1, "Architecture"
- Section 1.2, "Lifecycle"

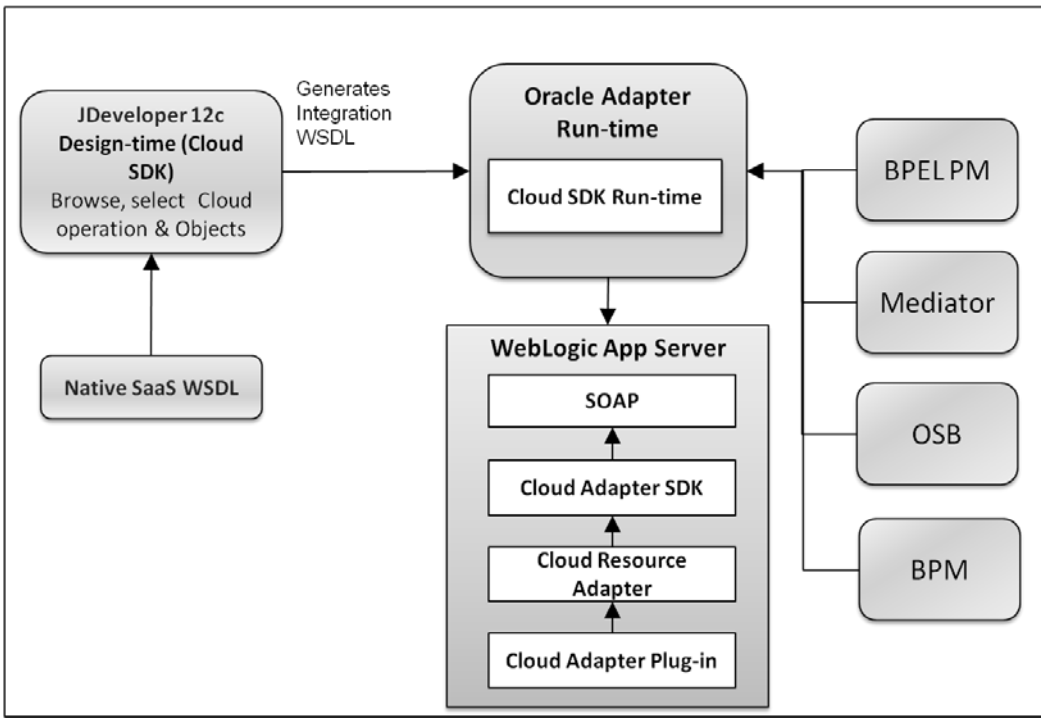
1.1 Architecture

Oracle Cloud Adapter framework and SDK are the new offerings provided by Oracle in Oracle SOA Suite. Figure 1-1 depicts the architecture of Oracle Cloud Framework. The Oracle Cloud Framework has two main components:

Design-time: Oracle Cloud Adapters are configured at design-time via the Adapter Configuration Wizard in JDeveloper. The Wizard leverages the underlying Cloud SDK to facilitate browsing of cloud application metadata and generation of project artifacts to be used at runtime.

Run-time: It includes Oracle run-time cloud SDK which helps in creation of cloud application (e.g. Salesforce.com) adapter run-time plug-in. Oracle WebLogic server is having Cloud Framework running, supported by Oracle Cloud Run-time SDK, which is a backbone of all cloud adapters. Cloud Framework uses one common JNDI name which is being referred by all cloud adapters.

Figure 1- 1 Architecture of Oracle Cloud Adapter



1.1.1 Architecture of Oracle Cloud Adapter for Salesforce.com

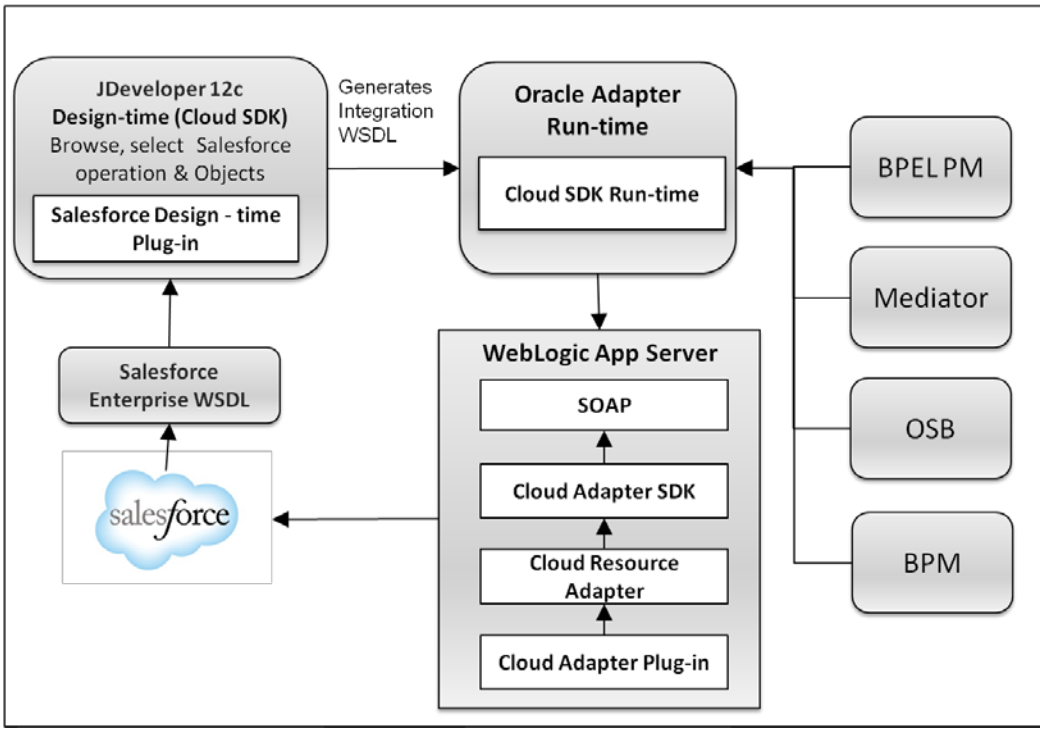
The Oracle Cloud Adapter for Salesforce.com was built using the Oracle Cloud Adapter Framework.

Design-time: The Salesforce.com Adapter is configured using the Adapter Configuration Wizard within JDeveloper. User can drag and drop the adapter from the component palette to the External References swim lane to start configuring the adapter. The Wizard enables the user to graphically browse and select Business Objects and Operations of interest for integration.

Run-time: The Runtime component of the Salesforce adapter implements the Cloud Runtime SDK to interact with Salesforce.com Enterprise WSDL SOAP APIs.

Figure 1-2 shows the Architecture of Oracle Cloud Adapter for Salesforce.com.

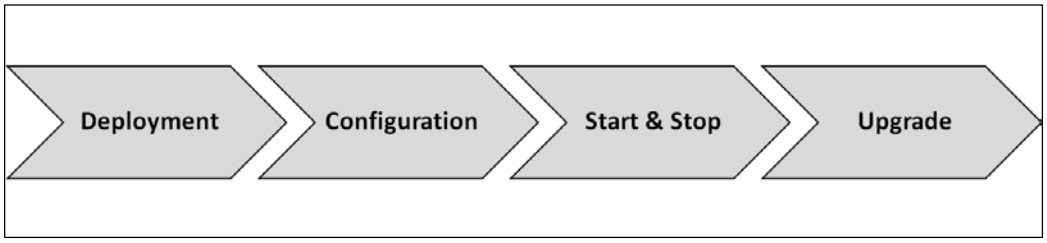
Figure 1- 2 Architecture of Oracle Cloud Adapter for Salesforce.com



1.2 Lifecycle

In general, the lifecycle of an Oracle Cloud Adapter for Salesforce.com instance includes the following four stages, as shown in Figure 1-3.

Figure 1- 3 lifecycle of Oracle Cloud Adapter for Salesforce.com



Deployment: Oracle Cloud Adapter for Salesforce.com gets installed and deployed as per the instructions provided in the section, "Deploy the Composite".

Configuration: Oracle Cloud Adapter for Salesforce.com installation and deployment happens with a default configuration provided by Oracle. If you want to change design-time configuration files, refer to the section, "Deploy the Composite".

Start and Stop of Adapter: Since Oracle Cloud Adapters architecture is different from JCA Adapters, the deployment section of WebLogic console does not show cloud adapters. The Cloud Adapter instances are stopped when the WebLogic server is shut down and similarly the instances are started when WebLogic server is booted.

Upgrade: Oracle will release the latest version of Oracle Cloud Adapter for Salesforce.com either in the form of OPatch or bundled with new versions of the Oracle SOA Suite. Oracle Cloud Adapter for Salesforce.com can be upgraded without impacting current functionalities. Future upgrades of Oracle Cloud Adapter for Salesforce.com will also have backward compatibility, so that the existing composites do not become obsolete in the upgraded environment.

Part II

Oracle Cloud Adapter for Salesforce.com

This part provides an overview of the Oracle Cloud Adapter for Salesforce.com. It contains the following chapters:

- Oracle Cloud Adapter for Salesforce.com
- Getting Started
- Oracle Cloud Adapter for Salesforce.com - Supported Features
- Design-Time: Using Oracle Cloud Adapter for Salesforce.com Configuration Wizard
- Integration with Different Service Components (BPEL/Mediator) in Oracle SOA Suite
- Configuring Outbound Processing Using Oracle Service Bus
- Configuring the Oracle Cloud Adapter for Salesforce.com on Oracle WebLogic Server
- Troubleshooting and Error Messages
- Migration Support
- Oracle Cloud Adapter for Salesforce.com Use Cases

1

Oracle Cloud Adapter for Salesforce.com

The Oracle Cloud Adapter for Salesforce.com enables integration to Salesforce CRM using Oracle SOA Suite and BPM.

This chapter contains the following topics:

- Section 1.1, "Overview"
- Section 1.2, "Supported Versions and Platforms"

1.1 Overview

Oracle Cloud Adapter for Salesforce.com is a key component within Oracle Fusion Middleware that enables Integration of On-Premise and SaaS applications with Enterprise, Unlimited or Developer Editions of Salesforce.com

1.1.1 Functional Overview

The Oracle Cloud Adapter for Salesforce.com allows seamless connection of different systems to Salesforce.com. It uses the Oracle SOA Suite to communicate with Salesforce.com, leveraging the SOAP API of Salesforce and supports all the major operations of SOAP API for standard as well as custom objects. The Salesforce.com WSDLs are polymorphic, for instance, the element sObject can refer to a plethora of objects, say account, contact, lead, etc. The support for this polymorphic behavior is limited in other options available in the market to integrate with Salesforce.com. The Oracle Cloud Adapter for Salesforce.com provides extensive support for this polymorphic behavior of Salesforce.com WSDLs through a simplified user interface. It enables you to define the operation you wish to perform in your API call and the object you wish to modify with this call. The integration WSDL that is generated is not polymorphic and contains the selected operation and object. The Oracle Cloud Adapter for Salesforce.com also provides a market-leading session management capability.

The adapter is compatible with the last six versions of Salesforce.com as described in section Salesforce.com "Supported Versions".

The Oracle Cloud Adapter for Salesforce.com is based on the Cloud Adapter Framework. The Cloud Framework relies on an underlying JCA framework which uses a common JNDI for all cloud adapters in Oracle SOA Suite 12c. The operations to be performed (CREATE, UPDATE, DELETE, etc.) are modeled using the design-time wizard. (For more information, refer to the section "
Design-Time: Using Oracle Cloud Adapter for Salesforce.com Configuration Wizard"). The inputs and outputs of the Oracle Cloud Adapter for Salesforce.com are XML; this allows the Oracle Cloud Adapter for Salesforce.com to be plugged into Oracle Fusion Middleware and enable the use of other SOA tools for XML transformations and assignments.

To connect to Salesforce.com, there are certain prerequisites:

- Enterprise WSDL. For more information, refer to the section "A.1 Generating the Enterprise WSDL"
- Valid Salesforce.com credentials. For more information, refer to the section "Oracle Cloud Adapter for Salesforce.com Connection Parameters".

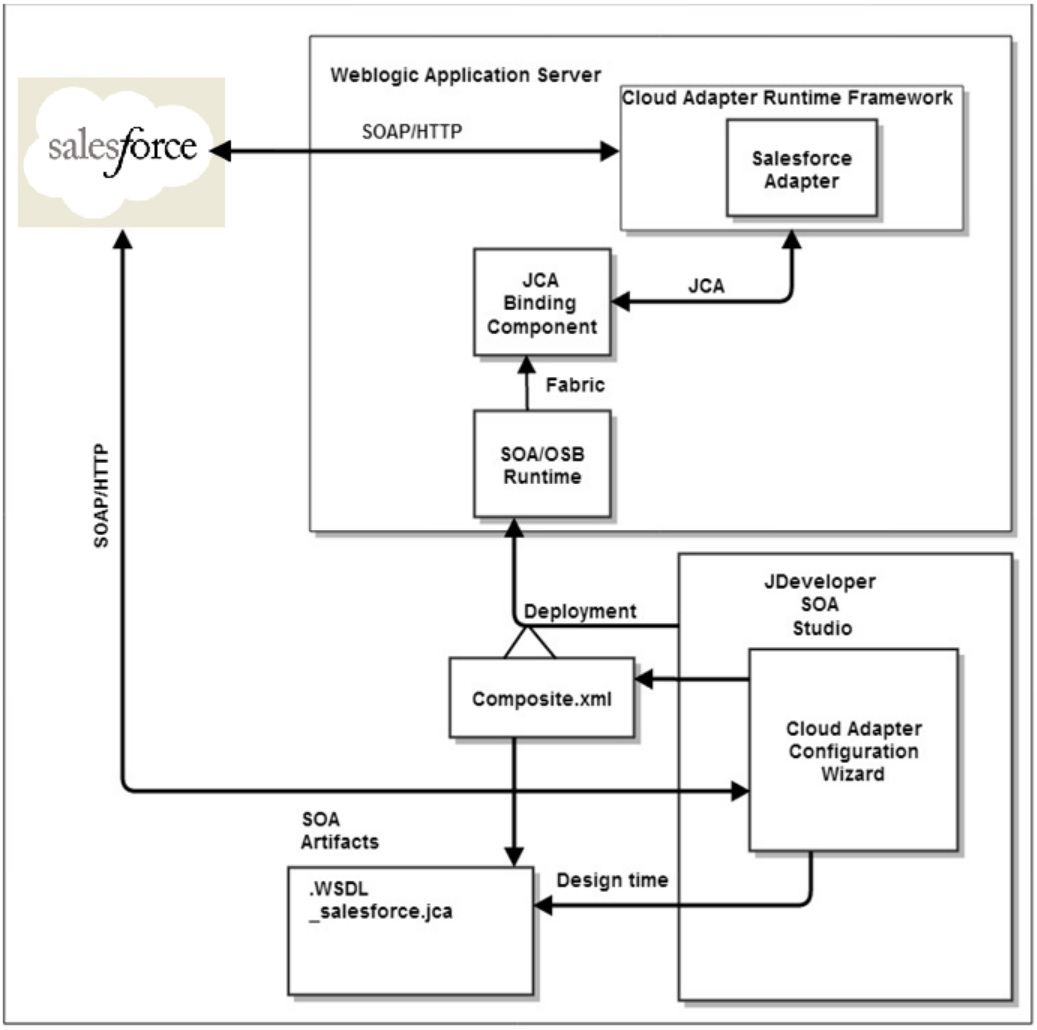
- Client certificates. For more information, refer to the section "Importing Salesforce.com Certificate into Client/Server".

The Oracle Cloud Adapter for Salesforce.com can be used at present, in an SOA/OSB/BPM process as described in section "Designing a Composite for Service Integration".

1.1.2 Design Overview

This section gives an overview about the design of Oracle Cloud Adapter for Salesforce.com. Figure 1-1 shows how the Oracle Cloud Adapter for Salesforce.com interacts with the various design-time and deployment artifacts.

Figure 1-1 How the Oracle Cloud Adapter for Salesforce.com Works



Description of Figure 1-1 How the Oracle Cloud Adapter for Salesforce.com works.

- The Oracle Cloud Adapter for Salesforce.com is an adapter which has been built using the Oracle cloud SDK. For more information on deployment, refer to the section, "Deploy the Composite".
- The Oracle Cloud Adapter for Salesforce.com can have multiple adapter instances; an adapter instance is basically a configured Oracle Cloud Adapter for Salesforce.com, which can connect with Salesforce.com and invoke the Salesforce Cloud operation which has been selected by the user during the configuration. Each adapter instance has its own artifacts i.e. Integration WSDL and JCA file.

- Because each adapter instance points to a single Salesforce operation, there is a one-to-one correspondence from adapter instances to Salesforce.com cloud operation. To see the list of supported operation, refer to the section "Supported SOAP API Operations".
- The adapter instances are part of the SOA composites. Each time the adapter wizard is run, it creates one instance of the Cloud Adapter. A Cloud Adapter instance consists of a jca file, a WSDL, and a reference element added to the composite.xml.

  For more information, refer to the section "Oracle Cloud Adapter for Salesforce.com Walkthrough".
- The artifacts generated in the above steps are used by the Oracle Cloud Adapter for Salesforce.com which is part of the Cloud Adapter Run-time framework for processing the request and response from Salesforce.com.

  Table 1-1 shows the Adapter Configuration Wizard Generated SOA Composite Adapter Artifacts.

Table 1-1 Adapter Configuration Wizard Generated SOA Composite Adapter Artifacts

File	Description
<serviceName>.wsdl	The Oracle Cloud Adapter for Salesforce.com generates this non-polymorphic integration WSDL for seamless integration to Salesforce.com. An abstract WSDL that defines, the name of operations and objects selected along with the request and response structures. Unlike the Enterprise WSDL available from Salesforce.com, it is not polymorphic and is therefore much more integration-friendly.
<serviceName>_salesforce.jca	The JCA file contains the internal implementation details used by the adapter during run-time. It contains the different interaction and connection properties used by the adapter. The operations describe the action that needs to be taken against the end point, such as Create, Update, etc. The contents of the file are determined by choices made during the Adapter Configuration.

1.2 Supported Versions and Platforms

This section provides an overview of the different versions and platforms of Salesforce.com as well as of different operating systems supported by the Oracle Cloud Adapter for Salesforce.com.

1.2.1 Salesforce.com

1.2.1.1 Supported Versions

The Oracle Cloud Adapter for Salesforce.com supports the previous six versions of Salesforce.com starting from the current version. The list of supported Salesforce.com Editions is given below:

- v25
- v26
- v27
- v28
- v29

- v30

1.2.1.2 Supported WSDLs

A Web Service Description Language (WSDL) file is required to access the Salesforce.com Web Service. It defines the available web service. It is used by the development platform to generate the API required to access Salesforce.com web service. The WSDL can be generated directly from Salesforce.com UI if one has access to the download page, or it can be requested from the organization's Salesforce administrator. Web Service Description Language (WSDL) contains all the relevant information required to invoke a web service. It is extensible and allows the description of endpoints and their messages irrespective of the network protocols and message formats. Oracle Cloud Adapter for Salesforce.com supports integration with Salesforce.com via the Enterprise WSDL.

Generate the most-recent WSDL for the organization by clicking on Setup>Develop>API. For steps on how to generate the Enterprise WSDL for your Salesforce.com organization, see section "A.1 Generating the Enterprise WSDL".

- **Enterprise WSDL** – Most enterprise users use this API to develop client applications for their organization. It is a strongly typed representation of the organization's data. It contains the information about data types, schema, and fields of the development environment. This allows for a tight integration between the Salesforce.com Web service and the WSDL. The Enterprise WSDL is object dependent, i.e. if custom objects or custom fields are created in the organization's Salesforce configuration, it changes. The Enterprise WSDL also contains the versions of the installed packages in the organization. This adds an extra step in WSDL generation where the user has to select the versions of managed packages.

  When generating the enterprise WSDL, keep in mind the following scenarios:

  - Addition of custom objects.
  - Modification of custom objects.
  - Addition of custom fields.
  - Modification of custom fields.
  - Changes/updates to installed packages.

  For all the above-mentioned scenarios, the WSDL file needs to be regenerated to permit access to these changes. Also, for any change in the installed packages, the WSDL needs to be regenerated with the specific package version. The new WSDL will have the objects and fields of only the packages selected during generation.

1.2.1.3 Supported API

SOAP API

To meet the needs of the complex business processes that an organization may need to orchestrate, Salesforce.com provides a number of ways for advanced administrators and developers to implement different functionalities. SOAP API is one such way.

SOAP API can be used to fetch and manipulate different records of standard as well as custom objects in Salesforce.com through operations like create, update, delete, query, etc. It supports all languages that support Web Services.

The organization must use Enterprise Edition, Unlimited Edition, or Developer Edition to be able to use SOAP API. An existing Salesforce.com customer can upgrade to either Enterprise or Unlimited Edition by contacting their account representative.

1.2.1.4 Support for Outbound

The Oracle SOA Suite Oracle Cloud Adapter for Salesforce.com does not support Salesforce making outbound synchronous calls to SOA Suite, that scenario must be implemented using the standard Web Service Binding in SOA Suite. It will be supported in the adapter in a future release. Automatic triggering of SOA Composite based services upon occurrence of SFDC events can be done by registering SOA Composite services end points manually within SFDC.

1.2.2 Oracle SOA Versions

Oracle Cloud Adapter for Salesforce.com is available in the following Oracle SOA Suite versions:

- 11g (11.1.1.7.0), alternatively referred as Oracle SOA Suite 11g PS6. Kindly refer the document User Guide for Oracle Cloud Adapter for Salesforce.com for more details.
- 12c (12.1.3.0.0): This document is intended for 12c version of SOA Suite.

2
Getting Started

This section provides a quick-start guide to enable the use of Oracle Cloud Adapter for Salesforce.com. This chapter explains the basic steps the user must follow to ensure that the Oracle SOA Suite 12c is properly configured for the Oracle Cloud Adapter for Salesforce.com. You must follow these steps to successfully connect with Salesforce.com and then proceed with implementing your integration. This chapter contains the following topics:

- Section 2.1, "Oracle Cloud Adapter for Salesforce.com Connection Parameters"
- Section 2.2, "IP Address Registration and Restrictions"
- Section 2.3, "Salesforce.com Permissions"
- Section 2.4, "Importing Salesforce.com Certificate into Client/Server"
- Section 2.5, "Enabling Oracle Cloud Adapter for Salesforce.com in Design-Time and Run-Time"
- Section 2.6, "Import the WSDL File into Your Development Platform"

1.1 Oracle Cloud Adapter for Salesforce.com Connection Parameters

The Oracle Cloud Adapter for Salesforce.com requires the user's Salesforce.com login credentials to successfully integrate with Salesforce.com.

The required connection parameters are:

- Username (in the form of an e-mail address).
- Password + Security token.
- CSF (Credential Store Framework) key.

Let us first begin with the description of connection parameter for Salesforce.com.

1.1.1 User Name

Salesforce.com provides every user in an organization a unique username. This username is in the form of an e-mail address. It could either be the same as the registered e-mail address of the user or some other username but always in an e-mail format.

For Example, if you have the registered e-mail id as *john.doe@oracle.com*, your username for Salesforce.com can be *john.doe@oracle.com* or you can choose another username in the form of an e-mail address like *john.doe@salesforce.com*

1.1.2 Password

The design of Salesforce.com APIs is such that the security token generated at the time of password reset is required to be appended at the end of the password. For example, if the

security token automatically generated by Salesforce.com is SSSSSSSSSS and the password is “password”, then the user is required to enter “passwordSSSSSSSSSS” to log in.

The security token can be obtained by changing the password or resetting the security token through the Salesforce.com UI. The security token is sent by mail to the e-mail address of the user registered with Salesforce.com for that particular organization. This token is valid until the user resets the security token for his/her account or changes the password.

Note: In case the user generates a new security token, then all the existing composites using the old security token in their passwords will need to be updated.

1.1.3 CSF Key

The Adapter uses the Credential Store Framework to retrieve the username and password required for Salesforce.com authentication. This key identifies a user’s Salesforce.com login credentials during design-time and has to be configured as exactly the same in Enterprise Manager on WebLogic Server under the CSF Map name “SOA” (Case sensitive).

For steps on how to configure CSF key in WebLogic Server, refer to the section “A.2 CSF Key in Enterprise Manager”

1.1.4 Downloaded Enterprise WSDL

A copy of the Enterprise WSDL of your Salesforce.com organization is needed for successful connection of the Oracle Cloud Adapter for Salesforce.com to Salesforce.com. To view steps on how to generate the enterprise WSDL, refer to the section “A.1 Generating the Enterprise WSDL”.

1.2 IP Address Registration and Restrictions

When a user tries to access Salesforce.com via a new IP address or machine, Salesforce.com requires a verification code to be entered for the user. This code is mailed to the user’s registered e-mail address. To circumvent this login challenge, a list of IP address ranges can be set for all users from which they can always log in without facing such issues.

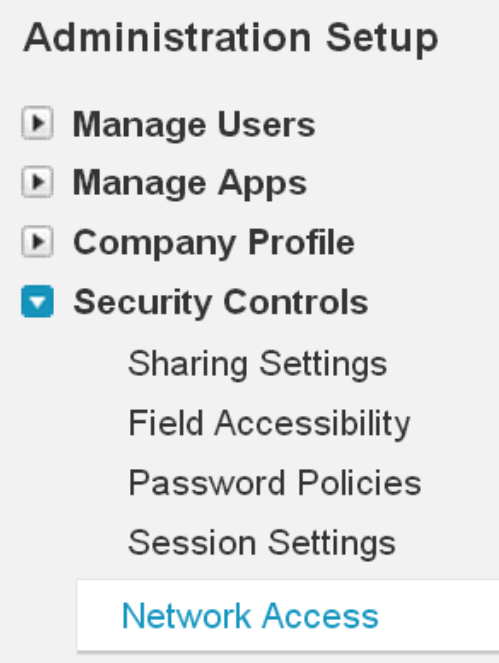
Restricting Login IP Ranges for the Organization

A list of IP addresses can be specified in Salesforce.com to protect the organization’s data and from these addresses, the users can always log in without receiving a login challenge:

To register the list of IP addresses, follow the steps given below:

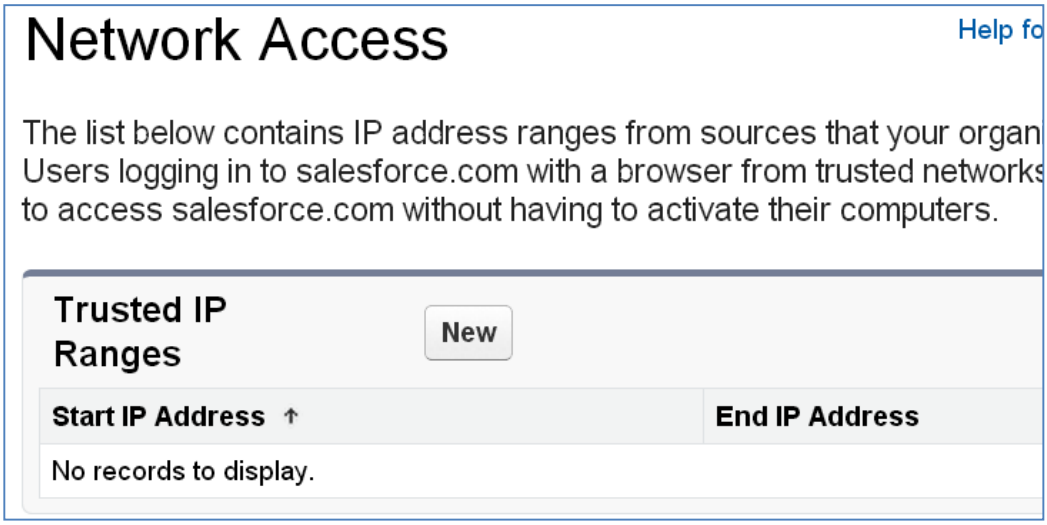
1. Log in to the salesforce.com using a valid user name and password.
2. Click on **User Menu** for user name, and then click on **Setup**.
3. Under **Administration Setup**, Expend **Security Controls** and click on **Network Access**, as shown in Figure 2-1.

Figure 2-1 Setup Window



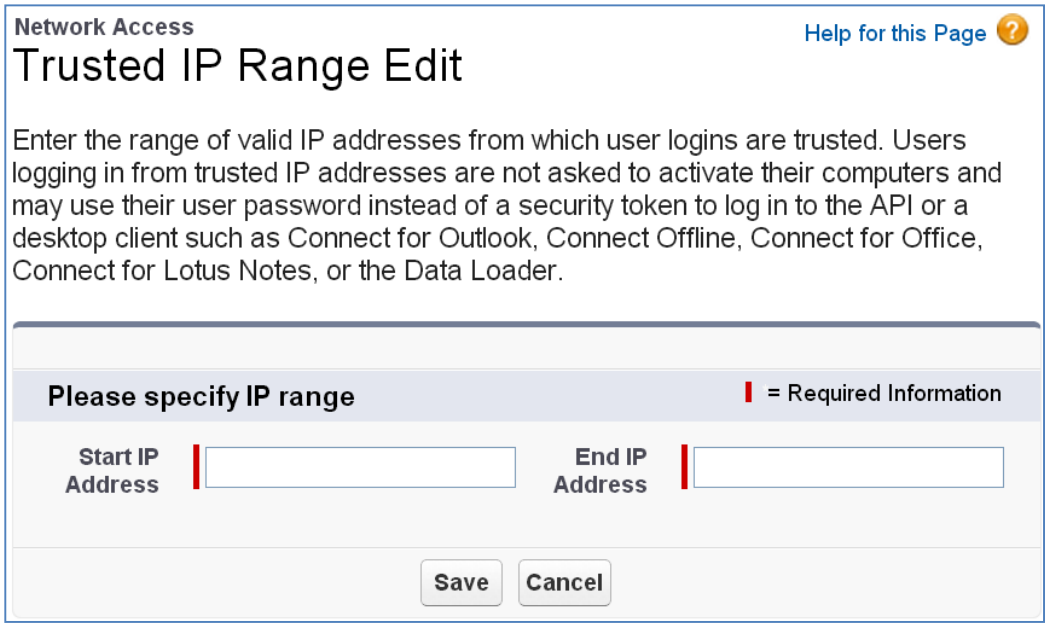
4. **Network Access** page is displayed, as shown in Figure 2-2.

Figure 2-2 Network Access Page



5. Click **New**, the **Trusted IP Range Edit** page is displayed, as shown in Figure 2-3.

Figure 2- 3 Trusted IP Range Edit Page



6. Enter a valid IP address in the **Start IP Address** field and a higher IP address in the **End IP Address** field.

The range of allowed IP addresses from which users can log in is defined by the start and end addresses. To allow logins from only one address, the administrator can enter that particular address. For example, to allow logins from only 125.12.3.0, enter 125.12.3.0 as both the start and end addresses. The start and end IP addresses in an IPv4 range must include no more than 33,554,432 addresses (2^{25}, a /7 CIDR block). Here are some examples of valid ranges:

- 0.0.0.0 to 1.255.255.255
- 132.0.0.0 to 132.255.255.255
- 132.0.0.0 to 133.255.255.255

However, ranges like 0.0.0.0 to 2.255.255.255 or 132.0.0.0 to 134.0.0.0 are too large.
The start and end IP addresses in an IPv6 range must include no more than 79,228,162,514,264,337,593,543,950,336 addresses (2^{96}, a /32 CIDR block). For example, the following range is valid: 2001:8000:: to 2001:8000:ffff:ffff:ffff:ffff:ffff:ffff. However, ranges like :: to ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff or 2001:8000:: to 2001:8001:: are too large.

7. Click **Save**.

When users log in to Salesforce.com via the API, Salesforce.com confirms that the login is authorized as follows:

1. Salesforce.com checks whether the user's profile has login hour restrictions. If login hour restrictions are specified for the user's profile, any login outside the specified hours is denied.
2. Salesforce.com then checks whether the user's profile has IP address restrictions. If IP address restrictions are defined for the user's profile, any login from an undesignated IP address is denied, and any login from a specified IP address is allowed.
3. If profile-based IP address restrictions are not set, Salesforce.com checks whether the user is logging in from an IP address they have not used to access Salesforce.com before:

- If the user's login is from an IP address in your organization's trusted IP address list, the login is allowed.
- If the user's login is not from a trusted IP address the login is blocked.

Whenever a login is blocked or returns an API login fault, Salesforce.com must verify the user's identity.

For access via the API or a client, the user must add their security token to the end of their password in order to log in.

1.3 Salesforce.com Permissions

The Cloud Adapter for Salesforce.com is based on the SOAP API of Salesforce.com. For the users to successfully make calls for different operations (create, update, deleted, query, etc.) they must have the requisite permissions for performing these operations on specific objects. This section provides an overview of the permissions the user must have to make the following calls:

- **Create:** You must have permission to create the particular object that is selected during design time. For example, if you are trying to create an account, you must have the permission to create new accounts.
- **Retrieve:** If you are trying to fetch the records of a certain object, you must have at least the read permission for that object. If you are trying to fetch a record of the Campaign object, you must have the permission to access Campaign object in Salesforce.com.
- **Update:** You must have permission to update the particular object that is selected during the design-time. For example, if you are trying to update an account, you must have the permission to change the account records.
- **Delete:** You must have the permission to delete the records of the object selected during the run-time. For example, if you have selected Lead object during design time, you must have the permission to delete Leads.
- **Upsert:** A user trying to use the upsert operation must have both create and update privileges for the said object to successfully make the upsert call.
- **ConvertLead:** A user trying to convert a lead to account and/or opportunity must have read and write permissions for the aforementioned objects.
- **Merge:** The merge operation works only on Account, Lead, and Contact objects. You must have full access to these objects.
- **Query:** This operation can be used to fetch records for several objects. To be able to successfully fetch these records you must have the permission to access these objects and their fields.
- **Search:** This operation is used to search records in the Salesforce.com organization and only the objects to which the user performing the operation has access to can be fetched. For instance, if you do not have access to Pricebooks, the records from Pricebook that match the search string will not be fetched.

1.4 Importing Salesforce.com Certificate into Client/Server

This section provides an overview of how to generate and import the Salesforce.com certificate for your organization into your client/server. The Oracle Cloud Adapter for Salesforce.com uses the certificate generated by Salesforce.com to establish a two-way SSL authentication. This enhances the security as the target of SOAP calls made by the Oracle Cloud Adapter for Salesforce.com receives the certificate and can use it to authenticate the request against its keystore.

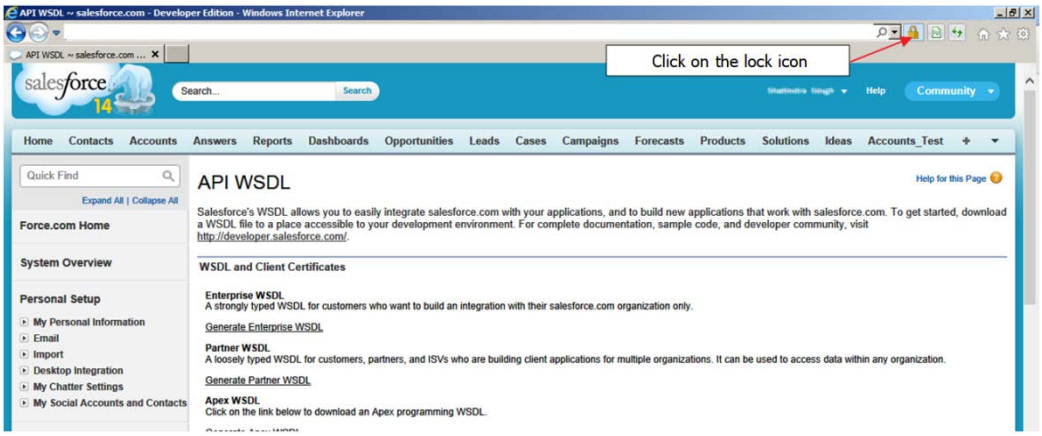
1.4.1 Generating Salesforce.com Certificate

The Salesforce.com Client certificate has to be downloaded from the Salesforce.com application user interface. This certificate has to be imported into the server for successful handshaking with Salesforce.com. The Oracle Cloud Adapter for Salesforce.com uses two-way SSL when interacting with Salesforce.com and hence requires this certificate. On the WebLogic server, these certificates are required to be imported.

To download the certificate, use the following procedure:

1. Log in to the salesforce.com using a valid user name and password.
2. Click on **User** menu for **user name**, and select **Setup**.
3. Under **App Setup**, Expand **Develop** and click on **API** to display the **WSDL download** page, as shown in Figure 2-4.

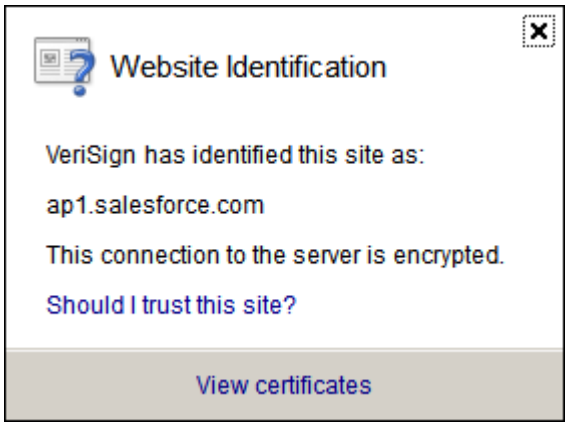
Figure 2- 4 API WSDL Download Page



4. Click on the **lock** icon as indicated in the above Figure 2-4. A small window will pop up. Click on the **view certificates**, as shown in Figure 2-5.

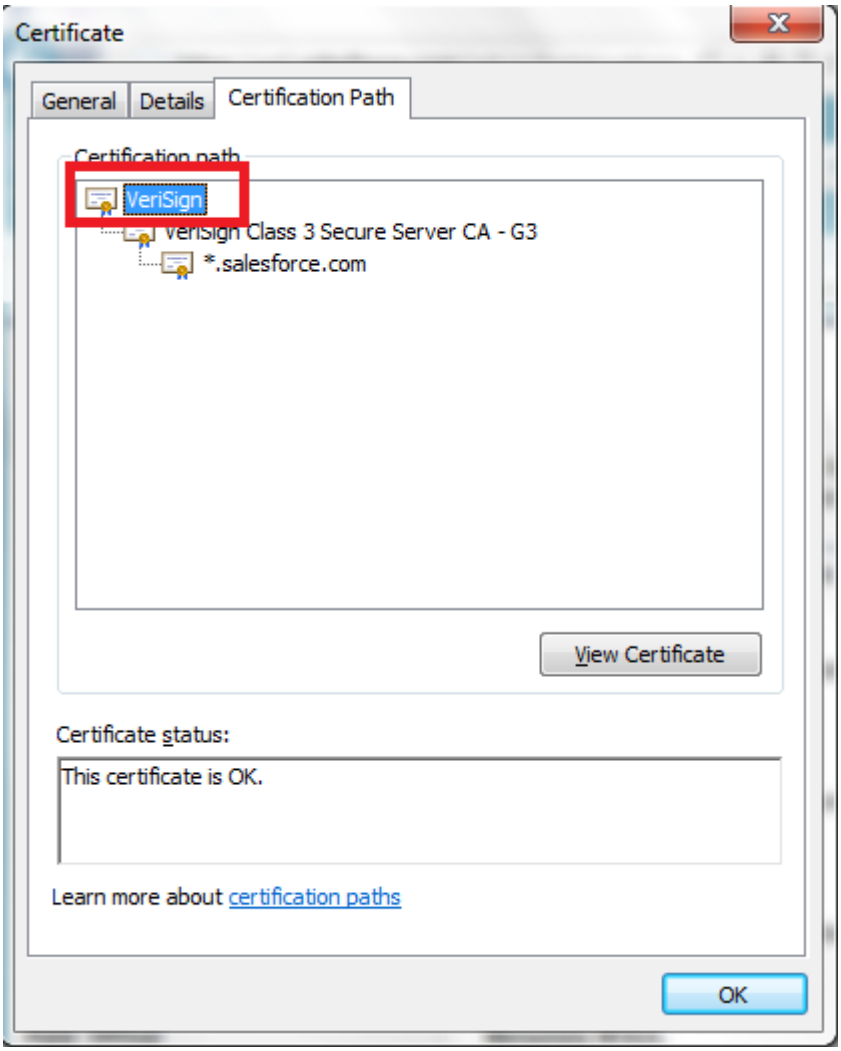
Note: Here, internet explorer (version 10) is used for demonstration. Experience may differ if you will be using other browser or other version of internet explorer. But idea here is to click on security lock to fetch the certificate information.

Figure 2- 5 View Certificates



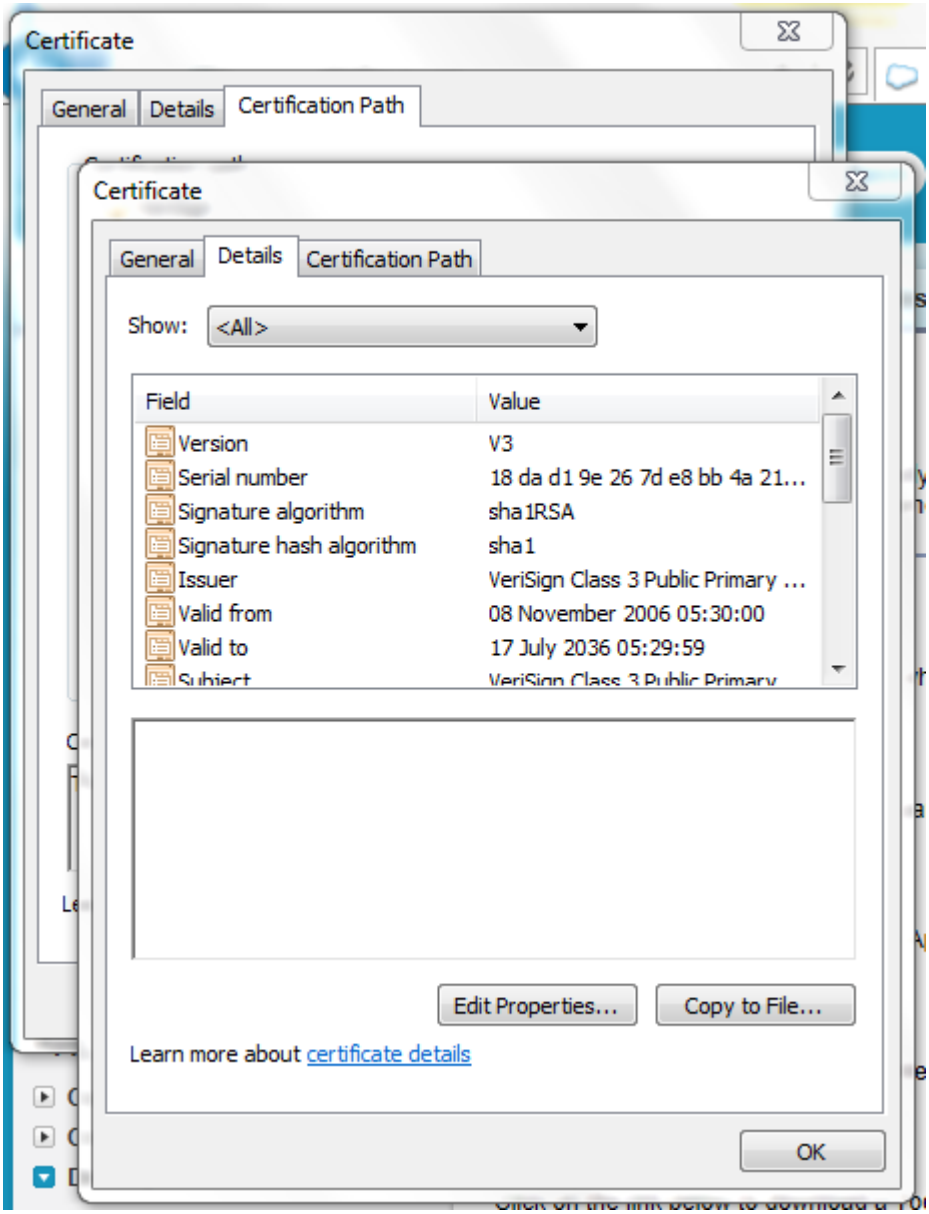
5. After clicking on the view certificates another window pops up providing the details of the certificate. Open the **Certification Path** tab and select VeriSign as shown in the screenshot below. The **View Certificate** button would be enabled, click the same to view certificate details.

Figure 2- 6 Selecting Certification Path



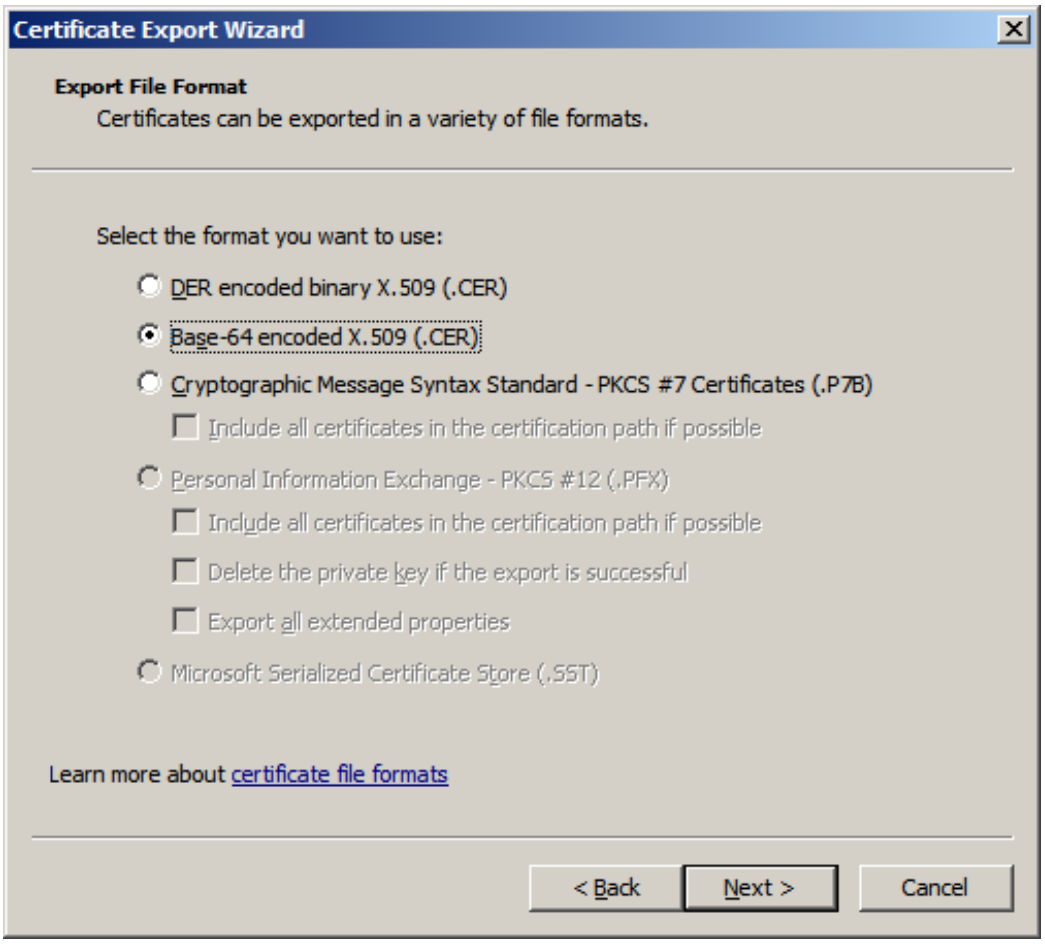
6. A new page opens above the existing page, open the **Details** tab and click on **Copy to File** button, as shown in Figure 2-7.

Figure2- 7 Details Tab



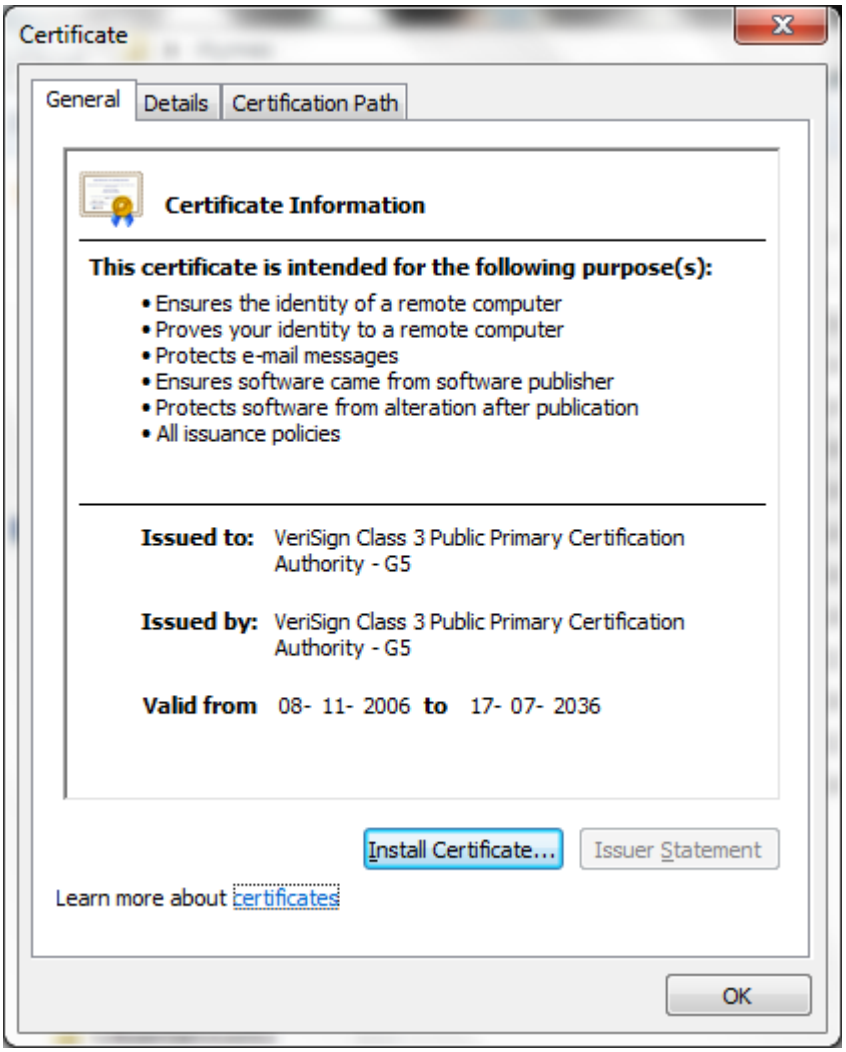
7. Now, Certificate Export wizard opens. Click on next button and select **Base-64 encoded X.509 (.CER)** option, as shown Figure 2-8.

Figure2- 8 Certificate Export Wizard



8. Click **Next** and browse to the location where you want to save the certificate. Provide an appropriate name to the certificate file and save it.
9. Double-click and open the Saved certificate, it should be similar to the screenshot provided below.

Figure 2- 9 Certificate Information

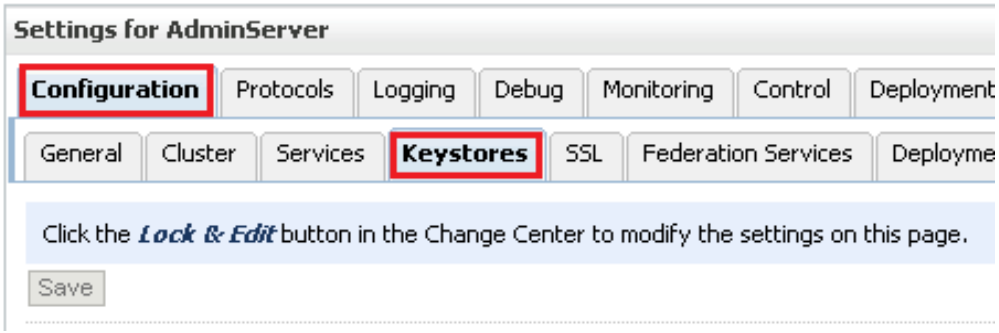


10. Import the downloaded certificate into your application server, and configure your application server to request the client certificate. The application server then checks that the certificate used in the SSL/TLS handshake matches the one you downloaded.

1.4.2 Importing Salesforce.com Certificate using Keytool

1. Provide a suitable location on your system to save the certificate authority. You would now add this certificate to Keystore.
2. Make sure that KSS for demo is disabled.
3. To disable KSS, go to the WebLogic console (http://url:port/console/).
4. Click on Domain → Security → Advanced page and unselect “Use KSS for Demo”.
5. Click **Save**.
6. To know the WebLogic server’s trust Keystore location: On the home page, proceed to **Servers** under **Environment** subsection.
7. Select the Admin server out of the list of servers displayed.
8. Proceed to the **Keystores** tab, as shown in Figure 2-10.

Figure 2- 10 Configuration Keystores Tab



9. Under **Keystores** tab, select the **Demo Trust Keystore** path. This is the path where our keystore resides.
10. It would be of the form {Middleware_Home}/wlserver/server/lib. Next, you need to import this certificate from the authority into WebLogic server's trust store.
11. If you are using a window based system, use a command prompt to navigate to the path mentioned in the above step.
12. At this path, run the following keytool command:

```
keytool -import -trustcacerts -alias SalesForceCA –file
<Filename with location> -keystore DemoTrust.jks -
storepass DemoTrustKeyStorePassPhrase
```

For e.g., assuming downloaded certificate is kept at the location as mentioned in the step 8 and name of the certificate is ***Salesforce.cer*** then keytool command would look like:

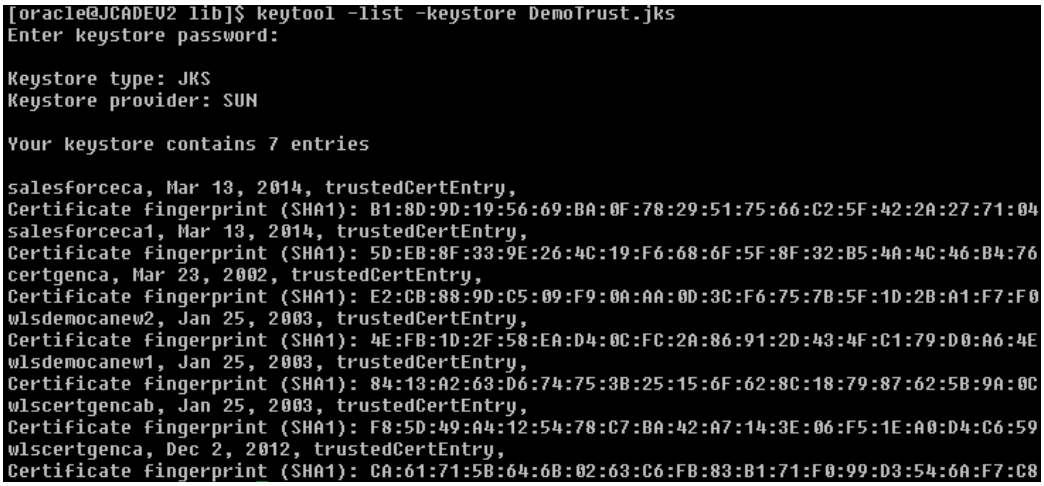
```
keytool -import -trustcacerts -alias SalesForceCA –file
Salesforce.cer -keystore DemoTrust.jks -storepass
DemoTrustKeyStorePassPhrase
```

13. A message **Certificate was added to keystore** is displayed, which confirms the successful import of the certificate. If you get a 'Certificate already exists in the Keystore message', enter 'Y'(yes) and proceed to import the certificate.
14. You can verify the same by enlisting all the certificates using the following command:

```
keytool –list –keystore DemoTrust.jks -storepass
DemoTrustKeyStorePassPhrase
```

15. The newly imported certificate appears as part of existing certificates in the keystore, as shown Figure 2-11.

Figure2- 11 Run Command Screen



16. Restart the server to bring the modifications made in the previous steps into effect.

16.1 Enabling Oracle Cloud Adapter for Salesforce.com in Design-Time and Run-Time

This section provides an overview that enables you to access the Oracle Cloud Adapter for Salesforce.com in both Design-Time and Run-Time.

The Oracle Cloud Adapter for Salesforce.com comes pre-installed with Oracle SOA Suite 12c unlike 11g PS6 where it has to be enabled using an OPatch for JDeveloper and WebLogic Server.

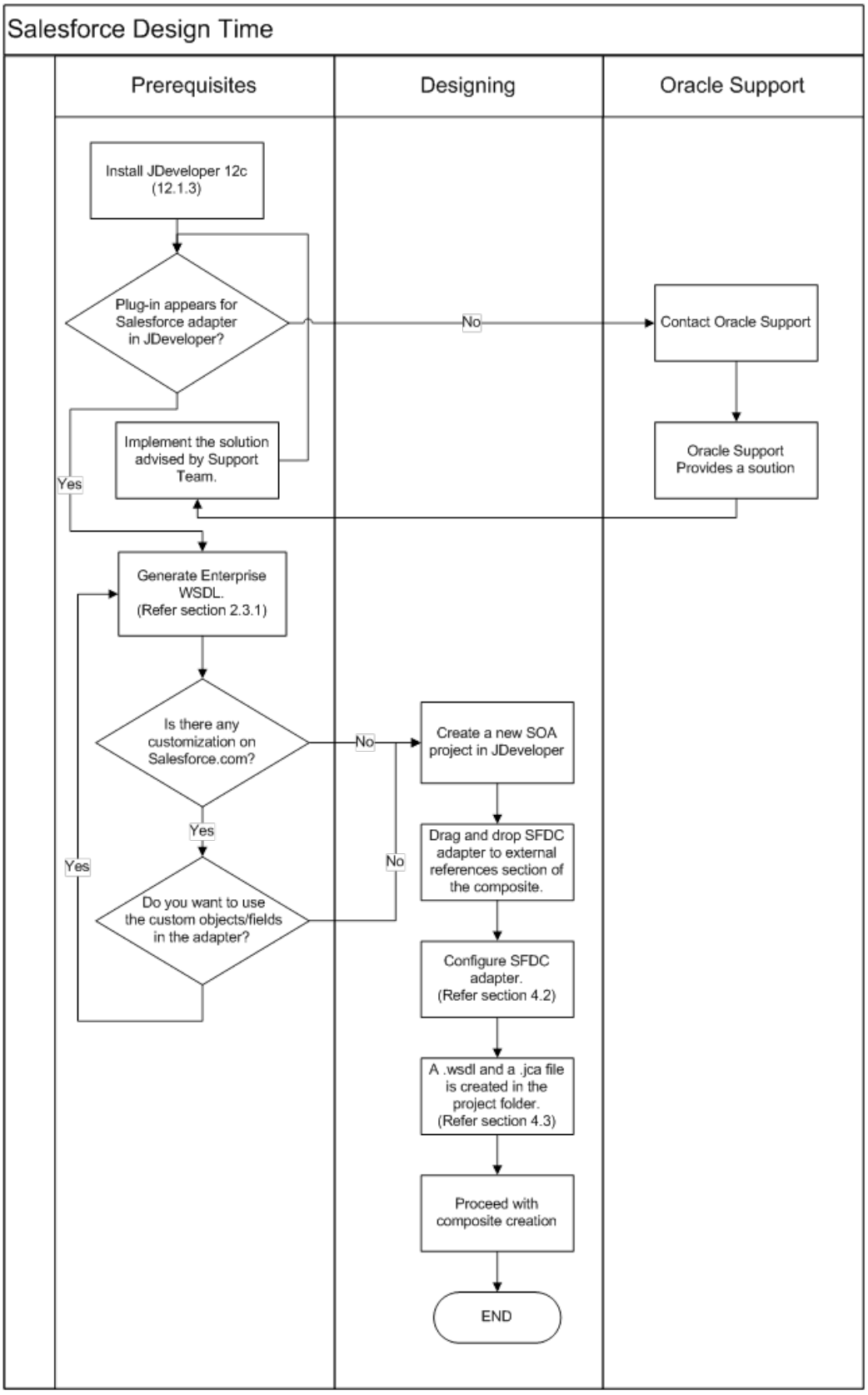
16.1.1 What are Design-Time and Run-Time?

The Cloud Adapter for Salesforce.com has two components:

1. **Design-Time** – This is the Adapter Configuration Wizard that allows an easy generation of artifacts that are needed to invoke and consume Web services. It consists of six simple configuration windows that create the artifacts needed to communicate with Salesforce.com.

Below flowchart diagram (Figure 2-12) explains the complete design-time lifecycle.

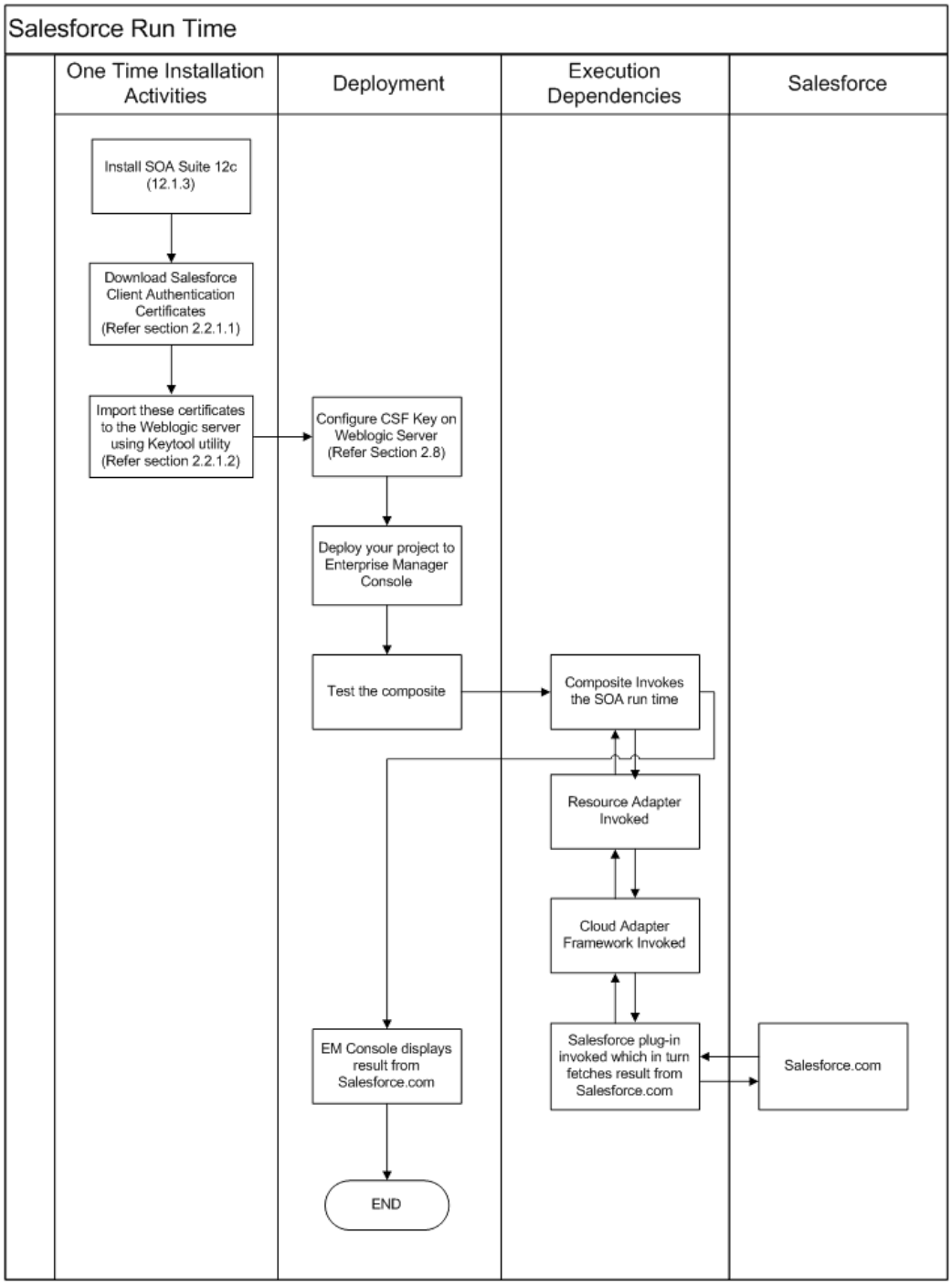
Figure2- 12 Design-Time Lifecycle



2. **Run-Time** – The run-time part of the adapter is utilized for delivering the information generated during the design-time to the service end point. This is where the actual invocation of Salesforce.com Web Service happens.

Below flowchart diagram (Figure 2-13) explains the complete run-time lifecycle.

Figure2- 13 Run-Time Lifecycle



16.2 Import the WSDL File into Your Development Platform

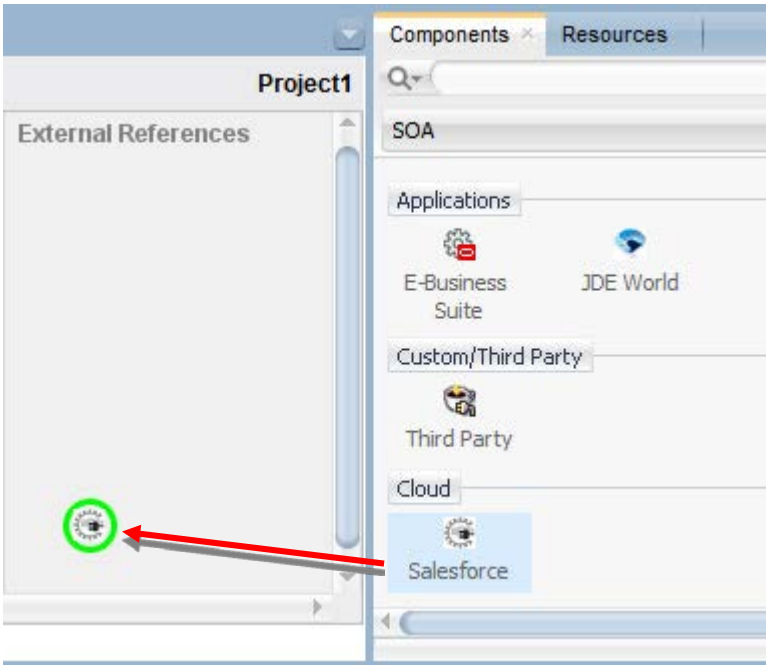
Once you have the WSDL file, it has to be imported into the development platform so that the development environment can generate the necessary objects for use in building client Web service applications in that environment. This section provides instructions for importing the WSDL into Oracle SOA Suite JDeveloper:

1. Open JDeveloper.
2. Drag and drop the **Salesforce Adapter** component from the Service Adapters pane to the **External References** pane, as shown in Figure 2–14.

OR

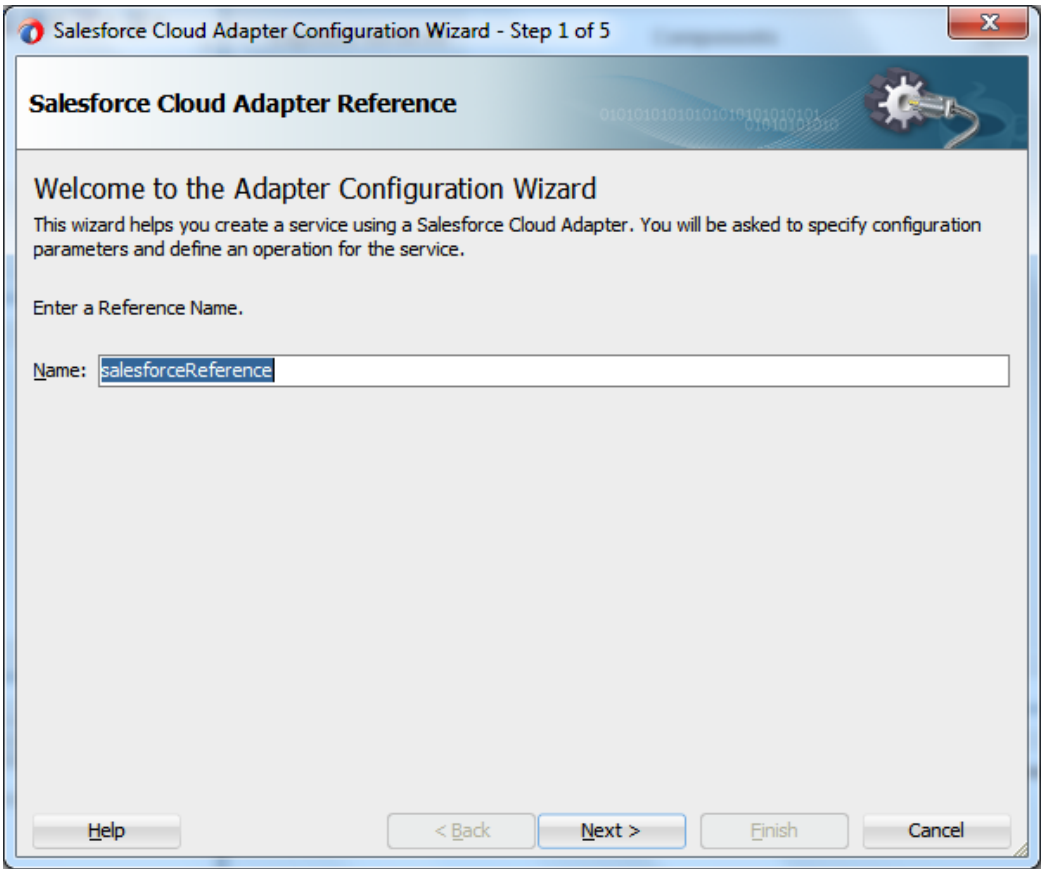
Right-click on the **External References** pane in JDeveloper and click on the **Salesforce Adapter**.

Figure 2- 14 Salesforce Adapter Component



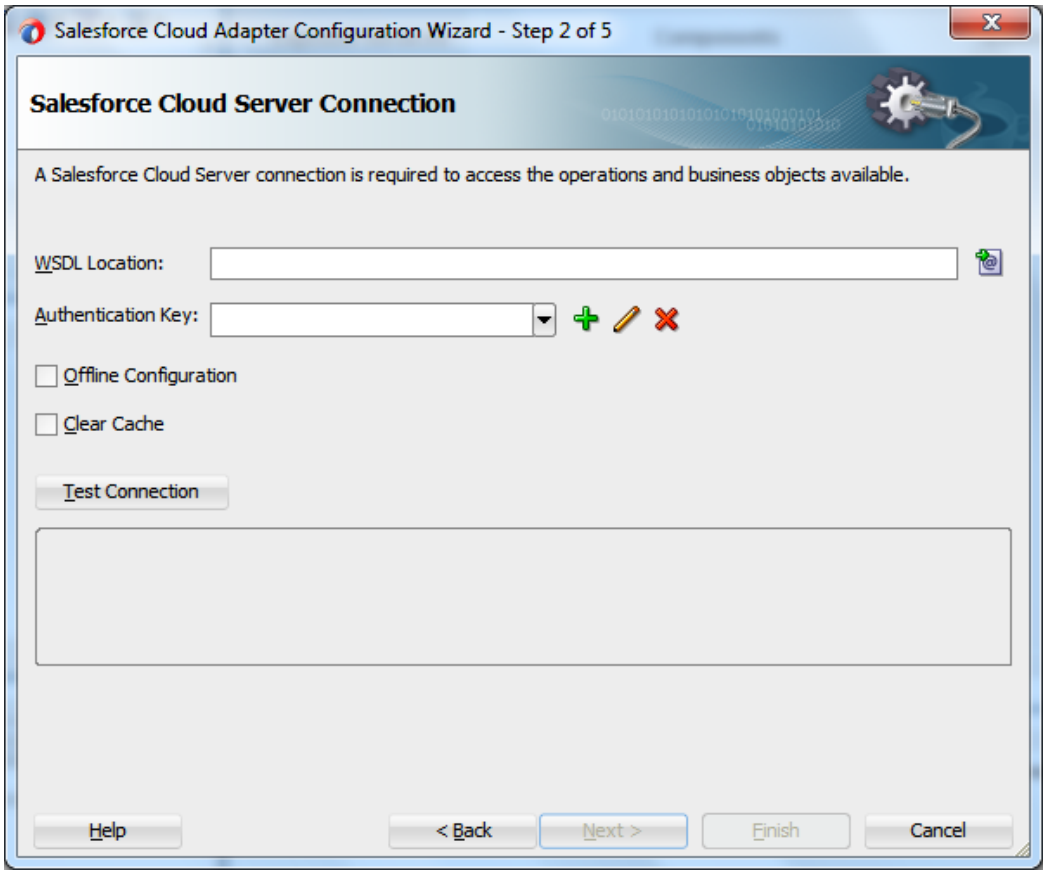
3. The **Salesforce Cloud Adapter Configuration Wizard- Welcome** page dialog is displayed, as shown in Figure 2-15.

Figure 2- 15 Salesforce Cloud Adapter Configuration Wizard



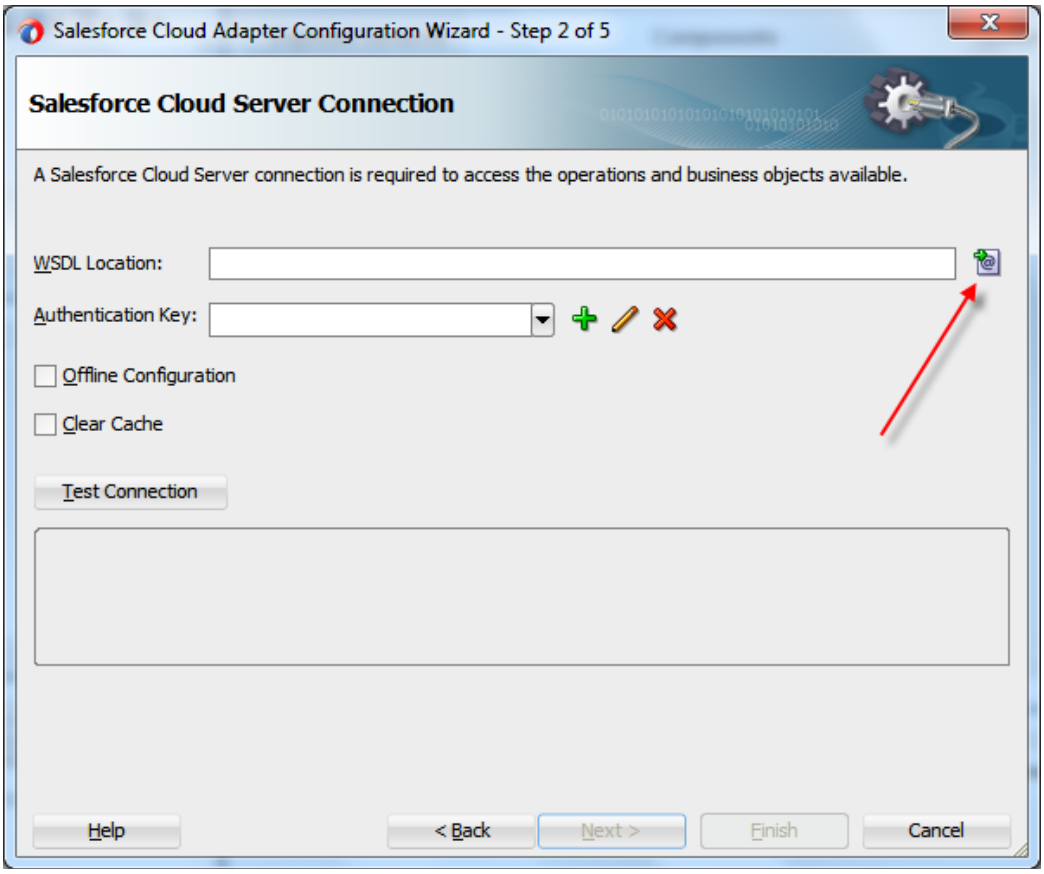
4. The **Name** box is already populated. It picks up these values from the cache. You can re-enter **Reference Name** if you want to use a different Reference Name.
5. Click **Next** to continue.
6. The **Salesforce Cloud Server Connection** page is displayed, as shown in Figure 2-16.

Figure 2- 16 Salesforce Cloud Server Connection Page



7. On the **Salesforce Cloud Server Connection** page, click the **Find existing WSDLs** icon, which is located to the right of the **WSDL Location** field, as shown in Figure 2–17.

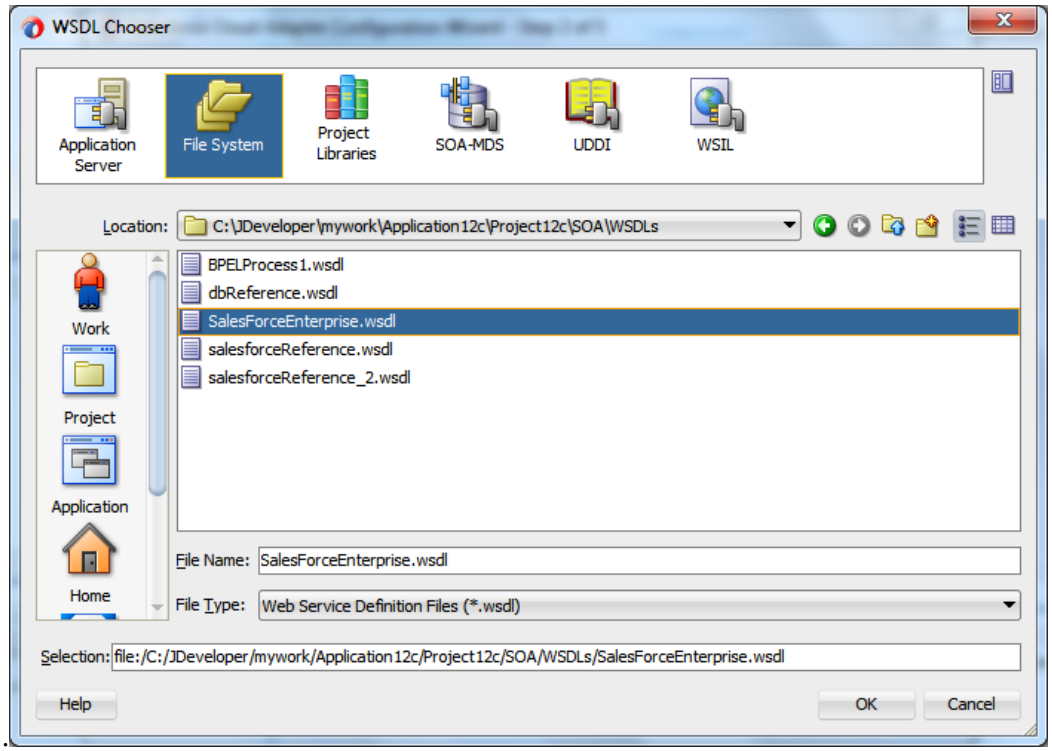
Figure 2- 17 Salesforce Cloud Server Connection Page



8. The **WSDL Chooser** dialog is displayed, browse and select the downloaded Enterprise WSDL either from the file system or resource palette (MDS) and click **OK**. Figure 2–18 shows the file system option.

For a walk-through of MDS, refer to the section "How to use MDS for importing WSDL into JDeveloper".

Figure 2- 18 WSDL Chooser



3

Oracle Cloud Adapter for Salesforce.com - Supported Features

The chapter offers a comprehensive guide to various Salesforce.com features supported by Oracle Cloud Adapter for Salesforce.com.

It contains the following topics:

- Section 3.1, "SOAP API"
- Section 3.2, "Supported SOAP API Operations"
- Section 3.3, "Salesforce.com SOAP Headers"
- Section 3.4, "Session Management"
- Section 3.5, "Handling Polymorphic Behavior of Salesforce.com Schema"
- Section 3.6, "Clear Cache"
- Section 3.7, "Offline Mode"
- Section 3.8, "Behaviors Expected When Using Offline Mode"
- Section 3.9, "Test Functionality"
- Section 3.10, "Suppress Response"
- Section 3.11, "Fault Handling"
- Section 3.12, "Salesforce.com Limit and Restriction Handling"

3.1 SOAP API

Nature of SOAP API calls are mentioned below:

- **Request and Response:** The adapter submits a request to Salesforce.com, and Salesforce.com processes the request and returns a response which the adapter handles.
- **Synchronous:** Every call to Salesforce.com SOAP API is synchronous, i.e. after invoking the adapter, calling process waits until it receives a response from the service. It does not support asynchronous calls.
- **Automatic Commit v/s Rollback:** Every operation that writes to a Salesforce.com object like create, upsert etc. is by default committed. Moreover, operations that write multiple objects in a single operation call treat every record as a separate transaction.

3.2 Supported SOAP API Operations

Following category of operations are supported by the Oracle Cloud Adapter for Salesforce.com, as shown in Table 3-1.

Table 3- 1 Supported Operations

Supported Category	Supported Operations
CORE	convertLead, getDeleted, getUpdated, merge, undelete, upsert
CRUD	create , retrieve, update, delete
MISC	getUserInfo, process
SOSL / SOQL	query,queryAll, search, queryMore

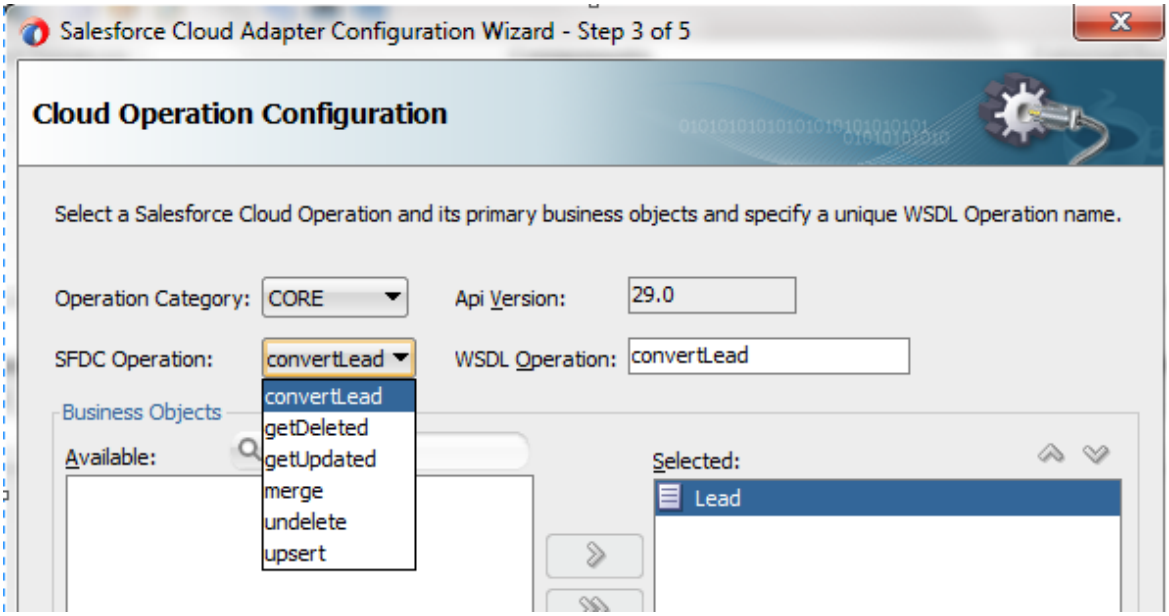
3.2.1 CORE Operations

This category contains all the core operations that are supported by Salesforce.com. The operation calls, falling under this category are:

- convertLead
- getDeleted
- getUpdated
- merge
- undelete
- upsert

Figure 3-1 shows the list of operation calls, falling under CORE category.

Figure 3- 1 SFDC Operation Calls



3.2.1.1 convertLead

- This operation converts a Salesforce.com Lead into an Account, Contact, and optionally into an Opportunity.

- The permissions required for convertLead operation are ‘Convert Leads’ and ‘Edit’ on Lead, in addition to ‘Create’ and ‘Edit’ on Account, Contact and Opportunity objects.
- In a simple and straightforward manner, a qualified lead on Salesforce.com is converted into a new or updated account, contact, and opportunity.
- An organization can have its own set of guidelines for determining when a lead is qualified, or it can follow the Salesforce.com default.

Supported Headers

- AllowFieldTruncationHeader
- DebuggingHeader
- EmailHeader
- LimitInfoHeader
- PackageVersionHeader

Figure 3-2 shows a sample request and response message for convertLead operation on Salesforce.com.

Figure 3- 2 Request and Response Message for convertLead Operation

3.2.1.2 getDeleted

- This call fetches the list of individual records that have been deleted in a particular time span.
- The specified endDate must sequentially follow the specified startDate.
- Only those records are returned on which user has access permissions.
- Only data modified in the last 30 days can be fetched, this is a Salesforce.com limitation.

Supported Headers

- LimitInfoHeader

Figure 3-3 shows a sample request and response message for getDeleted operation on Salesforce.com.

Figure 3- 3 Request and Response Message for getDeleted Operation

```
<?xml version="1.0" encoding="UTF-8"?><messages>
<Invoke1_getDeleted_InputVariable>
      <part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="parameters">
         <getDeleted xmlns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Automation12c/getDeleted/getDeletedAc
            <tns:startDate>2014-03-18T18:04:16+05:30</tns:startDate>
            <tns:endDate>2014-04-08T18:04:16.184+05:30</tns:endDate>
         </getDeleted>
      </part>
   </Invoke1_getDeleted_InputVariable>
<Invoke1_getDeleted_OutputVariable>
      <part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="parameters">
         <tns:getDeletedResponse xmlns:tns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Automation12c/getDele
            <tns:GetDeletedResult>
               <deletedRecords xmlns="urn:enterprise.soap.sforce.com">
                  <deletedDate>2014-04-08T10:00:45.000Z</deletedDate>
                  <id>0019000000shNkAAAU</id>
               </deletedRecords>
               <deletedRecords xmlns="urn:enterprise.soap.sforce.com">
                  <deletedDate>2014-04-08T10:00:45.000Z</deletedDate>
                  <id>0019000000shXjhAAE</id>
               </deletedRecords>
               <deletedRecords xmlns="urn:enterprise.soap.sforce.com">
                  <deletedDate>2014-04-08T10:00:45.000Z</deletedDate>
                  <id>0019000000shXjiAAE</id>
               </deletedRecords>
               <deletedRecords xmlns="urn:enterprise.soap.sforce.com">
                  <deletedDate>2014-04-08T10:00:45.000Z</deletedDate>
                  <id>0019000000shNYiAAM</id>
               </deletedRecords>
               <deletedRecords xmlns="urn:enterprise.soap.sforce.com">
                  <deletedDate>2014-04-08T10:00:45.000Z</deletedDate>
                  <id>0019000000shNsqAAE</id>
               </deletedRecords>
               <deletedRecords xmlns="urn:enterprise.soap.sforce.com">
                  <deletedDate>2014-04-08T10:00:45.000Z</deletedDate>
                  <id>0019000000shNxfAAE</id>
               </deletedRecords>
               <deletedRecords xmlns="urn:enterprise.soap.sforce.com">
                  <deletedDate>2014-04-08T10:00:45.000Z</deletedDate>
                  <id>0019000000t77xNAAQ</id>
               </deletedRecords>
               <deletedRecords xmlns="urn:enterprise.soap.sforce.com">
```

3.2.1.3 getUpdated

- This call fetches the list of individual records that have been updated (added or modified) in a particular time span.
- The specified endDate must sequentially follow the specified startDate.
- Only data modified in the last 30 days can be fetched, this is a Salesforce.com limitation.

Supported Headers

- LimitInfoHeader

Figure 3-4 shows a sample request and response message for getUpdated operation on Salesforce.com.

Figure 3- 4 Request and Response Message for getUpdated Operation

```
<?xml version="1.0" encoding="UTF-8"?><messages>
<Invoke1_getUpdated_InputVariable>
      <part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="parameters">
         <getUpdated xmlns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Automation12c/getUpdated/getUpdatedDemo" xmlns
            <tns:startDate>2014-03-18T17:57:14+05:30</tns:startDate>
            <tns:endDate>2014-04-08T17:57:14.104+05:30</tns:endDate>
         </getUpdated>
      </part>
   </Invoke1_getUpdated_InputVariable>
<Invoke1_getUpdated_OutputVariable>
      <part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="parameters">
         <tns:getUpdatedResponse xmlns:tns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Automation12c/getUpdated/getUp
            <tns:GetUpdatedResult>
               <ids xmlns="urn:enterprise.soap.sforce.com">0019000000s2XslAAE</ids>
               <ids xmlns="urn:enterprise.soap.sforce.com">0019000000s2XsnAAE</ids>
               <ids xmlns="urn:enterprise.soap.sforce.com">0019000000shK4nAAE</ids>
               <ids xmlns="urn:enterprise.soap.sforce.com">0019000000sh8jyAAA</ids>
               <ids xmlns="urn:enterprise.soap.sforce.com">0019000000shK52AAE</ids>
               <ids xmlns="urn:enterprise.soap.sforce.com">0019000000tBdH0AAK</ids>
               <ids xmlns="urn:enterprise.soap.sforce.com">0019000000tBdJQAA0</ids>
               <ids xmlns="urn:enterprise.soap.sforce.com">0019000000tBdJzAAK</ids>
               <latestDateCovered xmlns="urn:enterprise.soap.sforce.com">2014-04-08T11:14:00.000Z</latestDateCovered>
            </tns:GetUpdatedResult>
         </tns:getUpdatedResponse>
      </part>
   </Invoke1_getUpdated_OutputVariable>
</messages>
```

3.2.1.4 merge

- Merge operation is used to merge child record of an object into the master record.
- A maximum of three records can be merged into one call.
- A single merge call consists only of one business object. Merge doesn't work for multiple business objects in one call.
- Lead, Contact and Account are the only supported object types.
- The **masterRecord** field specifies the master record into which the child records are merged.

Supported Headers

- AllowFieldTruncationHeader
- AssignmentRuleHeader
- DebuggingHeader
- EmailHeader
- LimitInfoHeader
- MruHeader
- PackageVersionHeader

Figure 3-5 shows a sample request and response message for merge operation on Salesforce.com.

Figure 3- 5 Request and Response Message for merge Operation

3.2.1.5 undelete

- Undelete is used to recover already deleted records from Recycle Bin.
- Objects with undeletable property set as true, will be available for undelete.
- In an attempt to undelete an entity which is removed from the recycle bin as well, you will encounter the 'UNDELETE_FAILED' error, stating that 'An object could not be undeleted because it does not exist or has not been deleted.'

Supported Headers

- AllOrNoneHeader
- AllowFieldTruncationHeader
- DebuggingHeader

- LimitInfoHeader
- PackageVersionHeader

Figure 3-6 shows a sample request and response message for undelete operation on Salesforce.com.

Figure 3- 6 Request and Response Message for undelete Operation

```
<?xml version="1.0" encoding="UTF-8"?><messages>
<Invoke_undelete_InputVariable>
      <part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="parameters">
         <undelete xmlns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Automation12c/undeleteDemo/und
            <tns:ids>0019000000tBeKH</tns:ids>
         </undelete>
      </part>
   </Invoke_undelete_InputVariable>
<Invoke_undelete_OutputVariable>
      <part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="parameters">
         <tns:undeleteResponse xmlns:tns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Automation12c/
            <tns:UndeleteResult>
               <id xmlns="urn:enterprise.soap.sforce.com">0019000000tBeKHAA0</id>
               <success xmlns="urn:enterprise.soap.sforce.com">true</success>
            </tns:UndeleteResult>
         </tns:undeleteResponse>
      </part>
   </Invoke_undelete_OutputVariable>
</messages>
```

3.2.1.6 upsert

- Upsert is a combination of create and update operations.
- If a record already exists, this operation updates it, otherwise it creates a new record with the corresponding details.
- To avoid redundant records, it is better to use upsert operation instead of create operation.
- Contrary to insert and update operation, where you can process multiple objects in a single call, upsert operation lets you process only a single business object per call.

Supported Headers

- AllOrNoneHeader
- AllowFieldTruncationHeader
- AssignmentRuleHeader
- DebuggingHeader
- EmailHeader
- LimitInfoHeader
- MruHeader
- PackageVersionHeader

Figure 3-7 shows a sample request and response message for upsert operation on Salesforce.com.

Figure 3- 7 Request and Response Message for upsert Operation

```
<?xml version="1.0" encoding="UTF-8"?><messages>
<Invoke1_upsert_InputVariable>
      <part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="parameters">
         <upsert xmlns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Automation12c/upsertDemo/upsertA
            <tns:externalIDFieldName>Account_Ext_Id__c</tns:externalIDFieldName>
            <tns:Account>
               <ens:Account_Ext_Id__c>123456789</ens:Account_Ext_Id__c>
               <ens:Name>[illegible]</ens:Name>
            </tns:Account>
         </upsert>
      </part>
   </Invoke1_upsert_InputVariable>
<Invoke1_upsert_OutputVariable>
      <part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="parameters">
         <tns:upsertResponse xmlns:tns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Automation12c/up
            <tns:UpsertResult>
               <created xmlns="urn:enterprise.soap.sforce.com">true</created>
               <id xmlns="urn:enterprise.soap.sforce.com">0019000000tBebmAAC</id>
               <success xmlns="urn:enterprise.soap.sforce.com">true</success>
            </tns:UpsertResult>
         </tns:upsertResponse>
      </part>
   </Invoke1_upsert_OutputVariable>
</messages>
```

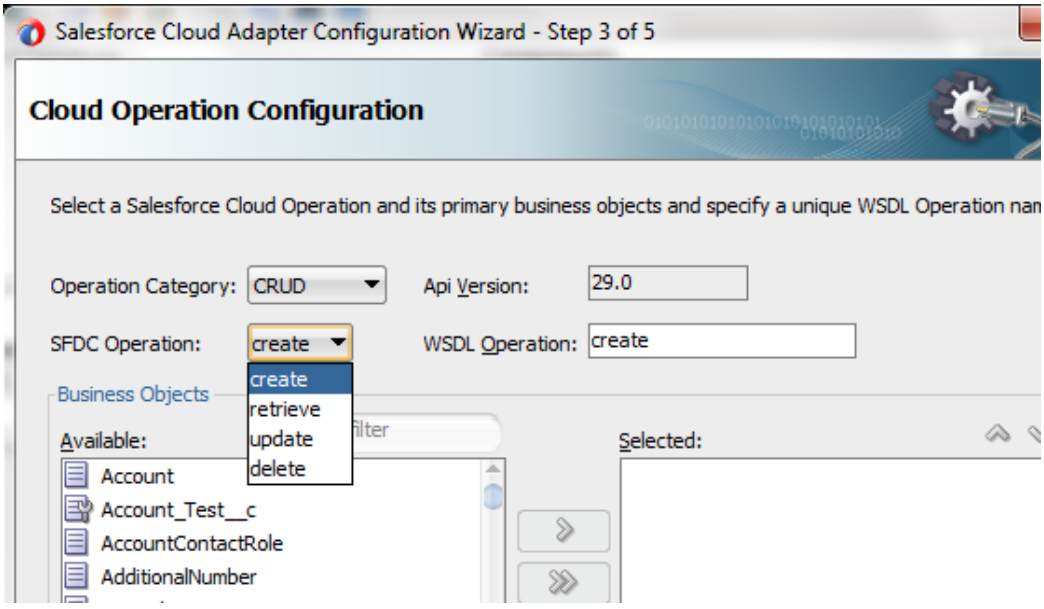
3.2.2 CRUD Operations

CRUD is an acronym for Create Retrieve Update and Delete. This category contains operations associated with manipulating the objects on Salesforce.com. The following operations fall under this category:

- create
- retrieve
- update
- delete

Figure 3-8 shows the operations falling under CRUD category.

Figure 3- 8 List of Operation under CRUD Category



3.2.2.1 create

- Create operation is used to add individual records to the organization's data.
- It can process records belonging to multiple business objects in a single call to Salesforce.com.

- It is similar to the Insert operation in SQL.

Supported Headers

- AllOrNoneHeader
- AllowFieldTruncationHeader
- AssignmentRuleHeader
- DebuggingHeader
- EmailHeader
- LimitInfoHeader
- MruHeader
- PackageVersionHeader

Figure 3-9 shows a sample request and response for create operation on Salesforce.com.

Figure 3- 9 Request and Response for CreateOperation

3.2.2.2 retrieve

- Fetches specific information for a sObject based on the ID of the object.
- The fields of the object, required to be fetched, are the inputs of the element ‘fieldList’ of the request message. As shown in Figure 3-10, “Origin” and “Status” are sent as fieldList in the request message.
- Retrieve call doesn’t fetch deleted records.

Supported Headers

- LimitInfoHeader
- MruHeader
- PackageVersionHeader
- QueryOptionsHeader

Figure 3-10 shows a sample request and response message for retrieve operation on Salesforce.com.

Figure 3- 10 Request and Response Message for retrieve Operation

3.2.2.3 update

- Update operation is used to update existing objects on Salesforce.com.
- This option can update multiple sObjects in one call.
- Only those objects are available under this operation whose updatable property is set to true.

Supported Headers

- AllOrNoneHeader
- AssignmentRuleHeader
- EmailHeader
- LimitInfoHeader
- MruHeader
- PackageVersionHeader

Figure 3-11 shows a sample request and response message for update operation on Salesforce.com.

Figure 3- 11 Request and Response Message for update Operation

3.2.2.4 delete

- Delete operation is used to delete one or more records on Salesforce.com.

- Operates on more than one sObject at one time.
- Similar to DELETE statement in SQL.
- Only those objects are available under this operation whose deletable property is set to true.

Supported Headers

- AllOrNoneHeader
- EmailHeader
- LimitInfoHeader
- PackageVersionHeader

Figure 3-12 shows a sample request and response message for delete operation on Salesforce.com.

Figure 3- 12 Request and Response Message for delete Operation

```
<?xml version="1.0" encoding="UTF-8"?><messages>
<Invoke1_delete_InputVariable>
      <part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="parameters">
         <delete xmlns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Automation12c/deleteDemo/deleteDemo" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
            <tns:ids>0019000000tBebm</tns:ids>
         </delete>
      </part>
   </Invoke1_delete_InputVariable>
<Invoke1_delete_OutputVariable>
      <part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="parameters">
         <tns:deleteResponse xmlns:tns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Automation12c/deleteDemo/deleteDemo">
            <tns:DeleteResult>
               <id xmlns="urn:enterprise.soap.sforce.com">0019000000tBebmAAC</id>
               <success xmlns="urn:enterprise.soap.sforce.com">true</success>
            </tns:DeleteResult>
         </tns:deleteResponse>
      </part>
   </Invoke1_delete_OutputVariable>
</messages>
```

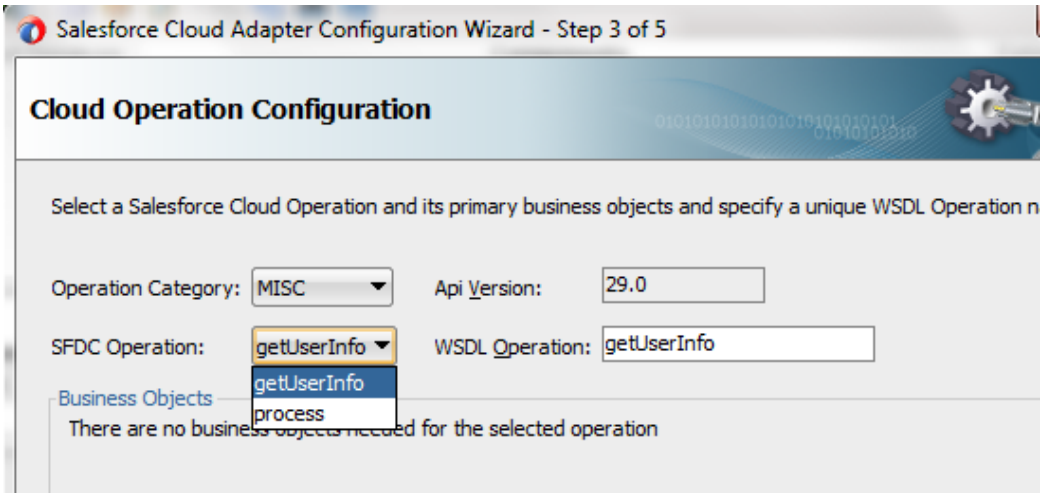
3.2.3 MISC Operations

This category contains all the miscellaneous operations supported by Salesforce.com. Following are the operations falling under this category:

- getUserInfo
- process

Figure 3-13 shows all the operations falling under this category.

Figure 3- 13 Options Available for MISC Category



3.2.3.1 getUserInfo

- The call fetches information of the user associated with the current session.
- Information returned from this call includes currency, profile, email and other user information.
- To obtain additional information about the user, you can also use the retrieve operation on User object providing the required fields in the fieldList element.
- Standard information pertinent to the current user is returned.

Supported Headers

- LimitInfoHeader

Figure 3-14 shows a sample request and response message for getUserInfo operation on Salesforce.com.

Figure 3- 14 Request and Response Message for getUserInfo Operation

```
<?xml version="1.0" encoding="UTF-8"?><messages>
<Invoke1_getUserInfo_InputVariable>
      <part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="parameters">
         <getUserInfo xmlns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Application7/getuserinfo/salesforceReference"/>
      </part>
   </Invoke1_getUserInfo_InputVariable>
<Invoke1_getUserInfo_OutputVariable>
      <part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="parameters">
         <ns1:getUserInfoResponse xmlns:ns1="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Application7/getuserinfo/salesforceRefer
            <ns1:GetUserInfoResult>
               <accessibilityMode xmlns="urn:enterprise.soap.sforce.com">false</accessibilityMode>
               <currencySymbol xmlns="urn:enterprise.soap.sforce.com" xsi:nil="true"/>
               <orgAttachmentFileSizeLimit xmlns="urn:enterprise.soap.sforce.com">5242880</orgAttachmentFileSizeLimit>
               <orgDefaultCurrencyIsoCode xmlns="urn:enterprise.soap.sforce.com" xsi:nil="true"/>
               <orgDisallowHtmlAttachments xmlns="urn:enterprise.soap.sforce.com">false</orgDisallowHtmlAttachments>
               <orgHasPersonAccounts xmlns="urn:enterprise.soap.sforce.com">false</orgHasPersonAccounts>
               <organizationId xmlns="urn:enterprise.soap.sforce.com">00D900000hIQnEAM</organizationId>
               <organizationMultiCurrency xmlns="urn:enterprise.soap.sforce.com">true</organizationMultiCurrency>
               <organizationName xmlns="urn:enterprise.soap.sforce.com">[redacted]</organizationName>
               <profileId xmlns="urn:enterprise.soap.sforce.com">00e90000001274tAAA</profileId>
               <roleId xmlns="urn:enterprise.soap.sforce.com" xsi:nil="true"/>
               <sessionSecondsValid xmlns="urn:enterprise.soap.sforce.com">43200</sessionSecondsValid>
               <userDefaultCurrencyIsoCode xmlns="urn:enterprise.soap.sforce.com">JPY</userDefaultCurrencyIsoCode>
               <userEmail xmlns="urn:enterprise.soap.sforce.com">[redacted]</userEmail>
               <userFullName xmlns="urn:enterprise.soap.sforce.com">[redacted]</userFullName>
               <userId xmlns="urn:enterprise.soap.sforce.com">005900000015SkBAAU</userId>
               <userLanguage xmlns="urn:enterprise.soap.sforce.com">en_US</userLanguage>
               <userLocale xmlns="urn:enterprise.soap.sforce.com">en_IN</userLocale>
               <userName xmlns="urn:enterprise.soap.sforce.com">[redacted]</userName>
               <userTimeZone xmlns="urn:enterprise.soap.sforce.com">[redacted]</userTimeZone>
               <userType xmlns="urn:enterprise.soap.sforce.com">Standard</userType>
               <userUiSkin xmlns="urn:enterprise.soap.sforce.com">Theme3</userUiSkin>
            </ns1:GetUserInfoResult>
         </ns1:getUserInfoResponse>
      </part>
   </Invoke1_getUserInfo_OutputVariable>
</messages>
```

3.2.3.2 process

- It submits an array of approval process instances for approval, or processes an array of approval process instances to be approved, rejected, or removed.
- Process operation either submits an approval process or processes an already submitted process.

Supported Headers

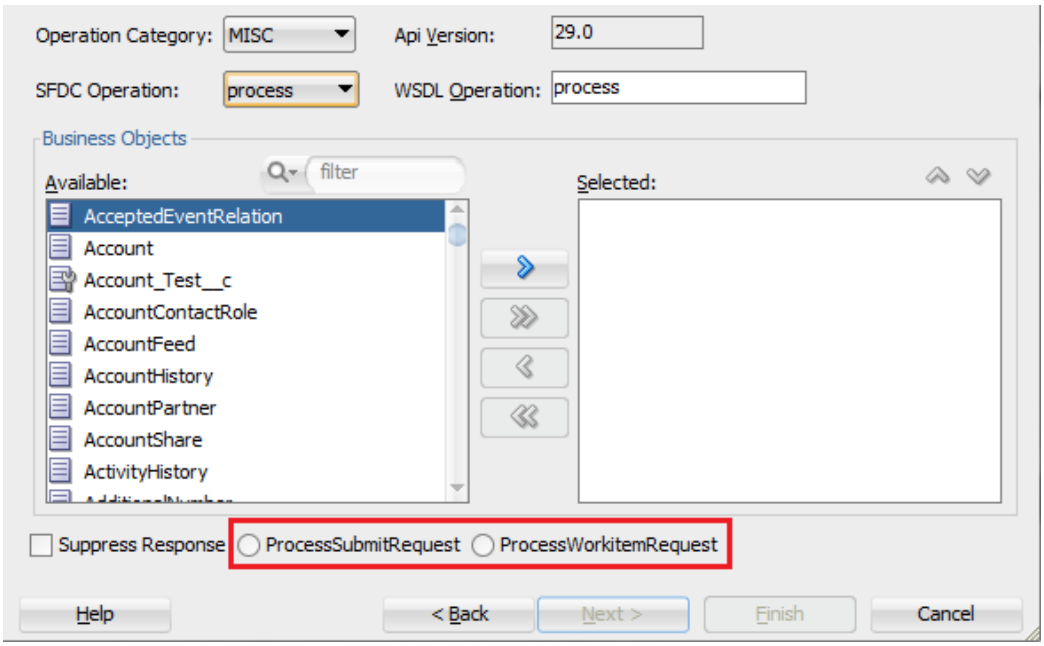
- LimitInfoHeader
- PackageVersionHeader

Process operation is divided into two parts:

- Process Submit Request
- ProcessWorkItemRequest

Two radio buttons appear once you select **process** as the operation, as shown in Figure 3-15.

Figure 3- 15 Radio Button under process Operation



ProcessSubmitRequest: This option is used to submit an object for approval. The response of this operation generates a **WorkItemId** and the actor (user) to whom this object is submitted for approval. The WorkItemId is a unique identifier of the object submitted for approval.

Figure 3-16 shows a sample request and response message for ProcessSubmitRequest operation on Salesforce.com.

Figure 3- 16 Request and Response Message for ProcessSubmitRequest Operation

ProcessWorkItemRequest: This option is used to process an object already submitted for approval. The actor can approve or reject a process. To perform the approval actions on the object already submitted for approval, the WorkItemId generated in ProcessSubmitRequest response is needed.

Figure 3-17 shows a sample request and response message for ProcessWorkItemRequest operation on Salesforce.com.

Figure 3- 17 Request and Response Message for ProcessWorkItemRequest Operation

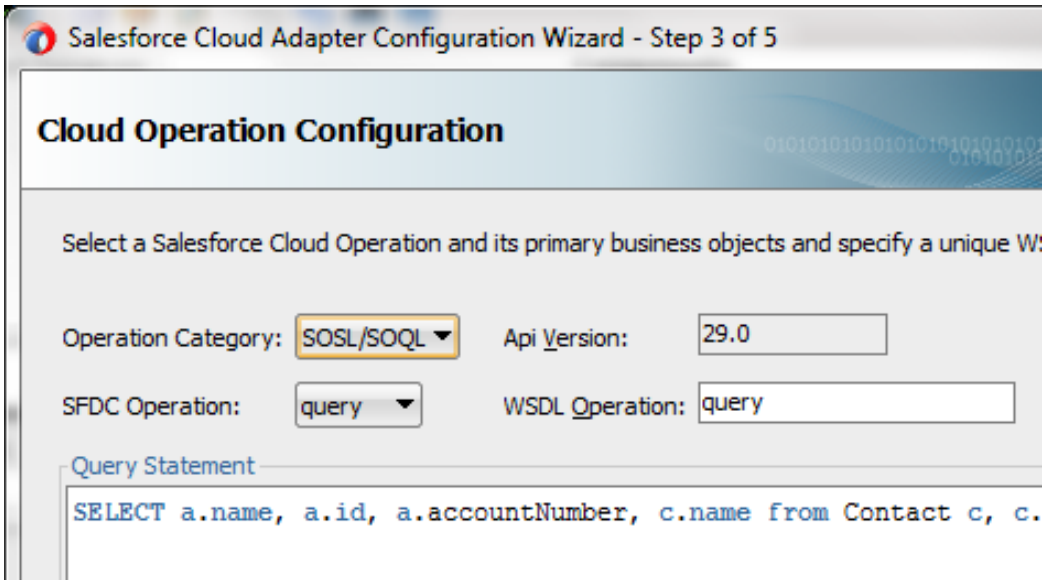
3.2.4 SOSL and SOQL Operations

SOSL and SOQL allow you to search your organization's Salesforce.com data for specific information. Following are the operations falling under this category:

- query
- queryAll
- search
- queryMore(supported internally)

Figure 3-18 shows the list of operations falling under SOSL/SOQL category.

Figure 3- 18 Option Available under SOSL/SOQL Category



3.2.5 query

- The query operation executes a query against a particular criteria and returns data that matches that particular criteria.
- Uses the SOQL (Structured Object Query Language).
- The query returns only those records that have not been deleted from your Salesforce.com account.
- Oracle Cloud Adapter for Salesforce.com provides functionality for using bind parameters in query operation. Using this functionality, you can dynamically provide an input to your query. For more information, refer to the section “Support for Bind Parameters”

Supported Headers

- LimitInfoHeader
- MruHeader
- PackageVersionHeader
- QueryOptionsHeader

Figure 3-19 shows a sample request and response message for query operation on Salesforce.com.

Figure 3- 19 Request and Response Message for query Operation

```
<?xml version="1.0" encoding="UTF-8"?><messages>
<Invoke1_query_InputVariable>
      <part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="parameters">
         <query xmlns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Application7/QueryProject1/salesforceReference_query"/>
      </part>
   </Invoke1_query_InputVariable>
<Invoke1_query_OutputVariable>
      <part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="parameters">
         <ns0:queryResponse xmlns:ns0="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Application7/QueryProject1/salesforceReference_query">
            <ns0:QueryResults>
               <done xmlns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Application7/QueryProject1/salesforceReference_query">true</done>
               <queryLocator xmlns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Application7/QueryProject1/salesforceReference_query" xsi:nil="true"/>
               <records xmlns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Application7/QueryProject1/salesforceReference_query" xsi:type="sf:Account">
                  <sf:Id xmlns:sf="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Application7/QueryProject1/salesforceReference_query">0019000000CBebmAAC</sf:Id>
                  <sf:Name xmlns:sf="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Application7/QueryProject1/salesforceReference_query">
               </records>
               <records xmlns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Application7/QueryProject1/salesforceReference_query" xsi:type="sf:Account">
                  <sf:Id xmlns:sf="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Application7/QueryProject1/salesforceReference_query">0019000000CBeKHAA0</sf:Id>
                  <sf:Name xmlns:sf="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Application7/QueryProject1/salesforceReference_query">
               </records>
               <size xmlns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Application7/QueryProject1/salesforceReference_query">2</size>
            </ns0:QueryResults>
         </ns0:queryResponse>
      </part>
   </Invoke1_query_OutputVariable>
</messages>
```

3.2.6 queryAll

- queryAll has a wider scope than query operation, in another word the syntax is same as query (SOQL) but it also fetches the deleted records present in the recycle bin.
- Oracle Cloud Adapter for Salesforce.com provides functionality for using bind parameters in queryAll operation. Using this functionality, you can dynamically provide an input to your queryAll. For more information, refer to the section "Support for Bind Parameters"
- The main difference between query and queryAll is that queryAll returns the records, even if they have been deleted and are present in the recycle bin, while query only returns the records that are currently a part of Salesforce.com organization's active data.

Supported Headers

- LimitInfoHeader
- QueryOptionsHeader

Figure 3-20 shows a sample request and response message for queryAll operation on Salesforce.com.

Figure 3- 20 Request and Response Message for queryAll Operation

3.2.7 search

- Based on a search string, the search operation fetches records from Salesforce.com.
- Uses the SOSL (Structured Object Search Language) to fetch records from Salesforce.com.
- Oracle Cloud Adapter for Salesforce.com provides a provision for using bind parameters in search. Using this functionality, you can dynamically provide a search string as an input to your search operation.

Supported Headers

- LimitInfoHeader
- PackageVersionHeader

Figure 3-21 shows a sample request and response message for search operation on Salesforce.com.

Figure 3- 21 Request and Response Message for search Operation

3.2.8 queryMore

- queryMore operation retrieves a next set of records against a specified query string.
- To use queryMore operation, you are required to provide queryLocator value that has been returned as a result of query or queryAll operation, and you can use that queryLocator value to use queryMore call.
- queryMore operation is supported internally so it is not visible in the operations page in Oracle Cloud Adapter for Salesforce.com configuration wizard.
- Since it internally uses the query and queryAll, the request and response structure for queryMore could be of either of them, depending upon which operation's queryLocator is called.

Supported Headers

The headers supported for query and queryAll operations hold to for queryMore operation.

Figure 3-22 shows how you can set the value of 'queryLocator' at the design-time.

Figure 3- 22 Set Value of 'queryLocator' at Design-Time

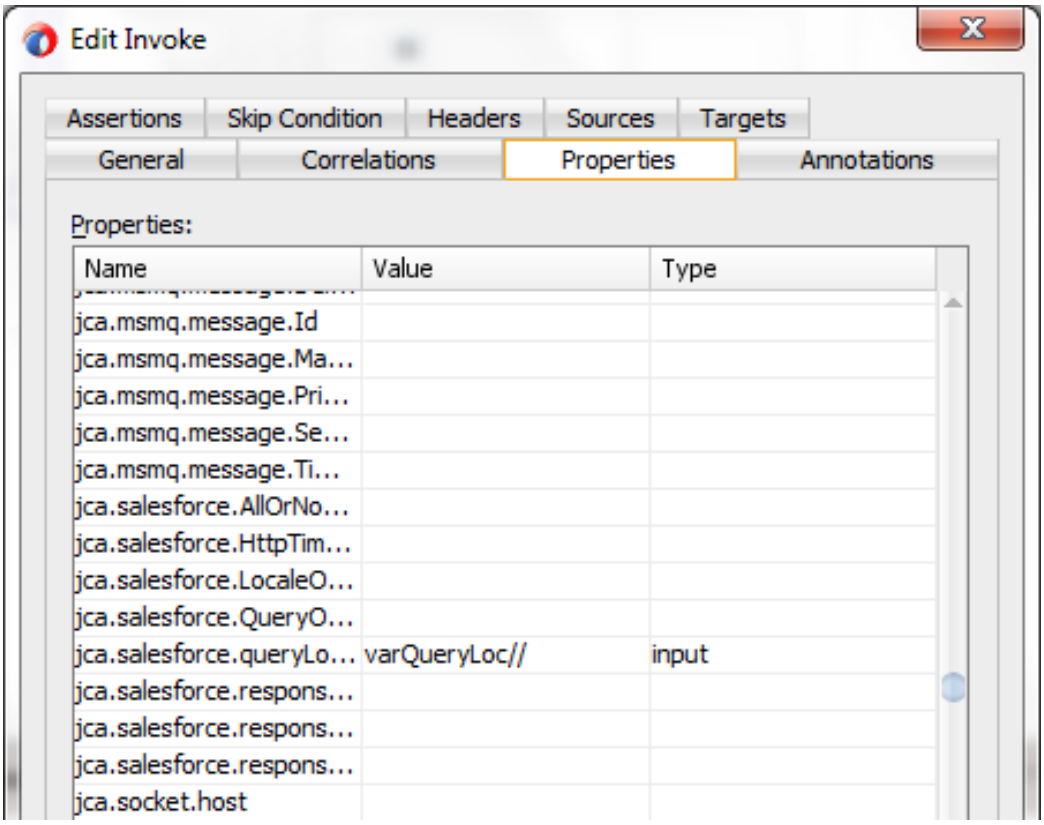
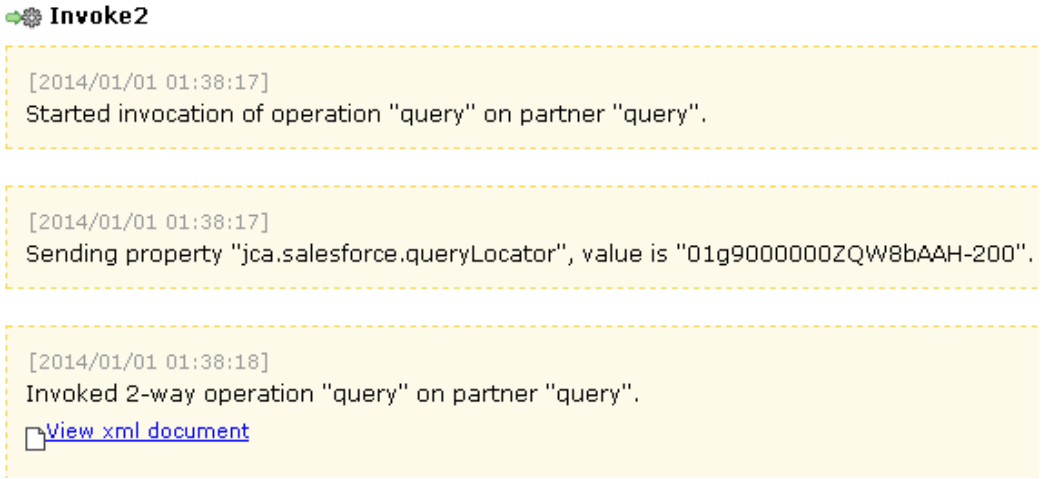


Figure 3-23 shows how the value of 'queryLocator' is passed at the run-time.

Figure 3- 23 Set Value of 'queryLocator' at Run-Time



3.3 Salesforce.com SOAP Headers

The section (Table 3-2) explains what all headers are maintained by every supported operation of Oracle Cloud Adapter for Salesforce.com. Salesforce.com puts into effect what SOAP headers are available for each operation. Moreover, this functionality is enforced by the adapter UI, that is, the headers available for a particular operation on the Headers and Properties page is in accordance to the recommended headers for that operation. For more

information on Salesforce.com headers, follow the link (http://www.salesforce.com/us/developer/docs/api/Content/soap_headers.htm).

Table 3- 2 Salesforce.com SOAP Headers

Operation Type	Operation Name	All or None Header	Allow Field Truncation Header	Assignment Rule Header	Debugging Header	Email Header	Limit Info Header	Mru Header	Package Version Header	Query Options Header
Core Operations	convertLead		Y		Y		Y		Y	
	getDeleted						Y			
	getUpdated						Y			
	merge		Y	Y	Y	Y	Y	Y	Y	
	undelete	Y	Y		Y		Y		Y	
	upsert	Y	Y	Y	Y	Y	Y	Y	Y	
CRUD Operations	create	Y	Y	Y	Y	Y	Y	Y	Y	
	retrieve						Y	Y	Y	Y
	update	Y		Y		Y	Y	Y	Y	
	delete	Y				Y	Y		Y	
MISC Operations	getUserInfo						Y			
	process						Y		Y	
SOSL and SOQL Operations	query						Y	Y	Y	Y
	queryAll						Y			Y
	queryMore						Y			Y
	search						Y		Y	

3.3.1 Request Headers

Headers that come under this category are the one that are being sent along with the request call made to Salesforce.com. Following is the explanation of each header that comes under this category:

3.3.1.1 AllOrNoneHeader

This header allows transactional behavior for Salesforce.com operations. This means if you set this header to “true” then the call to Salesforce.com will get committed only if it gets completed without any error, otherwise it will rollback. The default behavior is to commit partial records without any error.

This header was added in API version 20.0

Supported Operations:

Create(), update(), upsert(), delete(), undelete()

Fields :

Element Name	Type	Description
allOrNone	Boolean	**True**: If one record in a payload fails, all records are rolled back, i.e. none of them are committed on

		Salesforce.com. A record is committed only when all records in a payload are successfully written. **False**: Indicates if any record in a payload fails, only those records are rolled back, all other records are committed to Salesforce.com

3.3.1.2 AllowFieldTruncationHeader

This header enables the truncation behavior for the following fields, which are of string data type.

- anyType: anyType can be anyone from rest of the list.
- email
- picklist
- encryptedstring
- textarea
- mulitpicklist
- phone
- string

This header was added in API version 15.0.

Supported Operations:

Create(), update(), upsert(), undelete(), process(), merge(), convertLead()

Fields:

Element Name	Type	Description
allowFieldTruncation	Boolean	**True**: If a user attempts to enter a value of 25 characters in a field of 20 characters, first 20 records are inserted into the field and, the transaction is a success. **False**: If a user attempts to enter a value of 25 characters in a field of 20 characters, an error is thrown and the transaction does not commit.

3.3.1.3 AssignmentRuleHeader

The AssignmentRuleHeader specifies the assignment rule to use when creating or updating an Account, Case, or Lead. The assignment rule can be active or inactive in your Salesforce.com organization. The ID can be retrieved by querying the AssignmentRule object. If the ID is specified, then you do not need to specify useDefaultRule. The MALFORMED_ID exception is returned, if the value is not a correct ID and the call fails.

This element is ignored for accounts, because all territory assignment rules are applied.

A MALFORMED_ID exception is returned, if the value is not a correct ID and the call fails.

Supported Operations:

Create(), update(), merge(), upsert()

Fields:

Element Name	Type	Description
assignmentRuleId	ID	The ID of the assignment rule which you want to use. The Id is not validated by Oracle Cloud Adapter for Salesforce.com, whether it exists or

		not in Salesforce.com. The validation takes place during the run-time.
useDefaultRule	Boolean	**True**: The default (active) assignment rule is used. **False**: The default (active) assignment rule is not applied.

3.3.1.4 EmailHeader

This header enables you to specify whether a notification email should be sent or not.

Supported Operations:

Create(), update(),delete(), upsert()

Fields:

Element Name	Type	Description
triggerAutoResponseEmail	Boolean	**True:** It triggers auto-response rules for leads and cases. **False:** Auto-response rules for leads and cases are not triggered.
triggerOtherEmail	Boolean	**True**: The email is triggered outside the organization. **False**: The email is not triggered outside the organization.
triggerUserEmail	boolean	**True**: The email is triggered and sent to users in the organization. This email is triggered by a number of events like adding comments to a case or updating a task. **False**: The email is not triggered and sent to users in the organization.

3.3.1.5 DebuggingHeader

Logging level for debugging purposes, following are different levels that can be mentioned in this header:

Supported Operations:

create(), upsert(), undelete(), merge(), convertLead()

Fields:

Element Name	Type	Description
debugLevel	Logtype	The following list orders the log levels from least (NONE) to most verbose (DETAIL): NONE DEBUGONLY DB PROFILING CALLOUT DETAIL

3.3.1.6 MruHeader

Recent Items section in Salesforce.com shows most recently used items. In API version 7.0 or later, the list would not get updated by itself. The MruHeader must be used in order to update that list. Note that using this header may impact performance negatively.

Supported Operations:

Create(), update(), merge(), upsert(), query(), retrieve()

Fields:

Element Name	Type	Description
updateMru	Boolean	**True**: The list of most recently used item is updated on Salesforce.com. **False:** The list of most recently used item is not updated on Salesforce.com.

3.3.1.7 PackageVersionHeader

A Package version is basically the identification of components in a package. Package version has a specific format majorNumber.minorNumber.patchNumber. For example, 3.4.5,(where" 3" refers to majorNumber, "4" refers to minorNumber and "5" refers to patchNumber)

This header is used to specify package version for any installed package.

Supported Operations:

create(), retrieve(), update(), delete(), undelete(), merge(), upsert(), process(), query(), search(), convertLead()

Fields:

Element Name	Type	Description
majorNumber	Int	Major version number of a package version.
minorNumber	Int	Minor version number of a package version.
Namespace	String	Namespace of the managed package.

3.3.1.8 QueryOptions

This header is used to specify batch size for queries. Default value for batch size is 500. Minimum value for this is 200 and maximum value is 2000.

Supported Operations:

retrieve(), queryMore(), query()

Fields:

Element Name	Type	Description
batchSize	Int	The batch size of the number of records returned in a query call. The minimum size is 200. If you enter a value less than 200, for example, 40 and your actual query has a size of 1200, it would not throw any error, but would return 200 records. The maximum size is 2000. The configuration wizard doesn't let you enter a value more than 2000.

3.3.2 Response Headers

Headers that come under this category are ones that are being received along with the response message that is being sent by Salesforce.com. Following is the explanation of each header that comes under this category:

3.3.2.1 DebuggingInfo

This response header will be returned only if debugLevel request header has been sent along request payload to Salesforce.com.

Fields:

Element Name	Type	Description
debugLog	String	The log information returned from the adapter invocation. This header is part of the SOAP response once debugLevel is sent as a part of the SOAP request

3.3.2.2 LimitInfoHeader

This header provides the information about the limitations of API calls on per-day basis for organization. This response header is introduced in API version 29.0

Fields:

Element Name	Type	Description
current	string	The number of calls that have already been used in the organization.
Limit	String	Organization's limit for specified limit type.
Type	String	Limit information type specified in the header API REQUESTS—contains limit information about API calls for the organization.

3.4 Session Management

Oracle Cloud Adapter for Salesforce.com provides session management capabilities to maintain transactions related to a particular Salesforce.com user. Attempts have been made to reduce the number of calls to Salesforce.com either for a login call or for any subsequent calls to fetch metadata. With this reduction in the number of calls, the adapter responds faster and the limitations that Salesforce.com imposes on the number of calls are also taken care of.

Configuration for session support is provided in two phases.

- Design Phase
- Execution Phase

3.4.1 Design Phase

- **Login:** While creating an adapter service, a single login call is fired to Salesforce.com during the complete Salesforce Cloud Adapter Configuration Wizard cycle. Once the user provides an Enterprise WSDL and the CSF key in the connection page, only one login call is made to Salesforce.com till the completion of adapter configuration. The next login does not happen unless a CSF key for another

user is selected or a different Enterprise WSDL is selected. Even the 'Run Query Test Tool' follows the same session that was initialized during the connection configuration page.

- **Fetch Metadata:** Post the first fetch metadata call, subsequent metadata calls have been reduced considerably. For instance, based on the current user and version of the Enterprise WSDL, metadata is cached. Now, if you are configuring a new instance of Oracle Cloud Adapter for Salesforce.com for the same combination of user and version of Enterprise WSDL, instead of making a call to Salesforce.com for fetching metadata details, the Oracle Cloud Adapter for Salesforce.com will get the metadata from the cache. In case you want to bypass fetching data from cache and make a direct call to Salesforce.com, use the **Clear Cache** option on the connection page. Besides serving an immediate need, the metadata cached in the above step is also used when a user opts for **Offline Mode**.
- **Offline Mode:** This is a unique functionality offered by the Oracle Cloud Adapter for Salesforce.com in which the metadata information is saved in the cache and the adapter doesn't have to make any further calls, be it a login or fetch metadata call and yet, the user can successfully proceed with adapter and composite creation. Consequently, till the checkbox for offline configuration is checked, the complete adapter creation is handled by cache without a single login or metadata fetch call. For more details on how the Offline Mode works, refer to the section "Offline Mode".

Note: In case any changes (modified/created) have been made to your organization's business objects on Salesforce.com, use the latest Enterprise WSDL and uncheck and check **Offline Configuration** checkbox to ensure you have the updated metadata in cache.

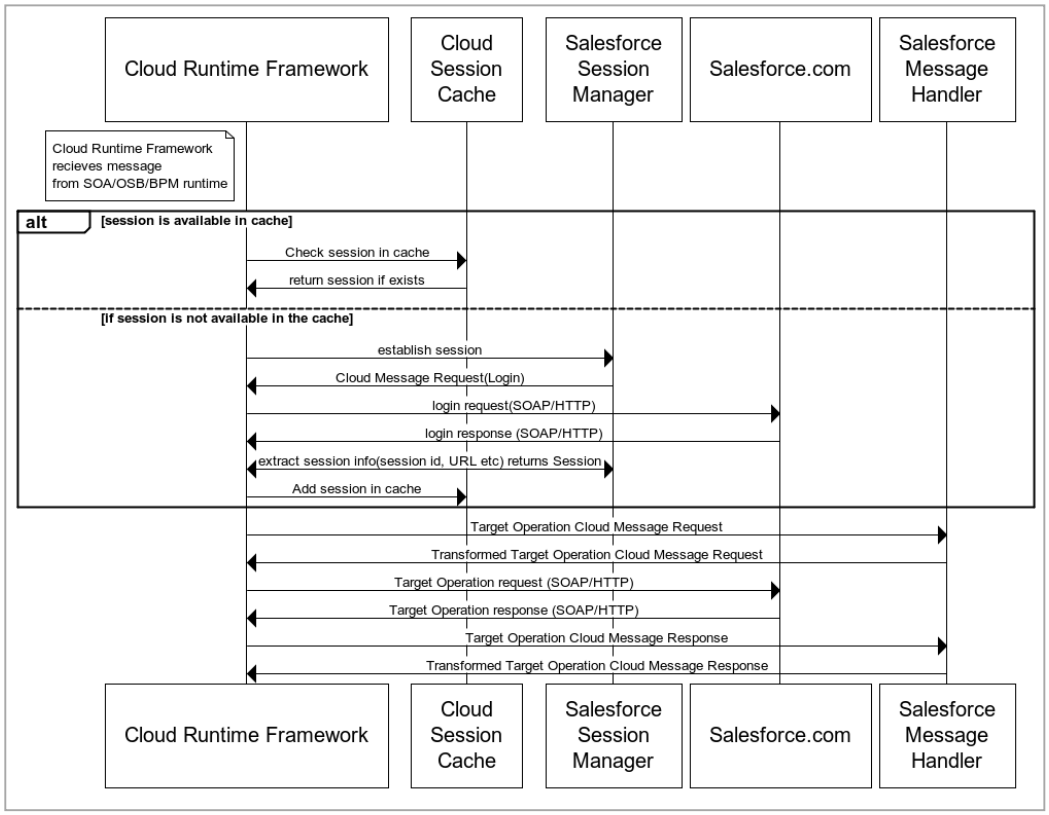
3.4.2 Execution Phase

During the execution phase, Cloud Runtime frame work receives the request message from SOA/OSB/BPM runtime.

Oracle Cloud Runtime framework takes care of the session management as it internally maintains a Session Cache (in the form of Map) which stores the session details. A session is uniquely defined for the user and version of the enterprise WSDL.

If a session is not found in Session Cache then framework establishes the session with Salesforce.com and then adds the session information in Session Cache. If we have different composites which are using the same credentials and the version of the enterprise WSDL then all threads would be provided the same session details. This helps in avoiding the additional login call to Salesforce as we already have the session details available in the cache. Figure 3-24 shows Execution Phase.

Figure 3- 24 Execution Phase



- A single login call made by the Oracle Cloud Adapter for Salesforce.com maintains the session across composites deployed in the server for a particular user and for a fixed version of Enterprise WSDL.
- During run-time as well, the fetch metadata calls are minimal.

You are no longer required to create a complex process, where you first need to invoke a login operation on Salesforce.com whose output payload provides two important details which must be supplied during each subsequent operation. These are:

- **ServerURL:** This is the URL that needs to be called for all subsequent operations (query, update etc.) for this user using the dynamic partner link concept.
- **Session ID:** This ID needs to be sent as part of header information for all operations post login.

Session management is now completely handled by the Oracle Cloud Adapter for Salesforce.com making the task of Salesforce.com operation invocation flawless and effortless.

3.5 Handling Polymorphic Behavior of Salesforce.com Schema

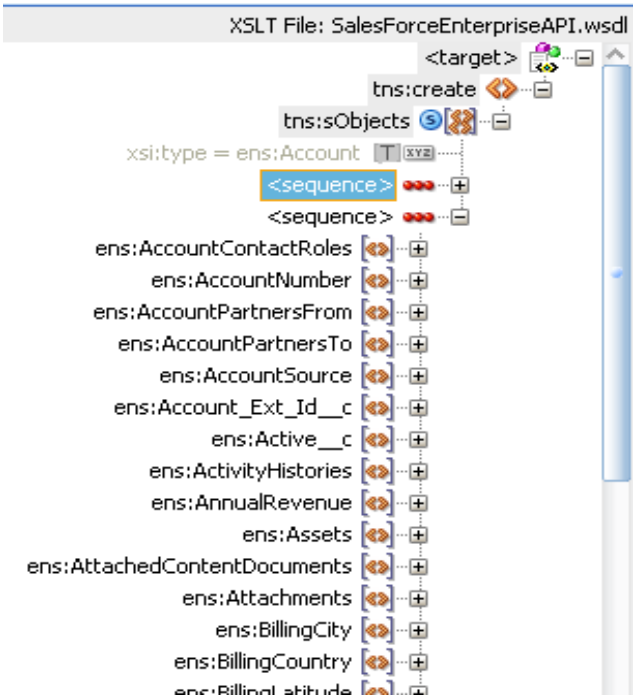
Due to the polymorphic nature of the Salesforce.com XML schema, it was cumbersome to write multiple objects in a single operation call to Salesforce.com using a Web service adapter. For instance, creation of a lead, account and a contact consisted of four steps, as mentioned below:

- You had to create an invoke activity for login operation.
- You had to create account.
- You had to create contact.
- You had to create lead.

All of this is done separately, however; with the introduction of Oracle Cloud Adapter for Salesforce.com to Oracle SOA Suite, integration to Salesforce.com has become a lot easier. Now the above mentioned four-step process has been reduced to a single step.

Figure 3-25 shows how only a single sObject was manipulated during the design-time.

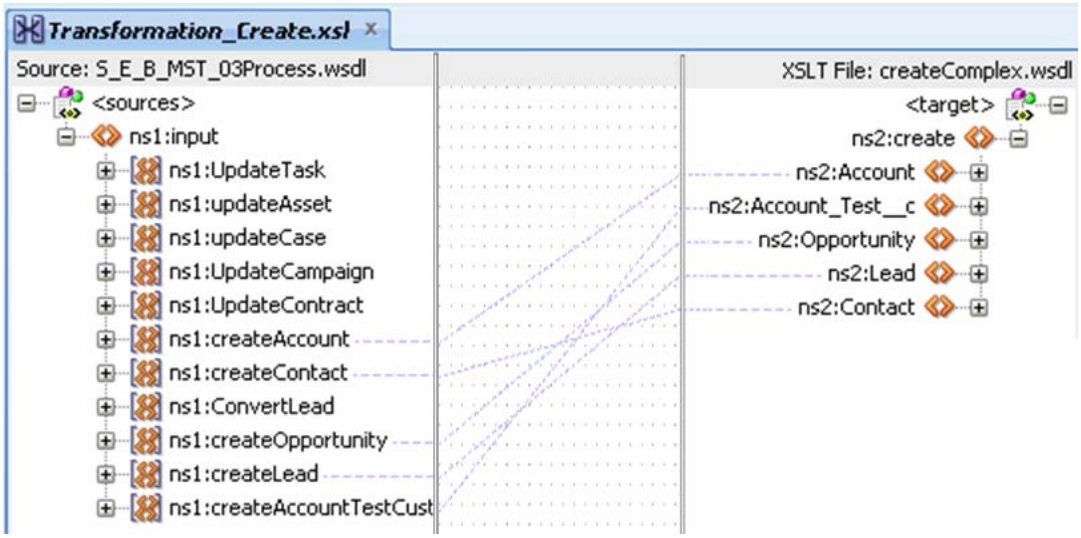
Figure 3- 25 Transformation Create Account



On the other hand, the polymorphic behavior of Salesforce.com schema is handled by Oracle Cloud Adapter for Salesforce.com and multiple sObjects can be written with a lot more ease, as shown in Figure 3-26.

Note that this behavior is possible only for operations where multiple insertion/writes are allowed by the Salesforce.com Enterprise API.

Figure 3- 26 Transformation Create Account



3.6 Clear Cache

To save the number of calls to Salesforce.com for fetching metadata, the adapter maintains a cache wherein all the metadata information is stored for future reference. However, in case any modification is done to objects on Salesforce.com and the same needs to be reflected in the adapter, the clear cache functionality does the task of deleting the old cache and fetching the latest metadata from Salesforce.com. For instance, for an already existing object on Salesforce.com, whose creatable property is by default “false”, if you update this property to true and clear the cache, the object should now appear in the list of **Available objects** under create operation.

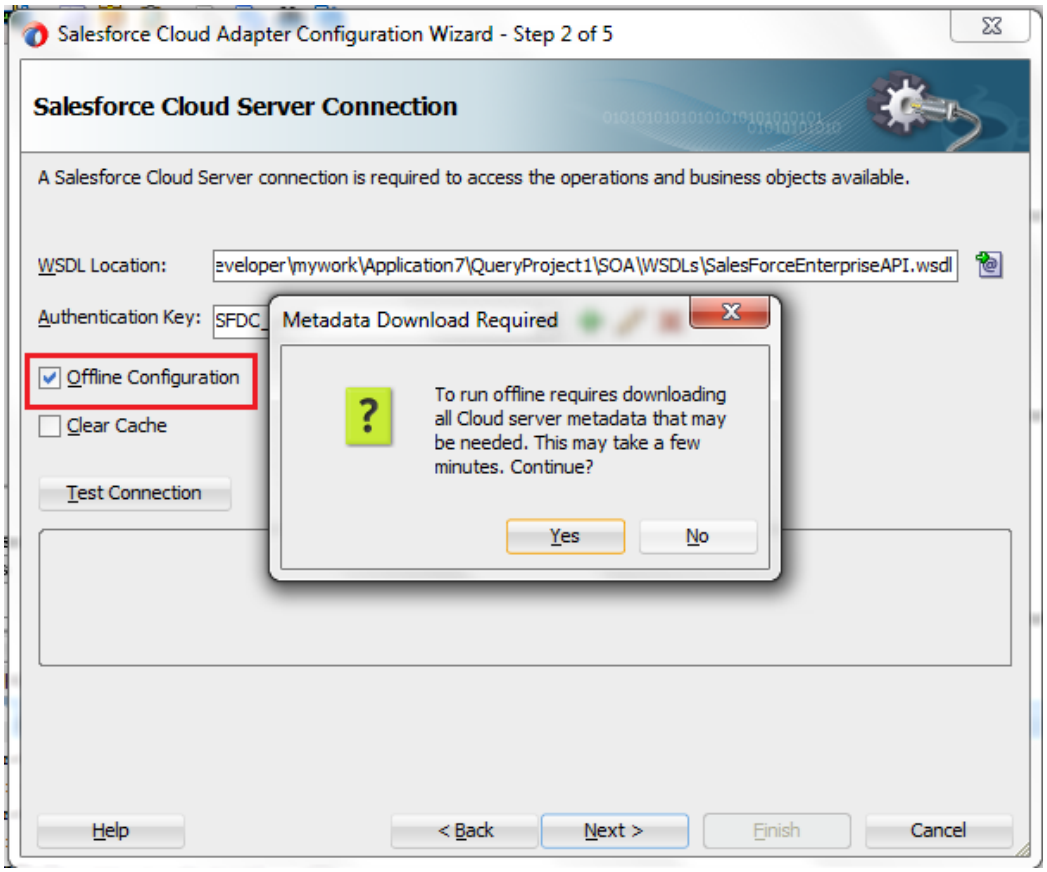
3.7 Offline Mode

As the name suggests, Offline Mode enables you to continue your adapter development even when the network connection is not available. It is a unique characteristic of the Oracle Cloud Adapter for Salesforce.com, and is a step forward in enabling seamless integration to Salesforce.com using the Oracle SOA Suite.

The metadata of your enterprise is downloaded and saved in the cache while you are online. For this, you need to go to the **Connection** page of your Oracle Cloud Adapter Configuration Wizard and select the checkbox for **Offline Configuration**, as shown in Figure 3-27.

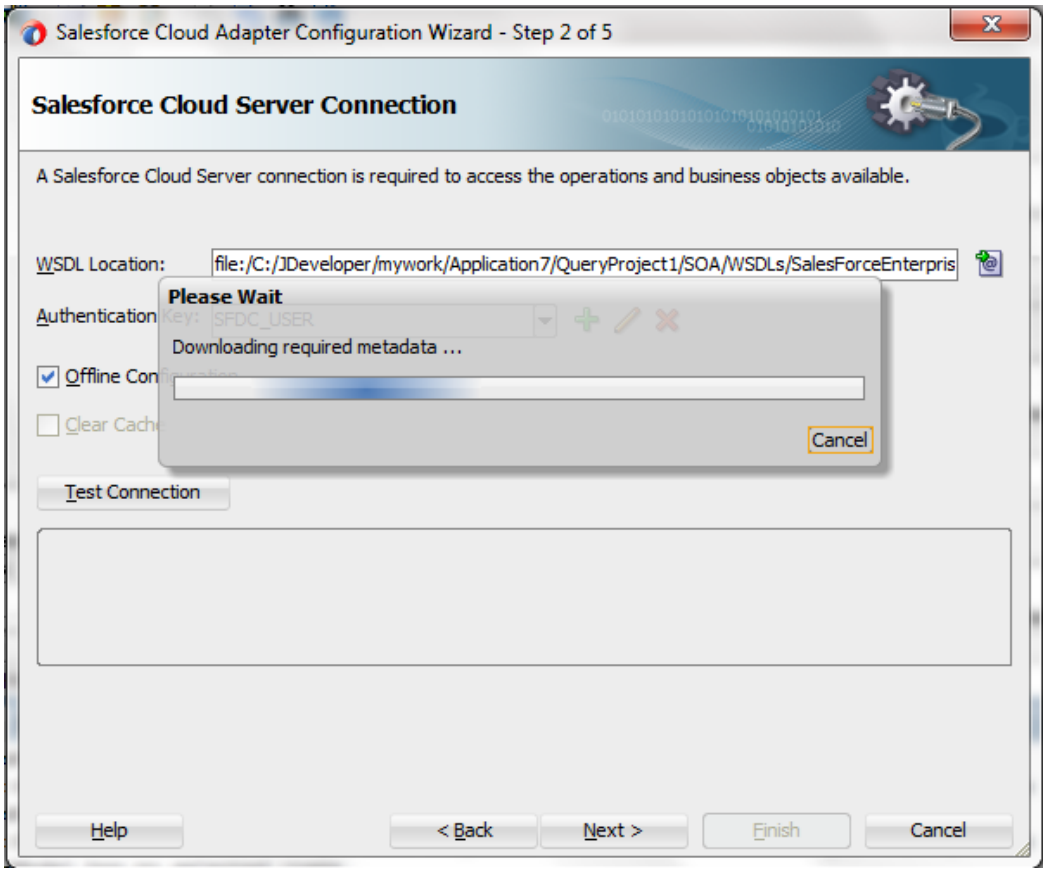
The moment you click the **Offline Configuration** checkbox, a ’Metadata Download Required’ message is displayed as “To run offline requires downloading all Cloud server metadata that may be needed. This may take a few minutes. Continue?” To confirm, click **Yes** else click **No**.

Figure 3- 27 Offline Configuration



On clicking '**Yes**', it starts downloading the metadata as shown in the Figure 3-28.

Figure 3- 28 Downloading Required Metadata



Now you will be able to proceed with the configuration wizard just like the normal configuration. Next time you create a new adapter, the checkbox would already be selected and you need to uncheck it ONLY if something has changed. This feature is very useful in case your organization's data changes rarely.

Selecting Offline Mode would make no difference to the execution of your composite on the Enterprise Manager Console, that is, the execution behavior of the composite remains same whether it is created with Offline Configuration checked or unchecked in the adapter configuration wizard.

3.8 Behaviors Expected When Using Offline Mode

Although the Offline Mode comes with numerous advantages, following are the limitations associated with it:

1. You will not be able to run the **Query Test** tool if the **Offline Configuration** checkbox is checked. In this case, if you click the **Query Test** tool button, a warning message is displayed that 'Query Test is not available in Offline mode'.
2. You will also not be aware to any new custom objects / custom attributes added by the organization while in offline mode. In this case, you need to uncheck the **Offline Configuration** checkbox and either check the 'clear cache' or 'offline configuration' checkbox to fetch the latest metadata.

3.9 Security Management

If the Salesforce.com API is invoked without using the Oracle Cloud Adapter for Salesforce.com, the credentials for login to Salesforce.com are sent with payload during login operation invocation. Instead, SFDC adapter uses CSF key to externalize the credentials from the SOA composites. Moreover, existing OWSM policies cannot be used to pass user credentials with the payload. Oracle Cloud Adapter for Salesforce.com provides security in the following ways:

- The Credential Store Framework translates the key into credentials and sends it over the network. We are relying upon SSL to have this encrypted to eliminate any possibility of eavesdropping.
- While creating the CSF key either in the configuration wizard or in the Enterprise Manager Console, the password characters are not exposed, thereby enforcing additional security.

For information on how to configure CSF key, refer to the section "A.2 CSF Key in Enterprise Manager".

3.10 Test Functionality

The test functionality is a distinctive feature of the Oracle Cloud adapter for Salesforce.com that is available in two forms – one in the form of testing the connection parameters, while the other is to validate and execute the SOQL/SOSL query. The section below talks more about these two features.

3.10.1 Design-Time Test Functionality

Design-time test functionalities include the following test functionalities:

1. **Test Salesforce.com Cloud Connection:** There is a **Test Connection** button on **Connection Configuration** page of Salesforce Cloud Adapter Configuration Wizard.

   By clicking on **Test Connection** button you will see below message:

   - Success! - For successful connection
   - Error! - For any exception during login

2. **Query Test Tool:** This tool helps to run and test query. By clicking on **Test** icon on **Operation Configuration** page of Salesforce Cloud Adapter Configuration Wizard, a test dialogue box appears with a **Query Statement** text box and **Results** box showing result of query executed. A provision is provided to add bind parameters to the query. A box corresponding to each bind parameter appears, wherein you can provide a value to the parameter and then test the query.

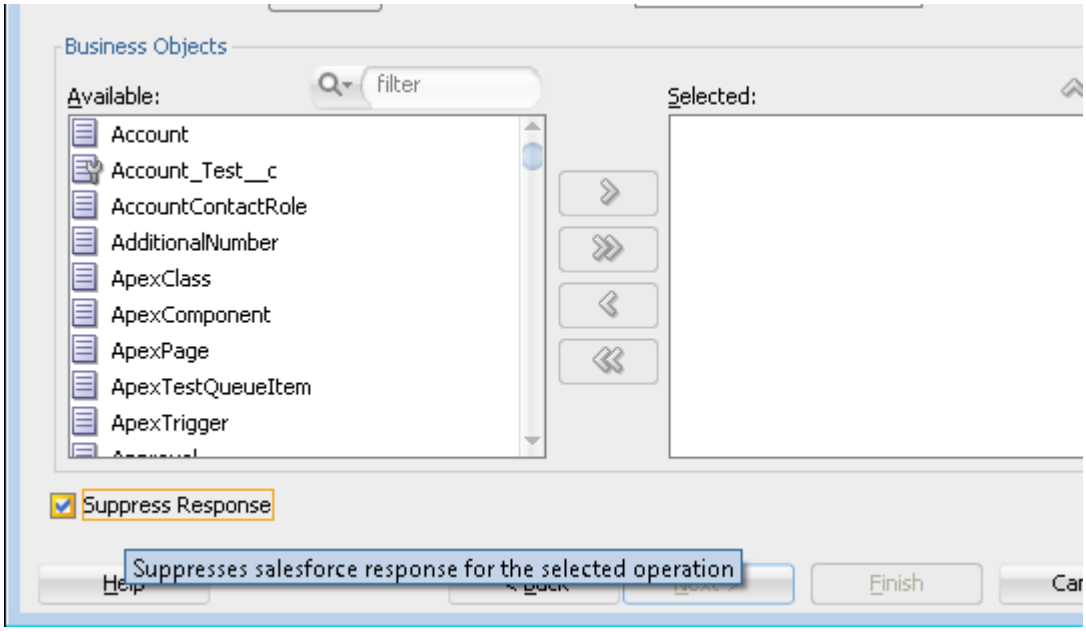
For more information, refer to the section "Design-Time: Using Oracle Cloud Adapter for Salesforce.com Configuration Wizard". Note that these capabilities are not available in offline mode.

3.11 Suppress Response

Figure 3-29 shows the **Suppress Response** feature of Oracle Cloud Adapter for Salesforce.com. It simulates a one-way / fire and forgets behavior in a synchronous call to Salesforce.com. The advantage of this functionality is that the calls are stateless and in case there is no business requirement to save the response returned from Salesforce.com, it saves the server from unwanted load. Also, the response is suppressed (is empty) either when a valid response is returned or when an exception is returned by Salesforce.com. However, if a fault is thrown by the adapter, it would be thrown to the client even if 'Suppress Response' is selected. The client can accordingly handle the fault thrown by the adapter. For more information on faults returned by the Oracle Cloud Adapter for Salesforce.com, refer to the section "Fault Handling".

Note: Suppress Response feature only simulates the one way behavior. The adapter run-time is still making a synchronous call and receiving the response. The client is going to be in suspended mode until then.

Figure 3- 29 Suppress Response



3.12 Fault Handling

Errors from Salesforce.com are returned in the following ways:

- **Salesforce Faults**: Error scenarios where the composite execution cannot proceed further and a **binding exception** is raised in the BPEL process.
- **Salesforce Exceptions**: Error scenarios where composite execution proceeds even if there are issues with committing some records. The records for which the transaction does not succeed returns a success status as 'false', while the records for which transaction commits to return a success status as 'true'.
- **Salesforce Host Unreachable**: Error scenarios in which the Salesforce.com host is not reachable, a **remote exception** is raised in business process.

The remote and binding faults are handled at the BPEL level using either fault policies or placing fault handlers in the process, i.e. using catch and catchAll blocks.

For exceptions returned from Salesforce.com, the error message is returned as part of the payload. Figure 3-30 shows the payload structure containing exception details.

Figure 3- 30 Payload Structure Containing Exception Details

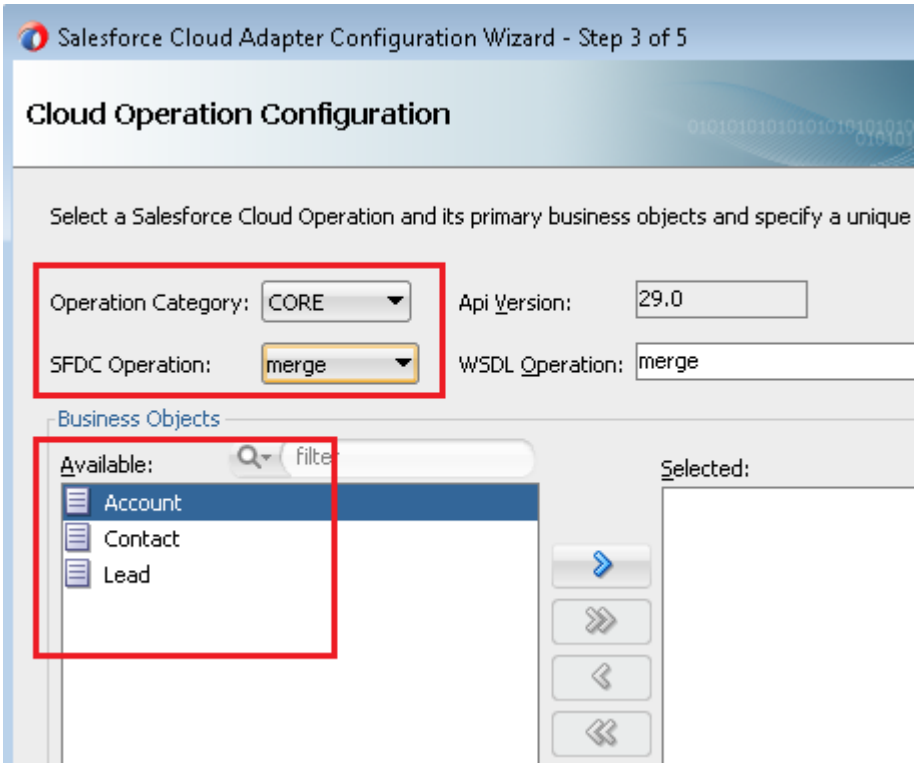
```
<xsd:complexType name="Error">
    <xsd:sequence>
        <xsd:element name="fields" type="xsd:string" nillable="true"
        minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="message" type="xsd:string"/>
        <xsd:element name="statusCode" type="tns:StatusCode"/>
    </xsd:sequence>
</xsd:complexType>
```

For Salesforce.com standard faults and exception details, refer to the section "Troubleshooting and Error Messages".

3.13 Salesforce.com Limit and Restriction Handling

Operation specific Object Selection: The list of objects available for a particular operation is dynamic and is updated separately for every operation. For instance, if you select **merge** operation under "Operation Category" **CORE**, only Account, Contact, and Lead objects are available in the list of "Available" under Business objects, as shown in Figure 3-31.

Figure 3- 31 Available Business Options in CORE Operation Category



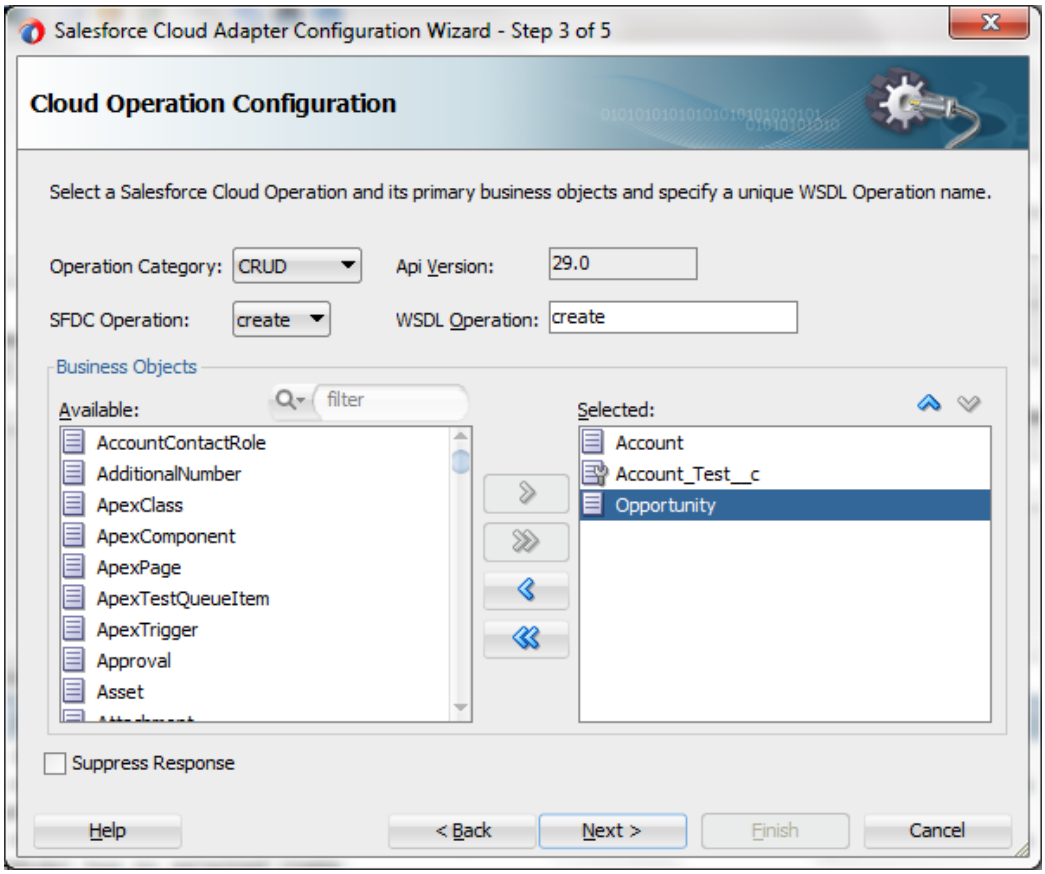
This section is further divided into the following subsections:

- Multiple Object Selection
- Single Object Selection
- Header Restrictions

Multiple Object Selection

For certain operations, there is a provision to manipulate more than one business object in a single call to Salesforce.com. An important example of this feature is the "**create**" operation, where you can select more than one object in one configuration wizard cycle, as shown in Figure 3-32.

Figure 3- 32 Create Operation in Cloud Operation Configuration Wizard



Single Object Selection

Operations that can operate only a single object in one call to Salesforce.com have a restriction in design-time that only a single object can be selected in the configuration wizard cycle. For example, in “retrieve” operation only a single object can be selected. As shown in Figure 3-33, the “Move Selected Values in Available List to Selected List” button (>) is disabled when you select more than one object.

Figure 3- 33 Retrieve Operation in Cloud Operation Configuration Wizard

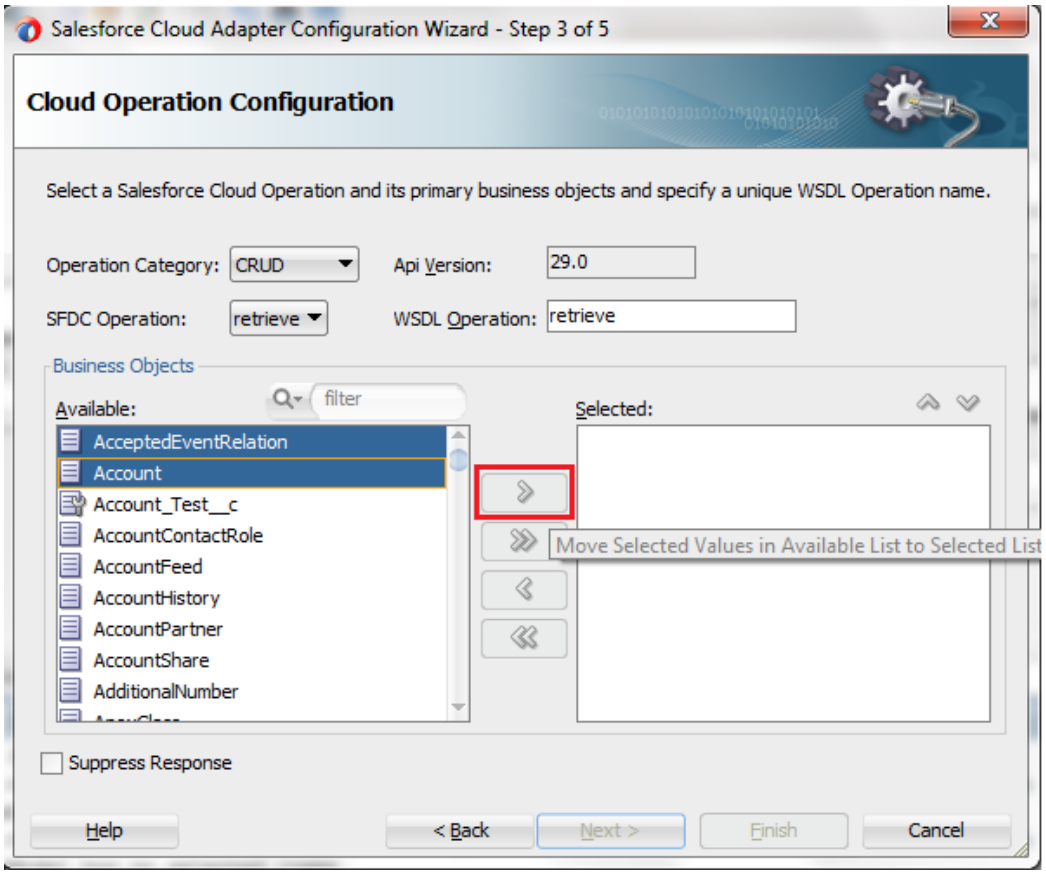


Table 3-33 shows an insight into the maximum number of objects you can select for a specific operation.

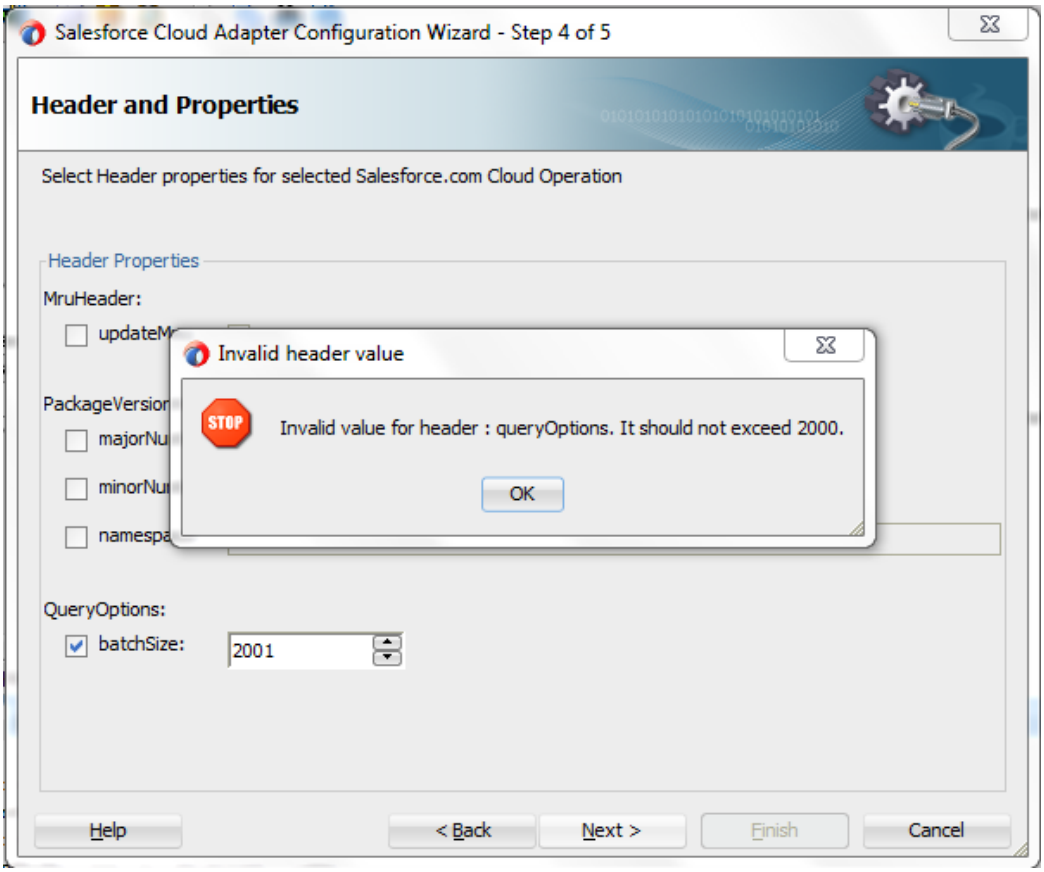
Table 3- 33 Maximum Number of Objects

#	Operation Name	Max Objects
1	create	10
2	retrieve	1
3	update	10
4	delete	Unbounded
5	upsert	1
6	undelete	Unbounded
7	convertLead	1
8	process	1
9	merge	1
10	getDeleted	1
11	getUpdated	1
12	getUserInfo	0

Header Restrictions

The Oracle Cloud Adapter for Salesforce.com saves the user from making any wrong selections at design-time, which may lead to erroneous conditions at run-time. For example, the maximum value for batch size in “Query” options header is 2000. The configuration wizard does not allow you to enter a value greater than 2000. If you enter a value which is greater than 2000, an error message is displayed, as shown in Figure 3-34.

Figure 3- 34 Invalid header value Message



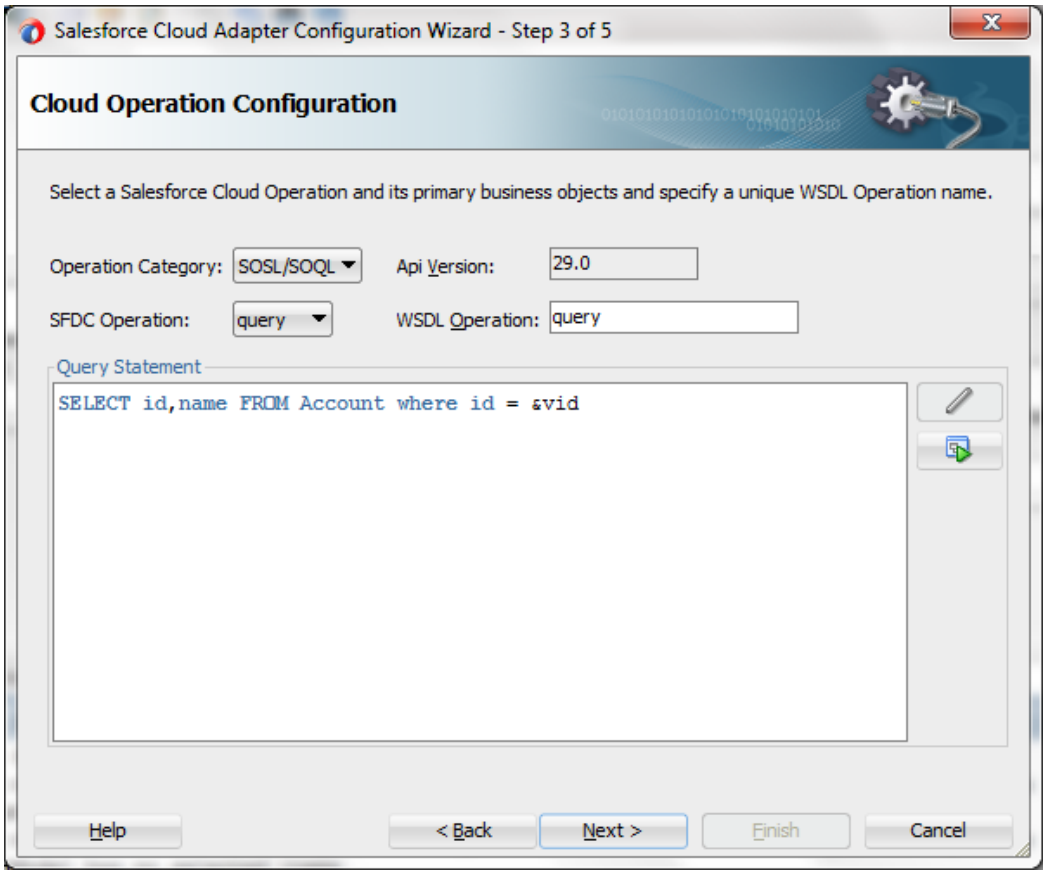
3.14 Support for Bind Parameters

The operations falling under the SOSL/SOQL category gives you the privilege to provide inputs to your query/search statement dynamically. This is made possible using the concept of bind parameters.

Consider the following query:

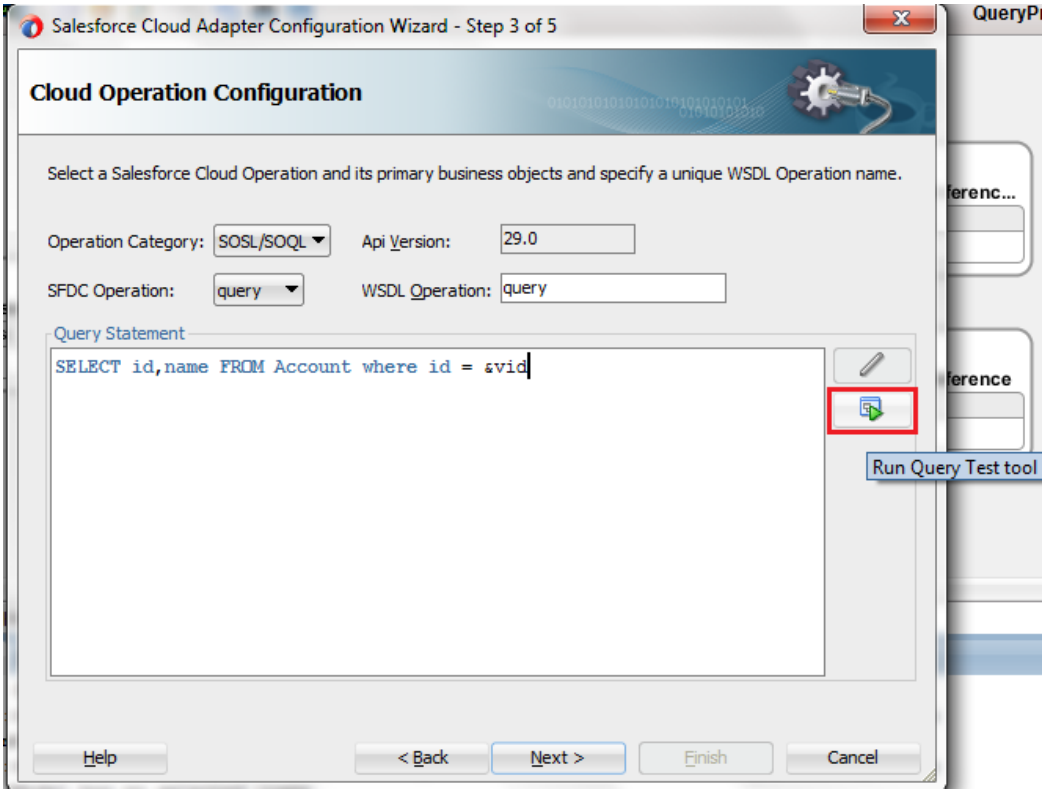
1. **Select id, Name from Account where id** = '0019000000sgbCW'
2. The Oracle Cloud Adapter for Salesforce.com gives you an option to give inputs to your query in the form of bind parameters, as shown in Figure 3-35.

Figure 3- 35 Give Inputs to your Query in the Form of Bind Parameters



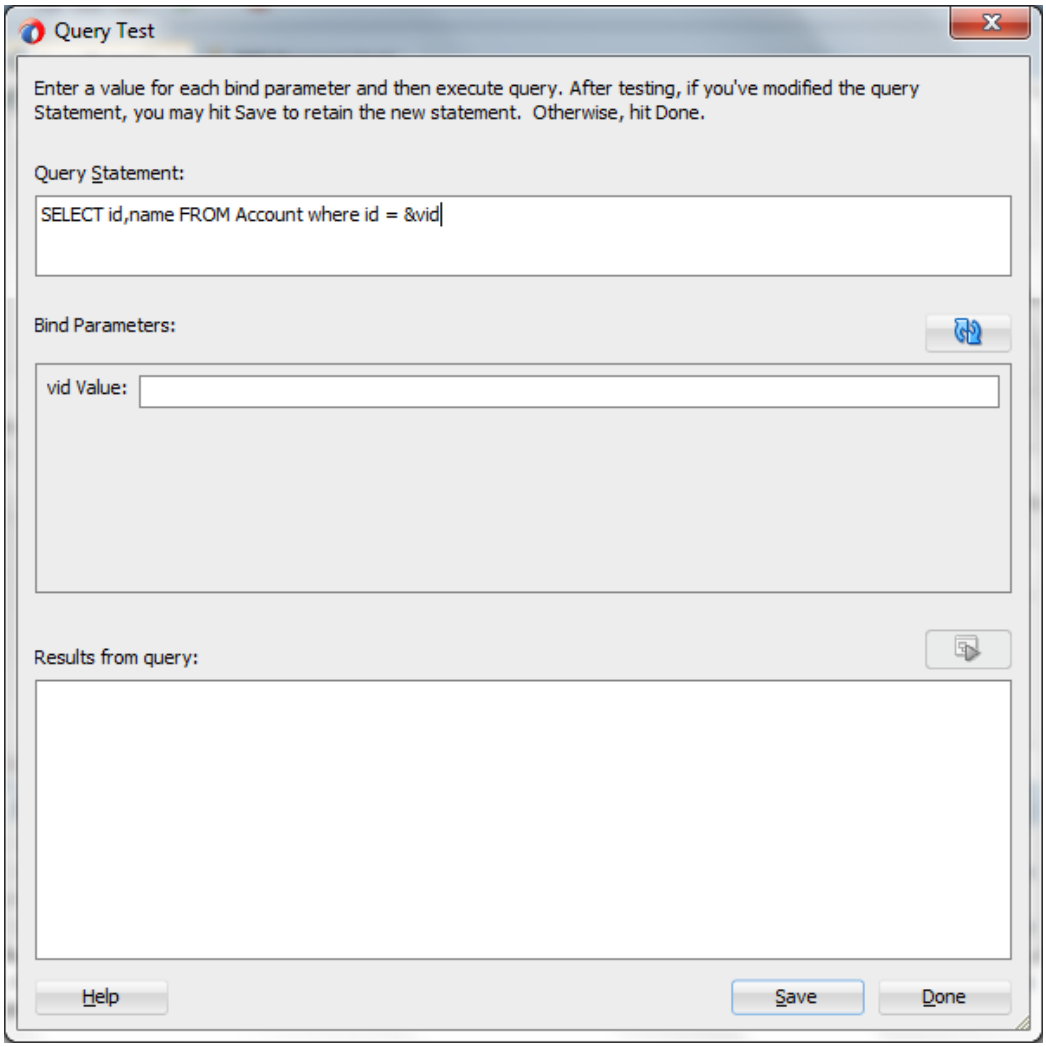
3. Here, the ampersand **'&'** symbol prefix to **vid** indicates that it is a bind parameter.
4. Click on the **Run Query Test tool** to see how it works for bind parameters, as shown in Figure 3-36.

Figure 3- 36 Run Query Test



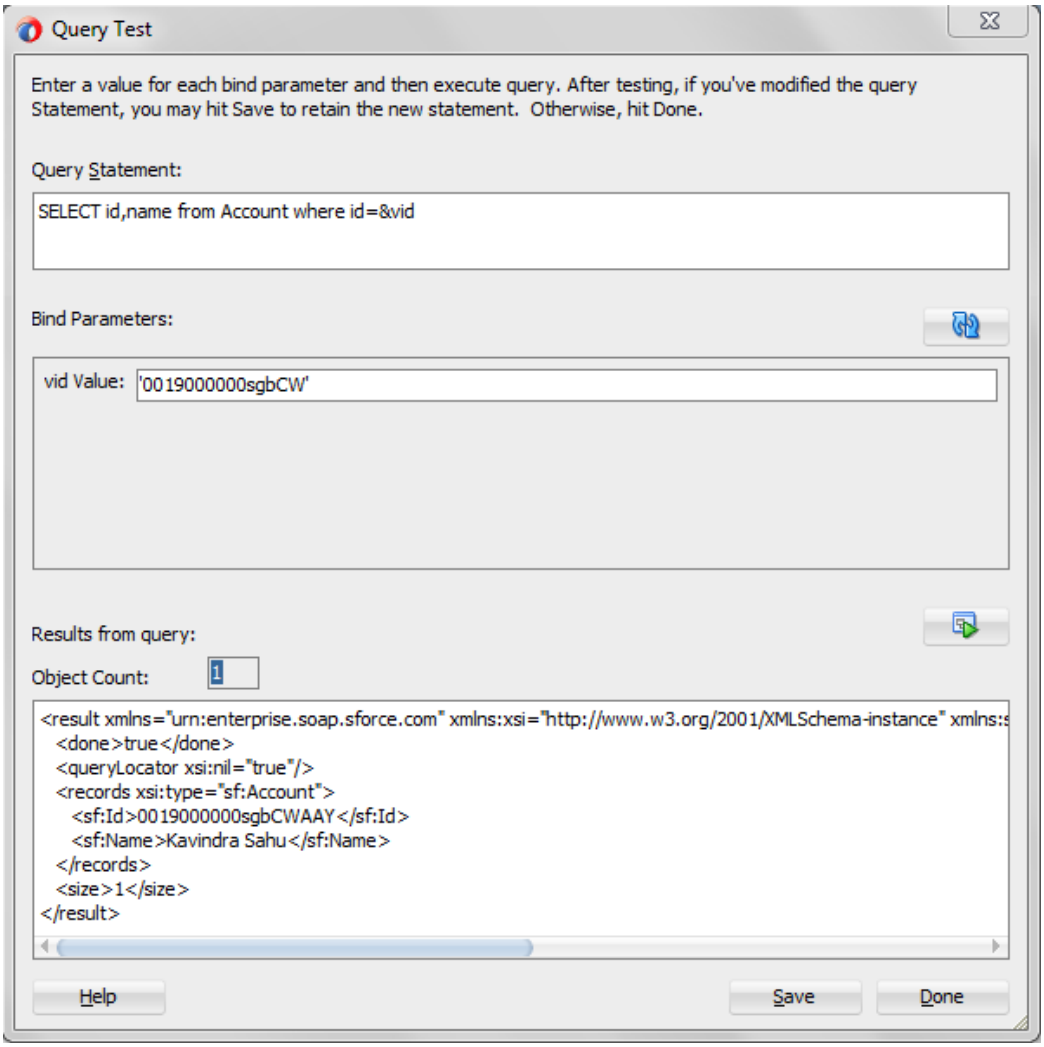
5. As shown in Figure 3-37, the **Run Query Test tool** prompts you to provide a value for the bind parameter, '**vid**' in this case.

Figure 3- 37 Provide a Value for the Bind Parameter



6. On providing an appropriate value to the bind parameter and clicking the '**Execute Query'** button, you can view the result set for that particular query, as shown in Figure 3-38.

Figure 3- 38 Result of the Query



7. The WSDL generated for this particular adapter contains the bind parameter as part of input schema, as shown in Figure 3-39.

Figure 3- 39 Input Schema

```
<xs:schema xmlns="http://xml.oracle.com/types/salesforceReference_2" elementFormDefault="qualified"
     xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://xml.oracle.com/types/salesforceReference_2"
     xmlns:ns3="http://xml.oracle.com/types/salesforceReference_2">
    <xs:complexType name="QueryParameters">
        <xs:all>
            <xs:element name="vid" type="xs:string" minOccurs="1" maxOccurs="1"/>
        </xs:all>
    </xs:complexType>
</xs:schema>
```

4

Design-Time: Using Oracle Cloud Adapter for Salesforce.com Configuration Wizard

This chapter defines how you can configure Oracle Cloud Adapter for Salesforce.com by walking through a complete Adapter Configuration Wizard scenario. Moreover, you will also get to know what artifacts are being generated after completing the Oracle Cloud Adapter for Salesforce.com Wizard.

It contains the following topics:

- Section 4.1, "Oracle Cloud Adapter for Salesforce.com Plug-in"
- Section 4.2, "Oracle Cloud Adapter for Salesforce.com Walkthrough"
- Section 4.3, "Design-time Artifact Generation"

4.1 Oracle Cloud Adapter for Salesforce.com Plug-in

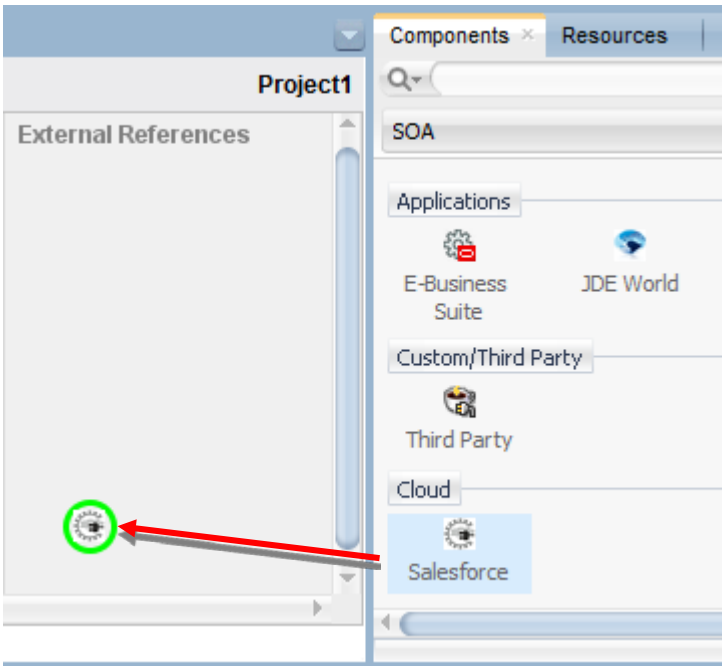
Oracle Cloud Adapter for Salesforce comes preinstalled with Oracle SOA Suite 12c (12.1.3) and is available under the Cloud subsection of Component Palette in JDeveloper.

4.2 Oracle Cloud Adapter for Salesforce.com Walkthrough

This section describes the Adapter Configuration Wizard and how you can define an Oracle Cloud Adapter for Salesforce.com by using the Adapter Configuration Wizard. Contrary to 11g wherein all adapters were part of the ‘Service Adapters’ subsection, 12c distinguishes adapters based on their usage. Salesforce adapter is visible under the ‘Cloud’ subsection, a section dedicated to cloud based adapters.

1. In the **Component Palette**, select **SOA**.
2. Under the **Cloud** subsection you should be able to find an icon for Oracle Cloud Adapter for Salesforce.com
3. Drag and drop **Salesforce** Adapter to the **External References** swim-lane in the composite.xml page, as shown in Figure 4-1.

Figure 4- 1 Salesforce Adapter



The **Salesforce Cloud Adapter Configuration Wizard** dialog is displayed.

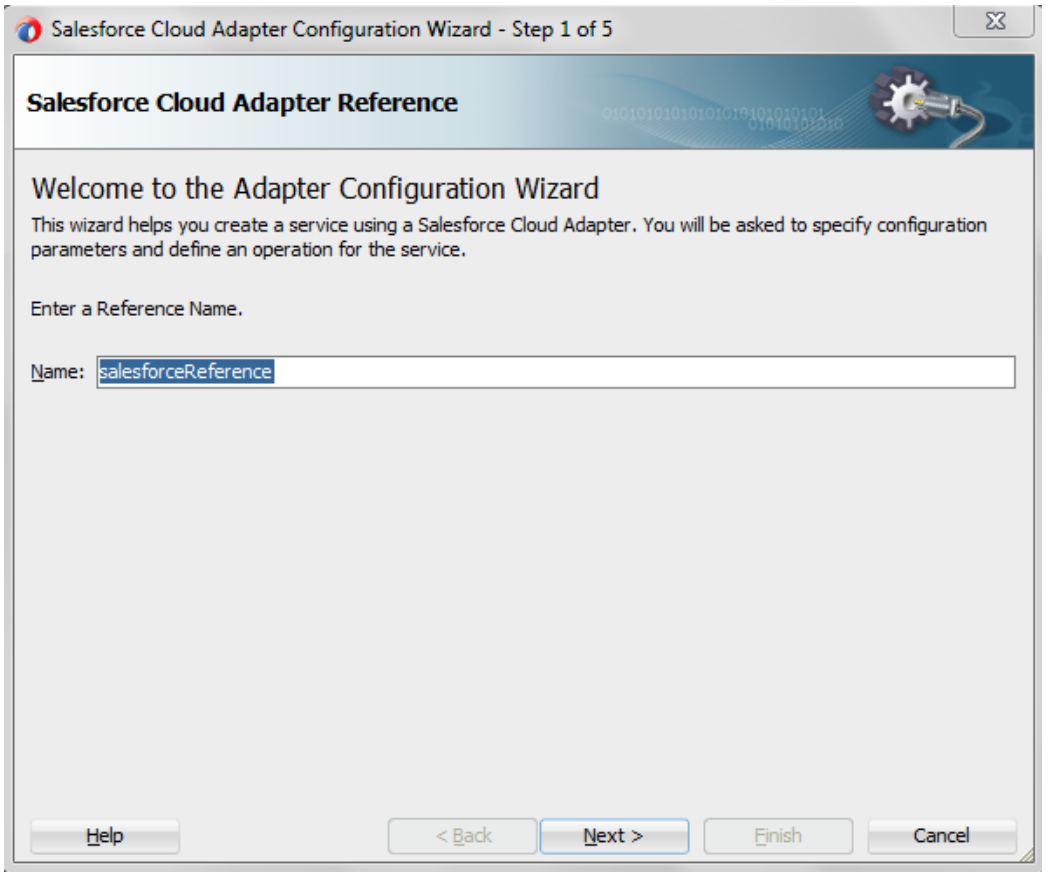
4.2.1 Welcome Page

The first page of Salesforce Cloud Adapter Configuration Wizard is – Welcome page, as shown in Figure 4-2.

Service Adapters enable your Oracle BPEL process or Oracle Mediator Component to interact with database tables, database queues, file systems, FTP servers, Java Message Services (JMS), IBM WebSphere MQ, Oracle applications, or Cloud applications.

Click **Next** to continue or **Cancel** to exit the wizard.

Figure 4- 2 Welcome Page

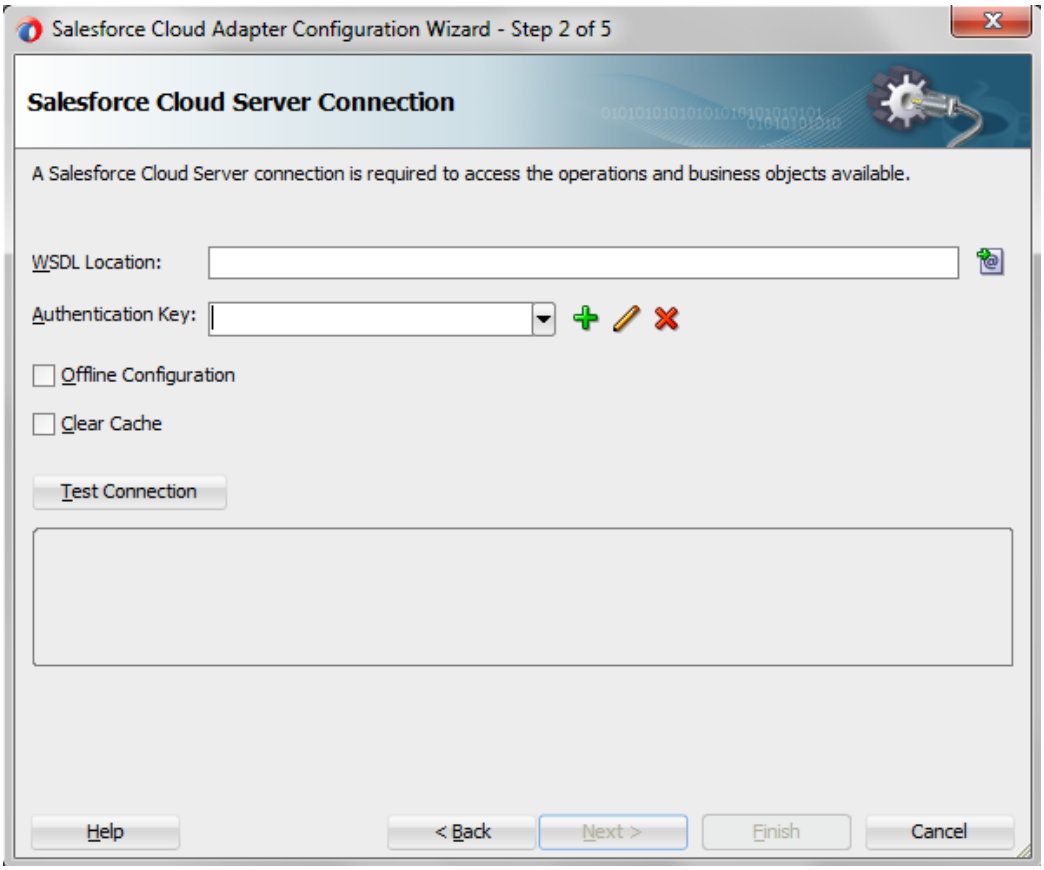


4.2.2 Salesforce Cloud Server Connection Page

The next page of the adapter configuration wizard is – **Salesforce Cloud Server Connection**, as shown in Figure 4-3.

This page enables you to either select an existing connection or create a new connection definition by providing an Authentication Key.

Figure 4- 3 Salesforce Cloud Server Connection Page



The Salesforce Cloud Server Connection page is arranged in the following sections:

- WSDL Location
- Authentication Key
- Offline Configuration
- Clear Cache
- Test Connection

4.2.2.1 WSDL Location

In the text box corresponding to WSDL location, provide the location of your Enterprise WSDL. Using the **Find existing WSDLs** button, browse for the Enterprise WSDL. Make sure you copy the file to your local project folder and use the same while configuring the adapter. Alternatively, you can provide the MDS location of the WSDL in this text box. For more information refer to the section "A.1 Generating the Enterprise WSDL" for WSDL generation. This field is mandatory.

Find existing WSDLs: Click **Find existing WSDLs** button to find/select the existing WSDLs, as shown in Figure 4-4.

Figure 4- 4 WSDL Location Option

4.2.2.2 Authentication Key

Authentication key enables you to specify and store the Authentication credentials in the Credential Store Framework. This field is mandatory. Select the Authentication key, this key is required to retrieve the Salesforce.com login credentials during run-time. It must match the CSF key configured on the WebLogic server. To see the steps to configure the CSF key on WebLogic server, see the section "CSF Key in Enterprise Manager".

Add a new credential: To create a new key, click on plus (+) button, as shown in Figure 4-5.

Figure 4- 5 Create Authentication Key

Once you click on the + icon, the **Add Credential** pop-up window appears, as shown in Figure 4-6.

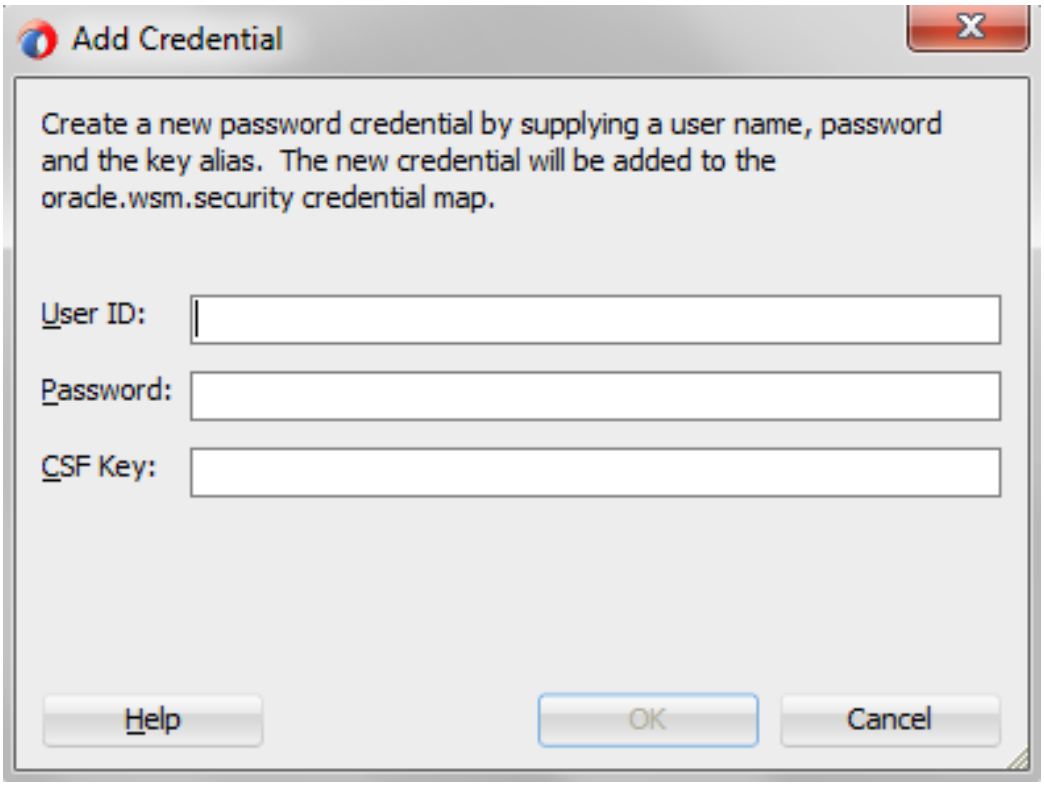
Add Credential / CSF Key Creation

Add Credential page is used to create a new password credential by supplying user name, password and key alias, as shown in Figure 4-6. The new credential will be added to the *oracle.wsm.security* credential map.

On this page the following information is required:

- **User ID:** Enter the Salesforce.com user ID (typically an email address).
- **Password:** Enter the Salesforce.com password + security token.
- **CSF Key:** A user defined (CSF) key which is associated with the specified username and password. Enter the CSF Key of your choice. Credential store framework requires a key which stores the username and password and prevents the user from rewriting the same at run-time. The CSF Key created in configuration wizard should match the CSF key created on Enterprise Manager console. For more information on how to create a CSF Key on Enterprise Manager Console, refer to the section "CSF Key in Enterprise Manager".

Figure 4- 6 Add Credential Page



All these parameters are needed for logging into the Salesforce.com.

Note: The CSF key view/edit functionality is currently not supported. It will be provided in a subsequent release.

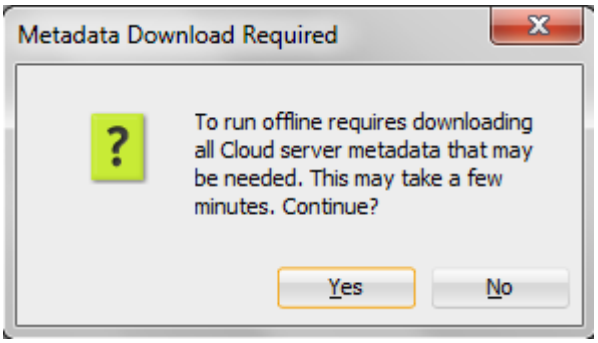
4.2.2.3 Offline Configuration

This option allows user to configure salesforce adapter in offline mode. As soon as user will select the offline configuration check box, wizard will make a call to Salesforce.com to fetch the metadata information about the salesforce objects. This would be stored in the cache which would be used when subsequent adapter instance would be created.

This feature helps in reducing the salesforce operation calls as adapter would be making use of the cache rather than the live salesforce call. In this mode adapter will not support test functionalities which requires direct call to Salesforce.com. Uncheck the same to resume working online.

When you select the **Offline Configuration** checkbox, below message appears, as shown in Figure 4-7.

Figure 4- 7 Metadata Download Required Message



4.2.2.4 Clear Cache

Check/Uncheck the **Clear Cache** checkbox to either refresh or retain the metadata in the cache. If this checkbox is selected, all the metadata in the cache will be refreshed. You should check 'Clear Cache' checkbox in order to get new custom objects (if any) or any other information from the newly generated WSDL. Refer to the section "A.1 Generating the Enterprise WSDL" for generating WSDL from Salesforce.com.

4.2.2.5 Test Connection Functionality

You can use **Test Connection** button to test the connection after creating or selecting the CSF Key. By clicking on **Test Connection** button, you will see the message as shown in Figure 4-8.

- Success! - For successful connection.
- Error! - For any exception during login.

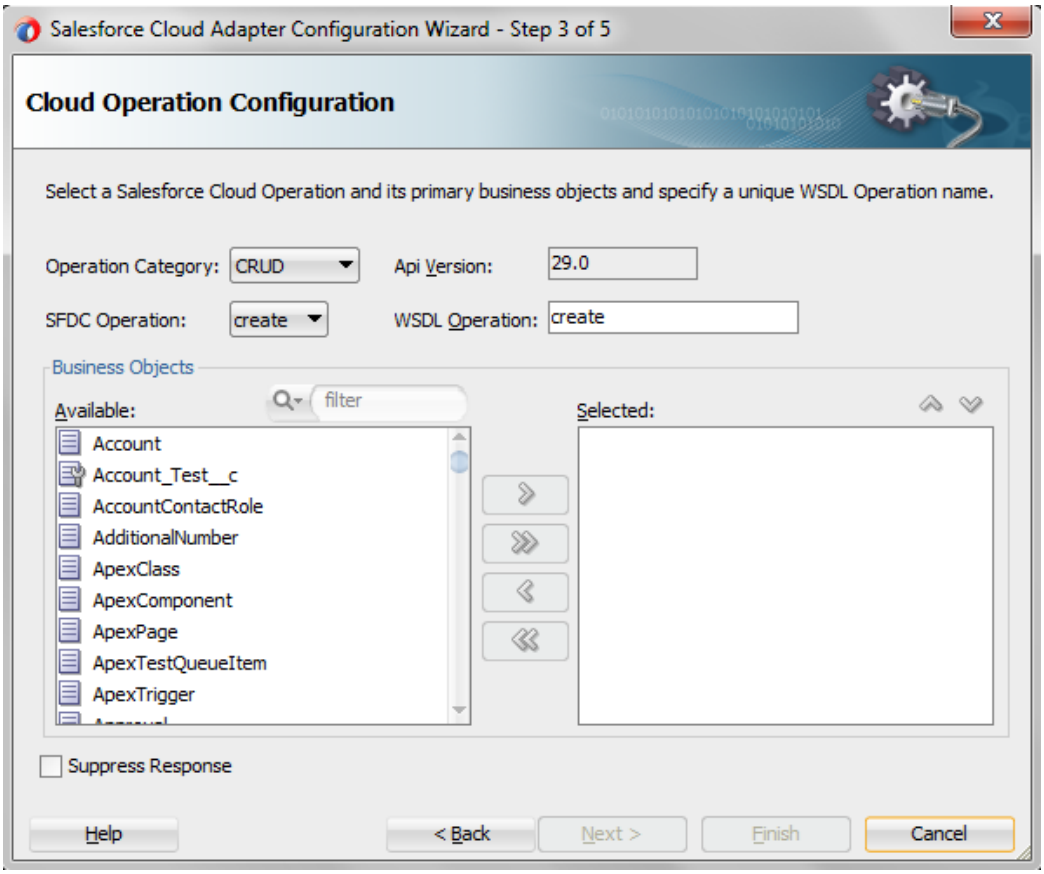
Figure 4- 8 Test Connection Option

Click **Next** to continue or **Cancel** to exit the wizard.

4.2.3 Salesforce.com Cloud Operation Configuration Page

The next page of Salesforce Cloud Adapter Configuration Wizard is – **Cloud Operation Configuration**, as shown in Figure -9.

Figure 4- 9 Cloud Operation Configuration page



The Cloud Operation Configuration page enables you to select operation and objects(s).

The operation page enables you to select from the different kinds of operation categories, and operations. Based on the operation selected, the list of objects will be displayed in the available object list text area. You can select one or multiple objects for a specified operation. You need to follow a logical sequence in selecting objects during design-time because the request structure that the adapter creates depends upon the order of selected objects. For example, if you want to create a lead, opportunity and an account in this order, these objects have to be selected accordingly; with lead on top, followed by opportunity and then account.

The **Cloud Operation Configuration** page is arranged in the following sections:

- Operation Category
- SFDC Operation
- API Version
- WSDL Operation
- Business Objects
- Suppress Response

4.2.3.1 Operation Category

This drop-down list groups a set of related Salesforce.com operations which the user can choose amongst. For more information, refer to the section "Supported SOAP API Operations".

4.2.3.2 SFDC Operation

This includes a number of operations, based on Operation Category selected.

SFDC Operation calls represent specific operations that the Oracle Cloud Adapter for Salesforce.com can invoke at run-time to perform tasks, for example:

- Query data in your organization.
- Add, update, and delete data.

For more information, refer to the section "Supported SOAP API Operations".

4.2.3.3 API Version

The API Version indicates which Salesforce.com WSDL (Web service) version you are using to get all the metadata information for Salesforce.com objects. This API version is read from the endpoint URL in service definition section at the bottom of the Salesforce.com WSDL.

4.2.3.4 WSDL Operation

The WSDL operation is a text field where you can define custom operation name for selected operation. This custom name will then be used in the integrated WSDL, generated after finishing the configuration wizard. If you have not provided any WSDL operation name, the selected operation name, which is the default value of this field, is used.

4.2.3.5 Suppress Response

Check this box if you do not need response during run-time. For example, in "create" operation, the IDs of created records are returned. If you do not need these for further usages then you can use the Suppress Response option.

4.2.3.6 Business Objects:

Available Objects

This area shows all the available Salesforce.com standard and custom objects that can be selected for the particular operation.

Selected Objects

This section shows all the Salesforce.com objects selected by you. The selected operation (create, update, delete, etc.) can be performed only on these objects by your adapter instance.

Filter Field

Filter field is used to search for a Salesforce.com object from the list of available objects.

When you click on the **filter** drop-down, following options are available.

- **Filter:** Select to filter the available objects.
- **Recent:** Select this option if you want to see the objects which have been used recently.

4.2.3.7 SOQL and SOSL Page

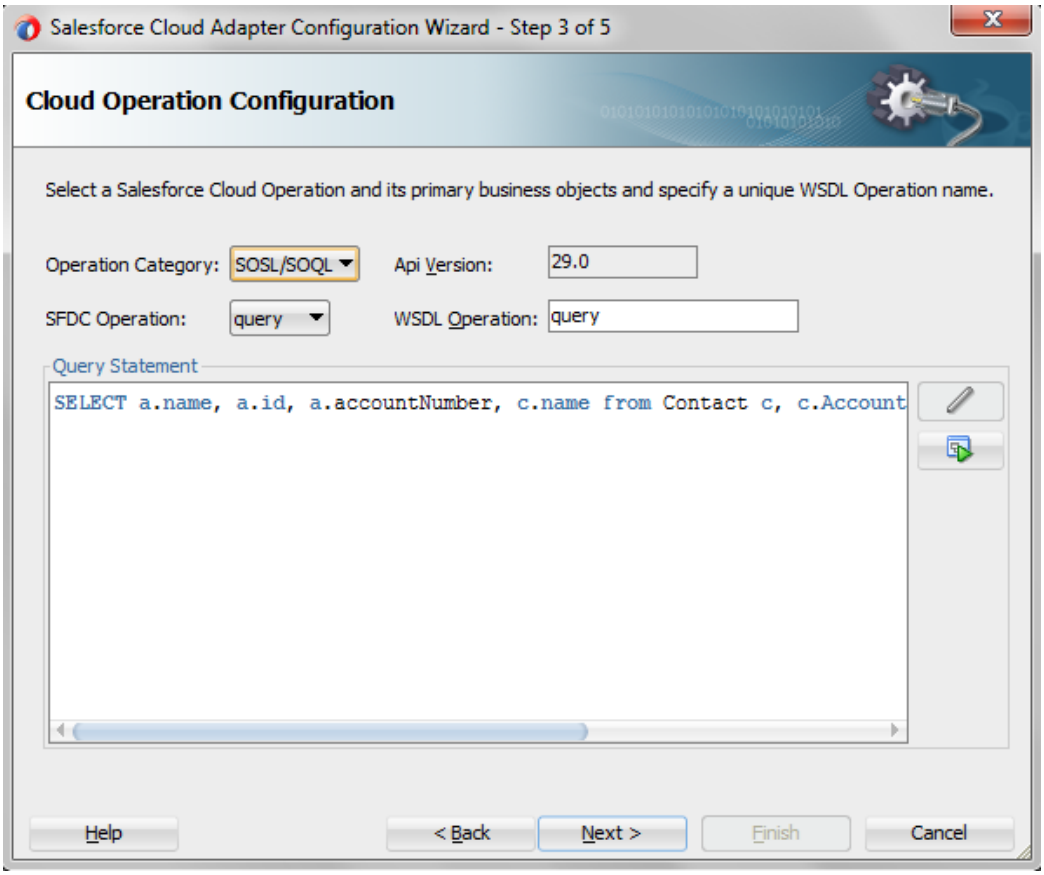
If you select SOSL/SOQL operation category, a query editor will open. Type the SOQL or SOSL statement depending on the operation selected, as shown in Figure 4-10.

SOSL/SOQL enables you to search your organization's Salesforce.com data for specific information.

Query Statement

Query statement is used to query data from Salesforce.com. Query statement text box consists of SOQL statements according to Salesforce.com. Refer to the section "SOSL and SOQL Operations" to understand how the operation works. For detailed information on SOQL/SOSL, follow this link (http://www.salesforce.com/us/developer/docs/soql_sosl/)

Figure 4- 10 Query Statement in SOQL and SOSL Option



Run Query Test Tool

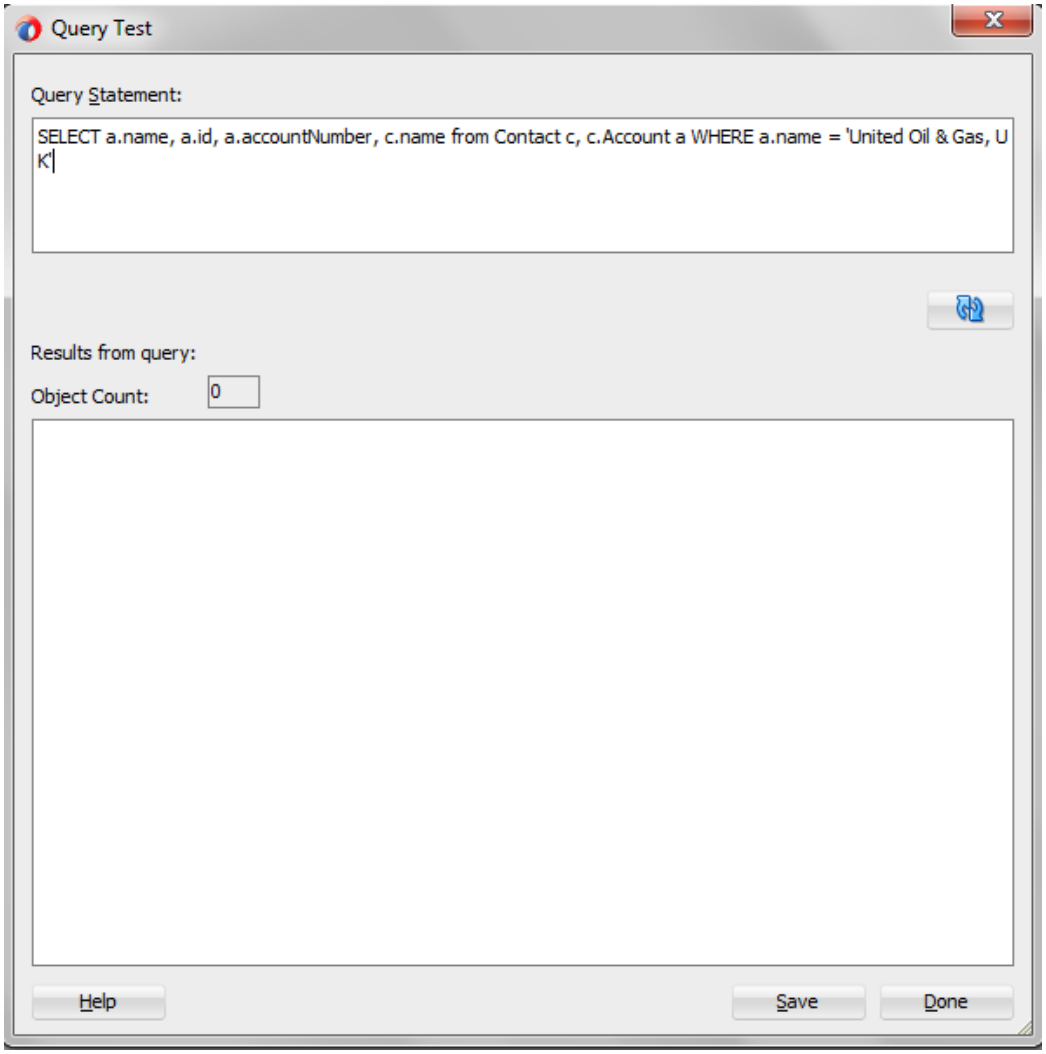
This tool helps you to test and verify the query. By clicking on **Test** icon, a test dialogue box appears.

The Query Test dialogue contains the following area:

Query Statement text box and **Results** box showing result of query executed. Result box can show results up to a maximum of 200 records. If the query has one or more bind parameters in it, then one or more input boxes will appear to provide values for these, as shown in Figure 4-11.

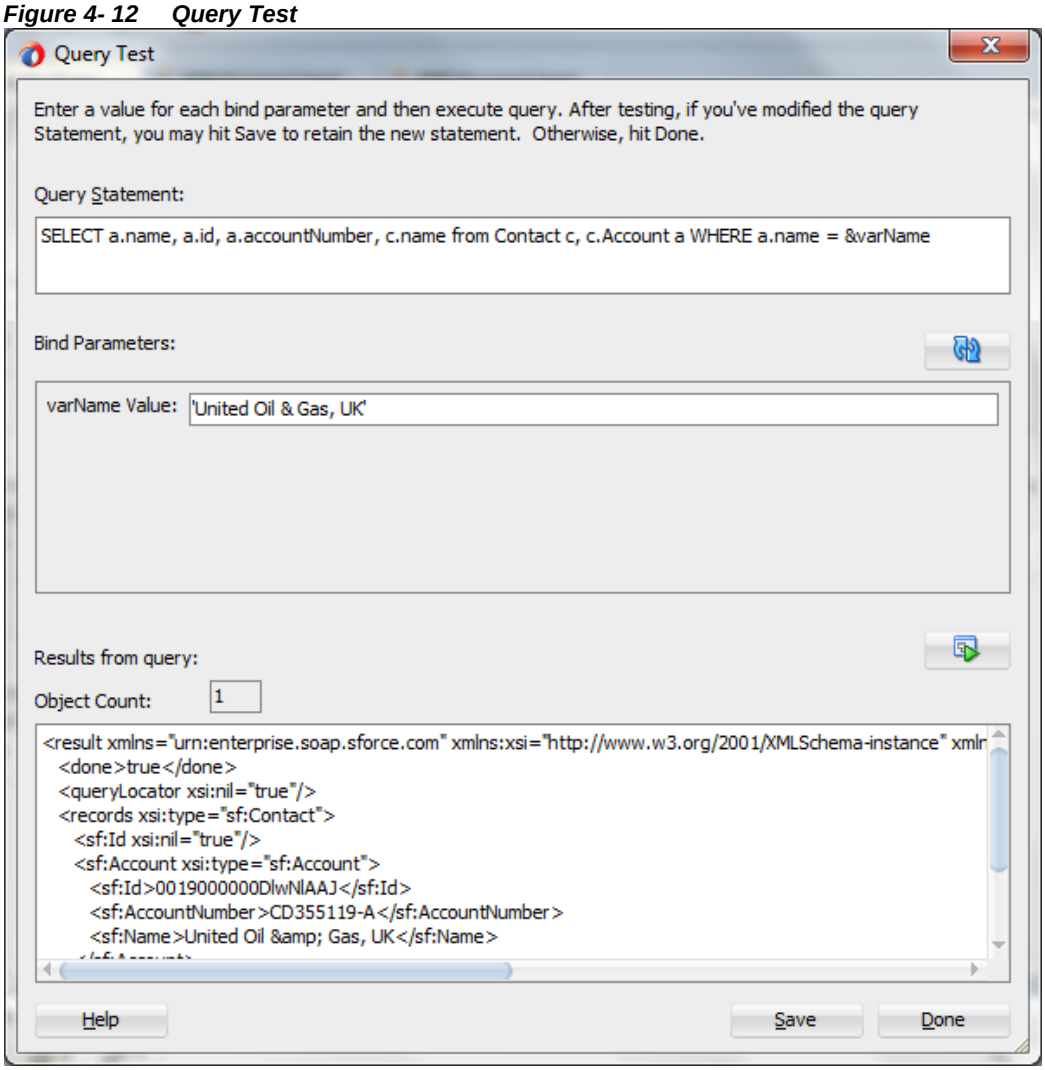
Execute Query: Execute Query button is used to execute the query.

Figure 4- 11 Query Test



Refresh Bind Parameters: Refresh Bind Parameters button is used to refresh Bind Parameters box, if there is any change in the number of bind parameters in the query statement. The screenshot below illustrates the usage of 'Bind Parameters'

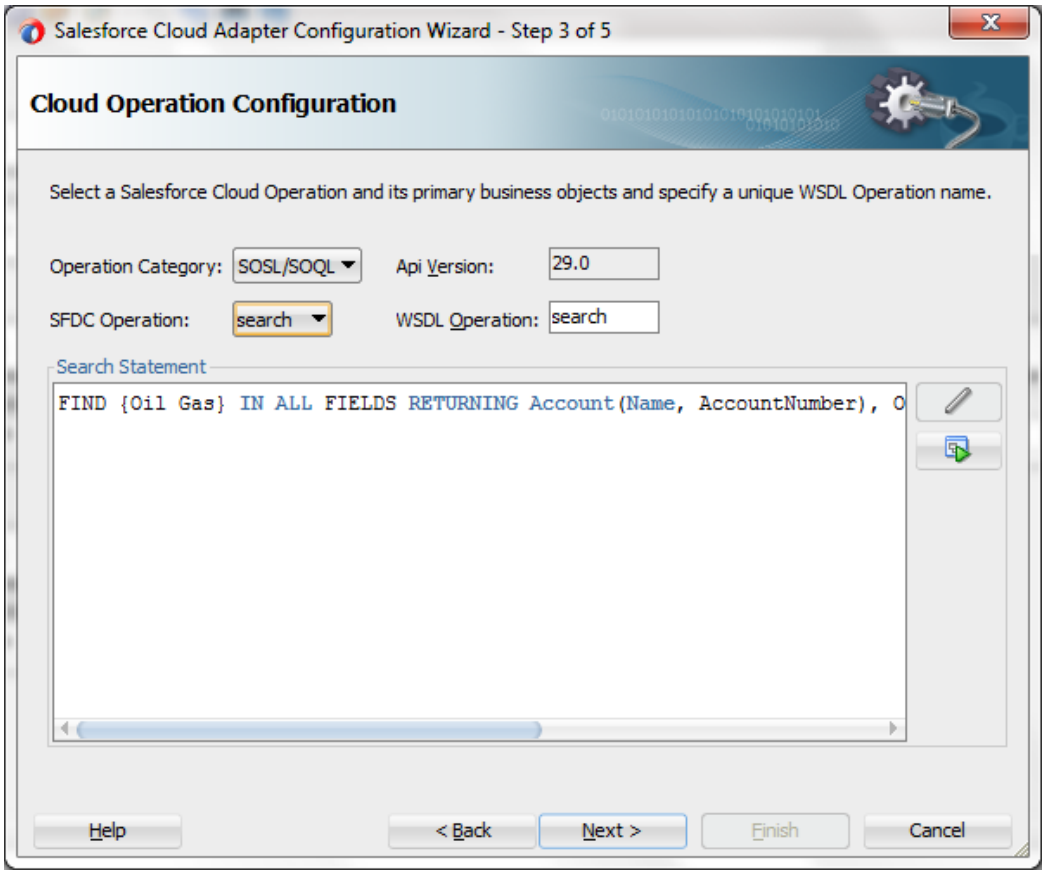
Figure 4- 12 Query Test



Search Statement

Salesforce object Search Statement (SOSL) is used to search specific data for your organization from Salesforce.com. It retrieves records for one or more objects. Analogous to the options described in previous section, Search statement also has buttons as provision for - Run Search test Tool, Execute Search and Refresh Bind Parameters.

Figure 4- 13 Cloud Operation Configuration



4.2.4 Header and Properties Page

The next page in Salesforce Cloud Adapter Configuration Wizard is – **Header and Properties** page, as shown in Figure 4-14.

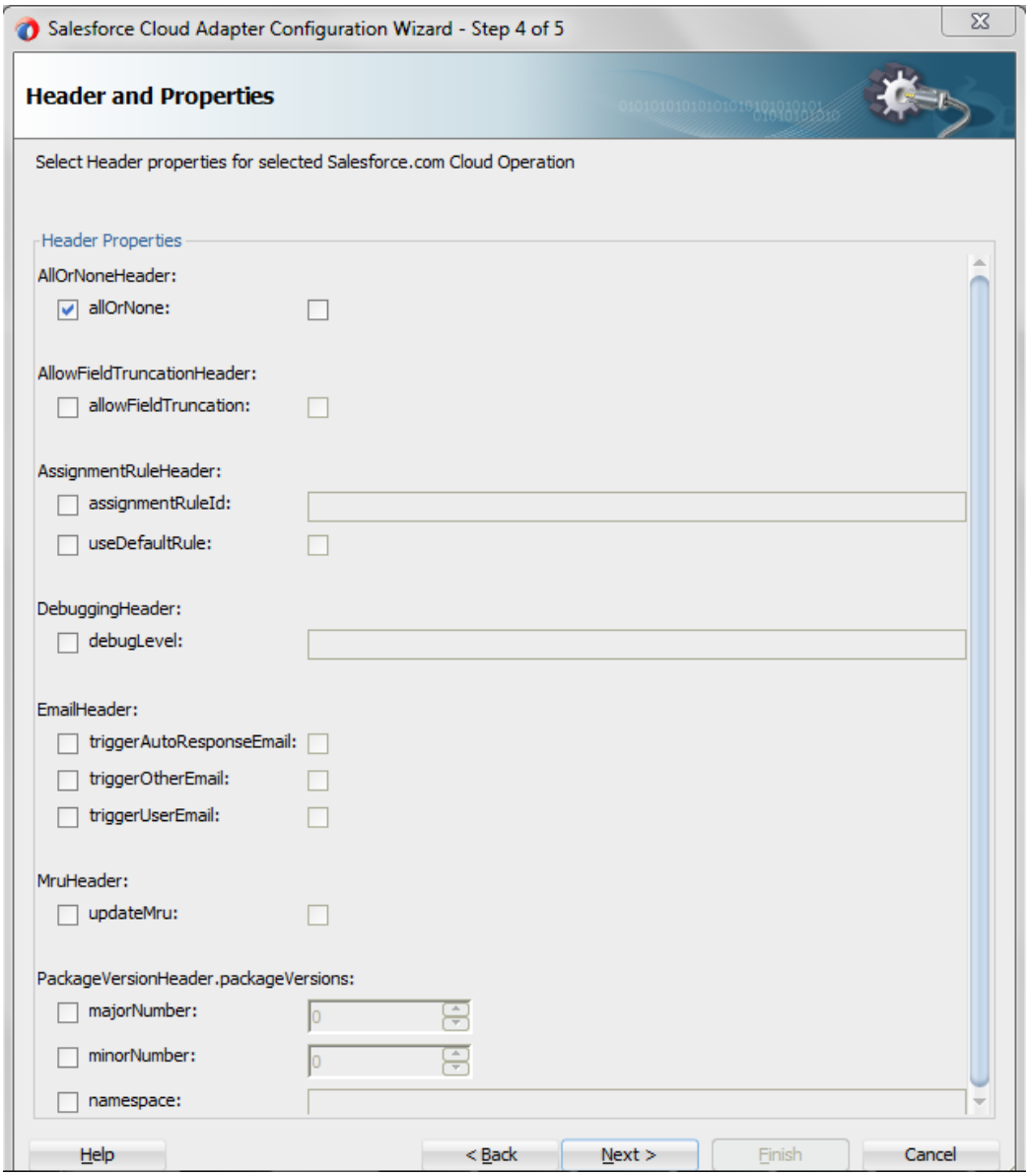
The **Header and Properties** page is used to select header properties for the selected Oracle Cloud Adapter for Salesforce.com Cloud Operation. The values defined in this page can be overridden by properties defined at the composite level or in the EM console. For a detailed description of run-time properties, see section “

Oracle Cloud Adapter for Salesforce.com Run-Time Properties”.

For more information on Salesforce.com headers, follow the link (http://www.salesforce.com/us/developer/docs/api/Content/soap_headers.htm).

For operation specific header information, refer to the section “Salesforce.com SOAP Headers”.

Figure 4- 14 Header and Properties Page (for create operation)



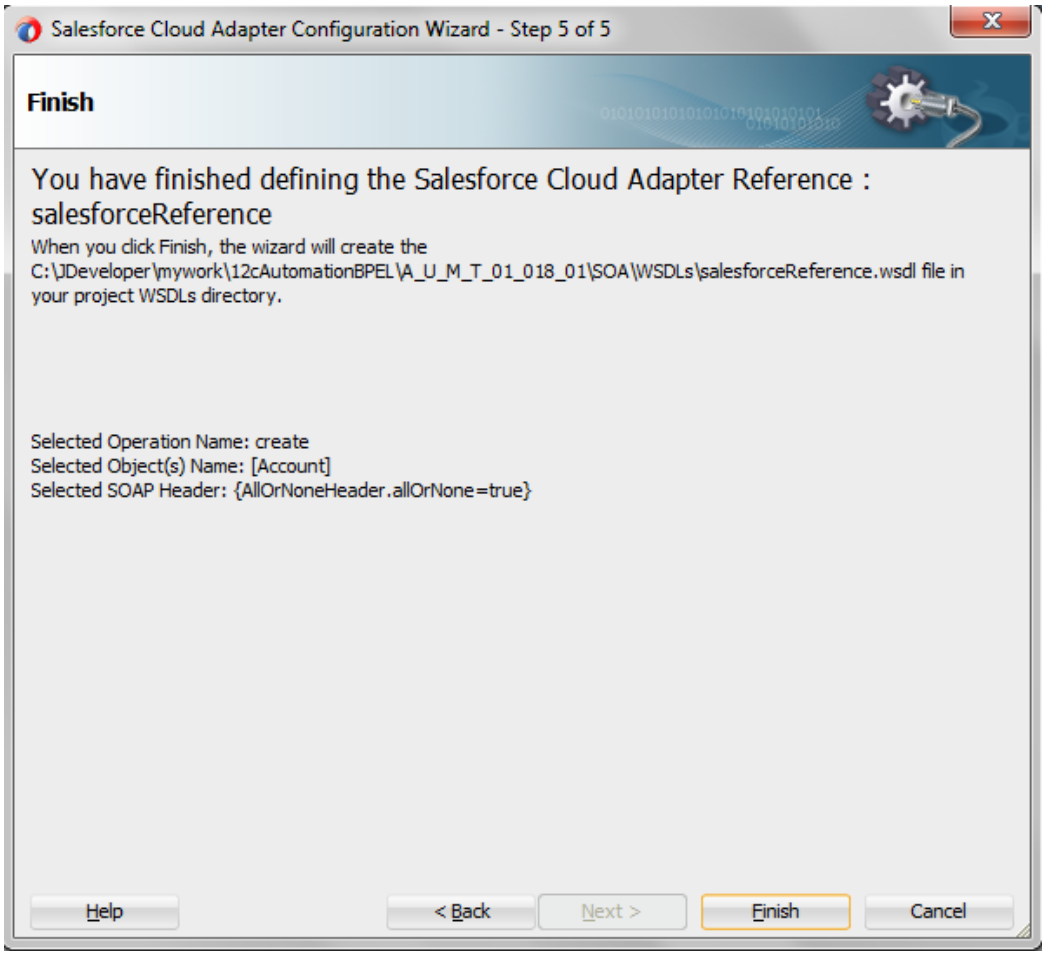
4.2.5 Finish Page

The next page in Salesforce Cloud Adapter Configuration Wizard is – **Finish** page. The **Finish** page summarizes the Oracle Cloud Adapter for Salesforce configuration..

When you complete the adapter configuration, a WSDL file named after the service name, you entered on the **Adapter Configuration Wizard - Service Name** page, appears in the **Application Navigator**, as shown Figure 4-15.

To finish adapter configuration, Click **Finish**.

Figure 4- 15 Finish Page



4.3 Design-time Artifact Generation

After clicking on Finish button on the last screen of Salesforce Cloud Adapter Configuration Wizard, composite.xml gets updated by adding a reference element along with jca file and integration WSDL file.

4.3.1 JCA File

The JCA file provides adapter configuration information for the service. A connection factory is specified so that the adapter run-time can connect to the Salesforce Cloud Servers, as shown in Figure 4-16.

Figure 4- 16 Sample JCA File contents for delete operation

```
<adapter-config name="samplePortType" adapter="Salesforce"
    wsdlLocation="../Projectl/sample.wsdl"
    xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
  <connection-factory location="cloud/CloudAdapter">
    <non-managed-connection managedConnectionFactoryClassName="oracle.cloud.connector.salesforce.SalesforceConnectionFactory">
      <property name="targetWSDLURL" value="EnterpriseWSDl_28.wsdl"/>
      <property name="csfkey" value="csfKey"/>
      <property name="csfMap" value="SOA"/>
      <property name="applicationVersion" value="28.0"/>
    </non-managed-connection>
  </connection-factory>
  <endpoint-interaction portType="samplePortType" operation="delete">
    <interaction-spec className="oracle.tip.adapter.cloud.CloudInteractionSpec">
      <property name="targetOperation" value="delete"/>
      <property name="operationPath" value=""/>
      <property name="oracle.cloud.rt.sfdcAdapterNamespace"
      value="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Applicationl/Projectl/sample"/>
      <property name="selectedObjects" value="Account"/>
      <property name="AllOrNoneHeader.allOrNone" value="true"/>
      <property name="oracle.cloud.rt.suppressResponse" value="false"/>
    </interaction-spec>
  </endpoint-interaction>
</adapter-config>
```

4.3.2 Integration WSDL

Integration WSDL is a simplified, abstract WSDL file as compared to the actual Salesforce WSDL. It has information in reference to selected operation and objects during configuration of the Oracle Cloud Adapter for Salesforce.com. Figure 4-17 and Figure 4-18 shows parts of the Sample Integration WSDL generated by Salesforce Adapter for delete operation.

Figure 4- 17 Integration WSDL part showing delete operation request and response schema

```
<xs:schema xmlns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Applicationl/Projectl/sample"
     elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema"
     targetNamespace="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Applicationl/Projectl/sample"
     xmlns:ns2="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Applicationl/Projectl/sample"
     xmlns:ns4="urn:enterprise.soap.sforce.com">
    <xs:import namespace="urn:enterprise.soap.sforce.com"/>
    <xs:element name="delete" type="ns2:deleteType"/>
    <xs:complexType name="deleteType">
        <xs:sequence>
            <xs:element name="ids" type="xs:ID"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="deleteResponse" type="ns2:deleteResponseType"/>
    <xs:complexType name="deleteResponseType">
        <xs:sequence>
            <xs:element name="DeleteResult" type="ns4:DeleteResult"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

Figure 4- 18 Integration WSDL part displaying delete operation details

```
<wsdl:message name="deleteRequestMessage">
    <wsdl:part name="parameters" element="nsl:delete"/>
</wsdl:message>
<wsdl:message name="deleteResponseMessage">
    <wsdl:part name="parameters" element="nsl:deleteResponse"/>
</wsdl:message>
<wsdl:message name="UnexpectedErrorFault">
    <wsdl:part name="parameters" element="fns:UnexpectedErrorFault"/>
</wsdl:message>
<wsdl:portType name="samplePortType">
    <wsdl:operation name="delete">
        <cloud:CloudOperation xmlns:cloud="http://xml.oracle.com/types" targetOperation="delete"/>
        <wsdl:input message="nsl:deleteRequestMessage"/>
        <wsdl:output message="nsl:deleteResponseMessage"/>
        <wsdl:fault name="UnexpectedErrorFault" message="nsl:UnexpectedErrorFault"/>
    </wsdl:operation>
</wsdl:portType>
```

5

Integration with Different Service Components (BPEL/Mediator) in Oracle SOA Suite

Oracle Cloud Adapter for Salesforce.com facilitates integration to various Salesforce.com objects via operations exposed by the Salesforce.com API. This chapter walks you through a simple integration wherein user creates a new account on Salesforce.com using Oracle Cloud Adapter for Salesforce.com. The scenario would enable you to create an easy and seamless integration to Salesforce.com using BPEL Process Manager.

This section describes Oracle Cloud Adapter for Salesforce.com concepts through a use case, which is a complete walkthrough of the Adapter configuration wizard. In addition, this use case also describes how by using the Adapter configuration wizard, you can access various operations available to your Enterprise on Salesforce.com, select objects specific to your business requirements, generate corresponding WSDL to expose the necessary operations etc. These services are consumed to define partner links that are used in the BPEL process. You use the Adapter configuration wizard to both create and edit adapter services.

This chapter contains the following topics:

- Section 5.1, "Overview"
- Section 5.2, "Configuring the CSF Key on Enterprise Manager Console"
- Section 5.3, "Designing a Composite for Service Integration"
- Section 5.4, "Configure Oracle Cloud Adapter for Salesforce.com"
- Section 5.5, " Integration with BPEL"
- Section 5.6, "Deploy the Composite"
- Section 5.7, "Test the Composite"

5.1 Overview

Account is one of the various standard objects provided by Salesforce.com. An account on Salesforce.com represents an individual or business associated with the business. Oracle Cloud Adapter for Salesforce.com allows you to create an account on Salesforce.com. The below scenario would provide you step by step instructions to accomplish the same.

To integrate with Oracle BPEL Process Manager, the organization's Enterprise WSDL should be available to the user. The underlying adapter services must be exposed as WSDL files, which are generated during design-time in the configuration wizard of the Oracle Cloud Adapter for Salesforce.com. For more information, refer to the section "A.1 Generating the Enterprise WSDL".

The generated WSDL files are used to design the appropriate BPEL processes for outbound adapter services. A completed BPEL process must be successfully compiled in JDeveloper and deployed to an SOA Suite server. Upon deployment to an SOA Suite server, every newly deployed process is viewable in the Oracle Enterprise Manager console, where you can run, monitor, administer BPEL processes, and monitor adapter events.

5.2 Configuring the CSF Key on Enterprise Manager Console

For steps to configure the CSF key on Enterprise Manager Console, refer to the section "CSF Key in Enterprise Manager".

5.3 Designing a Composite for Service Integration

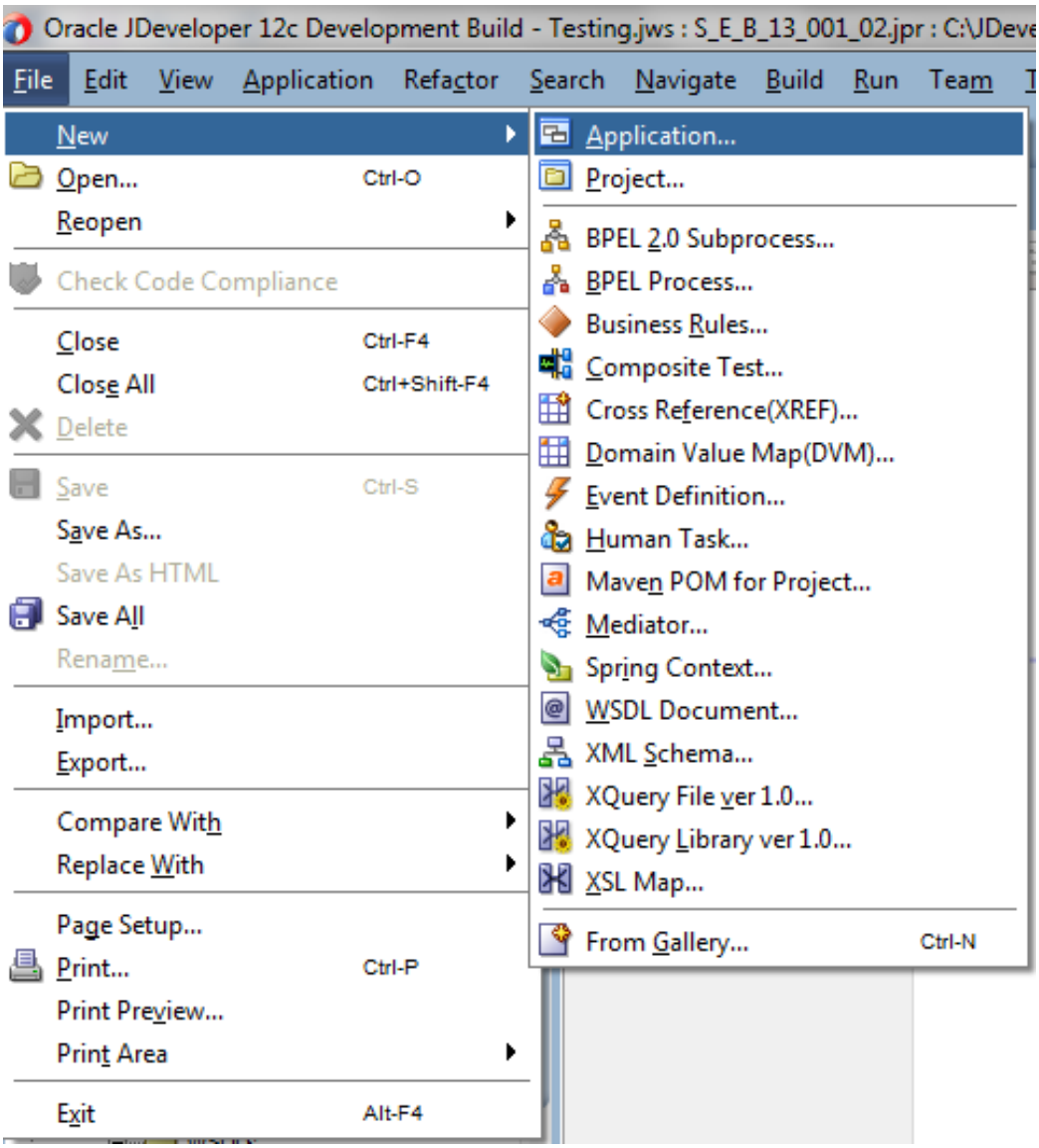
The steps mentioned below design a composite using the Oracle JDeveloper Studio Edition (12.1.3.0.0), which is a comprehensive tool for developing Oracle Fusion Middleware applications. It provides an easy-to-use Integrated Development Environment that let you create your integrations efficiently.

5.3.1 Define Composite for BPEL and Mediator

Perform the following steps to define a composite for BPEL and Mediator:

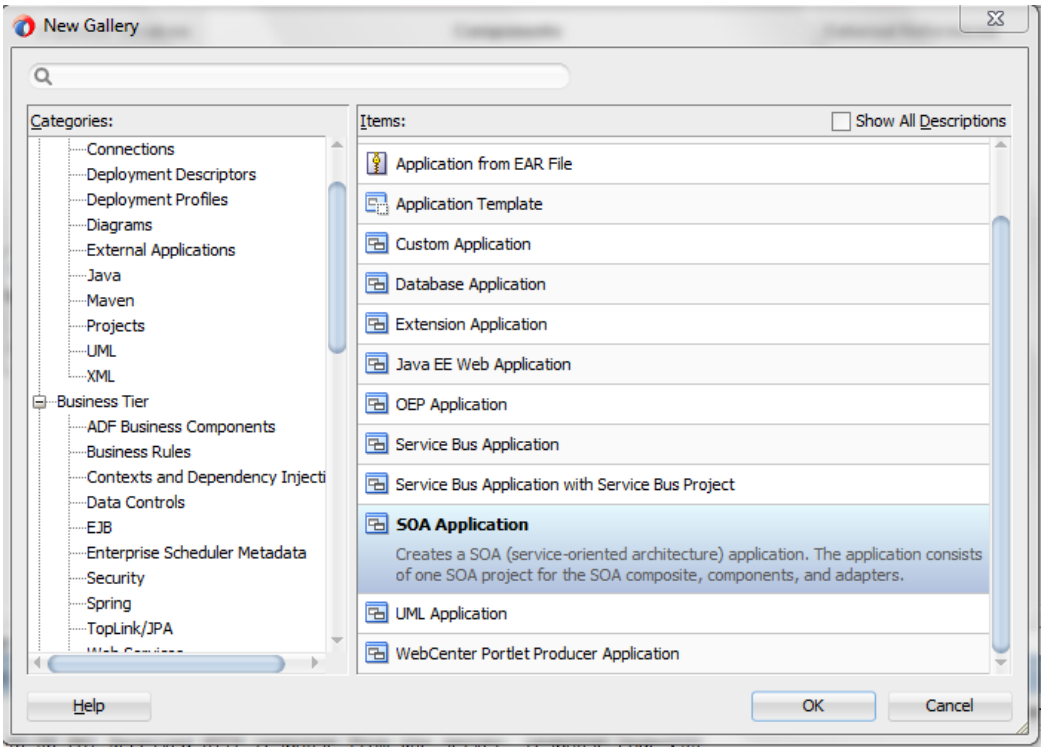
1. In the **File** menu of JDeveloper, click **New** and select **Application**.

Figure 5- 1 Navigation Window



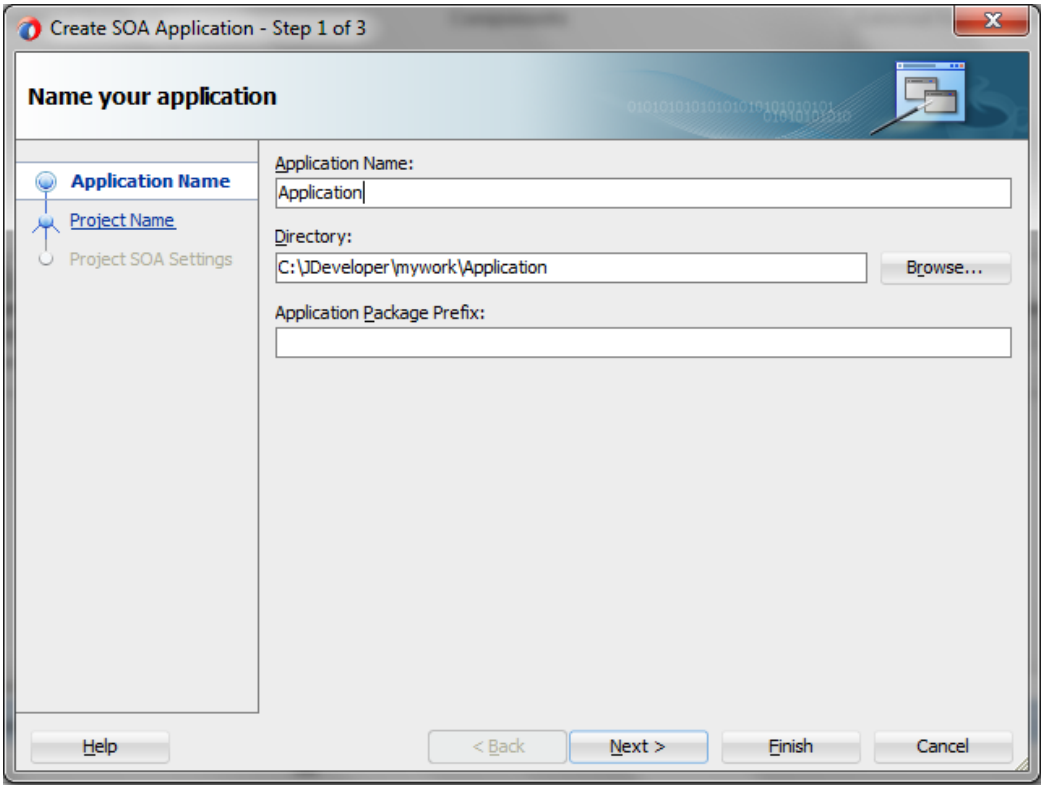
2. The **New Gallery** page is displayed. Select **SOA Application** from the **Items** list, as shown in Figure 5-2.

Figure 5- 2 Create SOA Application



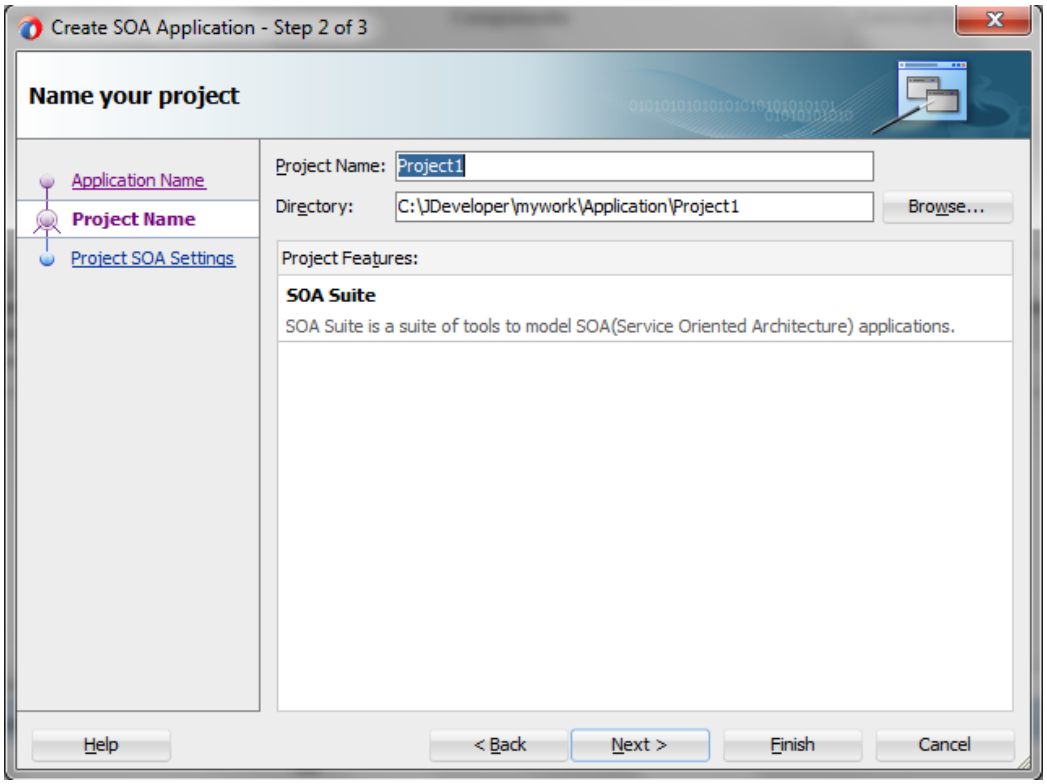
3. Provide a suitable name to your application, as shown in Figure 5-3.

Figure 5- 3 Name your application



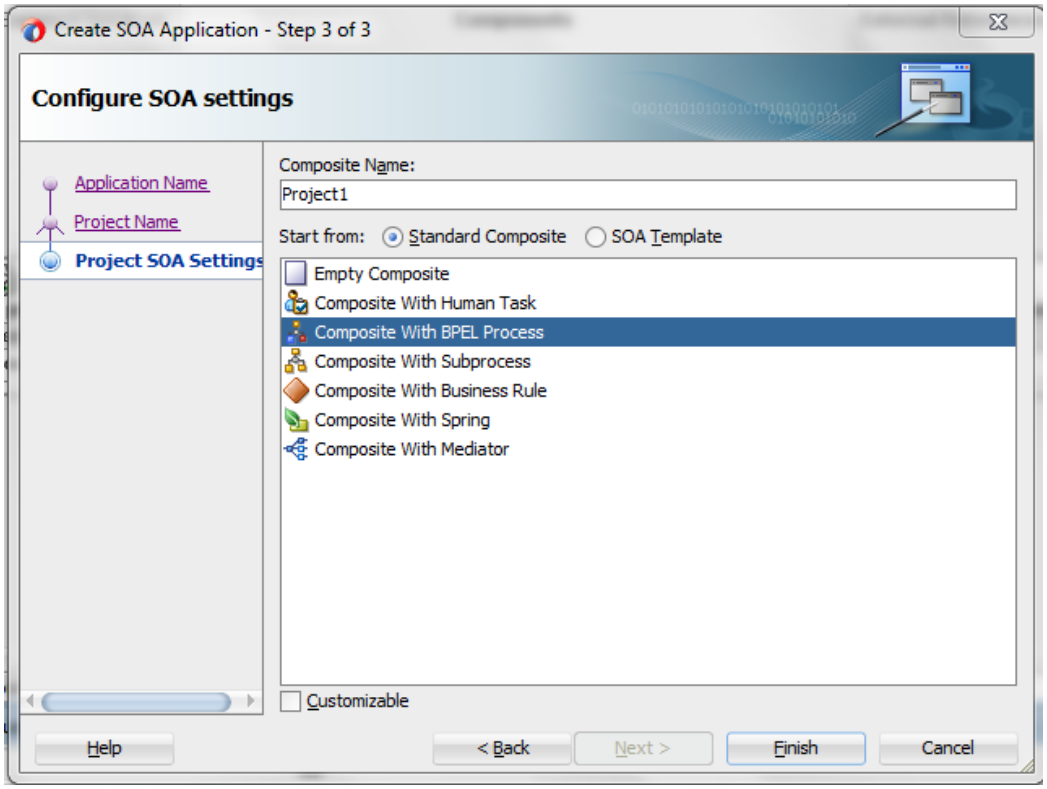
4. Click **Next** and provide a suitable name to your project, as shown in Figure 5-4.

Figure 5- 4 Name your project



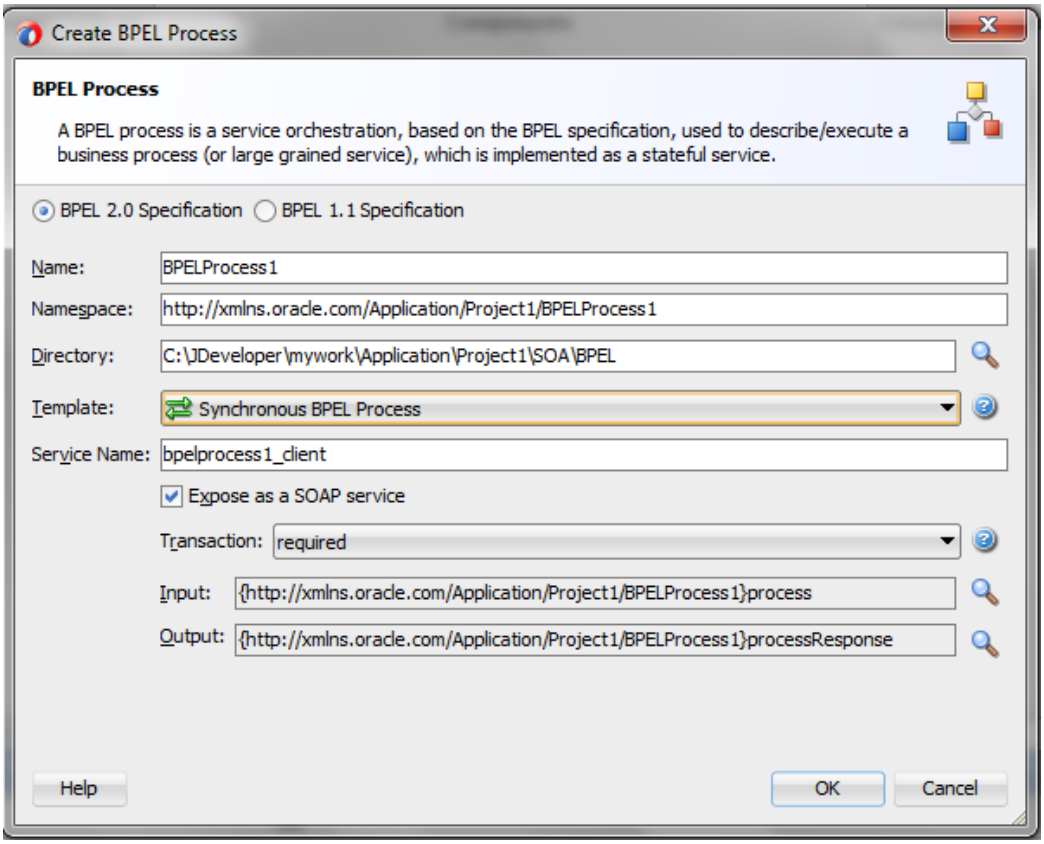
5. Click **Next.**
6. Select **Composite with BPEL Process** from **Standard Composite** list, as shown in Figure 5-5.

Figure 5- 5 Configure SOA Setting



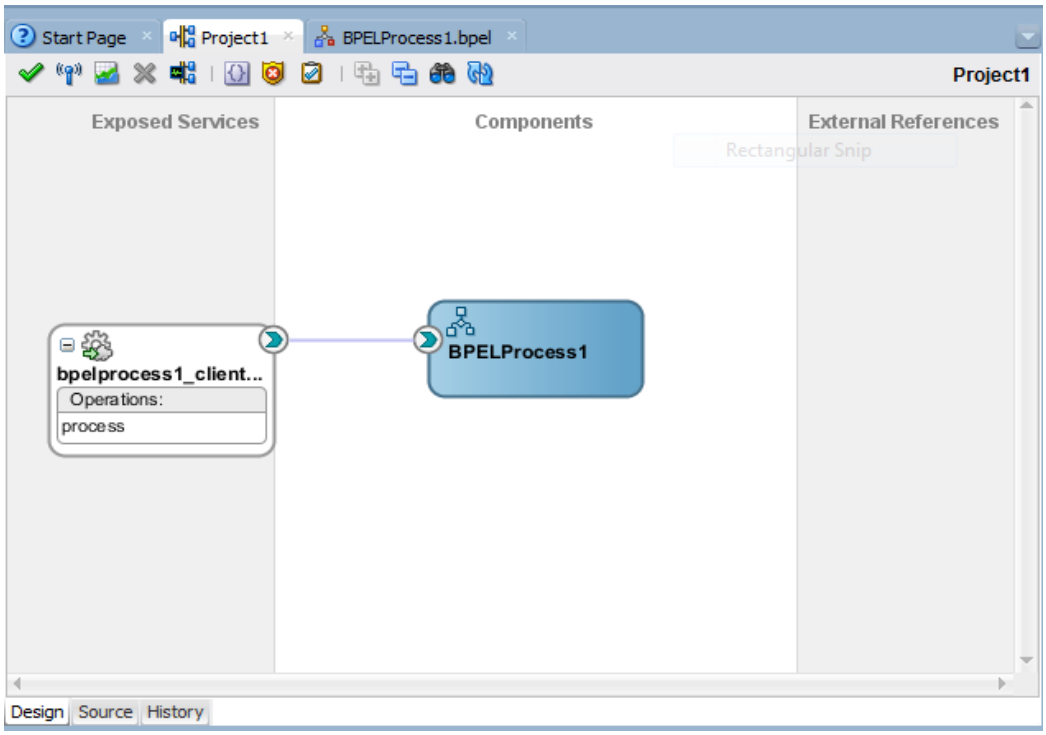
7. Click **Finish**.
8. Select the **Synchronous BPEL Process** from **Template** drop-down and click **OK**, as shown in Figure 5-6.

Figure 5- 6 Create BPEL Process



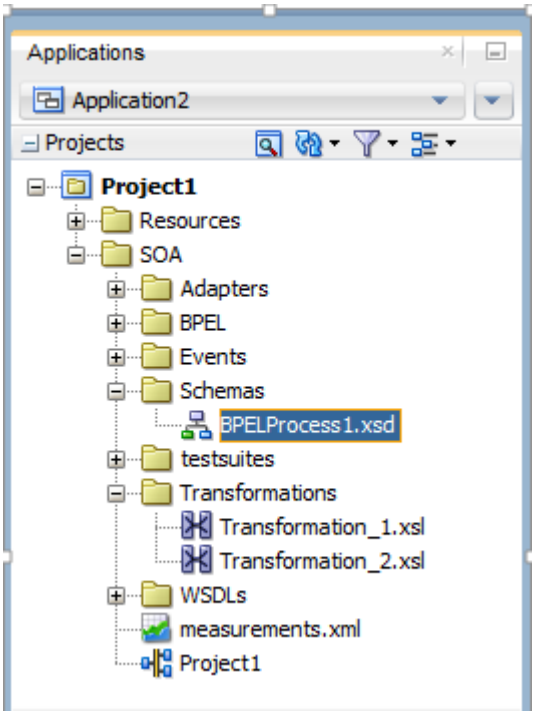
9. The composite.xml looks as shown in Figure 5-7.

Figure 5- 7 Composite.xml



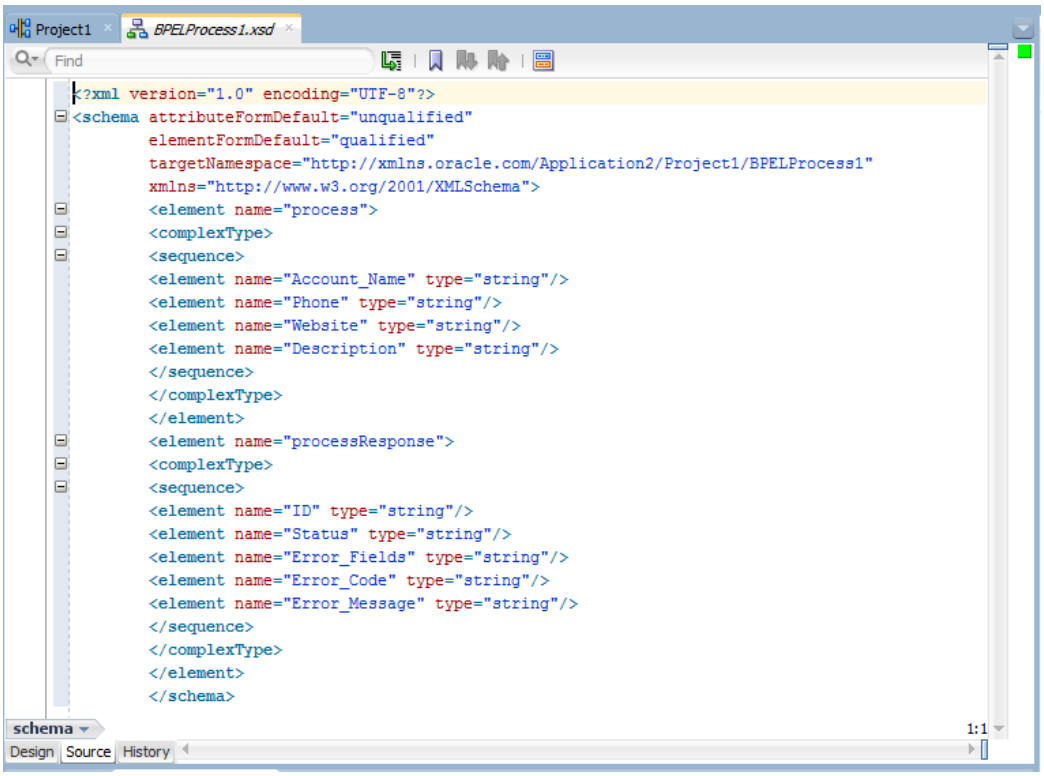
10. In the **Application Navigator**, under the **Schemas** folder, an XML schema file with the name **BPELProcess1.xsd** is displayed after BPEL process creation, as shown in Figure 5-8.

Figure 5- 8 XML Schema File



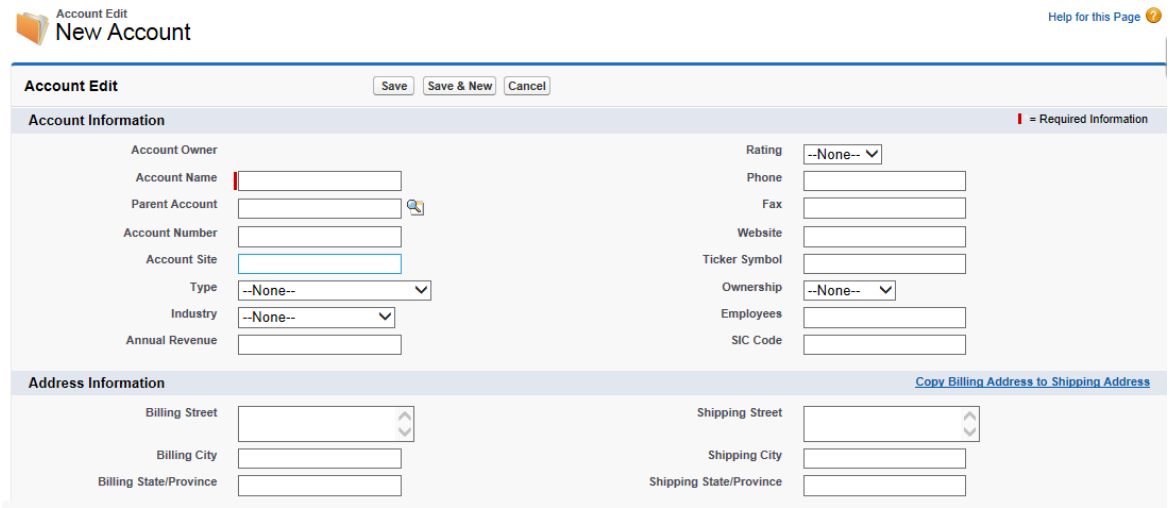
11. Edit this schema file as per your business requirement. It is the responsibility of the front-end application to enforce the data validations and to ensure that the input sent to SFDC via SOA is error free. The structure of the schema used in this use case is as shown in Figure 5-9.

Figure 5- 9 Edit XML Schema File



12. Figure 5-10 shows **Create Account** page on Salesforce.com. The fields with red mark are mandatory fields. This structure may vary for different organizations.

Figure 5- 10 Create Account Page on Salesforce.com

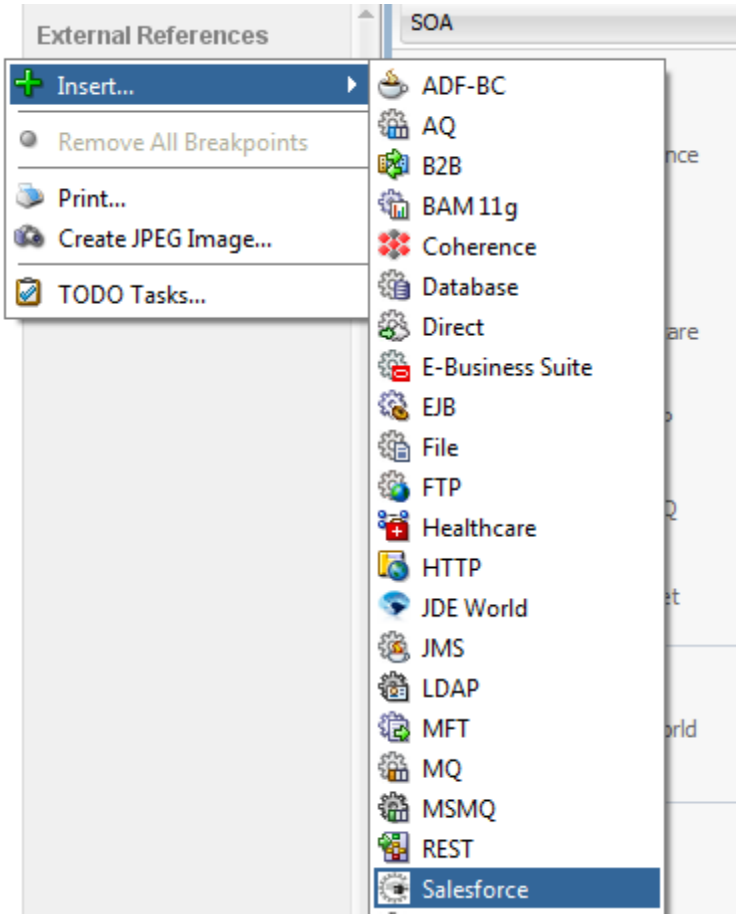


5.4 Configure Oracle Cloud Adapter for Salesforce.com

Perform the following steps to configure a New Oracle Cloud Adapter for Salesforce.com:

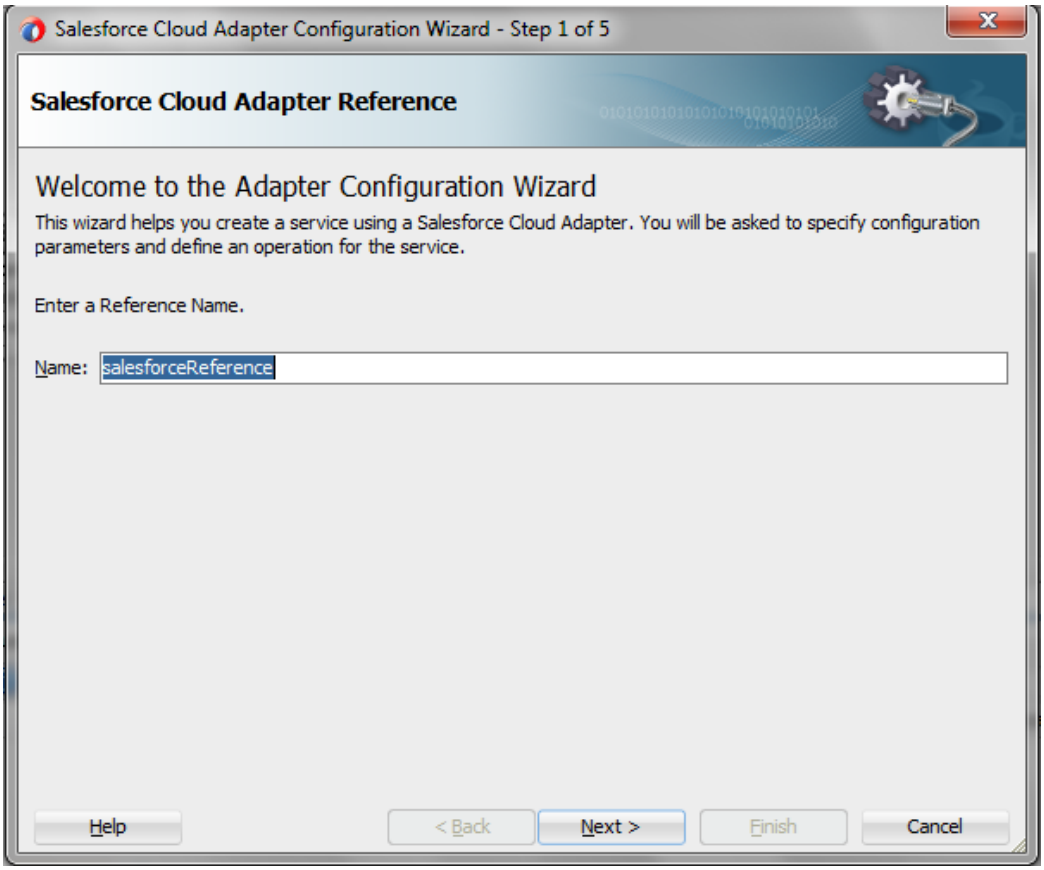
1. In **External References** swim lane of the composite.xml file, right-click and select **Salesforce** adapter, as shown in Figure 5-11.

Figure 5- 11 Salesforce Adapter



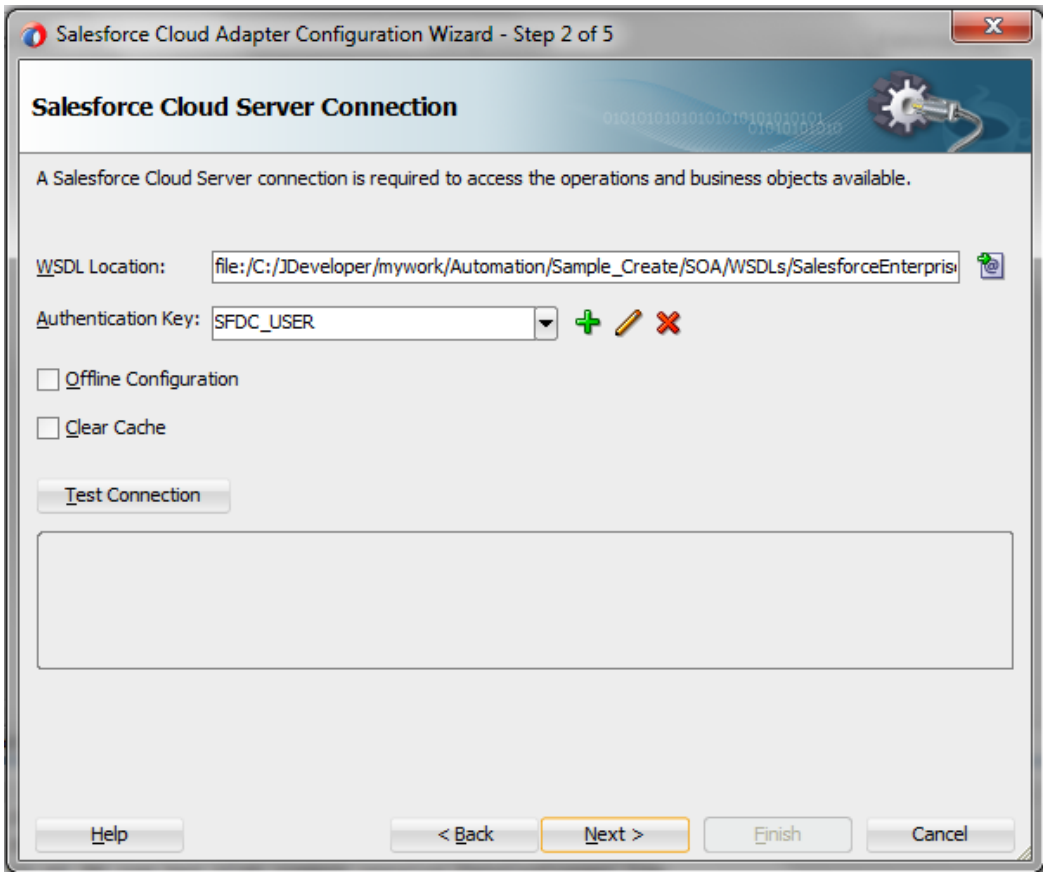
2. The Salesforce Cloud Adapter Configuration Wizard - Welcome page is displayed, as shown in Figure 5-12.

Figure 5- 12 Welcome Page



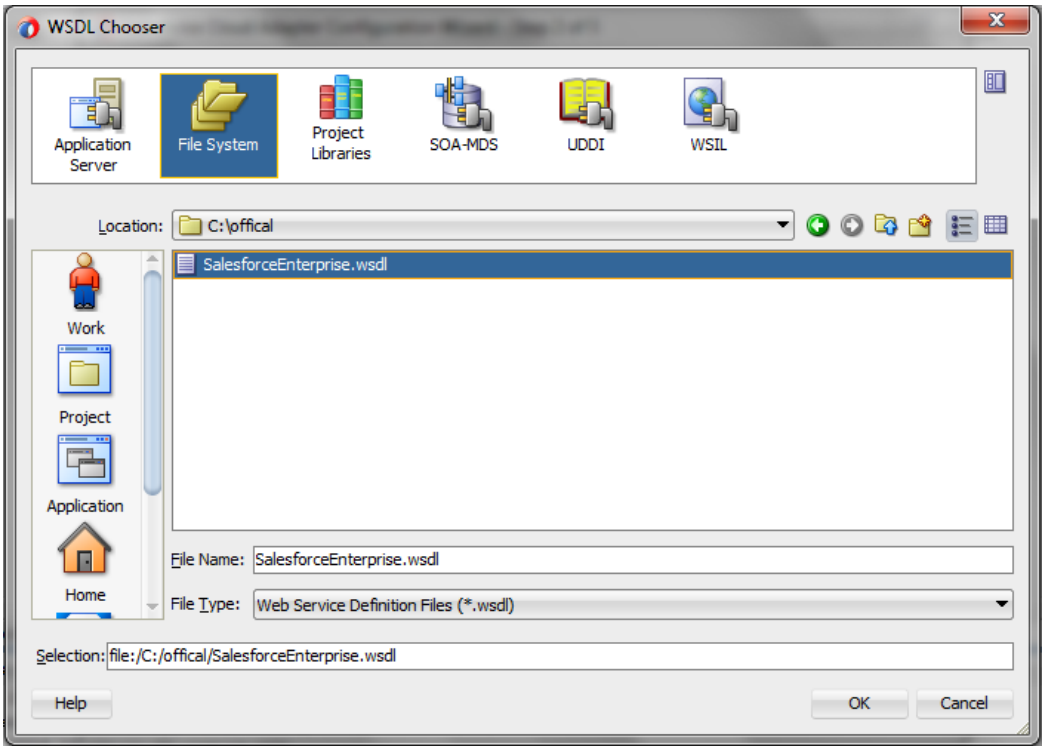
3. Click **Next**.
4. The **Salesforce Cloud Server Connection** page is displayed. The **WSDL Location** and **Authentication Key** text boxes are already populated. It picks up these values from the cache. You can re-enter these values. If you want to use a different value, click the **Find existing WSDLs** icon, which is located to the right of the **WSDL Location** field, as shown in Figure 5-13.

Figure 5- 13 Salesforce Cloud Server Connection Page



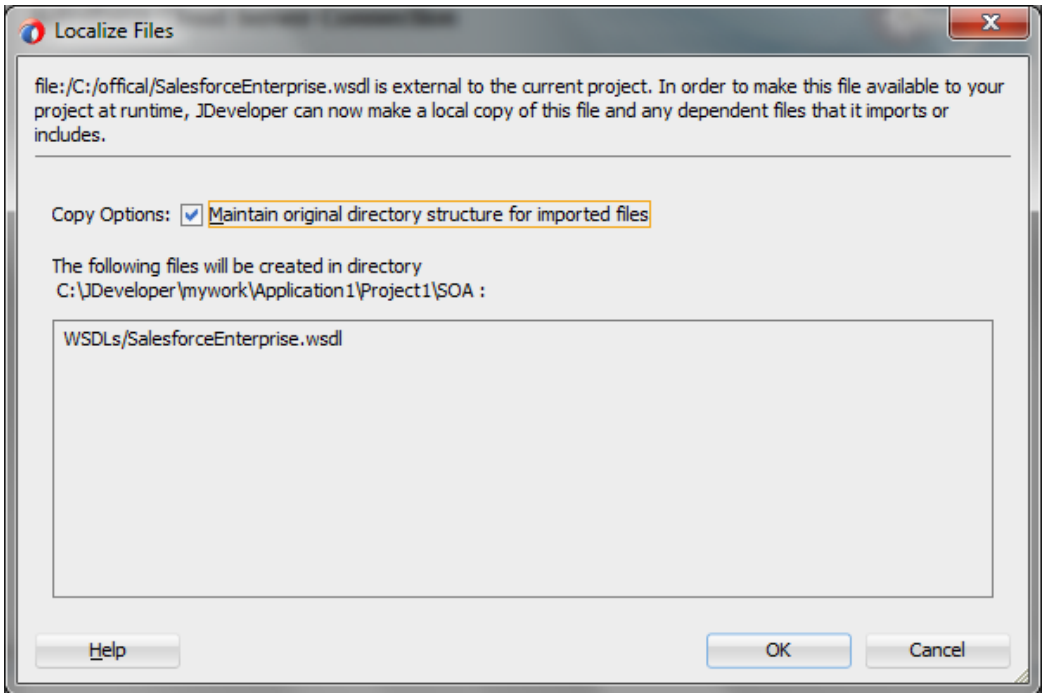
5. The **WSDL Chooser** dialog is displayed, browse and select the downloaded Enterprise WSDL and click **OK**, as shown in Figure 5-14.

Figure 5- 14 SOA Resource Browser



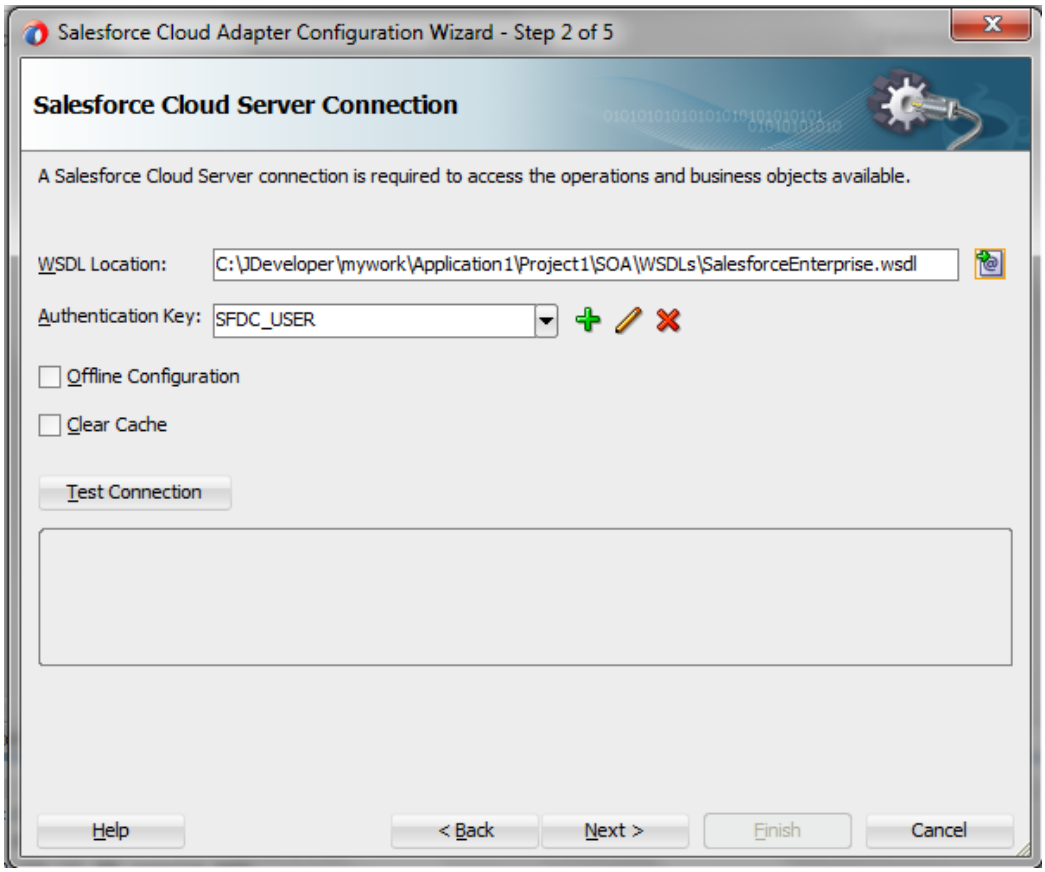
6. Click **OK**. The following screen appears as shown in Figure 5-15.

Figure 5- 15 Localize Files Dialog



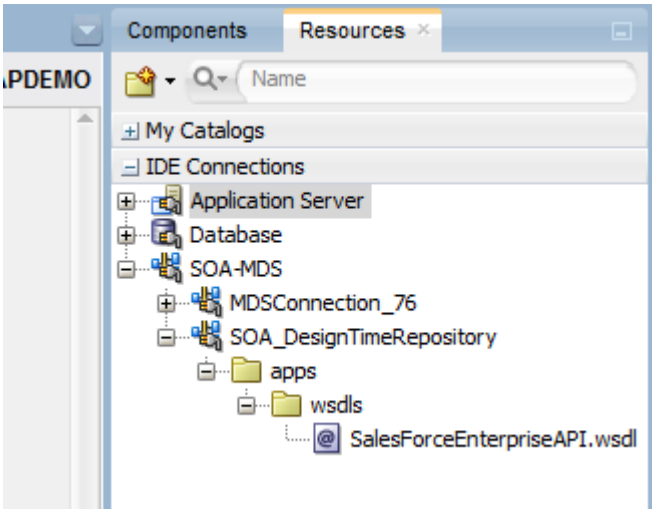
7. Click **OK**. You are returned to the Salesforce Cloud Server Connection page.

Figure 5- 16 Salesforce Cloud Server Connection Page



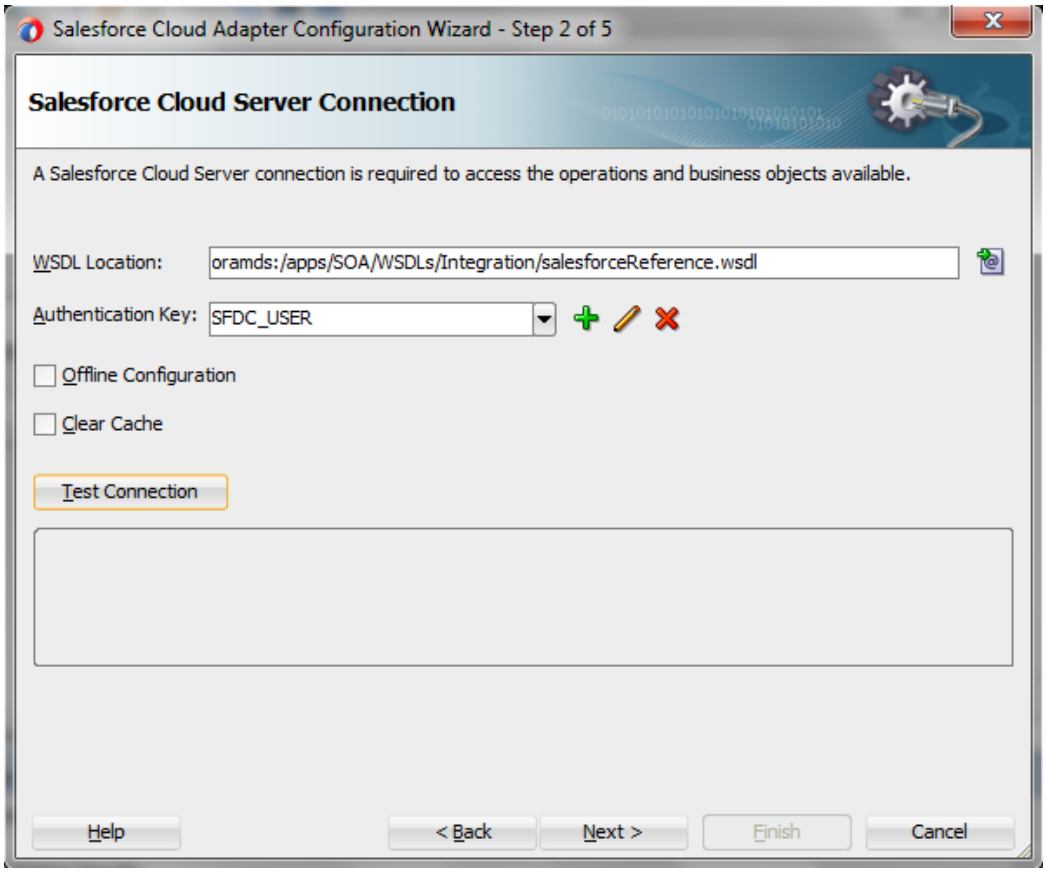
Note that as an alternative, you can store WSDL at an MDS location and access it, as shown in Figure 5-17.

Figure 5- 17 SOA Resource Browser



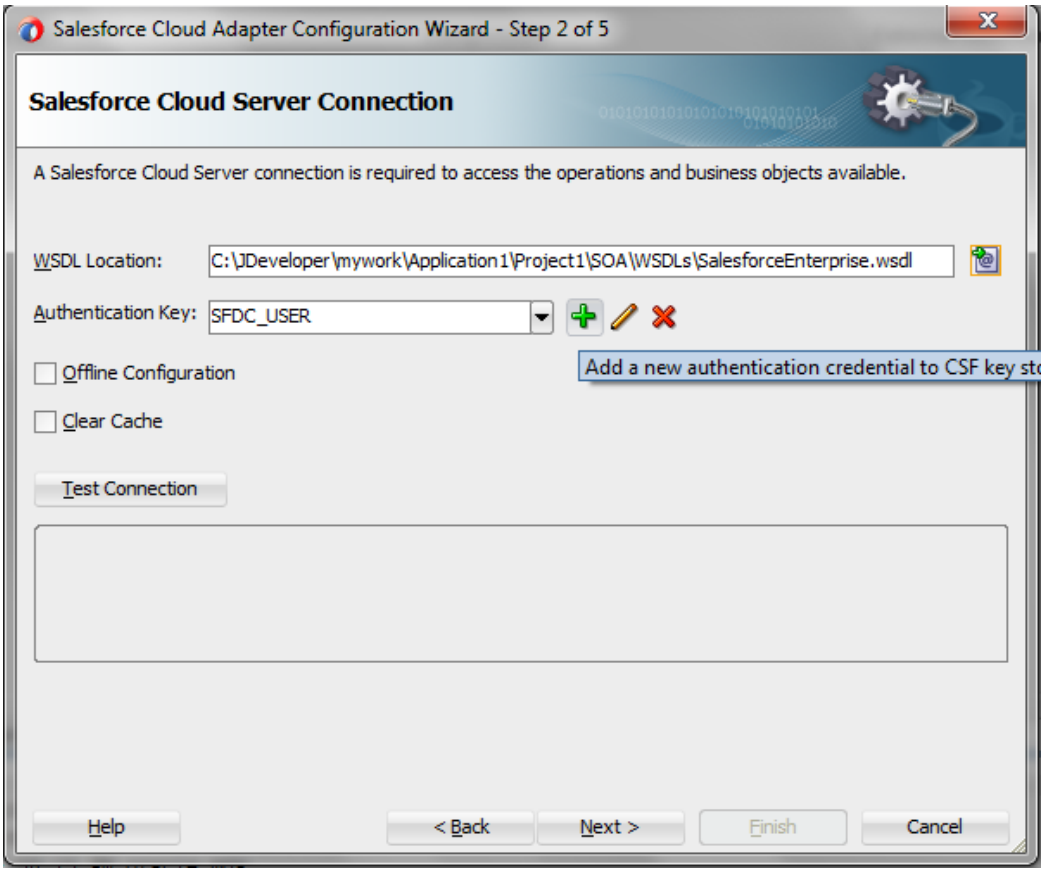
8. Traverse to IDE Connections → SOA-MDS. Select the appropriate SOA-MDS connection where you placed the Enterprise WSDL. Select the WSDL file to be used in the adapter configuration and click **OK**.
9. The WSDL location should be of the form 'oramds:/apps/SOA/WSDLs/Integration/SalesforceReference.wsdl', as shown in Figure 5-18.

Figure 5- 18 WSDL location



10. Click **OK**.
11. Click **"+"** button to create a new Authentication Key, as shown in Figure 5-19.

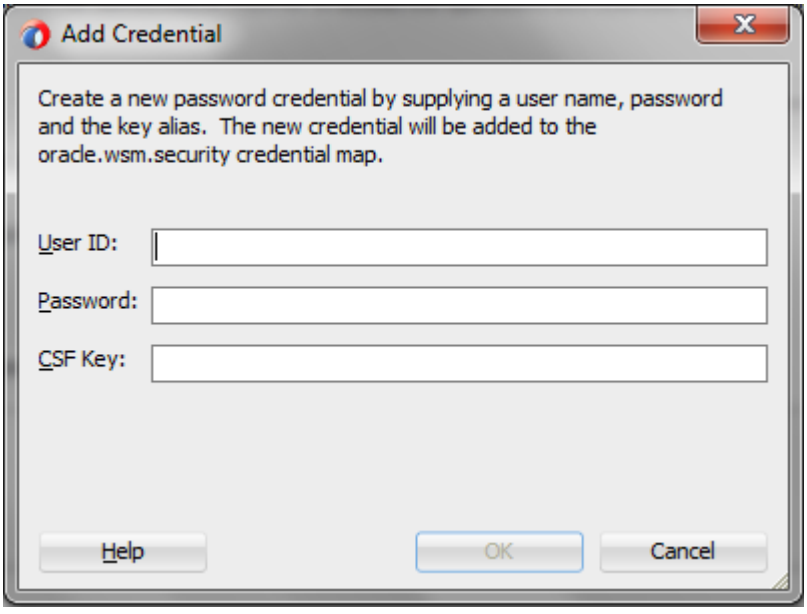
Figure 5- 19 Create a New Authentication Key



12. The **Add Credential** dialog box is displayed, as shown in Figure 5-20. Provide a suitable name and the Salesforce.com credentials and click **OK**.

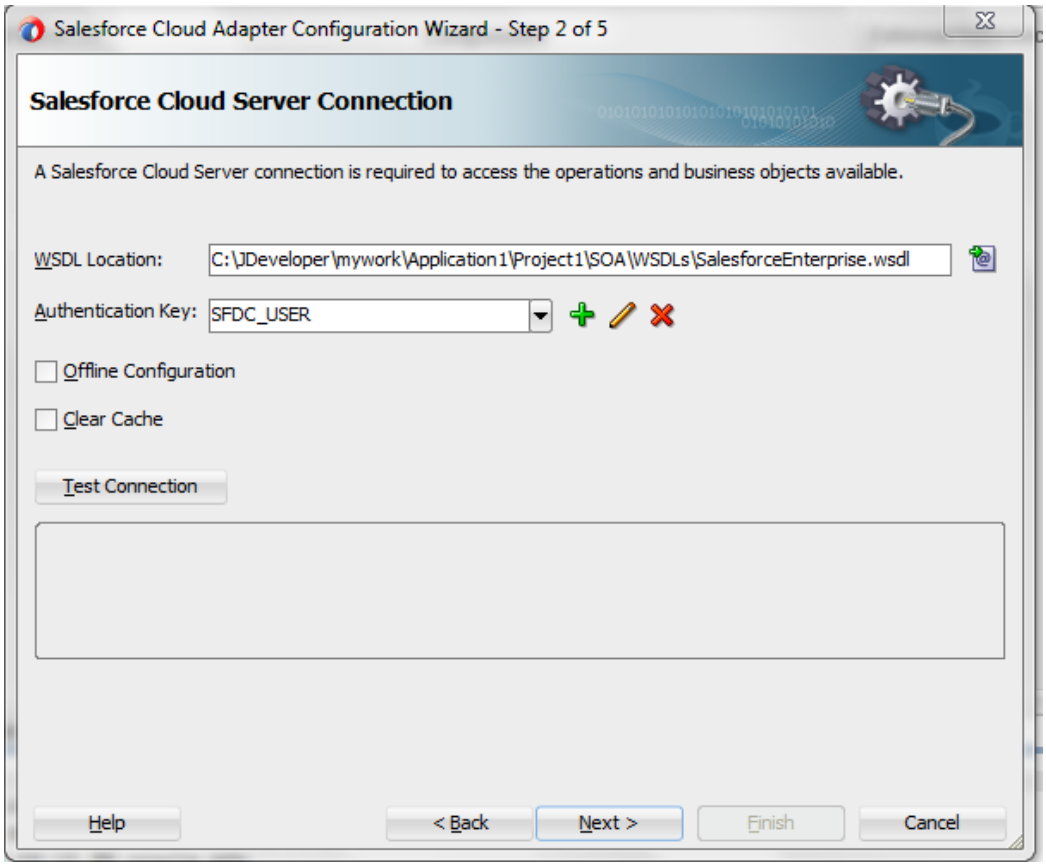
Note: The password should be a combination of Salesforce.com password and Salesforce.com Security Token.

Figure 5- 20 Add Credential



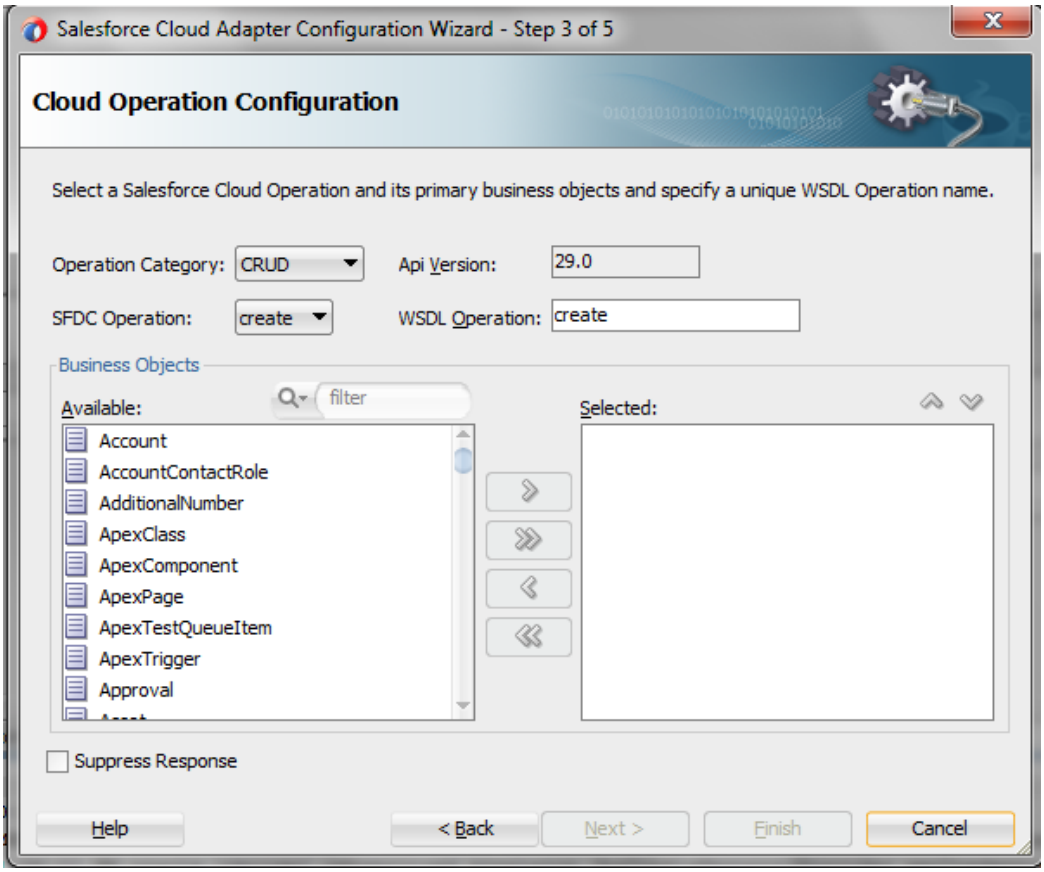
13. Click **Test Connection** button to validate the Authentication Key, as shown in Figure 5-21.

Figure 5- 21 Test Connection



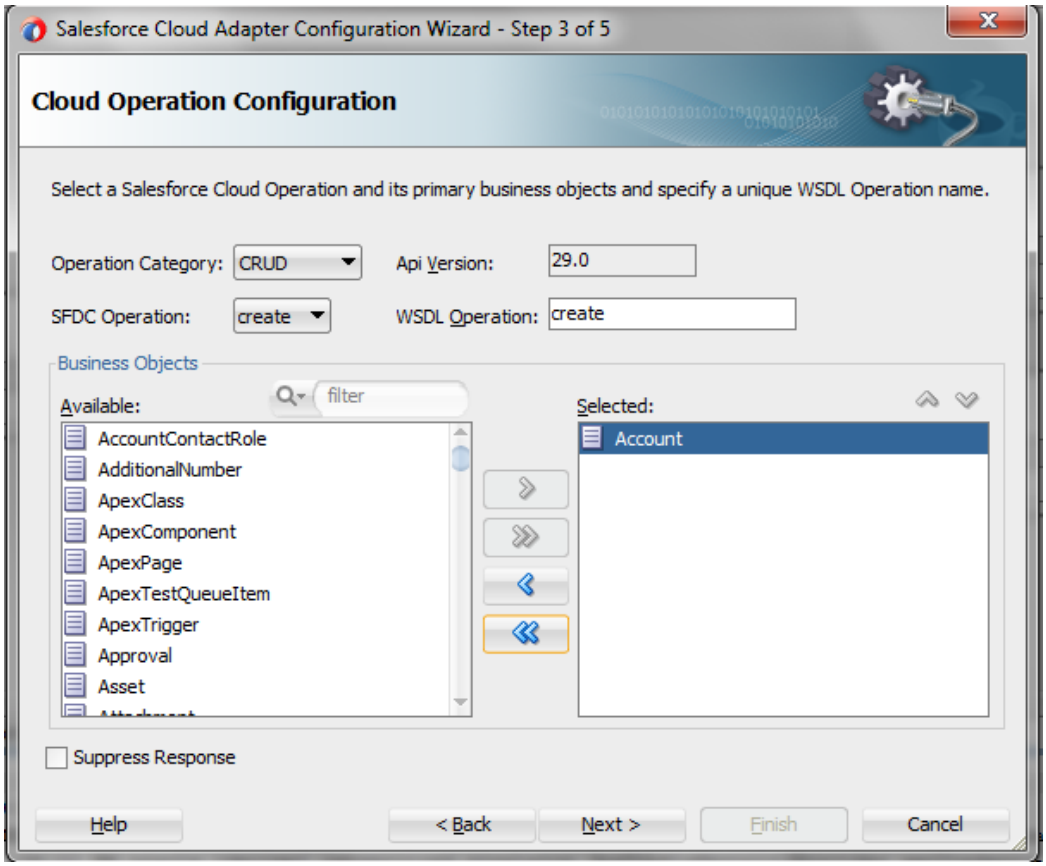
14. Click **Next**.
15. The **Cloud Operation Configuration** page is displayed, as shown in Figure 5-22.

Figure 5- 22 Cloud Operation Configuration Page



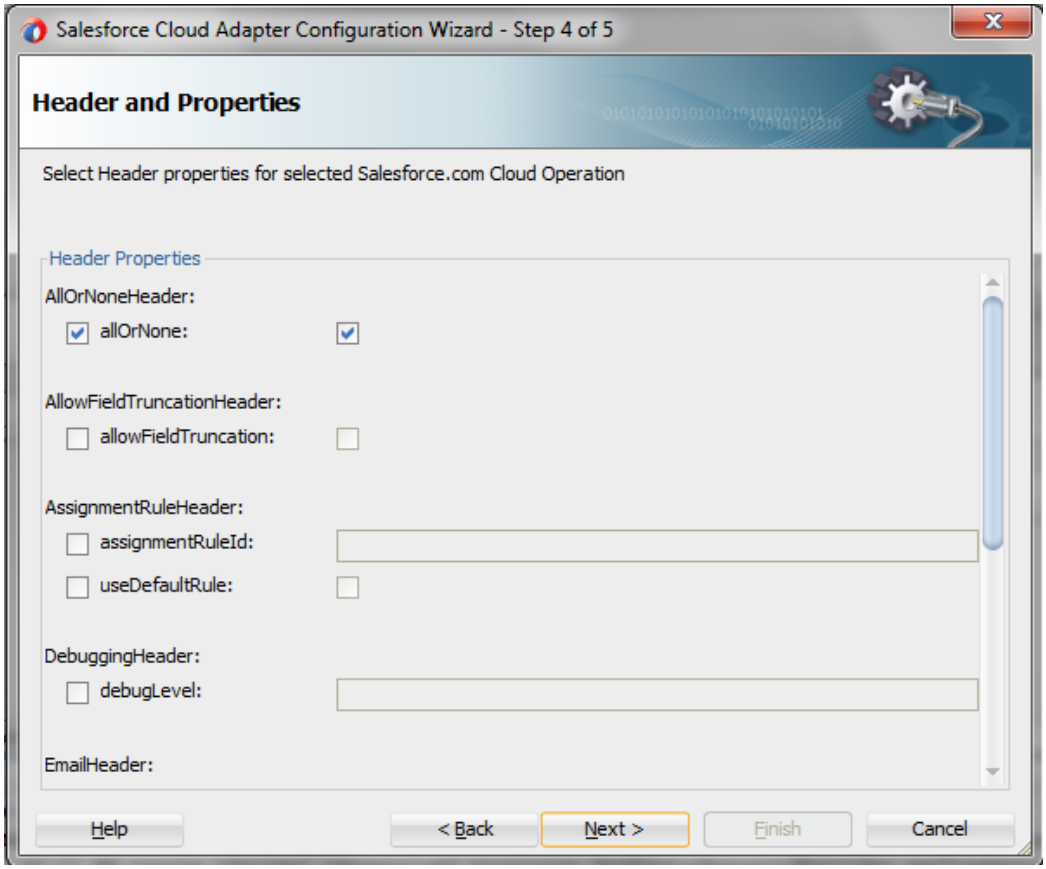
16. Since the scenario is to create an Account on Salesforce.com, select **Operation Category** as **CRUD** and **SFDC Operation** as **create**. Now move **Account** from the list of **Available** objects to the list of the **Selected** objects. The **WSDL Operation** by default is **create** (same as SFDC Operation). You can edit the same by providing an operation name suitable to your business requirement, as shown in Figure 5-23.

Figure 5- 23 Cloud Operation Configuration Page



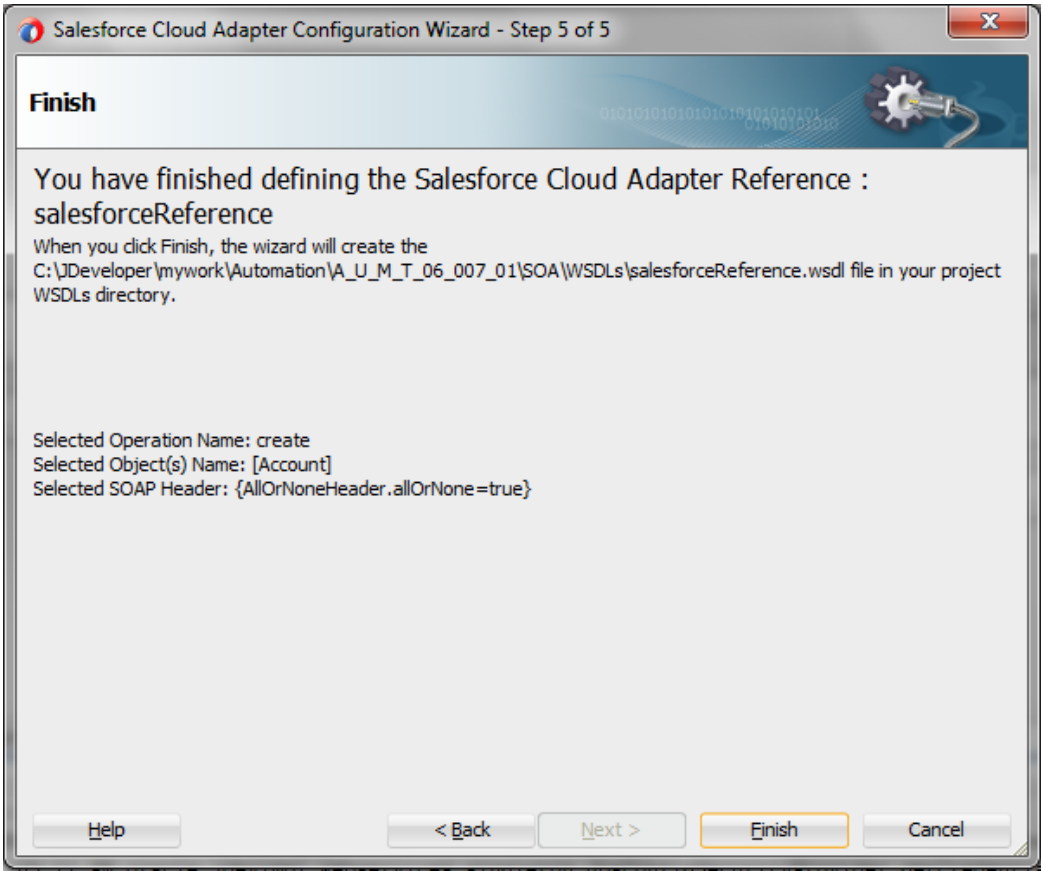
17. Click **Next**. The **Header and Properties** page is displayed, as shown in Figure 5-24.

Figure 5- 24 Header and Properties Page



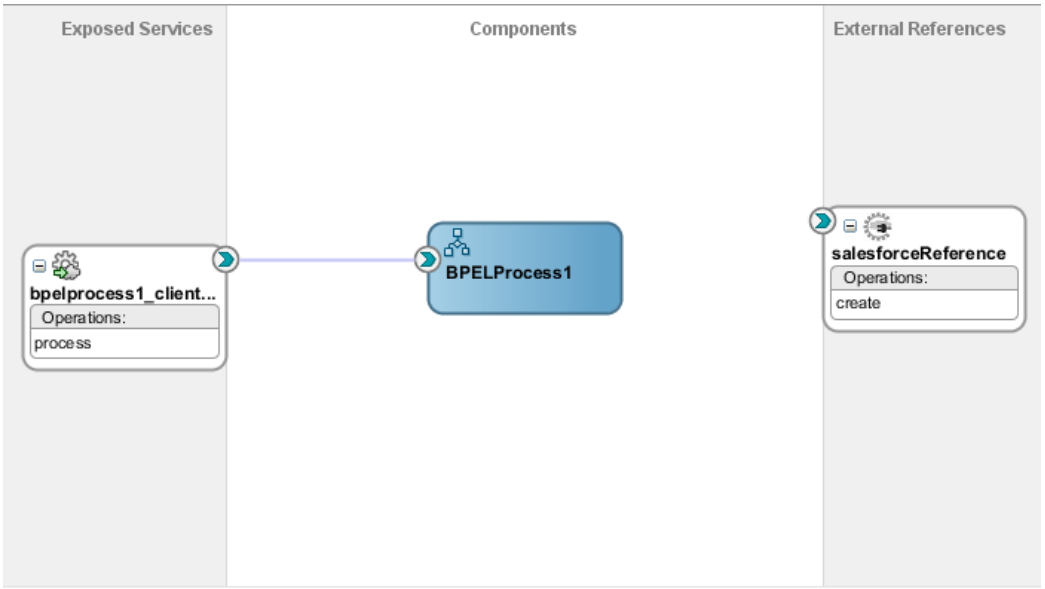
18. Select the appropriate header to your requirement. Headers displayed in this page depend on the operation selected in the previous page.
19. Click **Next**.
20. The finish page is displayed. It provides a complete summary of the operation selected, object on which the operation would operate and the headers selected for that operation, as shown in Figure 5-25.

Figure 5- 25 Finish Page



21. Click the **Finish** button to complete the Adapter Configuration Wizard.
22. After clicking on **Finish** button, the following screen appears, as shown in Figure 5-26.

Figure 5- 26 SFDC Cloud Account

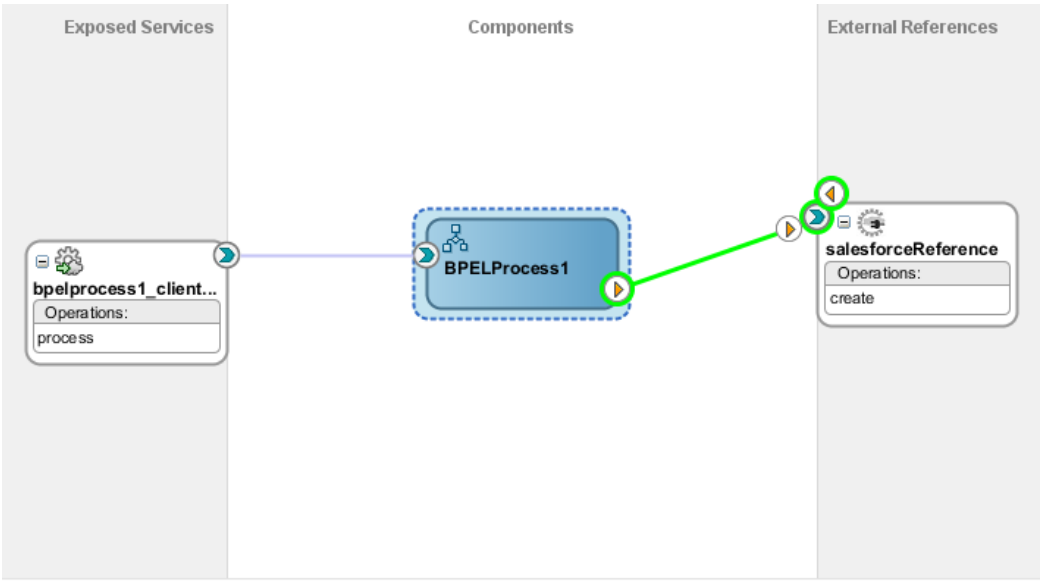


5.5 Integration with BPEL

Perform the following steps to integration with BPEL:

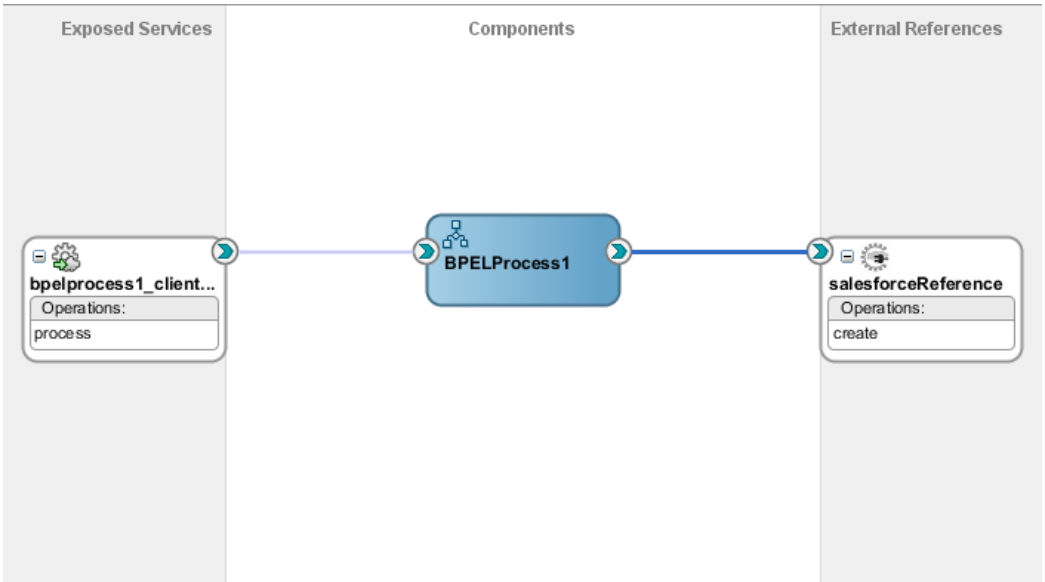
1. Connect **BPELProcess1** and **salesforceReference** via a wire, as shown in Figure 5-27.

Figure 5- 27 Wired BPELProcess1 and salesforceReference



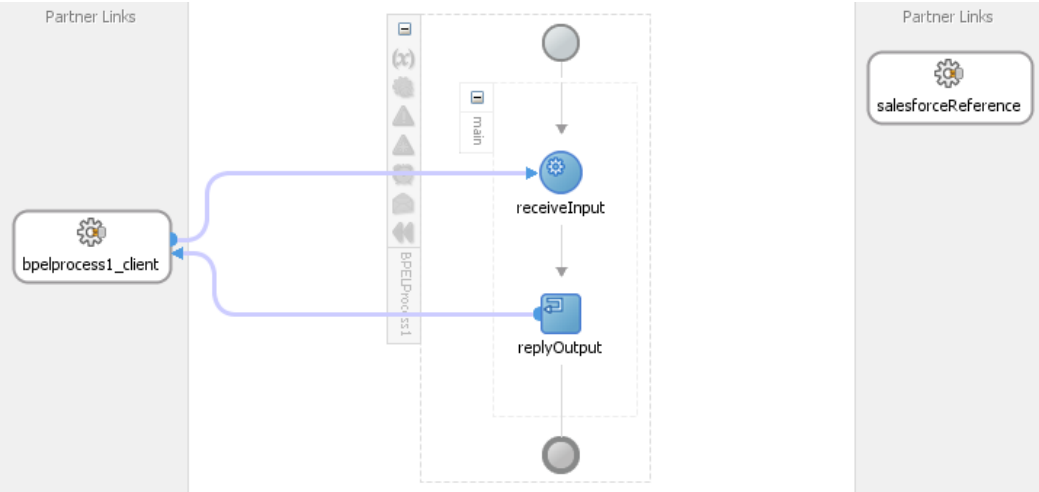
2. After wiring your composite looks like as shown in Figure 5-28.

Figure 5- 28 Wiring BPELProcess1 and salesforceReference



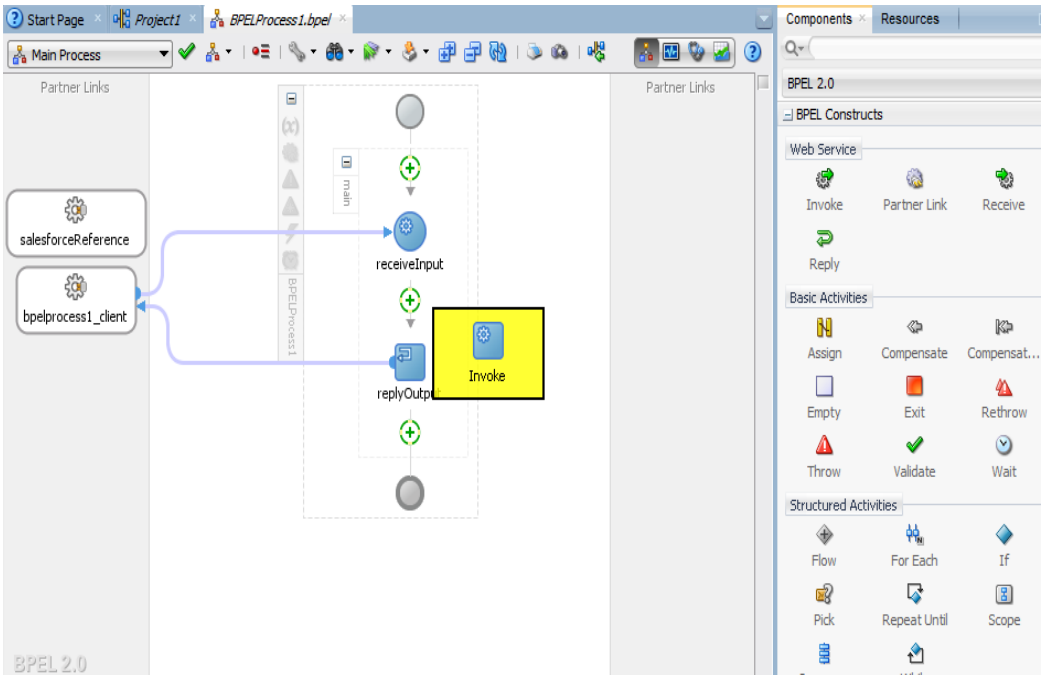
3. Double-click and open **BPELProcess1**. The salesforceReference adapter should be present as a part of the Partner Links, as shown in Figure 5-29.

Figure 5- 29 Open BPELProcess1



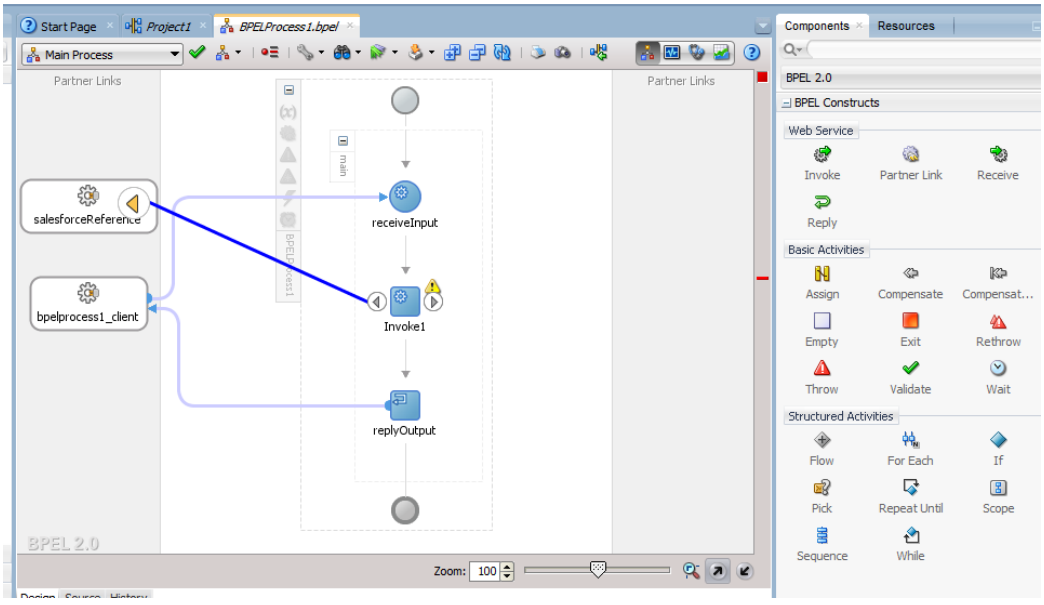
4. Add an invoke activity to invoke the salesforceReference Partner Link. To add the invoke activity, follow the below provided steps:
   a) Drag and drop the invoke activity from the BPEL constructs, as shown in Figure 5-30.

Figure 5- 30 Add invoke activity to invoke the salesforceReference Partner Link



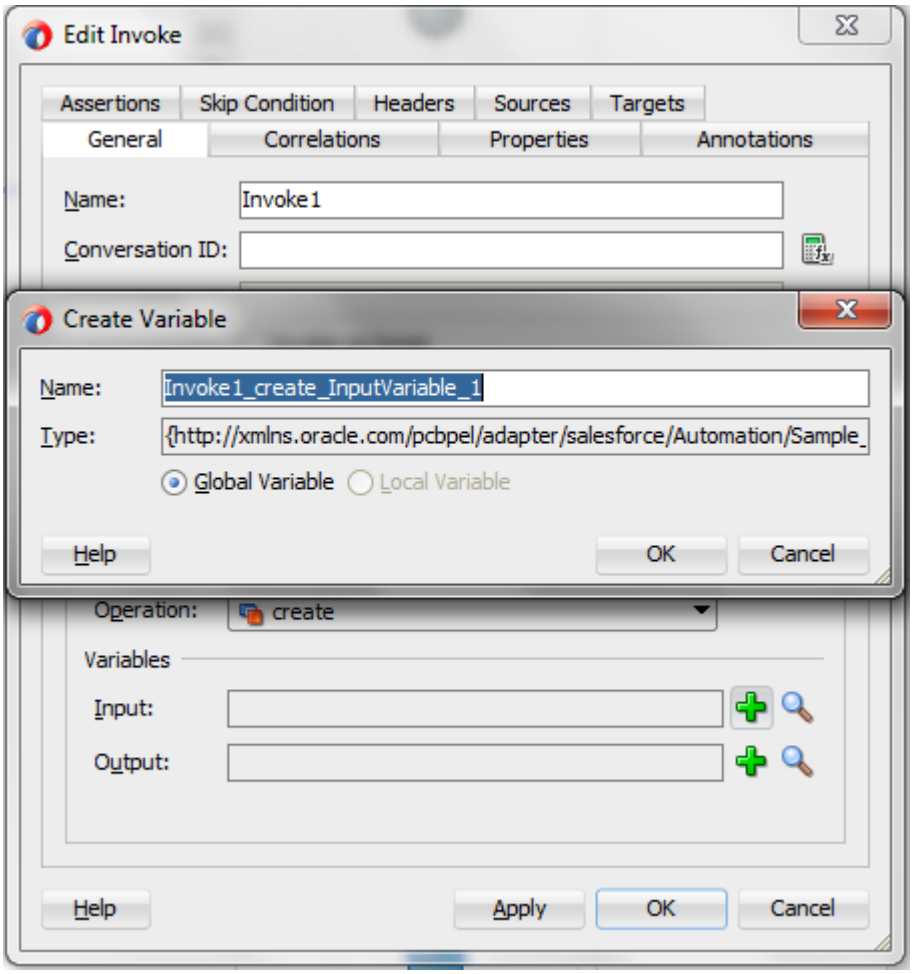
b) After dropping the Invoke property, wire the invoke activity to the salesforceReference partner link. Now, the composite will appear as shown in Figure 5-31.

Figure 5- 31 Composite Window



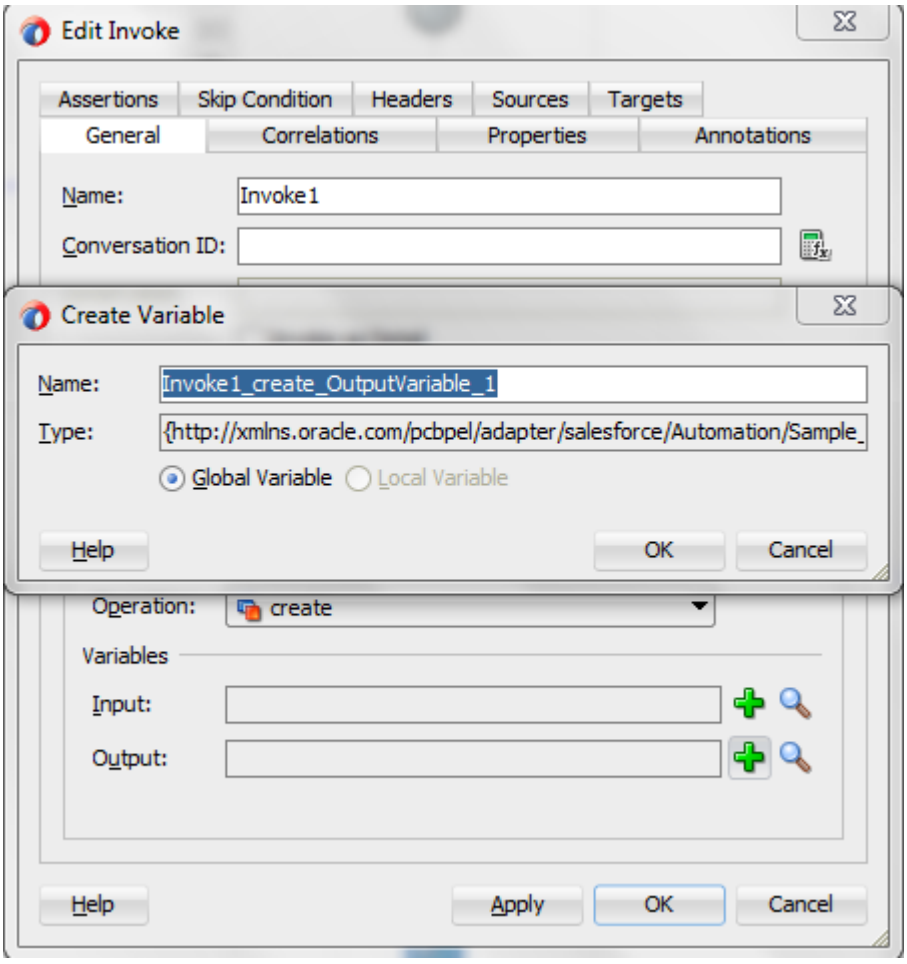
5. Create an input variable to the partner link by clicking the '+' button adjacent to **Input** text box in the **Variables** section. The **Create Variable** dialog is displayed, as shown in Figure 5-32.

Figure 5- 32 Create an Input Variable



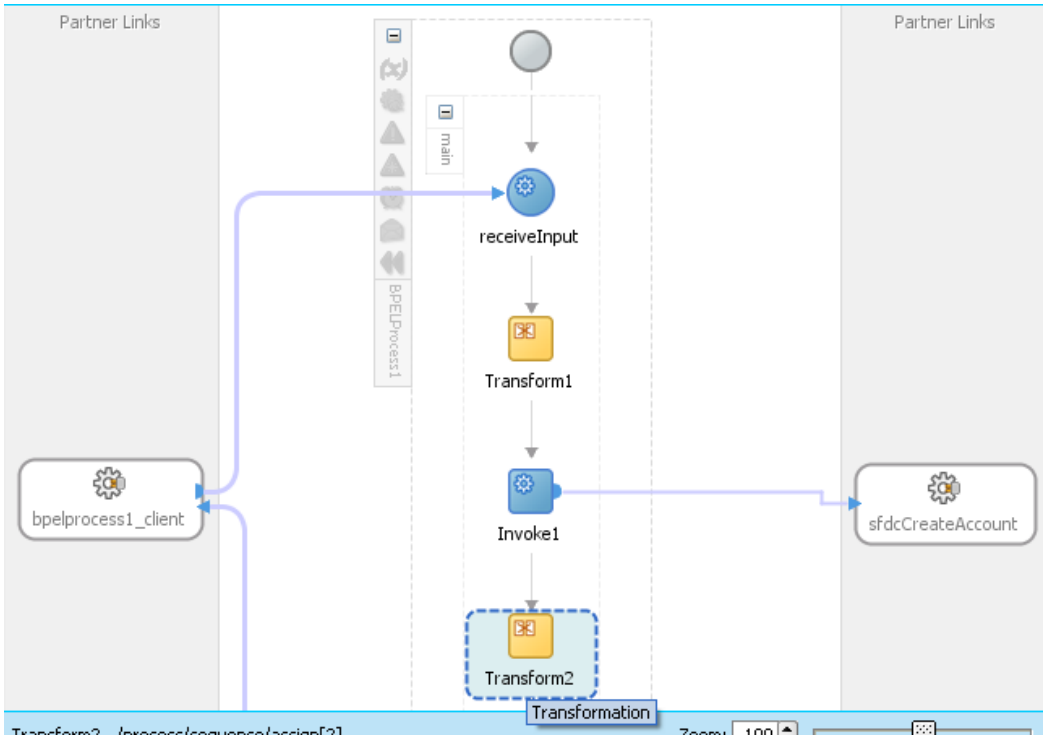
6. Create an output variable from the partner link by clicking the '+' button adjacent to **Output** text box in the **Variables** section. The **Create Variable** dialog is displayed, as shown in Figure 5-33.

Figure 5- 33 Create an Output Variable



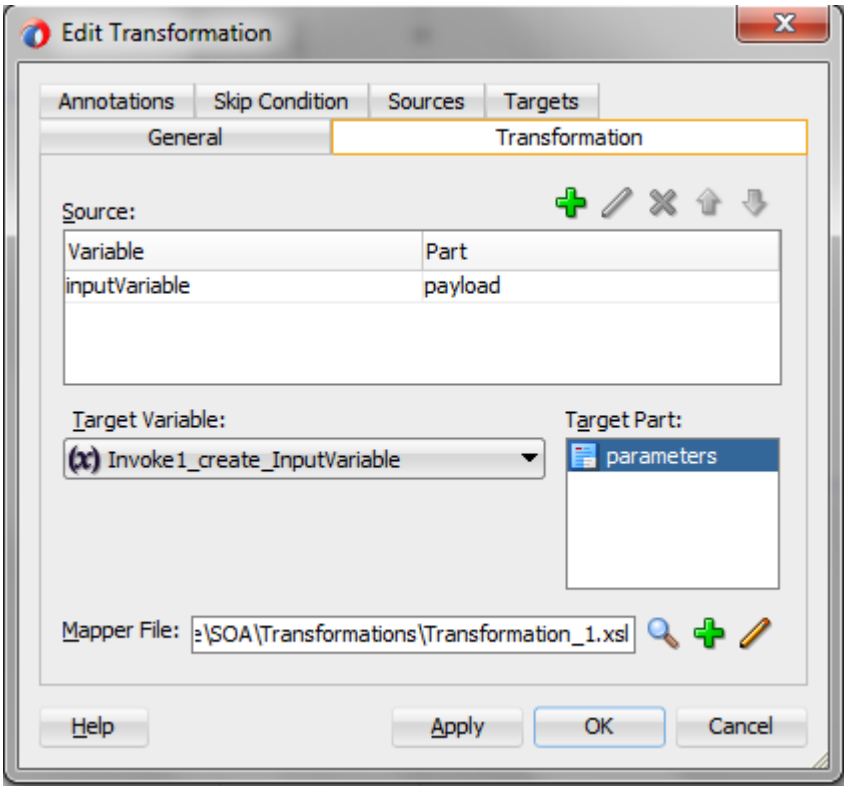
7. Introduce two transform activities, one prior to the invoke activity and another after it, as shown in Figure 5-34.

Figure 5- 34 Transform Activities



8. Map Transform1 values from receive activity's input variable to invoke activity's input variable, as shown in Figure 5-35.

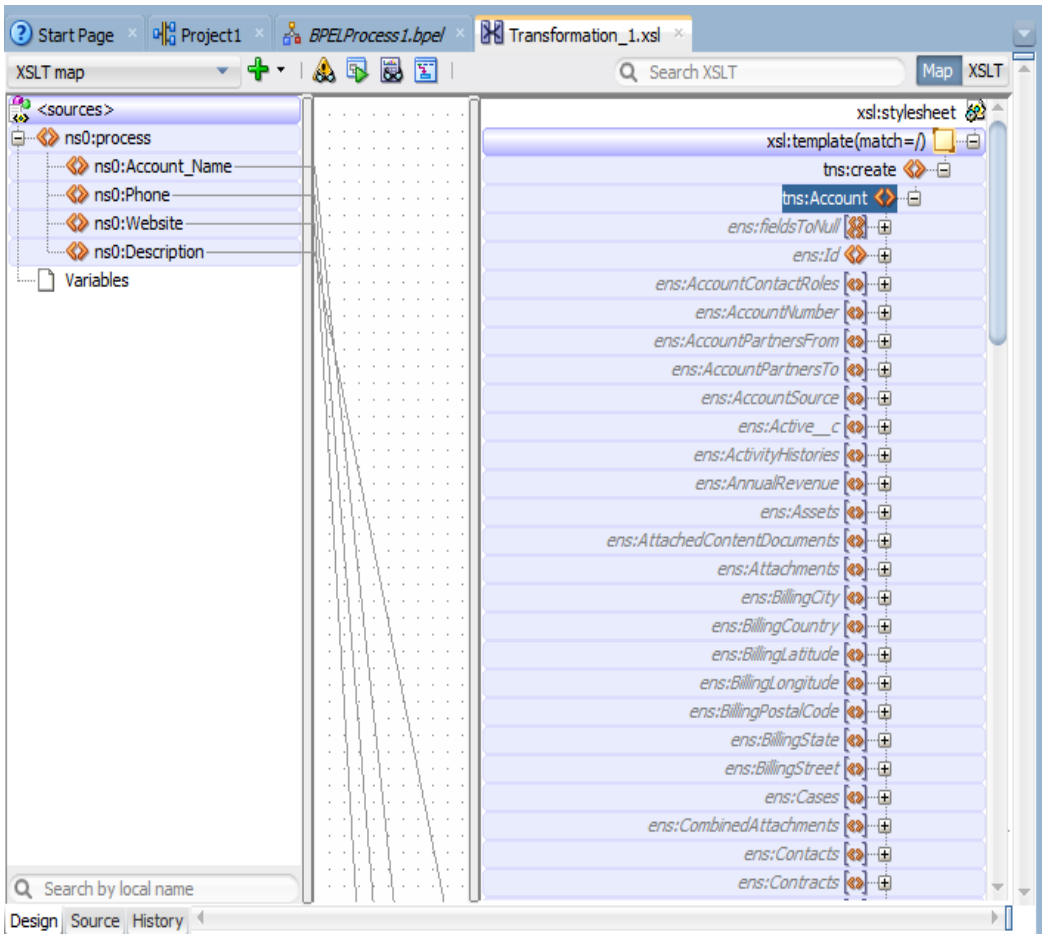
Figure 5- 35 Edit Transform



9. Click the '+' button adjacent to **Mapper File** text box to open the **Transformation_1.xsl** file.
10. Perform the mapping between **inputVariable** and **Invoke1_create_InputVariable**:

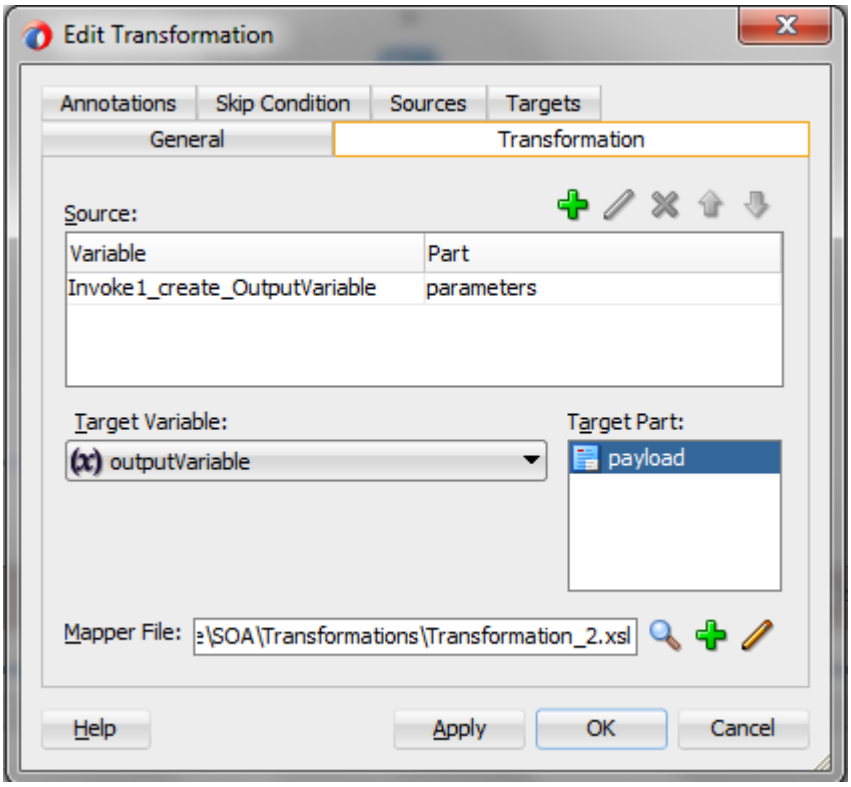
- Map Account_Name with Name.
- Map Phone with Phone.
- Map Website with Website.
- Map Description with Description, as shown in Figure 5-36.

Figure 5- 36 Mapping between inputVariable and Invoke1_create_InputVariable



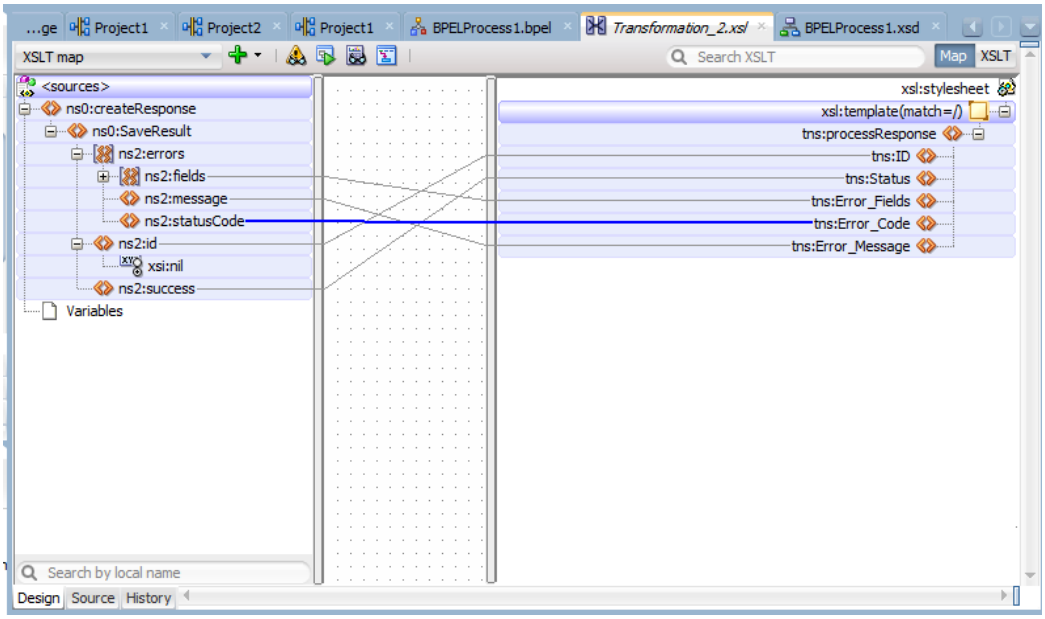
11. Map Transform2 values from invoke activity's output variable to reply activity's input variable, as shown in Figure 5-37.

Figure 5- 37 Edit Transform



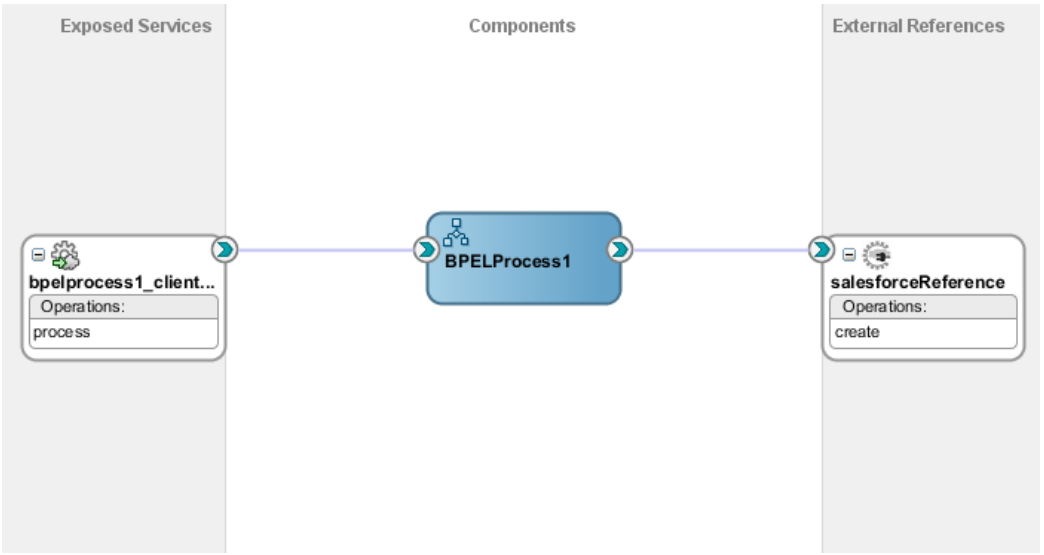
12. Perform the mappings for output variable, as shown in Figure 5-38.

Figure 5- 38 Mappings for Output Variable



13. This completes the project creation. The composite.xml looks, as shown in Figure 5-39.

Figure 5- 39 Composite.xml after Project Creation

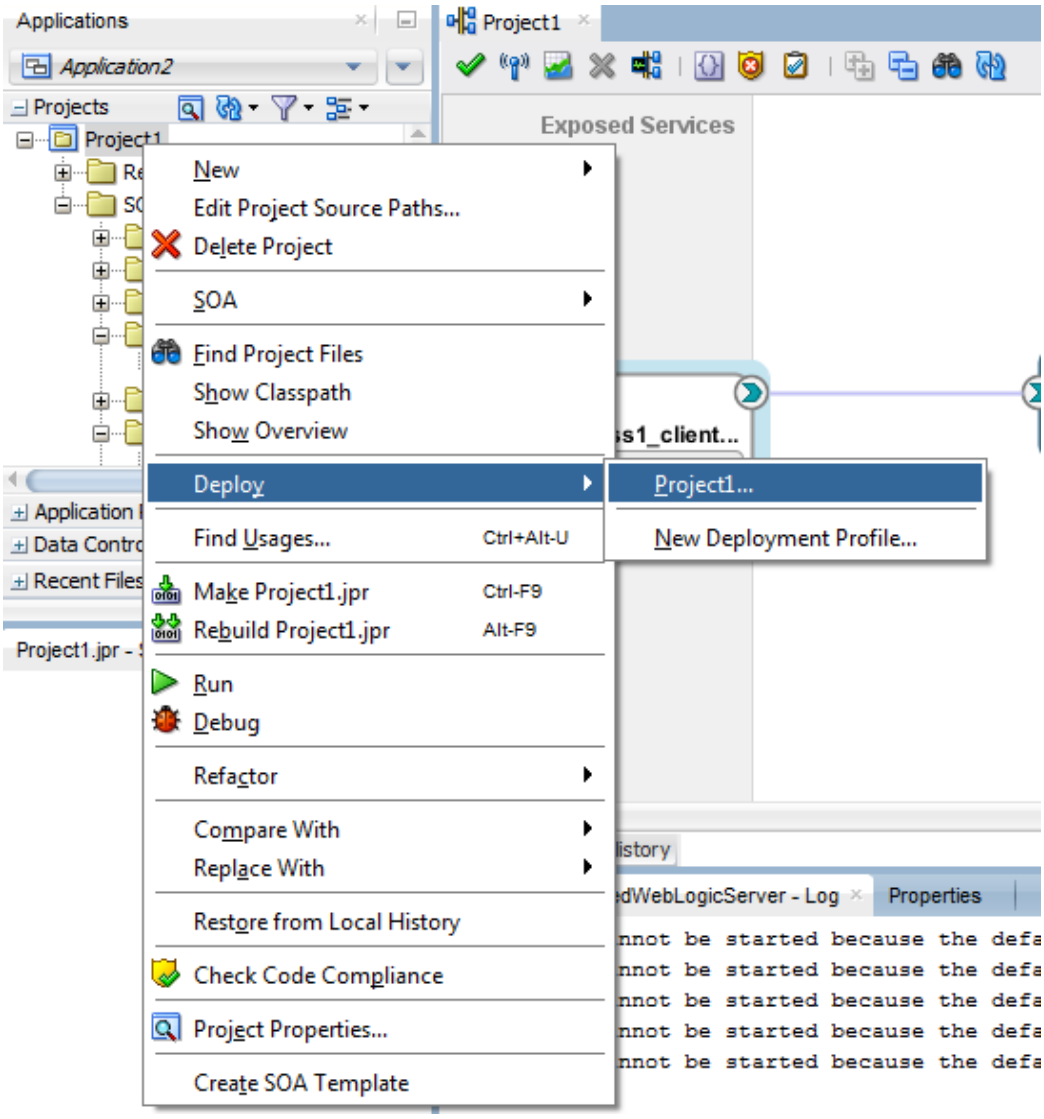


5.6 Deploy the Composite

Perform the following steps to deploy the composite.

1. In the **Application Navigator** pane, right-click **Project1** and select Deploy → Project1, as shown in Figure 5-40.

Figure 5- 40 Deploy the Composite



2. Select the **Deploy to Application Server** option and follow the instructions. Using this option, you can deploy the composite on the Application server after providing the details of the server.

5.7 Test the Composite

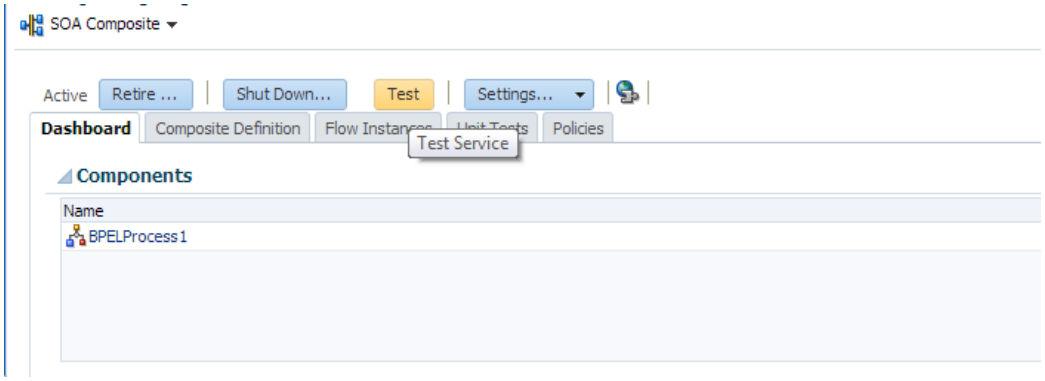
You can run and test the instances of deployed SOA composite applications from Oracle Enterprise Manager Grid Control Console. This enables you to manage a composite application, initiate and track an instance of a composite and to view detailed component instance audit trails. Perform the following process to test the composite:

5.7.1 Test the Outbound Process

Perform the following steps to test the Outbound process:

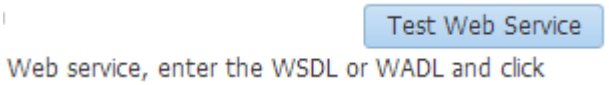
1. Login to Enterprise Manager Console for the server on which you have deployed your project.
2. Open **Project1** under the **Default** partition.
3. Click the **Test** button to test the Web service, as shown in Figure 5-41.

Figure 5- 41 Test the Web Service



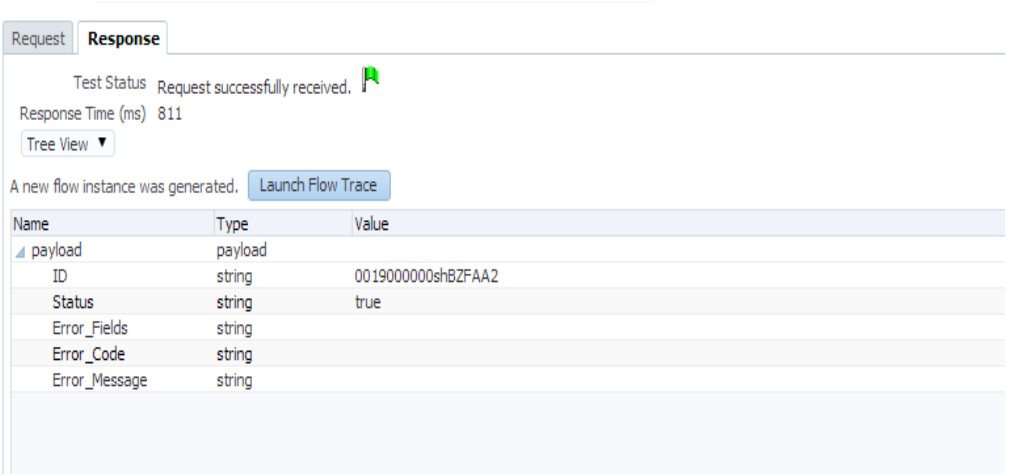
4. Provide the input payload and click the **Test Web Service** button, as shown in Figure 5-42.

Figure 5- 42 Test Web Service



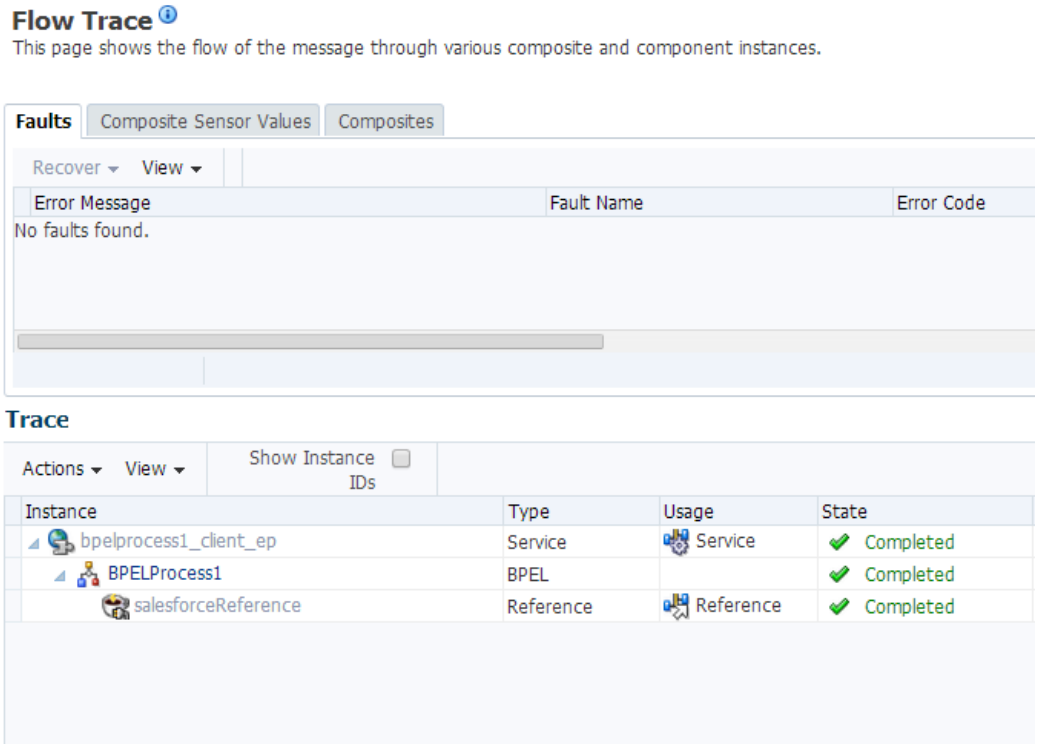
5. After successful execution, the response contains ID of the account created on Salesforce.com and **Status** as success, as shown in Figure 5-43.

Figure 5- 43 Test Status



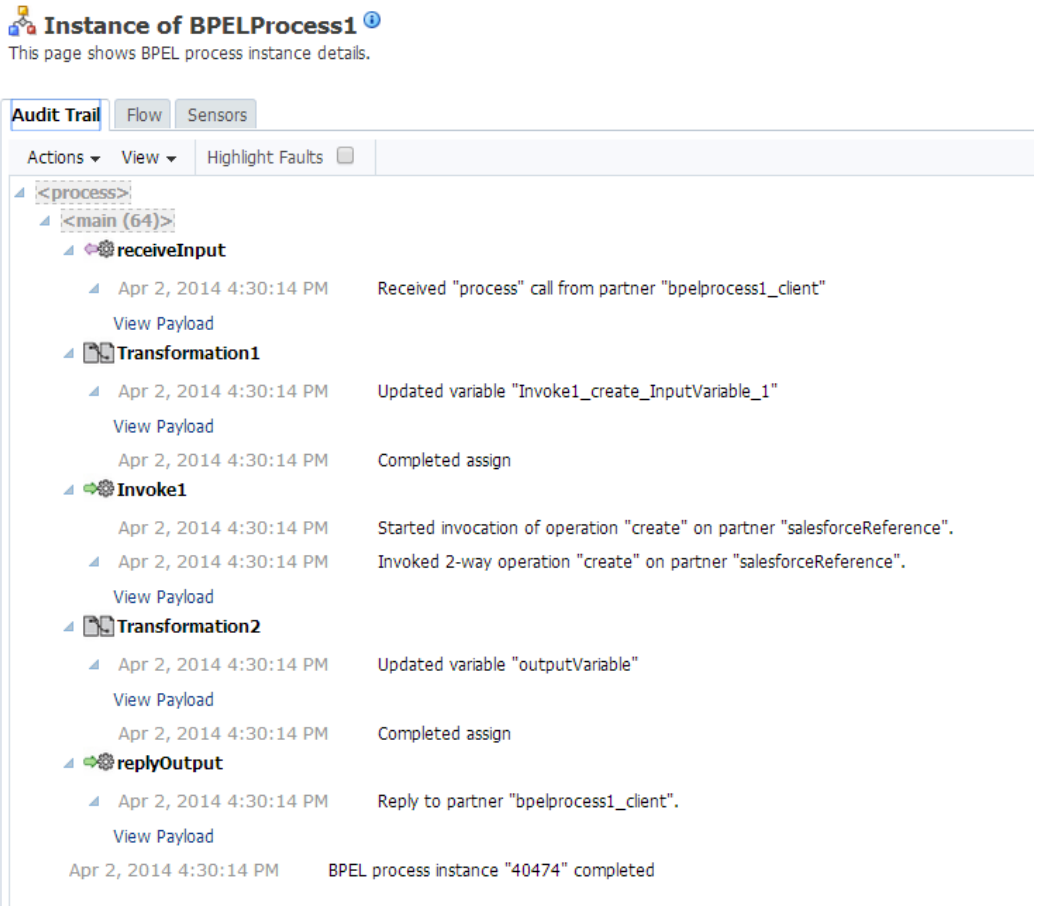
6. Click the **Launch Flow Trace** button to view the **Audit Trail**, as shown in Figure 5-44.

Figure 5- 44 Launch Flow Trace



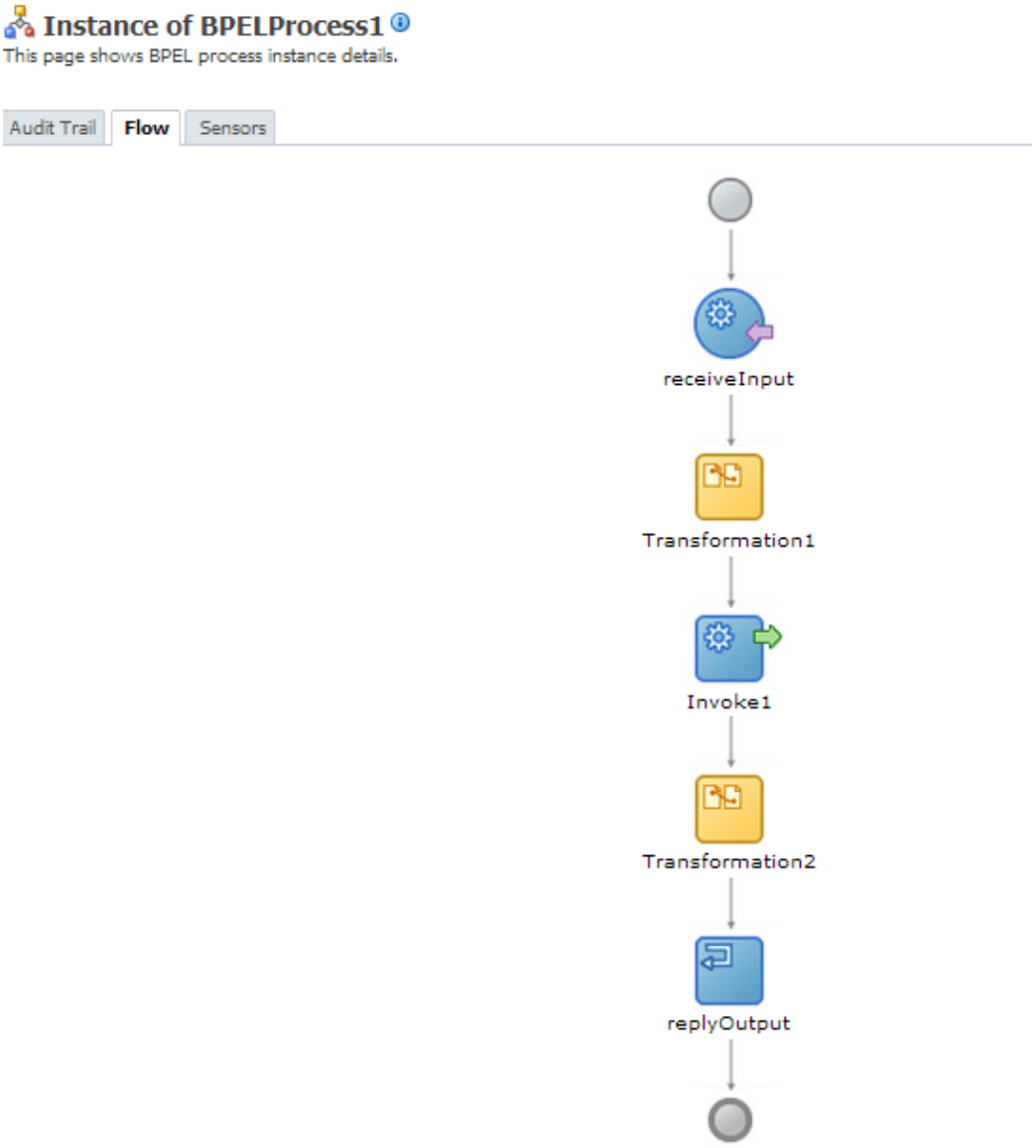
7. The **Audit Trail** will look like Figure 5-45.

Figure 5- 45 Audit Trail



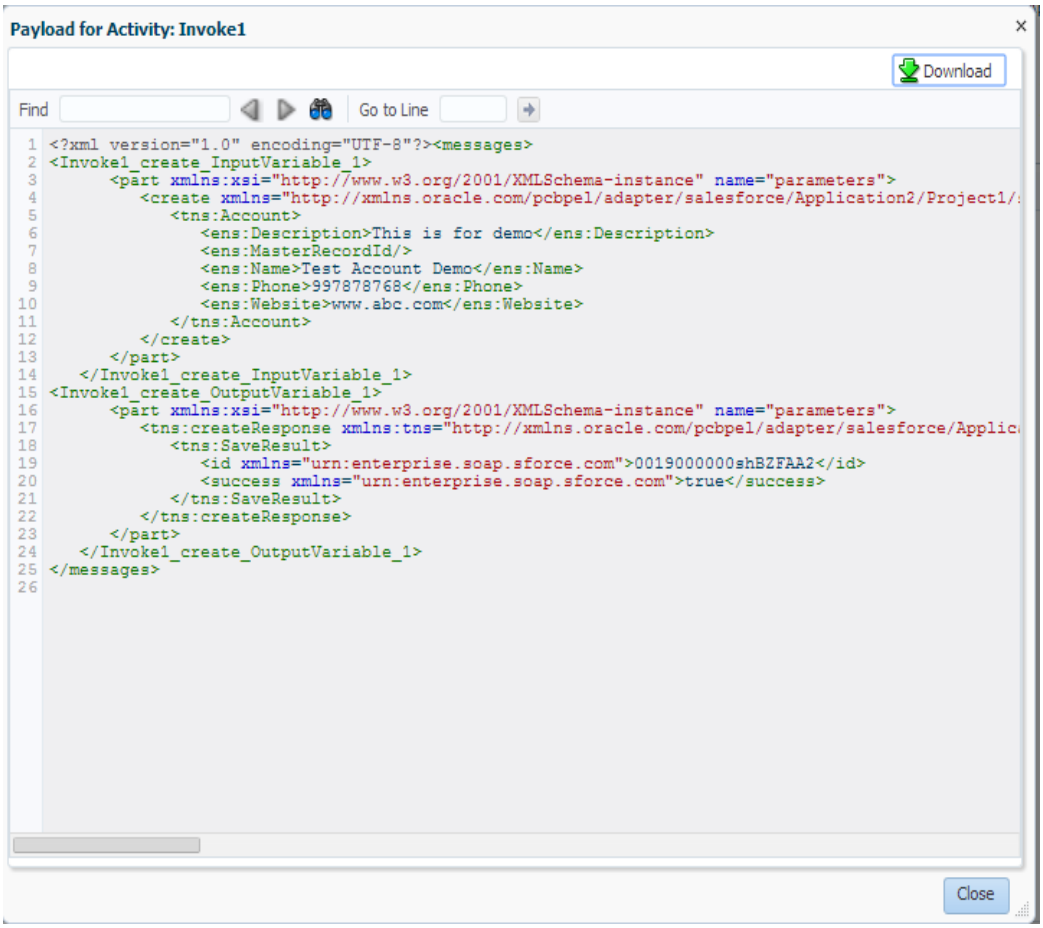
8. Click on **Flow** tab, The Flow tab is shown in Figure 5-46.

Figure 5- 46 Flow Tab



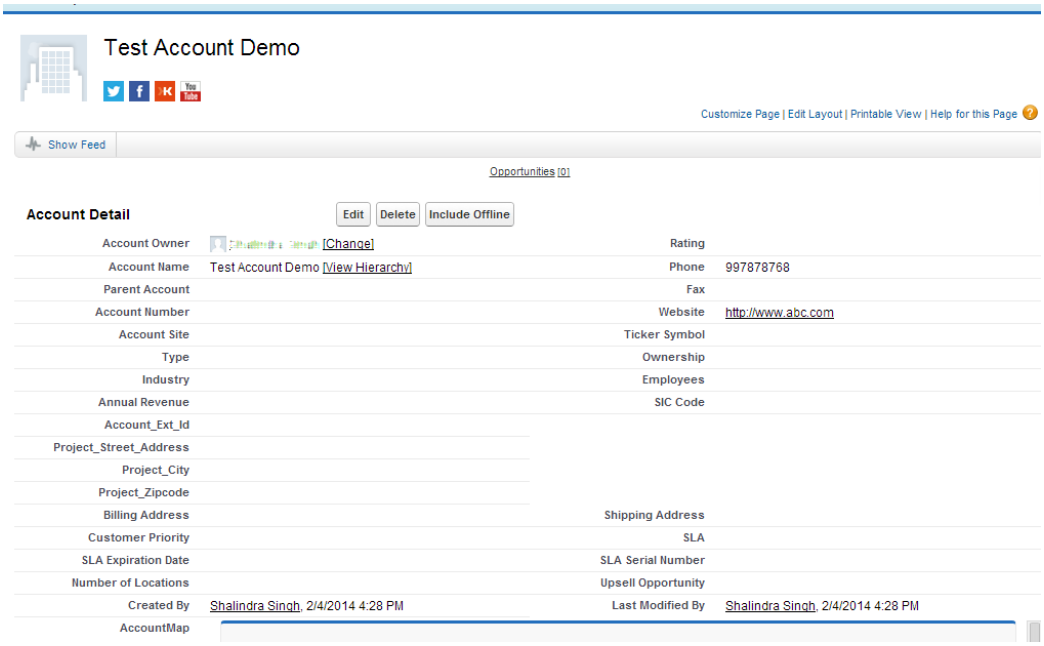
9. The Invoke activity of the process is shown in Figure 5-47.

Figure 5- 47 Invoke Activity



10. This completes the Create Account scenario. The success of the task can be verified on Salesforce.com using the ID returned as response of create call, as shown in Figure 5-48.

Figure 5- 48 Create Account



6

Configuring Outbound Processing Using Oracle Service Bus

Oracle Service Bus (OSB) make use of the Oracle Cloud Adapter for Salesforce.com via the Oracle SOA Suite components and JDeveloper IDE. This chapter describes the process of creating OSB projects using JDeveleoper IDE or using OSB console, deploying the OSB services to an OSB domain, and finally testing the OSB services.

This chapter contains the following topics:

- Section 6.1, "Overview of Application Adapter Integration with Oracle Service Bus"
- Section 6.2, "Creating Outbound Processes Using Oracle Service Bus"

6.1 Overview of Application Adapter Integration with Oracle Service Bus

Oracle Service Bus can be leveraged to access the APIs exposed by Salesforce.com to achieve application integration via the Oracle Cloud Adapter for Salesforce.com. OSB uses the Oracle SOA Suite components and JDeveloper to employ the Oracle Cloud Adapter for Salesforce.com. This chapter will use the same business case of creating an Account in Salesforce.com as in Chapter 5:
Integration with Different Service Components (BPEL/Mediator) in Oracle SOA Suite.

6.2 Creating Outbound Processes Using Oracle Service Bus

Oracle Service Bus interacts with the Oracle Cloud Adapter for Salesforce.com through the ‘Oracle Cloud Adapter for Salesforce.com Artifacts’ generated using the SOA components. This section describes the process of generating the Oracle Cloud Adapter for Salesforce.com artifacts and creating OSB services based on these artifacts.

Oracle Service Bus provides two ways of designing OSB projects. One is JDeveloper IDE, and second option is to create projects and services directly on the Oracle Service Bus console. This chapter explains both ways of creating OSB projects and services.

6.2.1 Creating OSB Projects Using OSB Console

This section explains how to create OSB projects using OSB console. For this you first need to generate Oracle Cloud Adapter for Salesforce.com artifacts using JDeveloper 12c, and then those artifacts would be used while creating OSB projects and services using OSB console.

To create the Salesforce.com artifacts to be used by the OSB services, you need to use JDeveloper 11g in preview mode. JDeveloper can be started in preview mode using prompt and by passing ***-J"-Dsoa.preview.mode=true"*** as parameters as shown below:

$Middleware_Home\jdeveloper\jdev\bin> jdev.exe *-J"-Dsoa.preview.mode=true"*

For information on how to test the OSB projects created using OSB console, refer to the section "Testing OSB project from Service Bus Console".

Generating Oracle Cloud Adapter for Salesforce.com Artifacts:

To create OSB projects using OSB Console, you need to use Oracle Cloud Adapter for Salesforce.com artifacts generated using JDeveloper 12c.

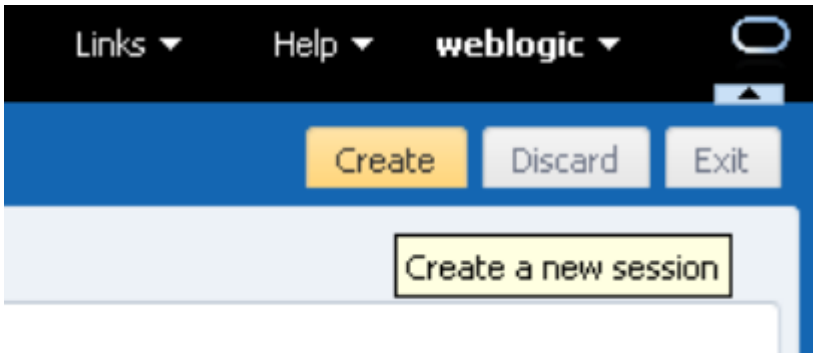
To create a BPEL composite, refer to the section "Designing a Composite for Service Integration". Next, to configure the Oracle Cloud Adapter for Salesforce.com, refer to the section "Configure Oracle Cloud Adapter for Salesforce.com". Save the Oracle Cloud Adapter for Salesforce.com artifacts in a directory. You will use the artifacts generated in section 5.4 "Configure Oracle Cloud Adapter for Salesforce.com" in creating OSB business services in OSB console.

Creating OSB projects and services using the OSB console:

Follow the steps given below to create OSB projects and services using the OSB console.

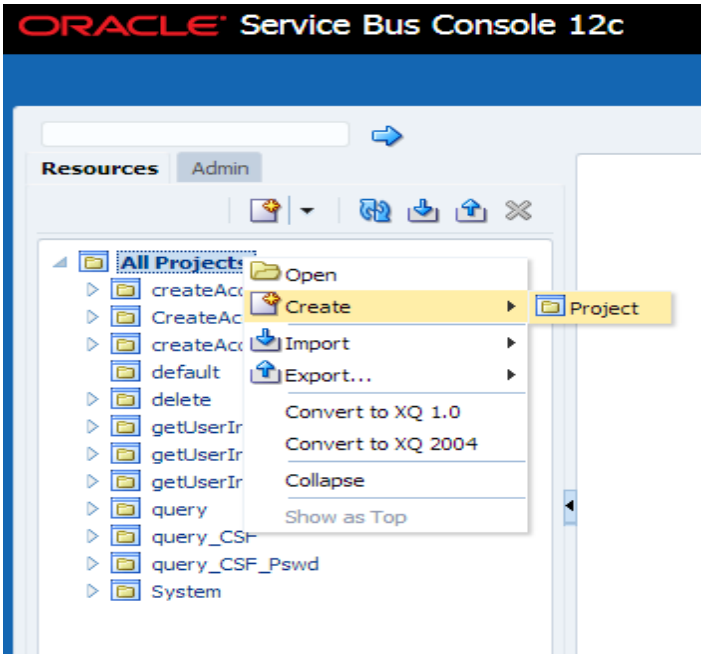
1. Log in to OSB Console: *<host:port/sbconsole>*
2. On right side of webpage, click on **Create**, as shown in the Figure below. This opens a new session in the OSB console to enable you to make changes.

Figure 6- 1 Create an OSB Session



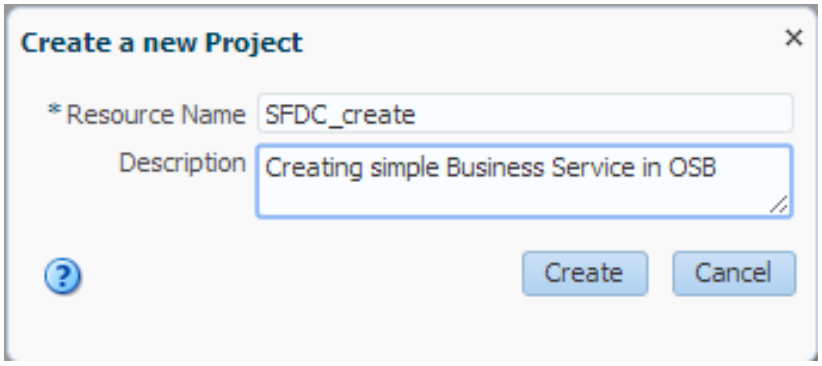
3. Enter a new project name, right-click on **All Projects** and click **Create - Project**, as shown in the Figure below.

Figure 6- 2 Add a New Project to OSB



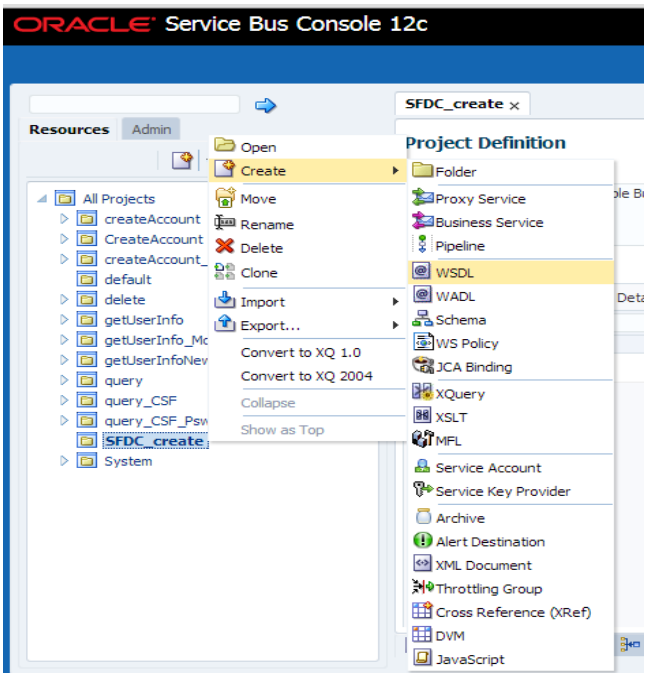
4. A dialog box will appear with title **Create a new project.**
5. Enter the project name in the **Resource Name** field, and provide the description in **Description** field, as shown in the Figure below.

Figure 6- 3 Create a New Project



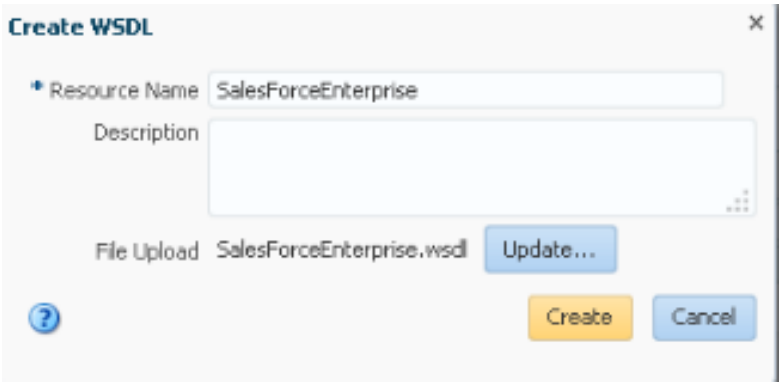
6. Click on **Create**. A new project named SFDC_Create appears in under **All Projects**.
7. Import the Oracle Cloud Adapter for Salesforce.com artifacts in this project. Right-click on the Project name, Select **Create** and then **WSDL** as shown in the Figure below.

Figure 6- 4 Choose WSDL Resource



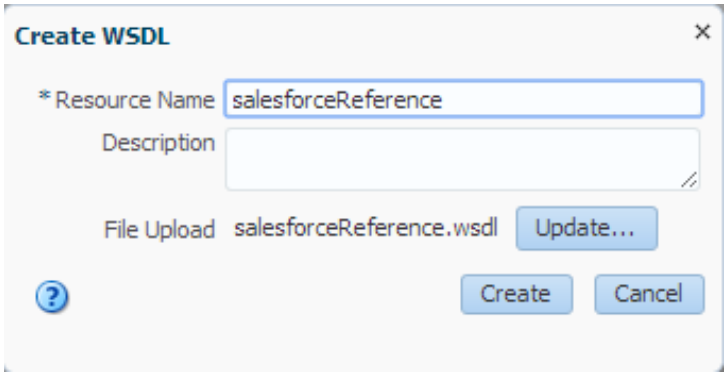
8. WSDL Creation Page is displayed. Click on **Browse** and browse to the directory where the artifacts received from the JDeveloper 12c are stored. Choose the Oracle Cloud Adapter for Salesforce.com WSDL, as shown in the Figure below.

Figure 6- 5 Create WSDL



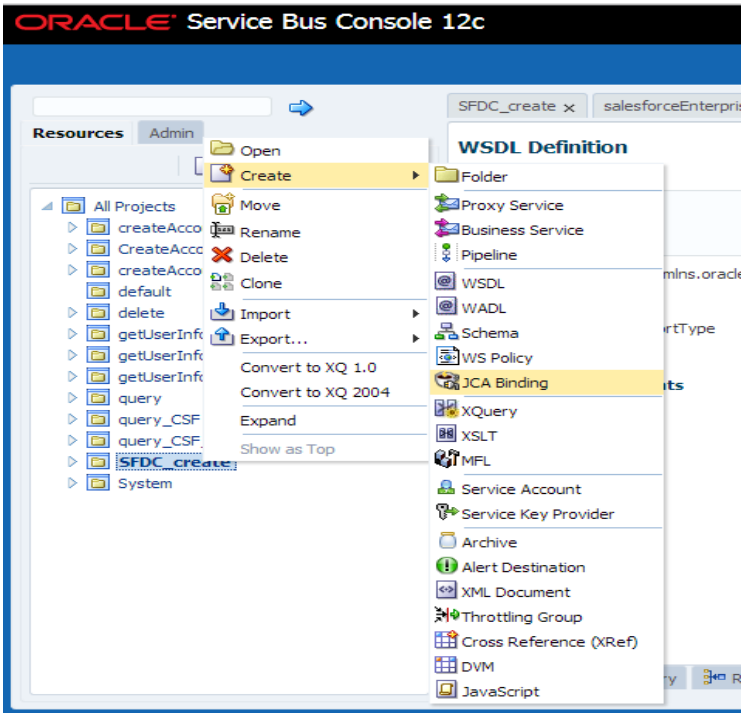
9. Click **Create**.
10. Again right-click on the Project name, from the menu select **Create** and then **WSDL**. Now on the WSDL Creation Page, browse for salesforceReference WSDL as shown in the Figure below.

Figure 6- 6 Create WSDL



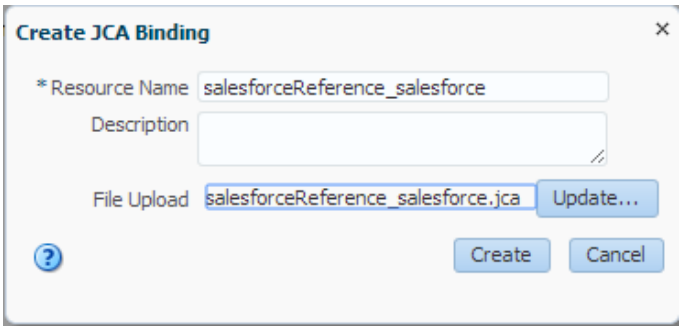
11. Click **Create.**
12. Select **JCA Binding** from the **Create** drop-down list under **Project Name**, as shown in the Figure below.

Figure 6- 7 Select JCA Binding from Create Resource list



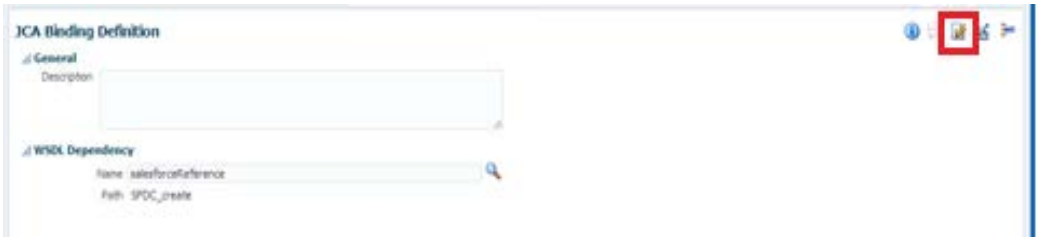
13. The JCA Binding creation dialog box is displayed. Click on **Browse** and browse to the directory where the artifacts received from the JDeveloper 12c are stored. Choose the Oracle Cloud Adapter for Salesforce.com JCA file, as shown in the Figure below.

Figure 6- 8 Create Adapter JCA Binding



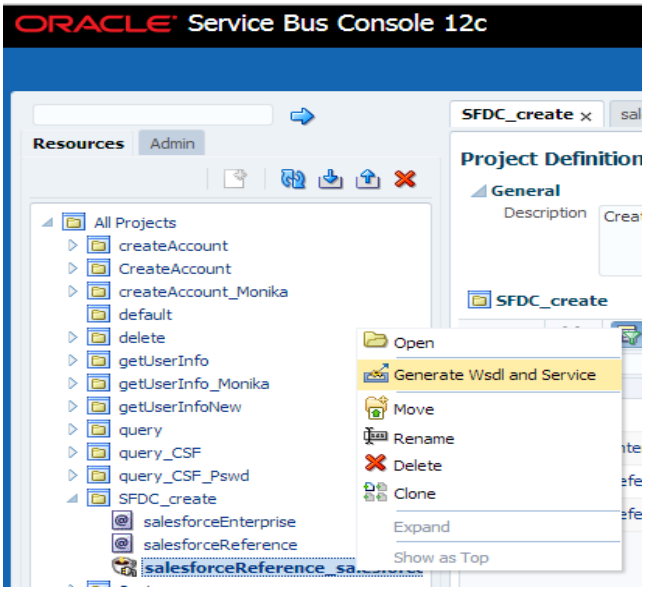
14. Click **Create**. You are returned to the project page.
15. You may get an error message as “The JCA Binding ‘salesforceReference_salesforce’ was successfully created with validation errors. View the JCA Binding/Conflicts to see detailed diagnostic messages.” This error is thrown because JCA binding cannot find the associated WSDL.
16. To rectify the above error, click on the JCA file created and then click on **Edit**.
17. Click **Browse**. Search for the WSDL you created and click **Submit** after choosing the WSDL, as shown in the Figure below.

Figure 6- 9 Edit JCA Binding References



18. Click **Save**.
19. Go back to the project folder.
20. Right-click on the JCA Binding just created, and click on “Generate WSDL and Service” option as shown in the Figure below.

Figure 6- 10 Generate WSDL and service from JCA Binding

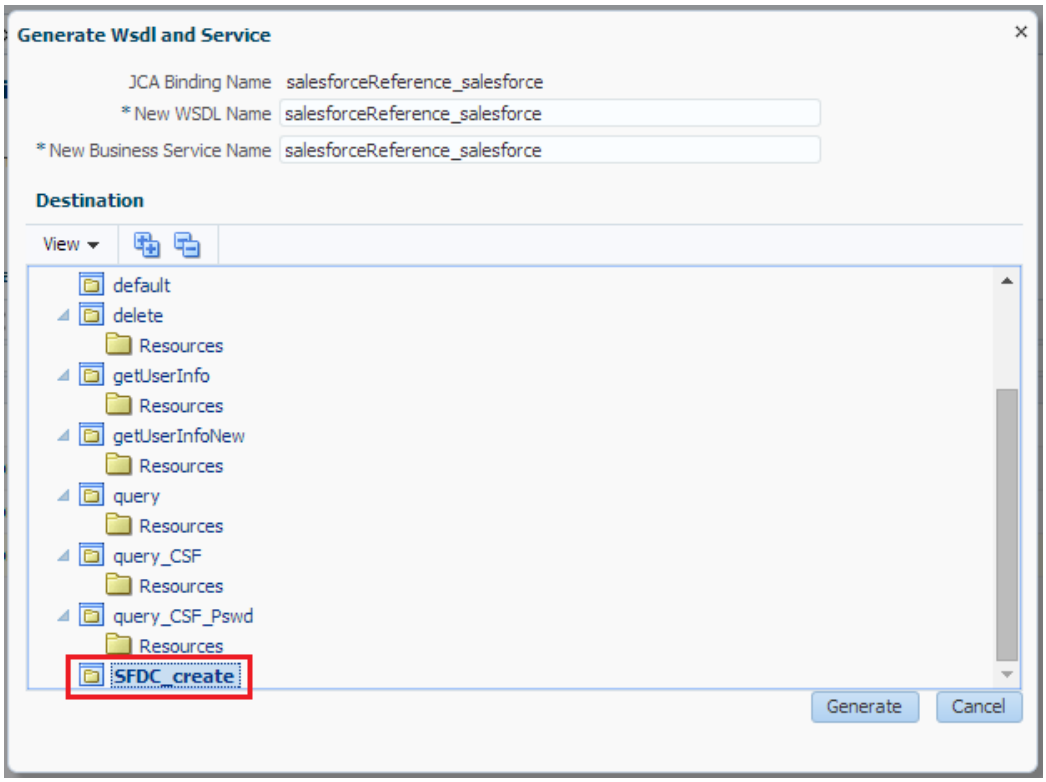


21. Enter a new name for the WSDL name in the **New WSDL Name** field and the service name in the **New Service Name** field.

> **Note:** Choose the correct location for the new WSDL and service to be generated.

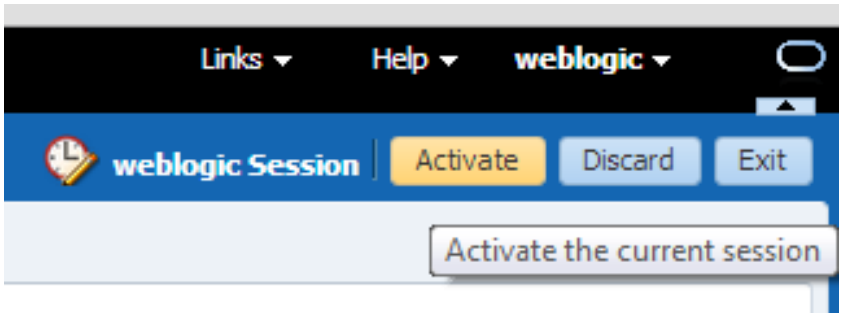
22. Click **Generate**, as shown in the Figure below.

Figure 6- 11 Generating WSDL and Business Service for Salesforce



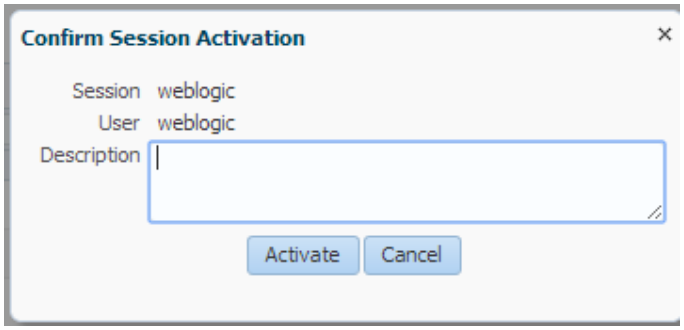
23. The new WSDL and the new business service are generated.
24. Activate the OSB session by clicking on **Activate** in the right corner, as shown in the Figure below.

Figure 6- 12 Activate Session



25. A **Confirm Session Activation** dialog box will appear. Click on **Activate** to activate the session as shown in the Figure below.

Figure 6- 13 Confirm Session Activation



Click **Activate** on the Activate Session page. Once the session is activated, all the generated artifacts and the services are deployed to the OSB server.

6.2.2 Creating OSB Projects Using JDeveloper

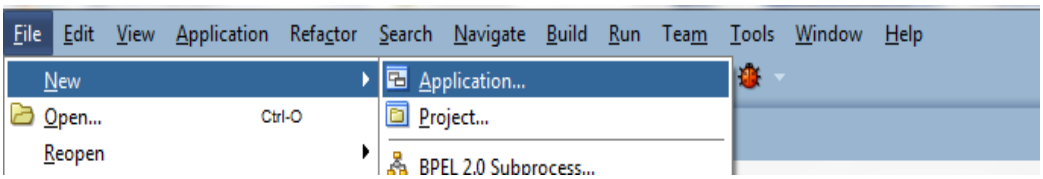
This section explains how to create OSB projects using JDeveloper 12c. It includes creating an empty composite for OSB, then defining an OSB outbound process and finally deploying that OSB project on server.

6.2.2.1 Create an Empty Composite for OSB

Perform the following steps to create an empty composite for OSB:

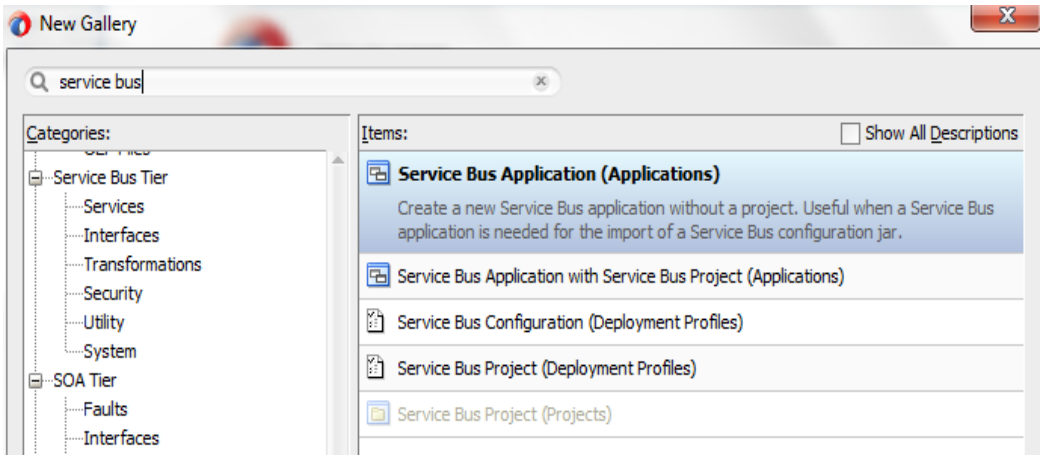
1. Create a new OSB application, Select **File → New → Application**, as shown in the Figure below.

Figure 6- 14 New Application Page



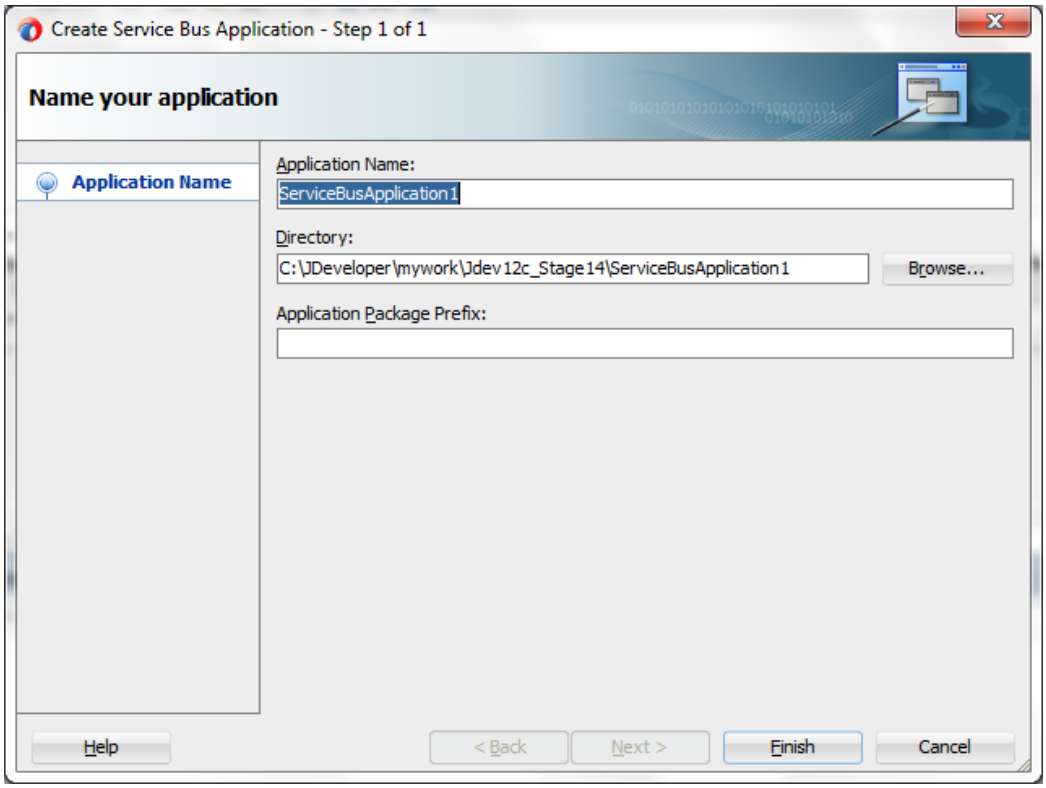
The New Gallery page is displayed as shown in Figure 6- 14.

Figure 6- 15 Choose Application



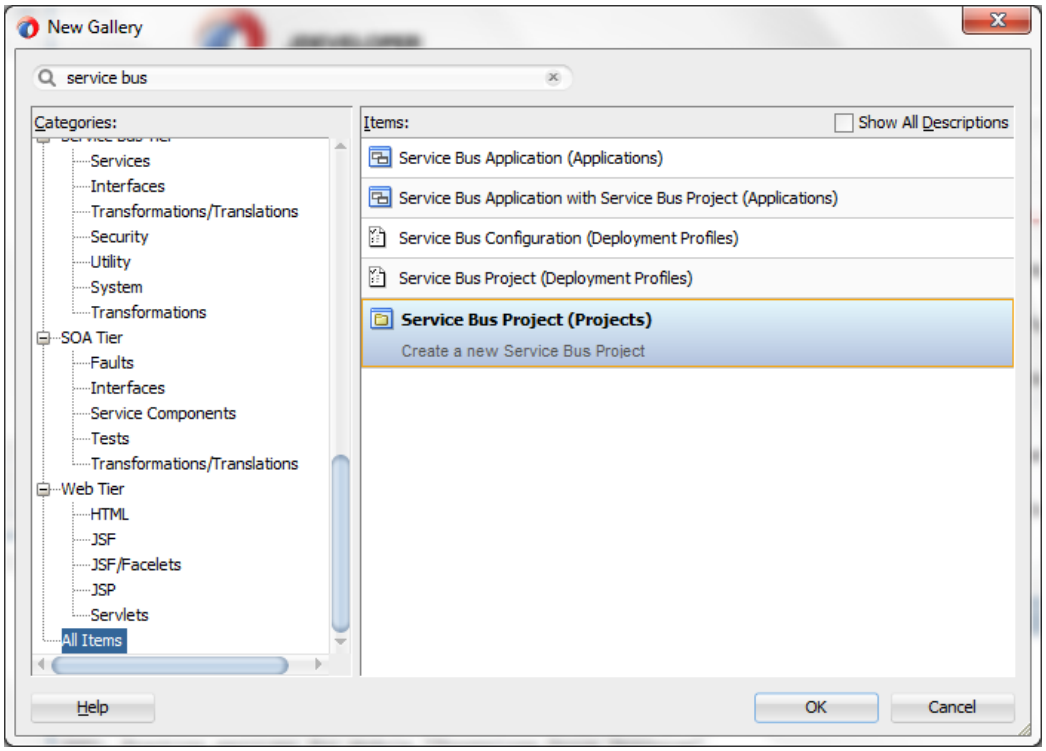
2. Enter a name for the new SOA Application and click **Finish**, as shown in the Figure below.

Figure 6- 16 Name Your Application



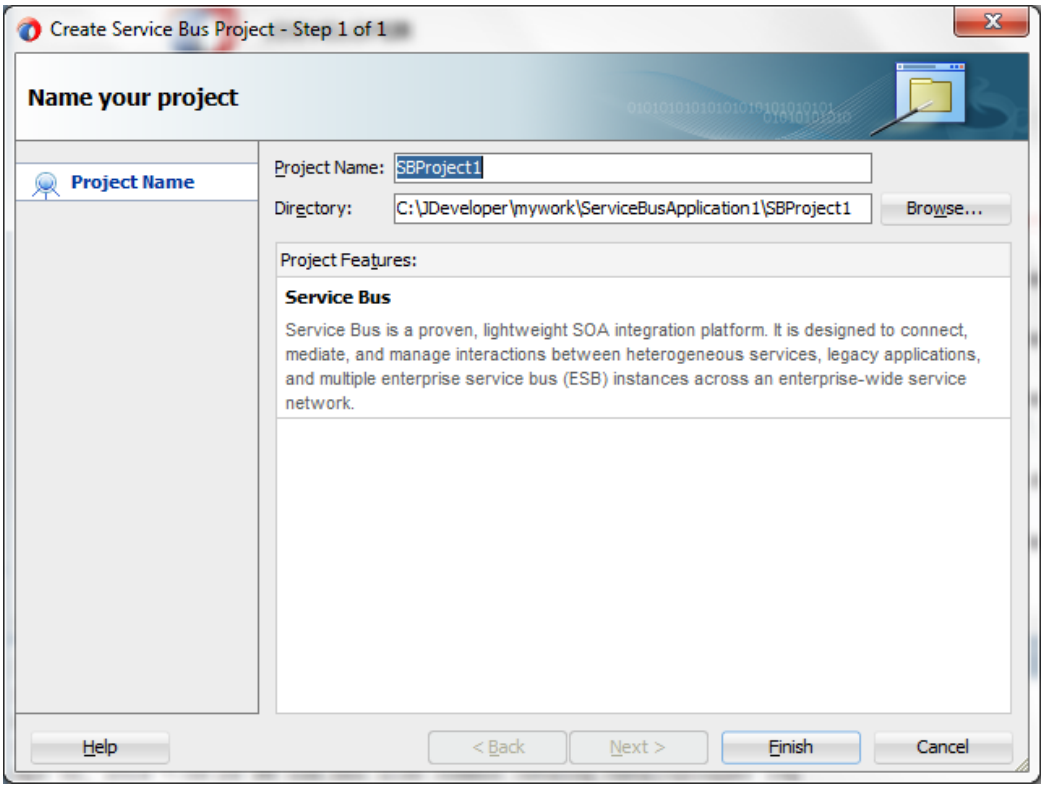
3. Create a new OSB application, Select **File → New → Project**, as shown in the Figure below.

Figure 6- 17 Create a new Project



4. The Name Your Project page is displayed, as shown in the Figure below.

Figure 6- 18 Name Your Project



5. Click **Finish**.

6.2.2.2 Define an OSB Outbound Process

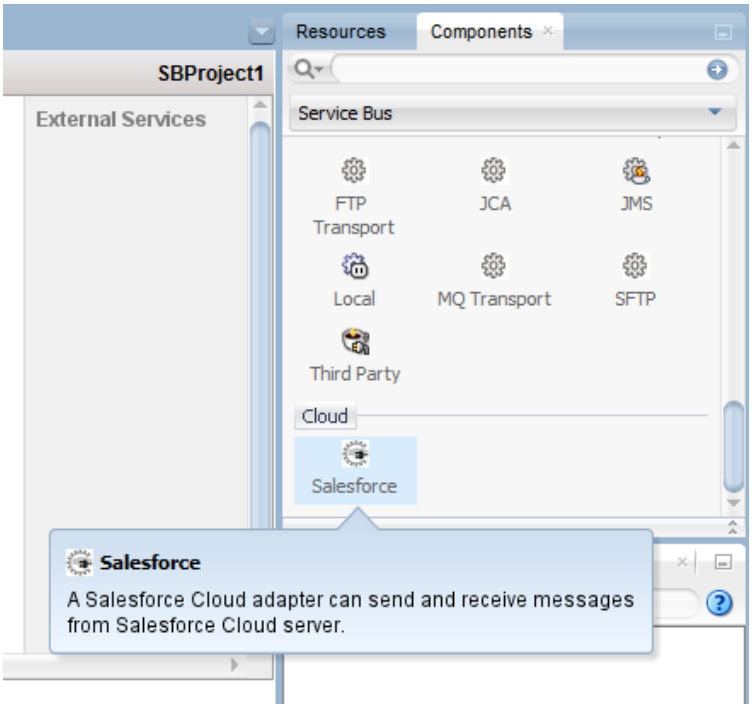
This section describes how to define an OSB outbound process for the Salesforce.com integration using Oracle Cloud Adapter for Salesforce.com, which consists of the following stages:

1. Configure a Salesforce Adapter Component.
2. Configuring an Outbound OSB Process Component.

Configure a Salesforce Adapter Component

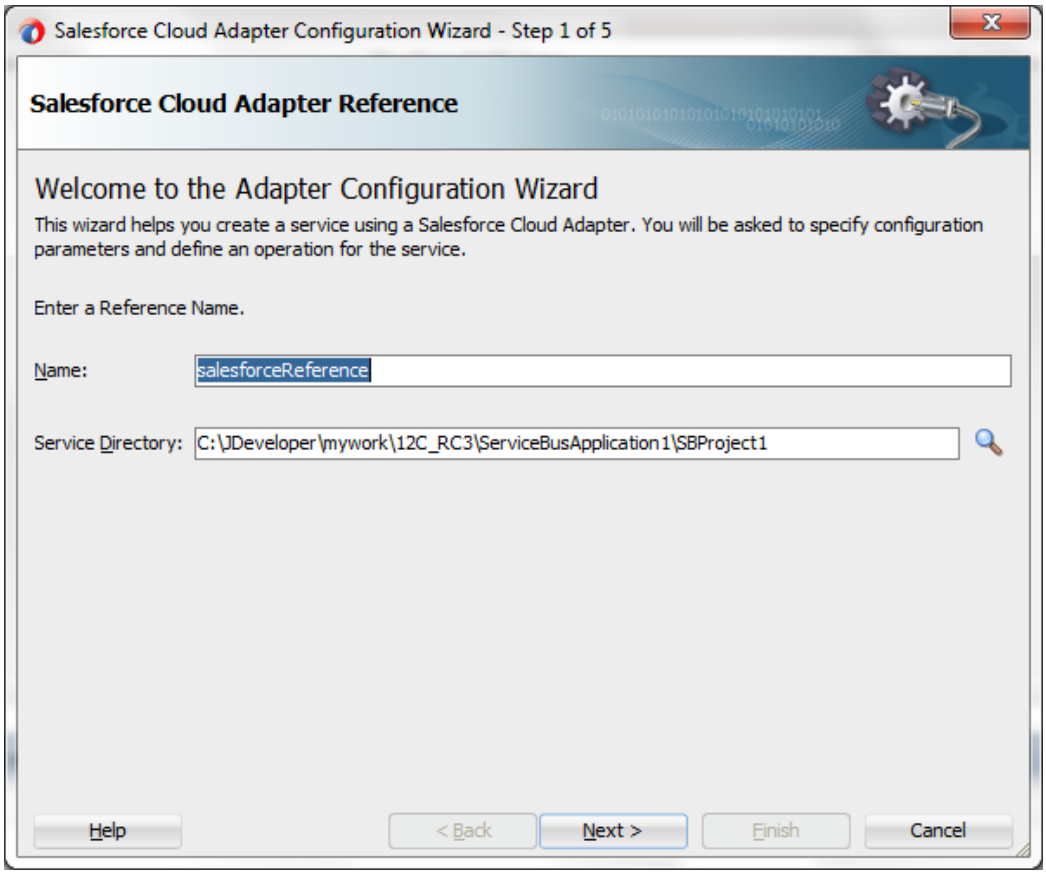
1. Open JDeveloper.
2. Drag and drop the Salesforce Adapter component from the **Resources Components** pane to the **External Service** pane, as shown in the Figure below.

Figure 6- 19 Salesforce Adapter Configuration Wizard



The **Welcome** page of the Adapter configuration wizard is displayed, as shown in the Figure below.

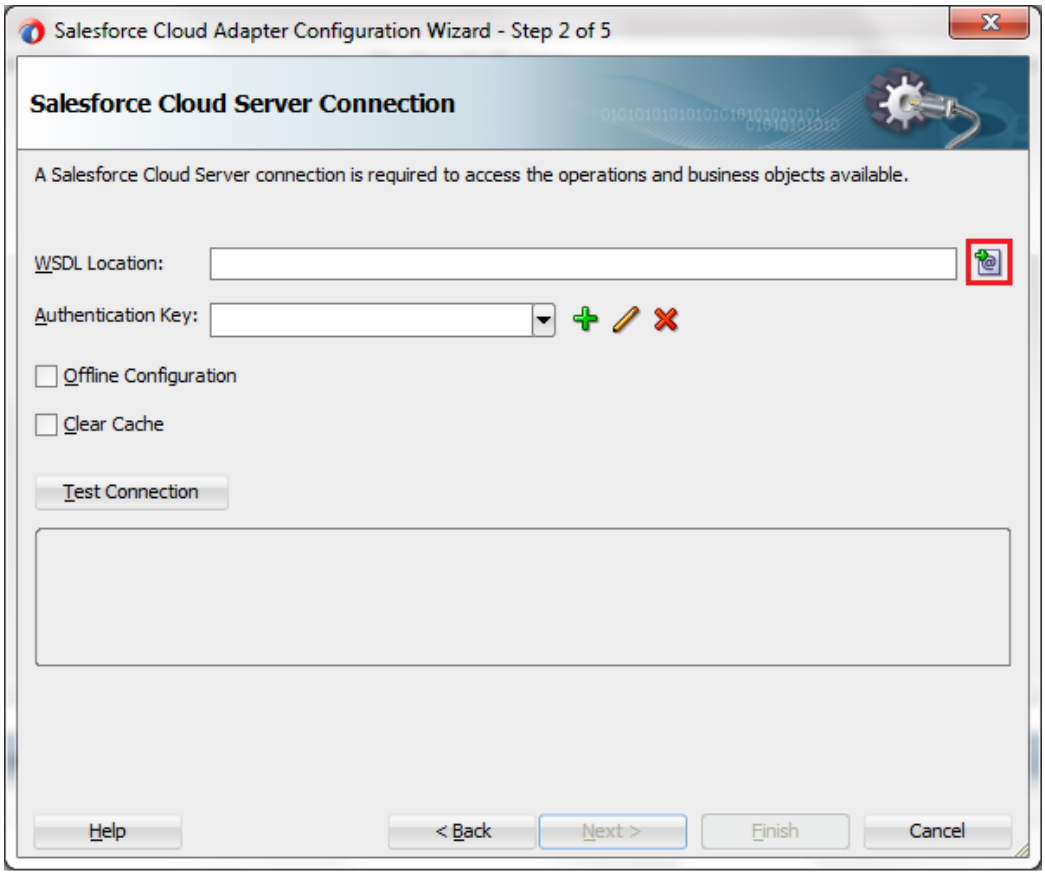
Figure 6- 20 Welcome Page



3. Enter a reference name for the Salesforce Adapter reference in the **Name** field and then click **Next**.

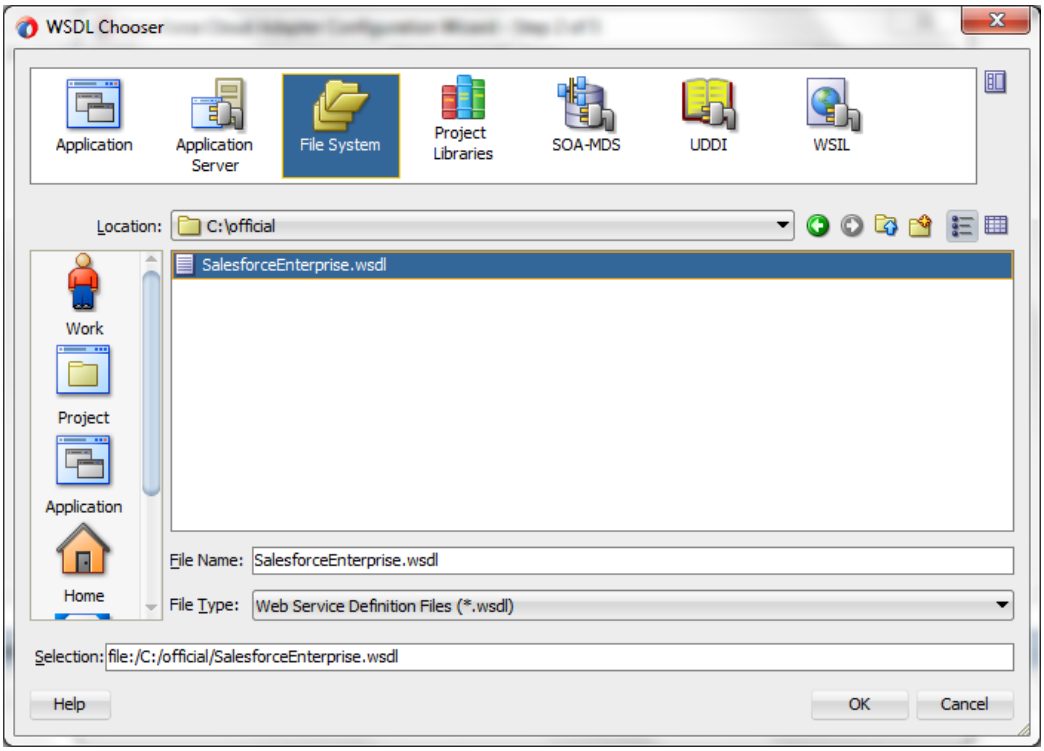
4. On the Connection Information page, browse for the Enterprise WSDL location by clicking on the browse button as highlighted in the Figure below.

Figure 6- 21 Connection Page



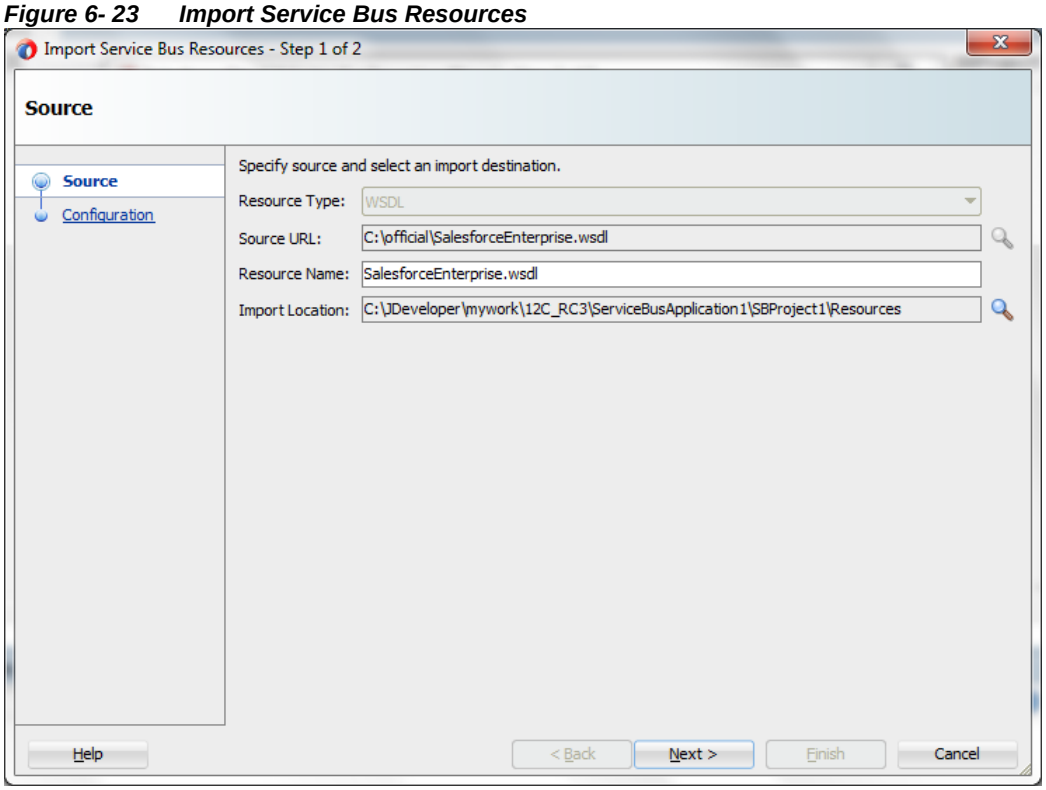
5. The **WSDL Chooser** dialog is displayed. Browse and select the downloaded Enterprise WSDL and click **OK**, as shown in the Figure below.

Figure 6- 22 SOA Resource Browser



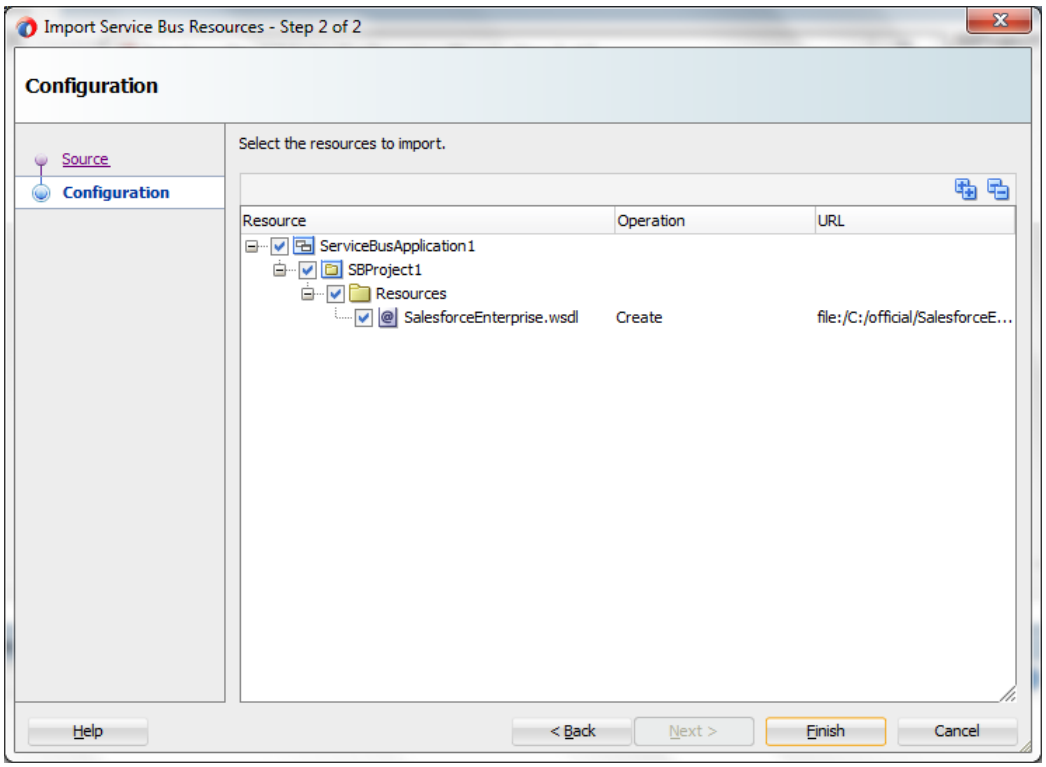
6. Click **OK**. The following screen appears as shown in the Figure below.

Figure 6- 23 Import Service Bus Resources



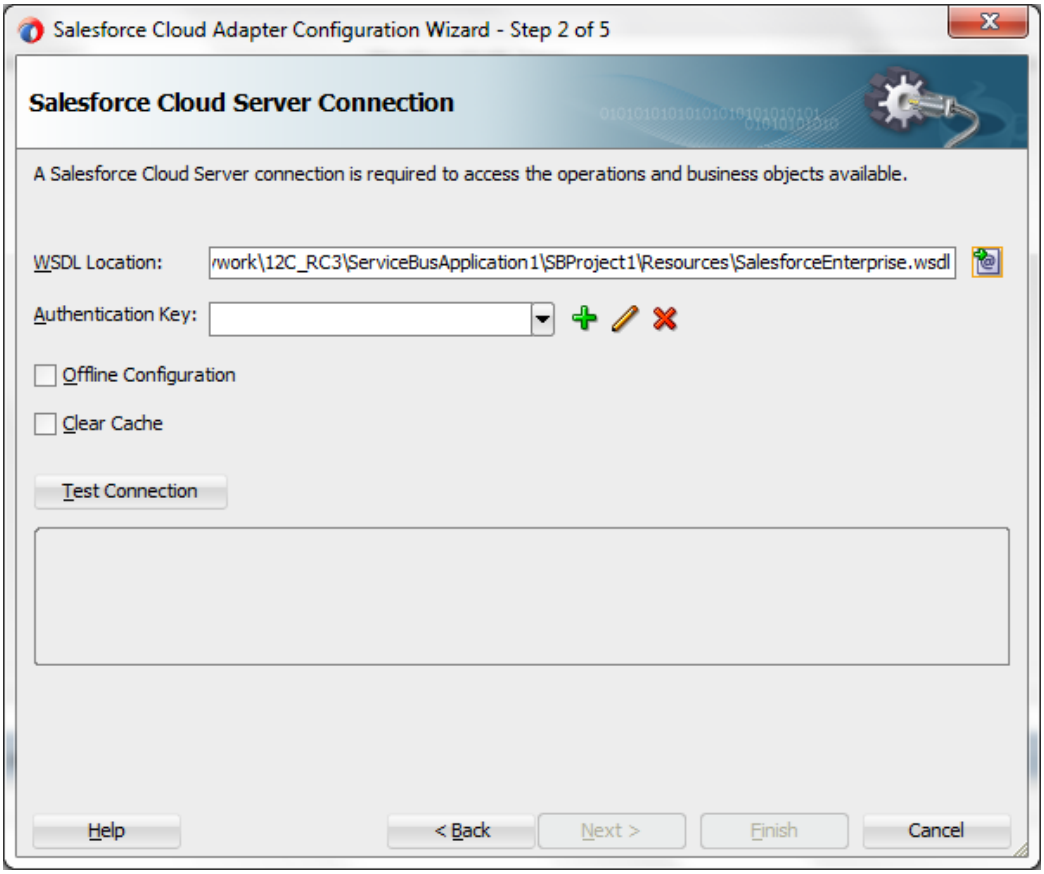
7. Click **Next**. The following screen appears as shown in the Figure below.

Figure 6- 24 Import Service Bus Resources



8. Click **Finish.** You are returned to the Salesforce Cloud Server Connection page.

Figure 6- 25 Connection Page

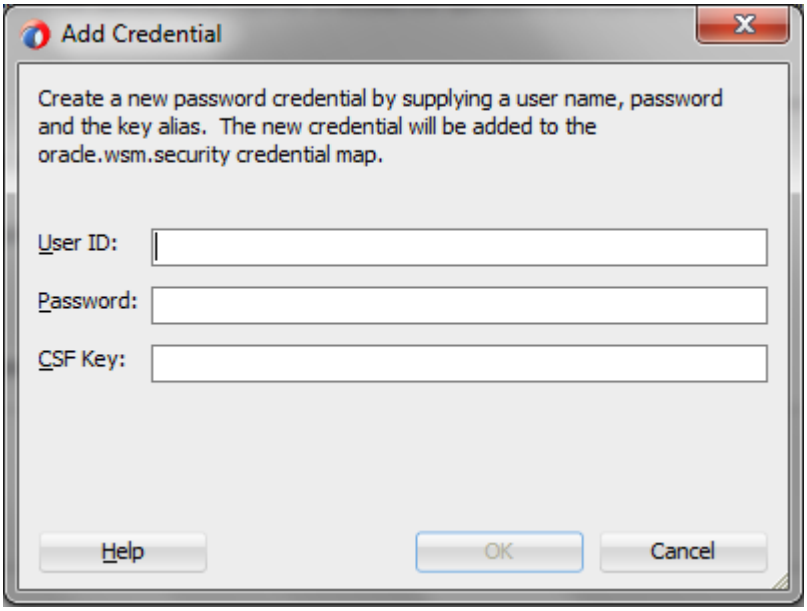


9. Select an authentication key from the dropdown, if available, or create a new Authentication key by clicking on "+" button.

10. The **Add Credential** dialog box is displayed, as shown in the Figure below. Provide a suitable name and the Salesforce.com credentials and click **OK**.

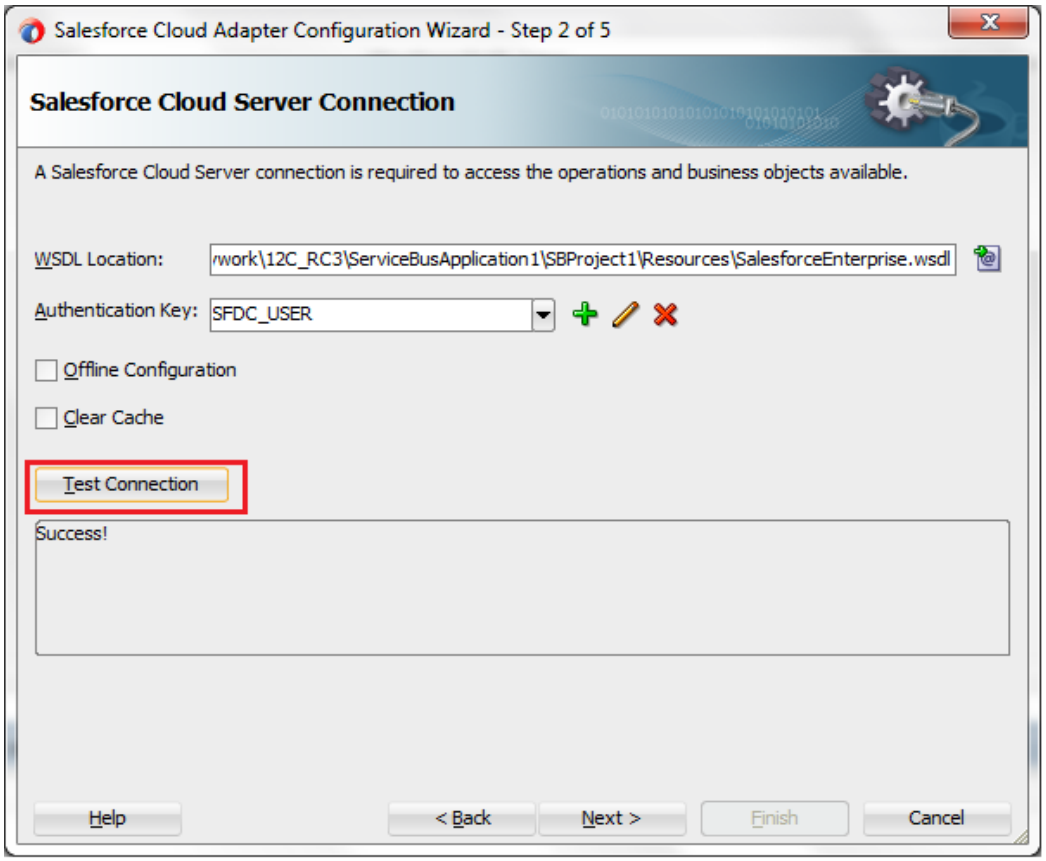
Note: The password should be a combination of Salesforce.com password and Salesforce.com Security Token.

Figure 6- 26 Add Credential



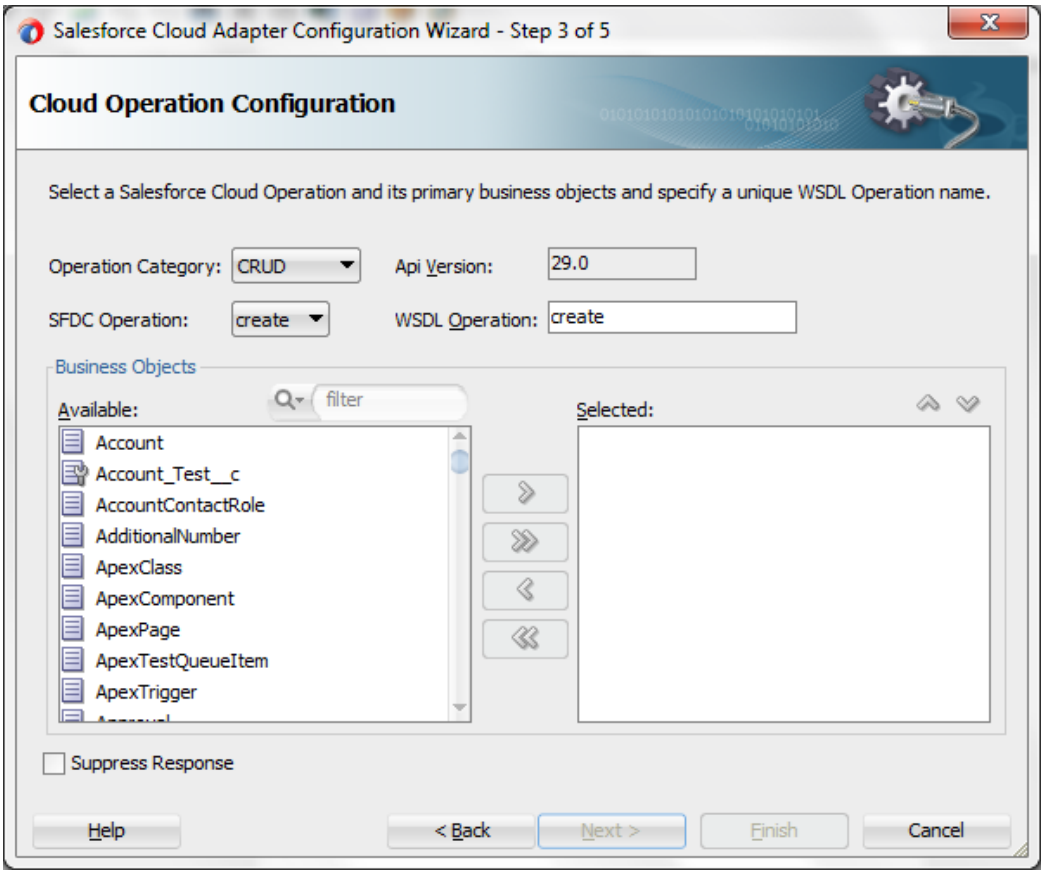
11. Click **Test Connection** button to validate the Authentication Key, as shown in the Figure below.

Figure 6- 27 Connection Page



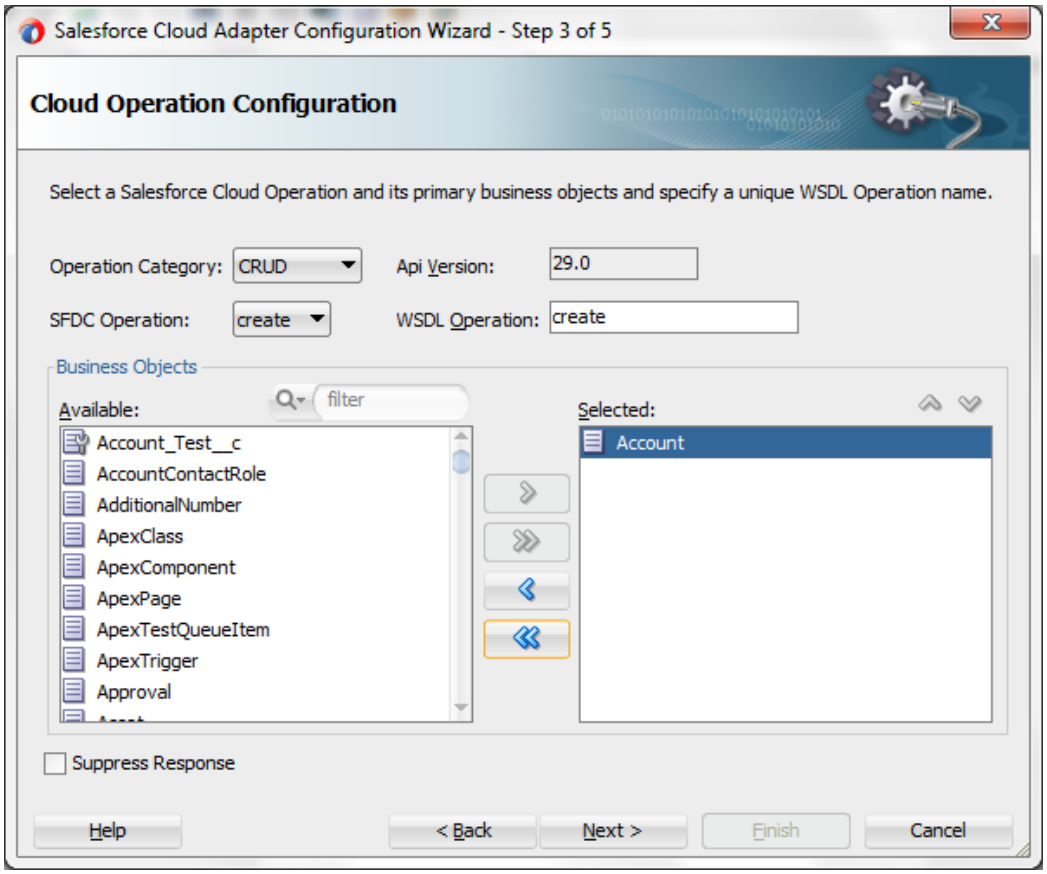
12. Click **Next.** The Operation Configuration page is displayed, as shown in the Figure below.

Figure 6- 28 Operation Configuration Page



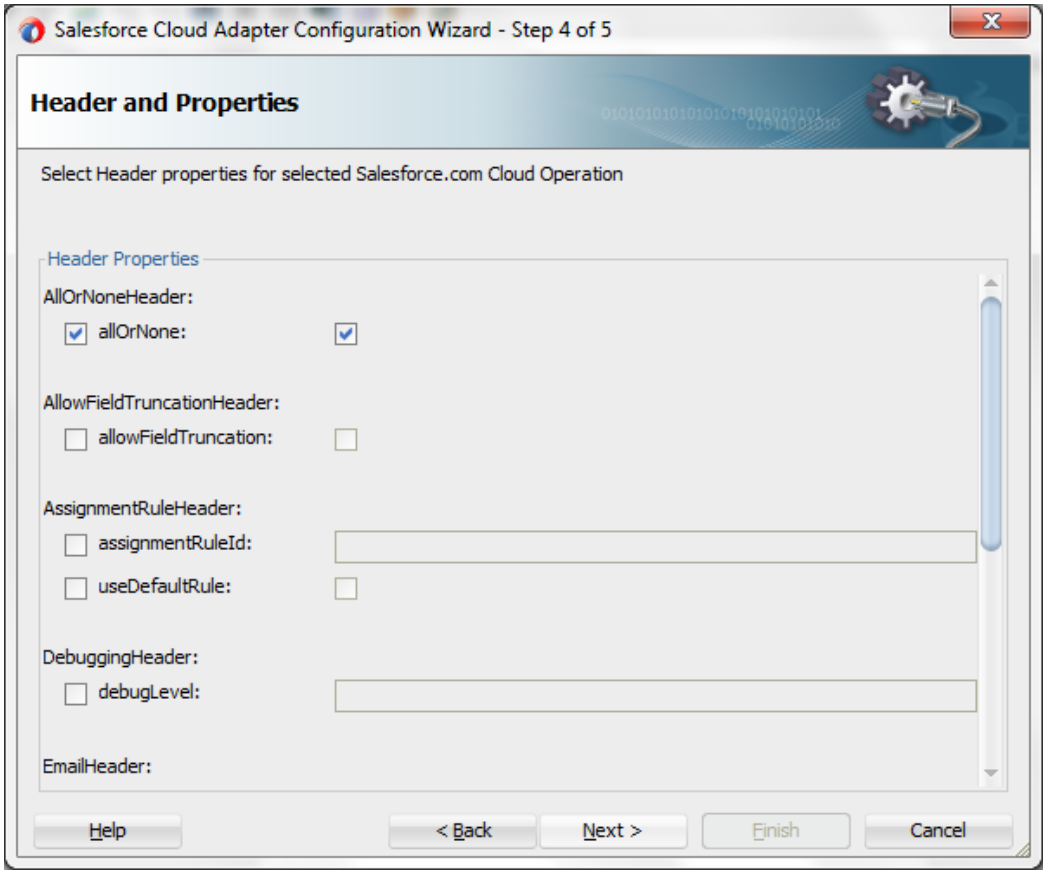
13. The default operation is create. Select the **Account** object from the Available objects list, and move to the selected objects list as shown in the Figure below.

Figure 6- 29 Operation Configuration Page



14. Click **Next**. The Header and Properties Page is visible as shown in the Figure below.

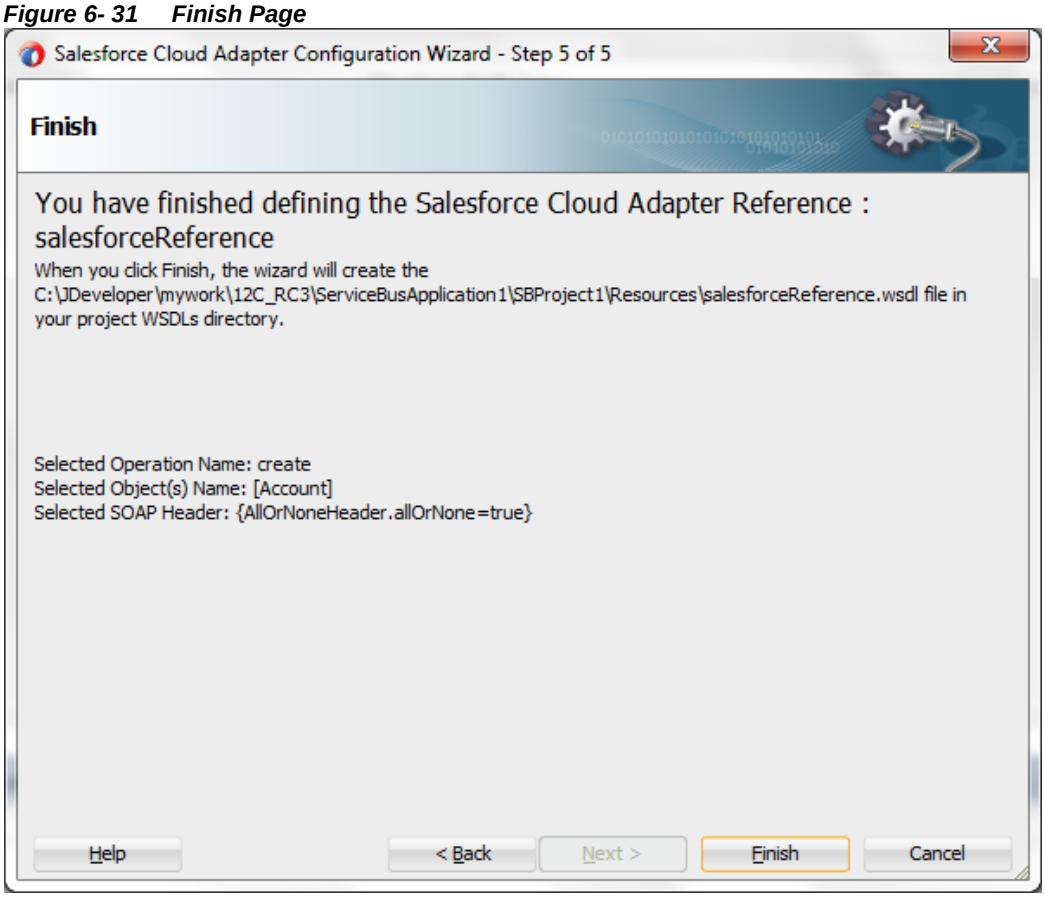
Figure 6- 30 Headers Page



15. Select any header and provide its value.
16. Click **Next**.

The Finish page is displayed, as shown in the Figure below.

Figure 6- 31 Finish Page

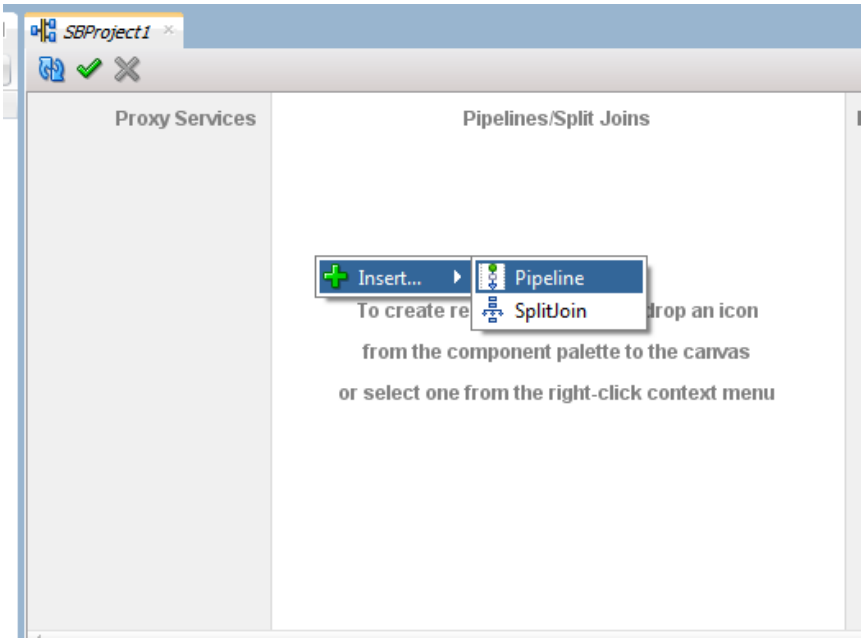


Configuring an Outbound OSB Process Component

Perform the following steps to configuring an Outbound OSB Process Component:

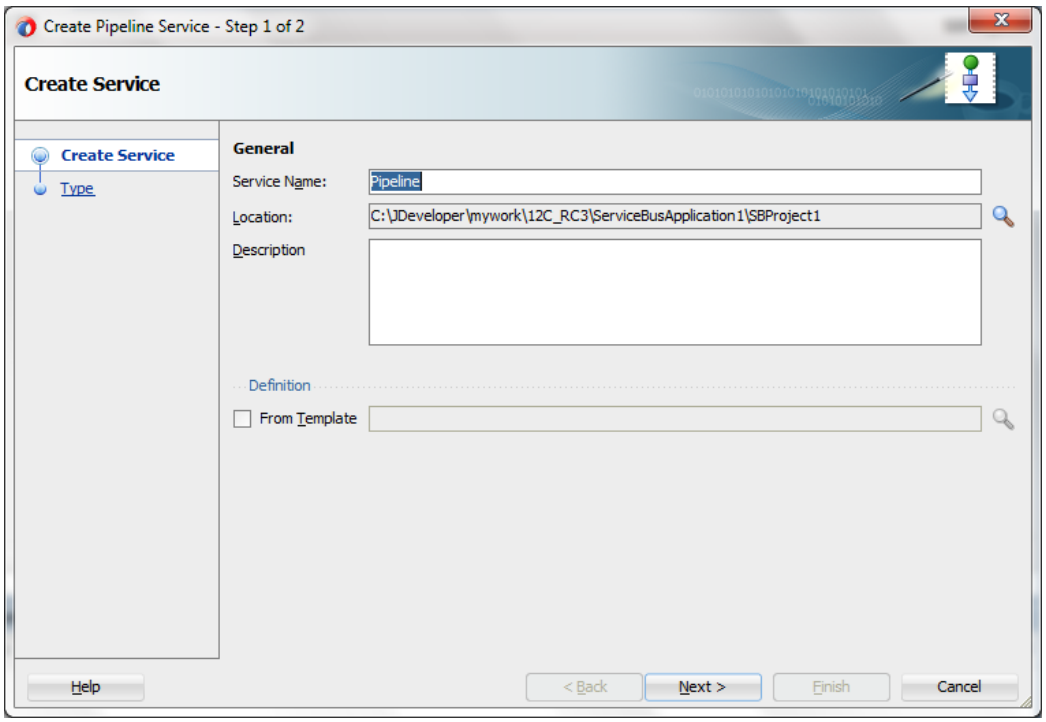
1. Right-click on **Pipeline/Split Joins** pane and click on **Insert** and then **Pipeline** as shown in the Figure below.

Figure 6- 32 Pipeline Component



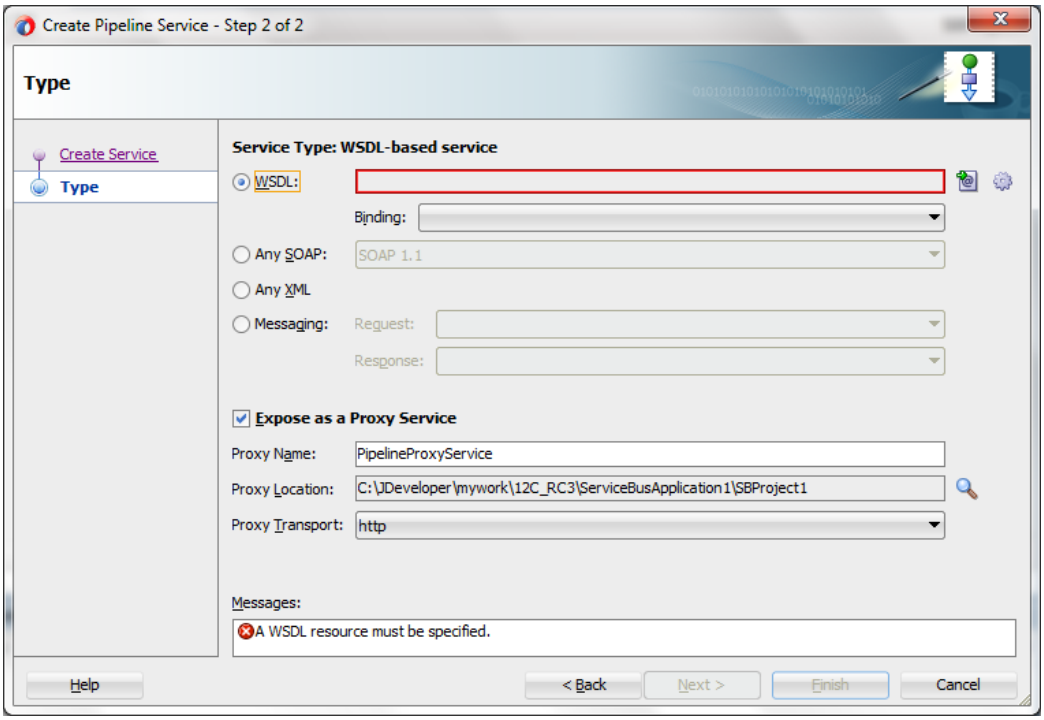
The Create Pipeline Service dialog is displayed, as shown in the Figure below.

Figure 6- 33 Create Pipeline Service



2. In the **Service Name** field, enter a name to identify the pipeline name and select the corresponding location of the project.
3. Click **Next** and select the **Service Type** as WSDL, as shown in the Figure below.

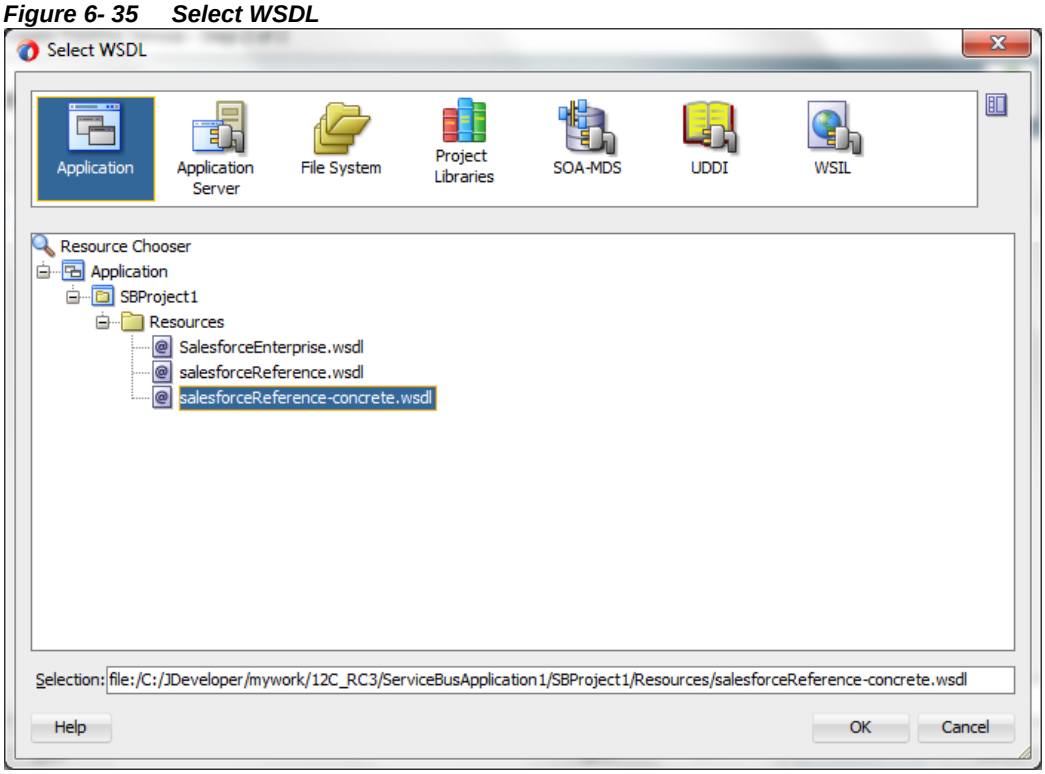
Figure 6- 34 Create Pipeline Service



4. Click **Browse** icon which is located to the right of the WSDL URL, to select the WSDL from the file system.

5. Select the appropriate WSDL file from the Application → Resources, as shown in the Figure below.

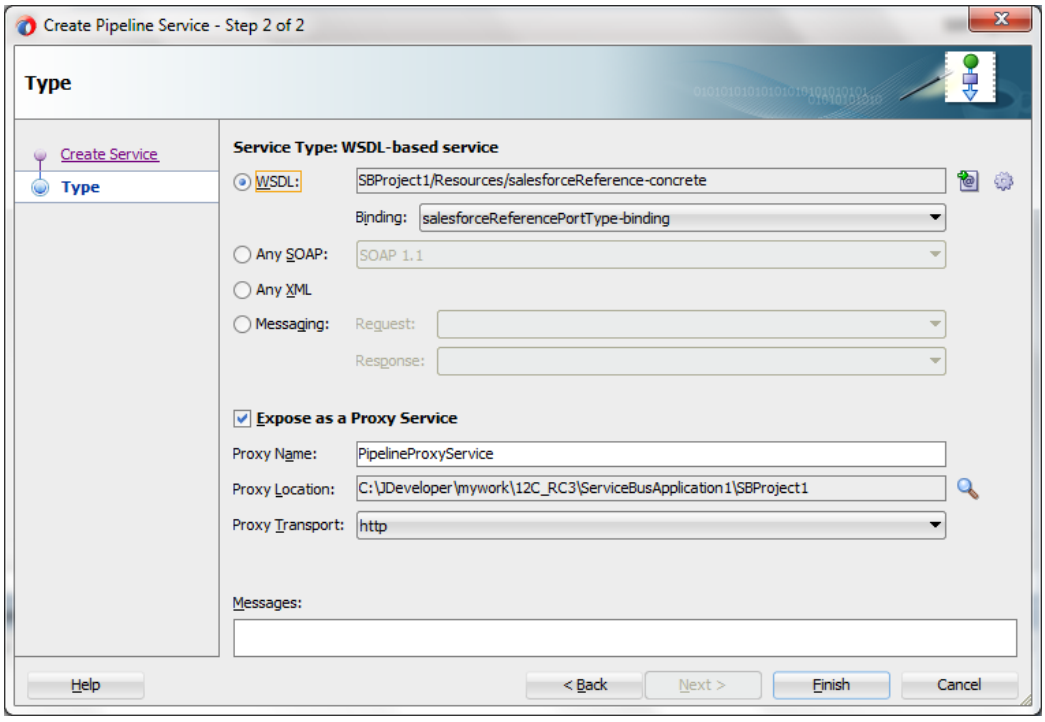
Figure 6- 35 Select WSDL



6. Click **OK**.

The selected WSDL and corresponding binding is displayed, as shown in the Figure below.

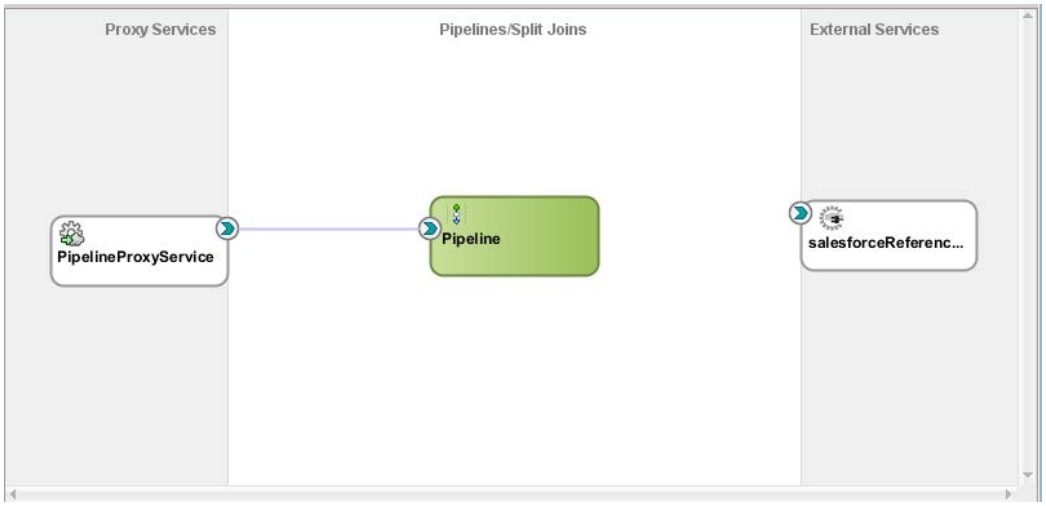
Figure 6- 36 Create Pipeline Service



7. Select checkbox for **Expose as a Proxy Service**.
8. Select **Proxy Transport** as **http**.
9. Click **Finish**.

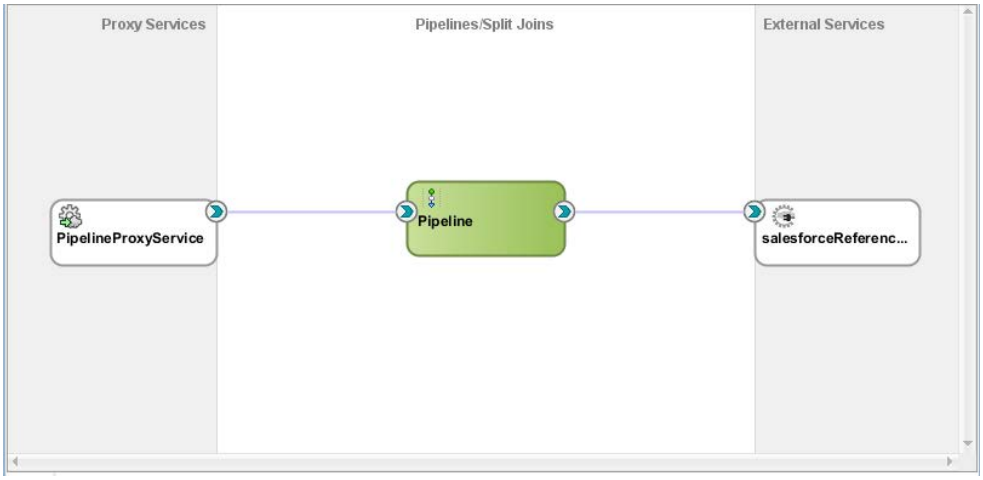
The Pipeline component is displayed as shown in the Figure below.

Figure 6- 37 Pipeline Component



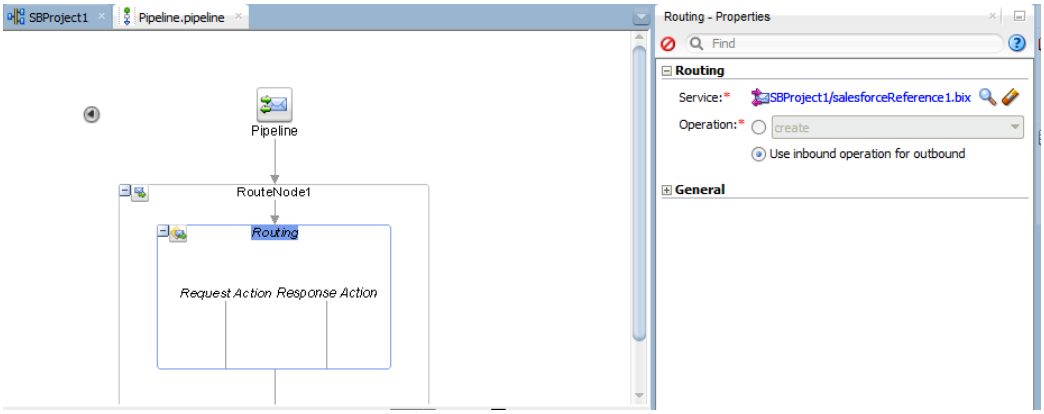
10. Connect **salesforceReference** to the **Pipeline**, as shown in the Figure below..

Figure 6- 38 Pipeline Component



11. Open the pipeline which shows the default routing. Verify the service and corresponding operation will be displayed in the Routing Properties, as shown in the Figure below.

Figure 6- 39 Routing Properties



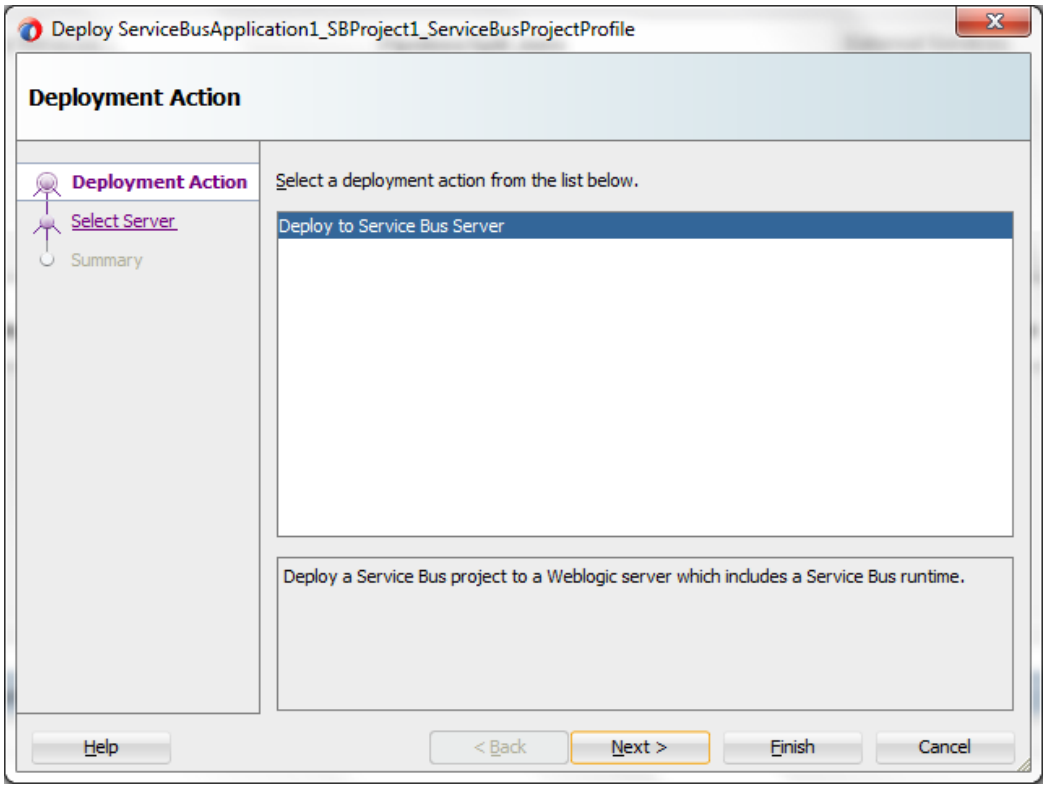
The outbound endpoint is ready to be deployed.

6.2.2.3 Deploying Outbound OSB Process

Perform the following steps to deploy the outbound OSB Process:

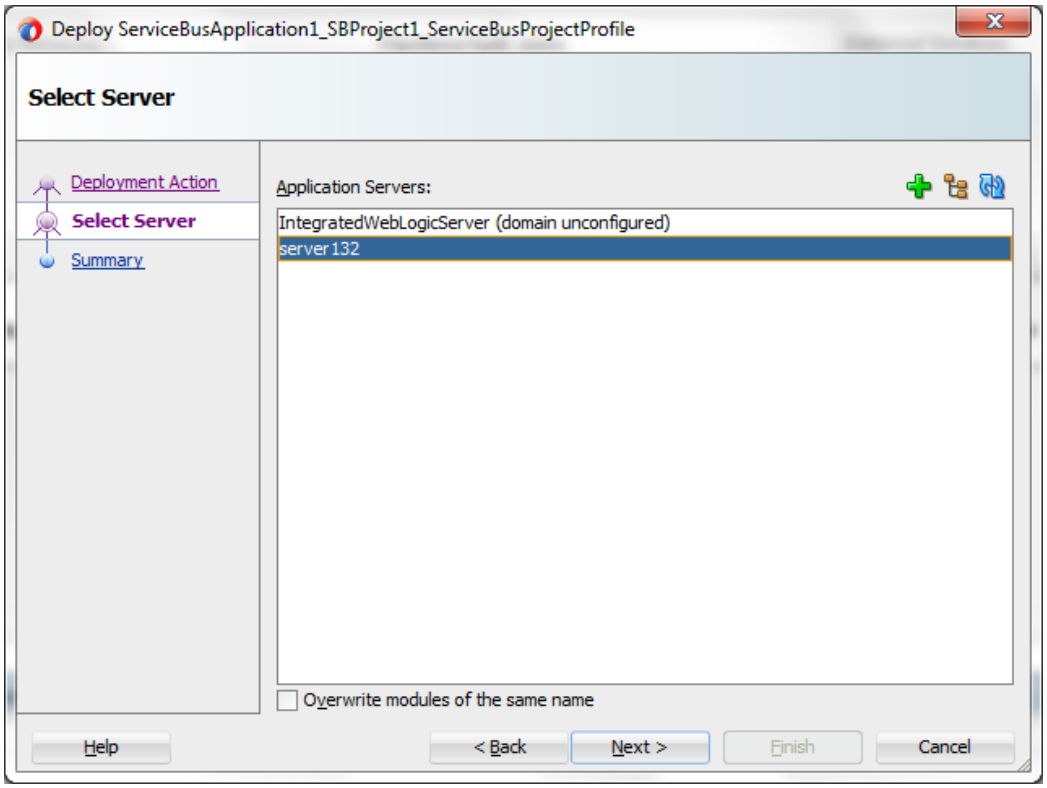
1. Select the project and **Deploy to Service Bus Server**, as shown in the Figure below.

Figure 6- 40 Deployment Action Page



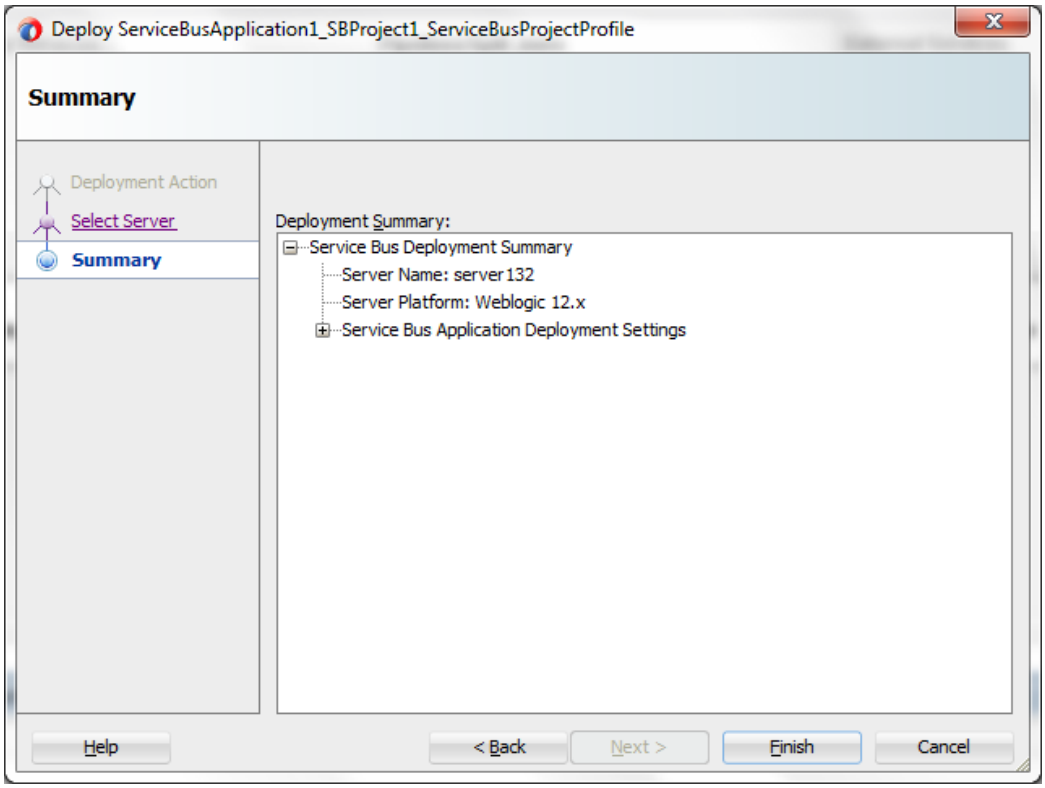
2. Select the already configured Application Server and click **Next**, as shown in the Figure below.

Figure 6- 41 Select Server Page



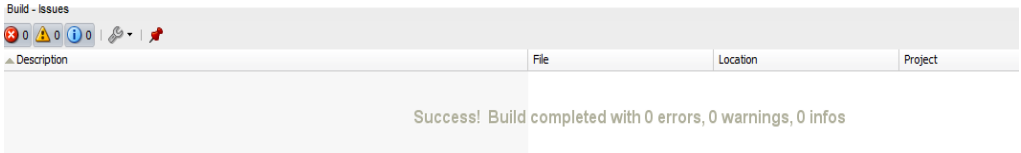
3. Check the deployment summary and click **Finish**, as shown in the Figure below.

Figure 6- 42 Summary Page



4. The Project is successfully deployed, as shown in the Figure below.

Figure 6- 43 Success Message Page



The successfully deployed project can be tested from service bus console.

6.3 Testing OSB project from Service Bus Console

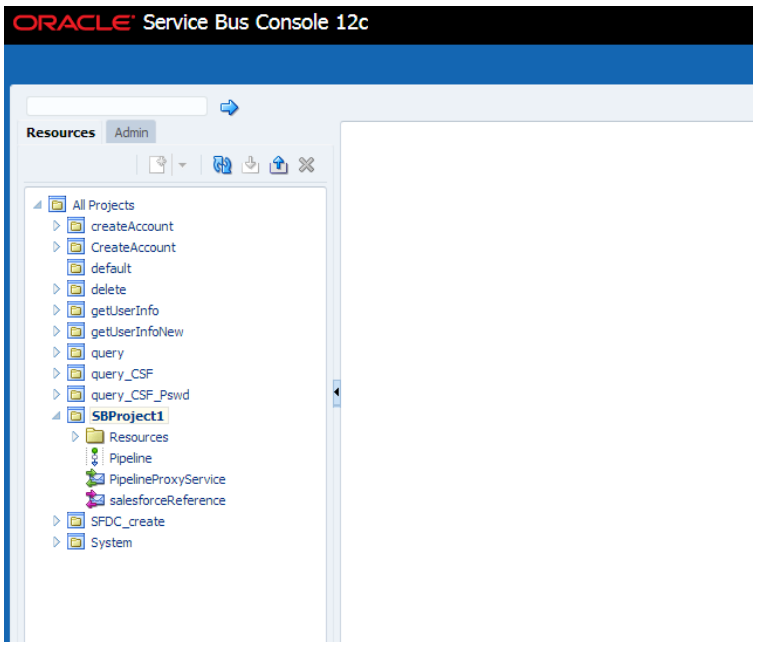
Follow the steps given below to test OSB project form Service Bus Console:

1. Open the **Service Bus Console** and enter user ID and Password, as shown in the Figure below.

Figure 6- 44 Service Bus Console

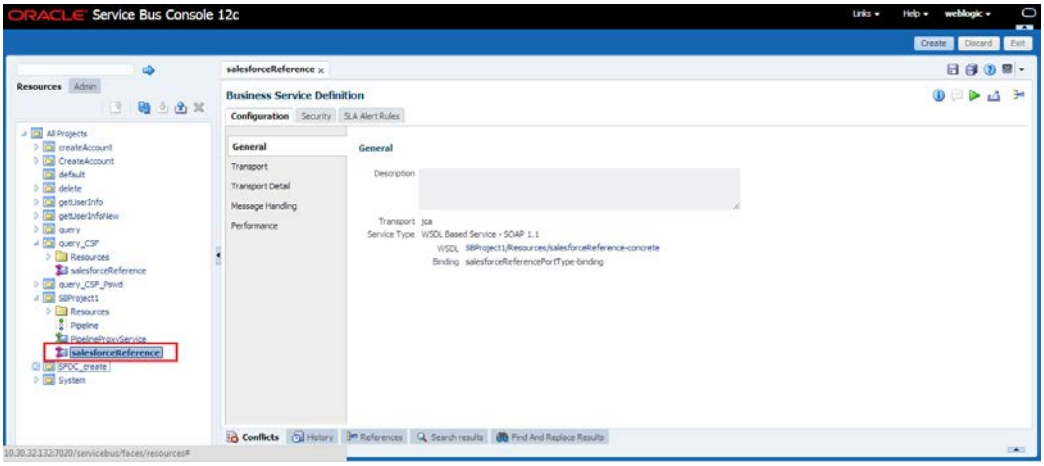
2. All the deployed projects are displayed under **All Projects**, as shown in the Figure below.

Figure 6- 45 Service Bus Console



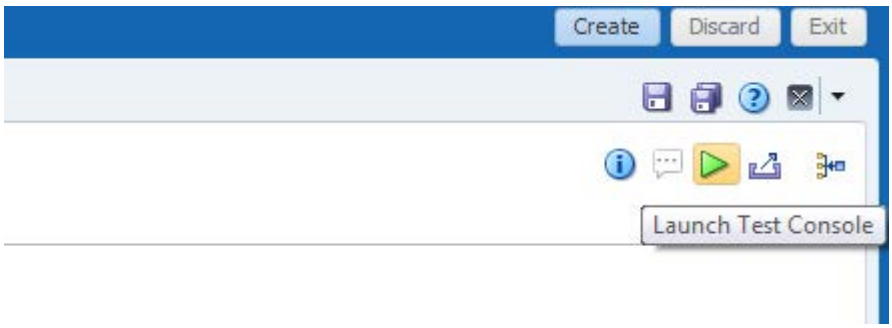
3. Open the project you want to test and click on Business Service of that project. For example, **salesforceReference** in this case, as shown in the Figure below.

Figure 6- 46 Business Service Definition



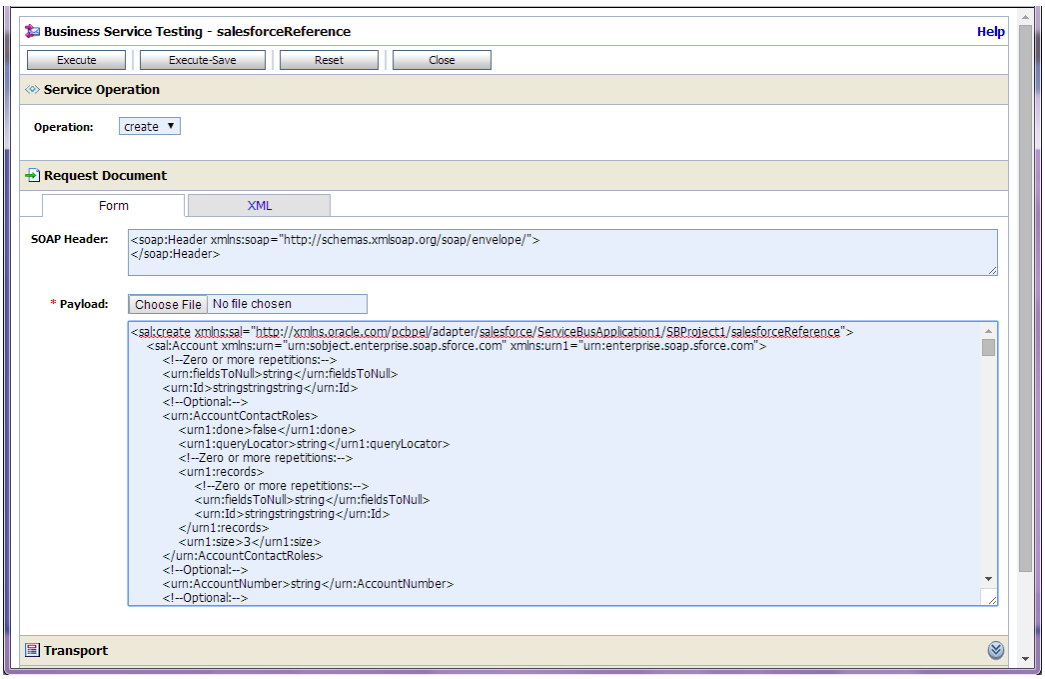
4. Option for **Launch Test Console** (Green arrow button) is displayed for testing the outbound endpoint, as shown in the Figure below.

Figure 6- 47 Launch Test Console



5. Launching Test Console opens new window displaying Business Service and the operation to test along with **Execute**, **Execute-Save**, **Reset** and **Close** button, as shown in the Figure below.

Figure 6- 48 Business Service Testing Page



Request Document section contains the Request Payload.

6. Provide the input and click on the **Execute** button.

This would send the payload to Salesforce and the response is displayed under **Response Document** section.

7

Configuring the Outbound Processing Using BPM

Oracle Cloud Adapter for Salesforce.com allows you to seamlessly integrate with Salesforce.com through the operations exposed by the Salesforce.com API. This chapter walks you through a simple integration wherein user creates a new account on Salesforce.com using Oracle Cloud Adapter for Salesforce.com. The scenario will enable you to create an easy and seamless integration to Salesforce.com using BPM.

This section describes how you can use Oracle Cloud Adapter for Salesforce.com to create the artifacts according to your business requirements, i.e. select salesforce.com objects and operations in the adapter configuration wizard. These generated artifacts are consumed in the BPM process. You can use the Adapter configuration wizard to both create and edit the Adapter services.

7.1 Overview

Account is one of the various standard objects provided by Salesforce.com. An account on Salesforce.com represents an individual or business associated with your organization. Oracle Cloud Adapter for Salesforce.com allows you to create an account on Salesforce.com. The scenario shown in the following sections will walk you through the same using BPM.

To integrate using a BPM process, the organization's Enterprise WSDL should be available to the user. The underlying adapter services must be exposed as WSDL files, which are generated during design-time in the configuration wizard of the Oracle Cloud Adapter for Salesforce.com. For more information, refer to the section "A.1 Generating the Enterprise WSDL".

The generated WSDL files are used to design the appropriate BPM services for outbound adapter services. A completed BPM process must be successfully compiled in JDeveloper and deployed to a SOA Suite server. Upon deployment to a SOA Suite server, every newly deployed process can be seen in the Oracle Enterprise Manager console, where you can run, monitor, administer BPM processes, and monitor adapter events.

7.2 Configuring the CSF Key on Enterprise Manager Console

For steps to configuring the CSF key on Enterprise Manager Console, refer to the section "CSF Key in Enterprise Manager".

7.3 Designing a Composite for Service Integration

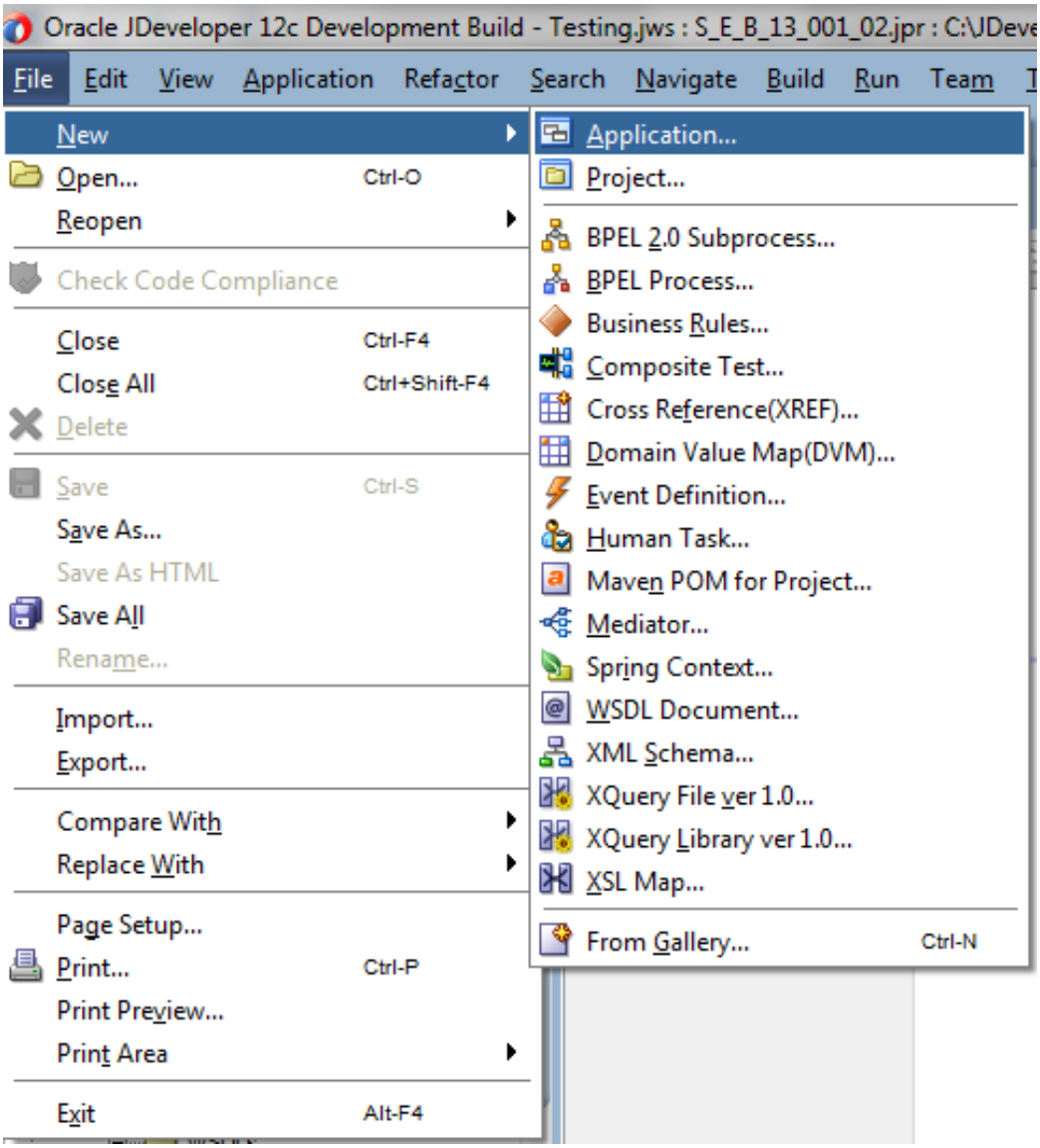
This section walks you through the steps to configure the Oracle Cloud Adapter for Salesforce.com with a BPMN process in Oracle SOA Suite and to deploy it to the WebLogic server. The composite can then be executed from the Enterprise Manager Console.

7.3.1 Define Composite for BPM

Perform the following steps to define a composite for BPM

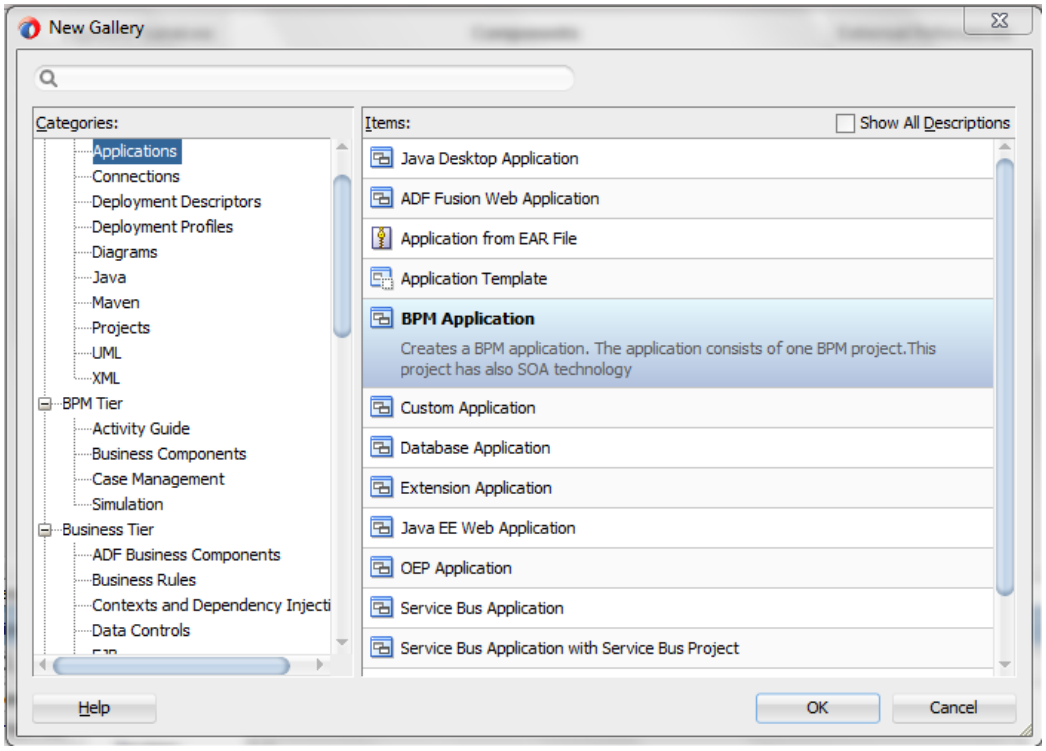
1. In the **File** menu of JDeveloper, click **New** and select **Application**.

Figure 7- 1 Navigation Window



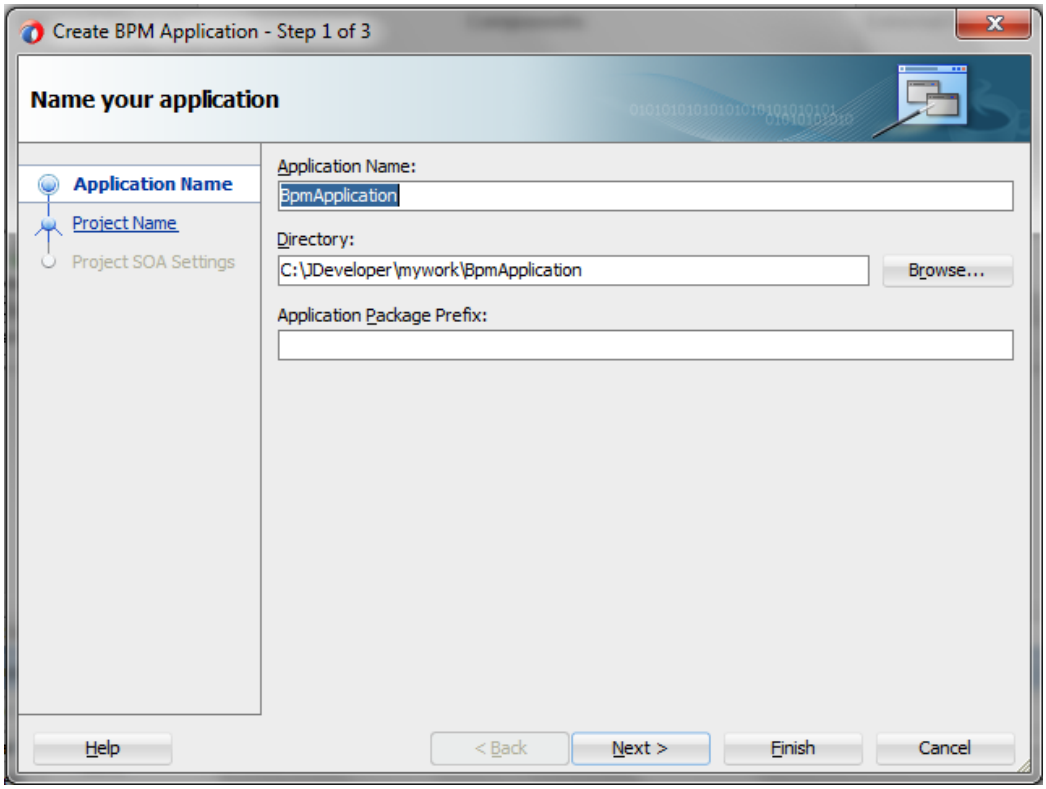
2. The **New Gallery** page is displayed. Select **BPM Application** from the **Items** list, as shown in the Figure below.

Figure 7- 2 Create BPM Application



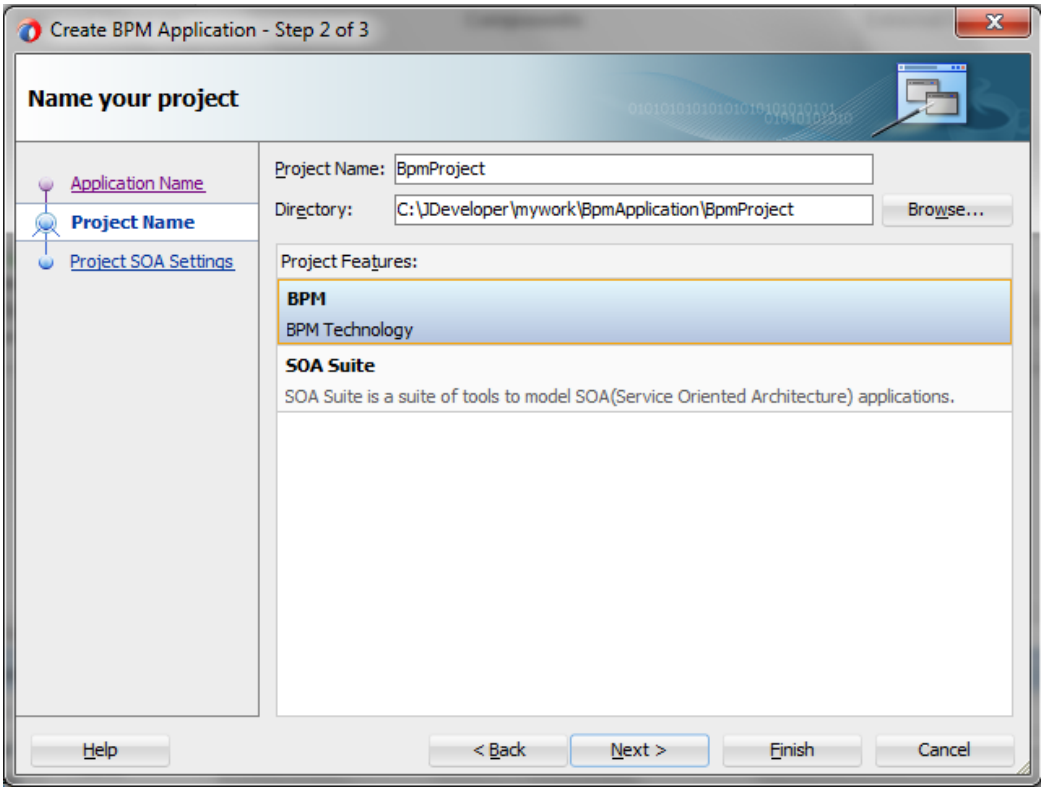
3. Provide a suitable name to your application, as shown in figure below.

Figure 7- 3 Name your application



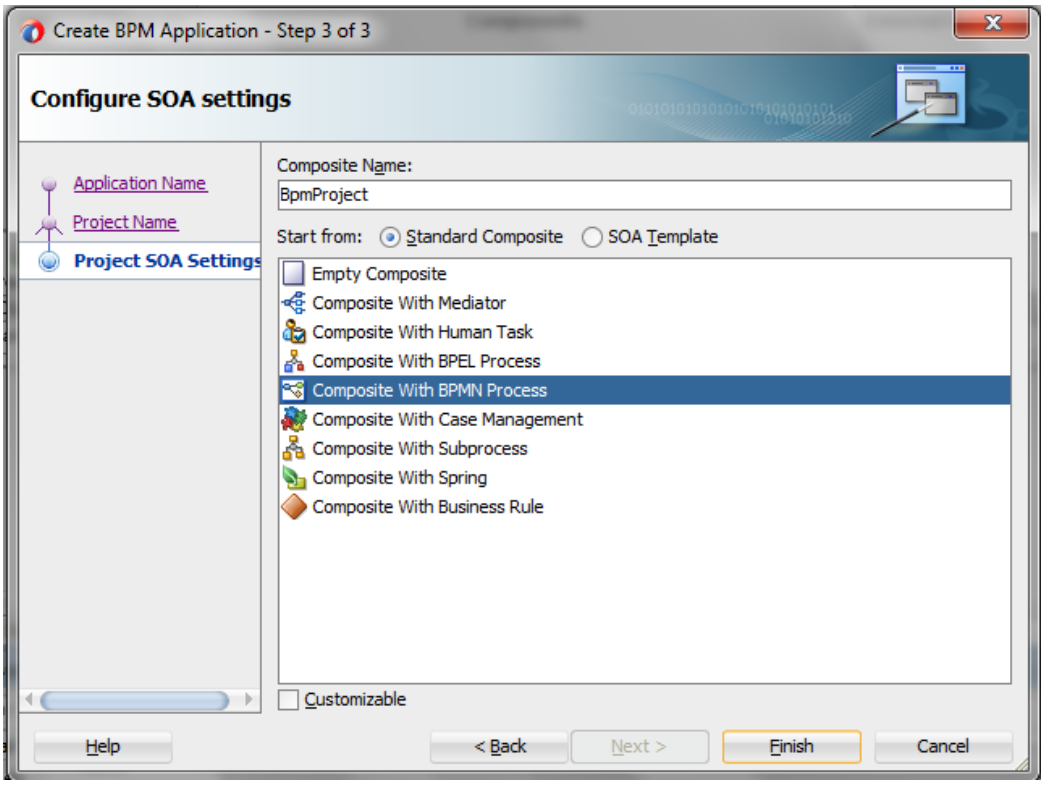
4. Click **Next** and provide a suitable name to your project, as shown in figure below.

Figure 7- 4 Name your project



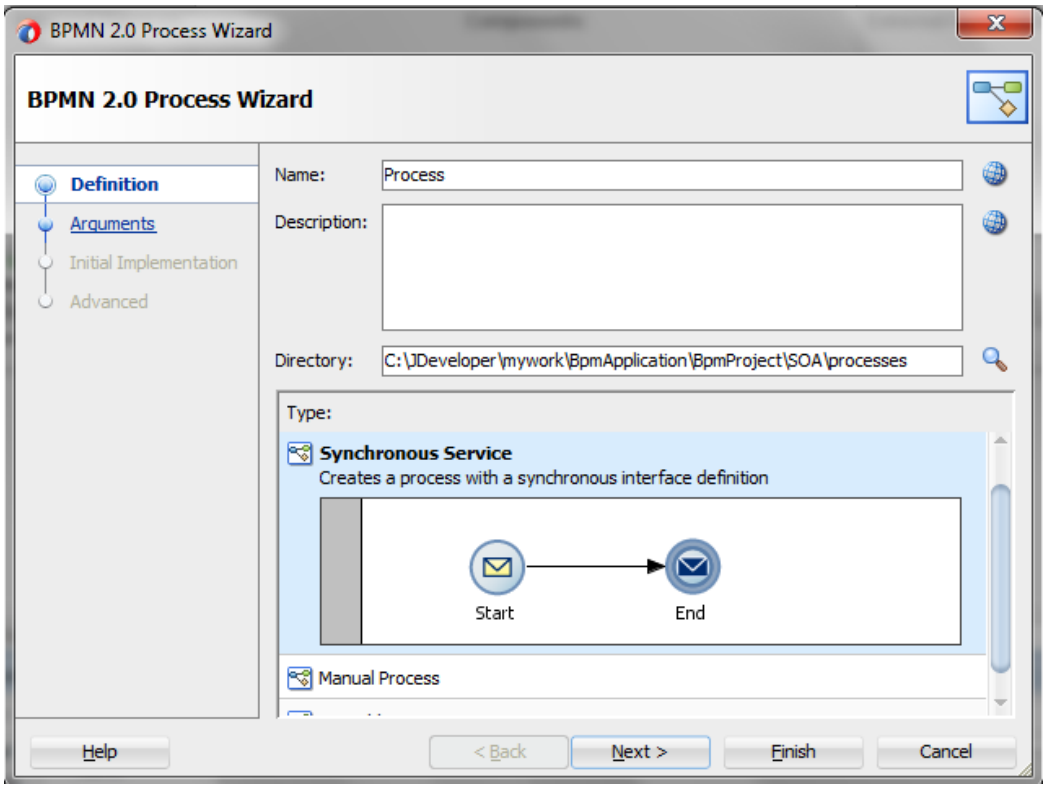
5. Click **Next.**
6. Select **Composite with BPMN Process** from **Standard Composite** list, as shown in the Figure below.

Figure 7- 5 Configure SOA Setting



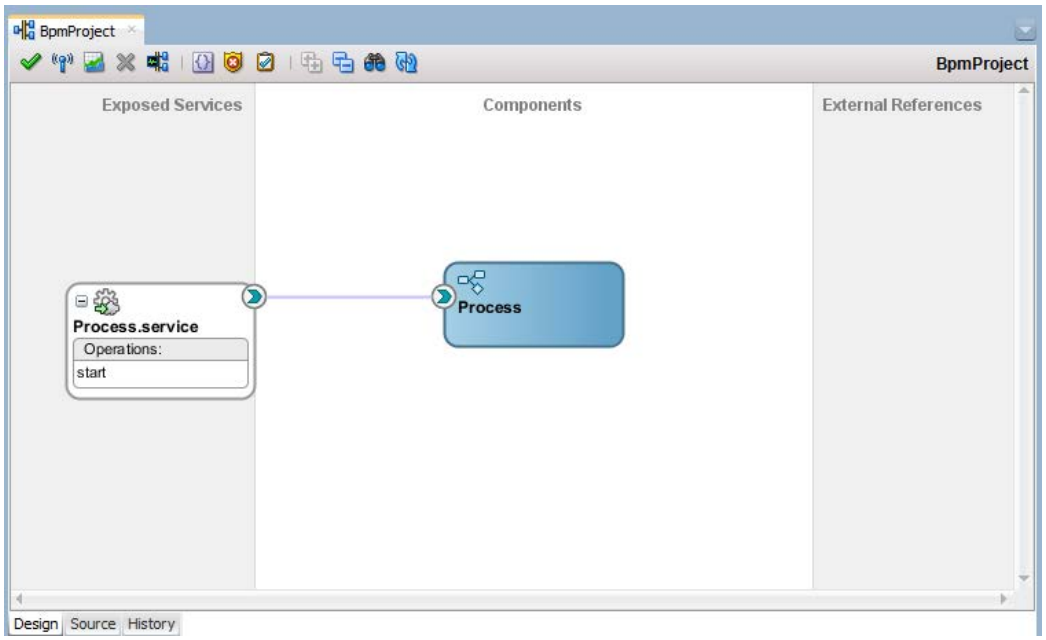
7. Click **Finish**.
8. Select the **Synchronous Service** from **Type** section and click **Finish**, as shown in the Figure below.

Figure 7- 6 Create BPM Process



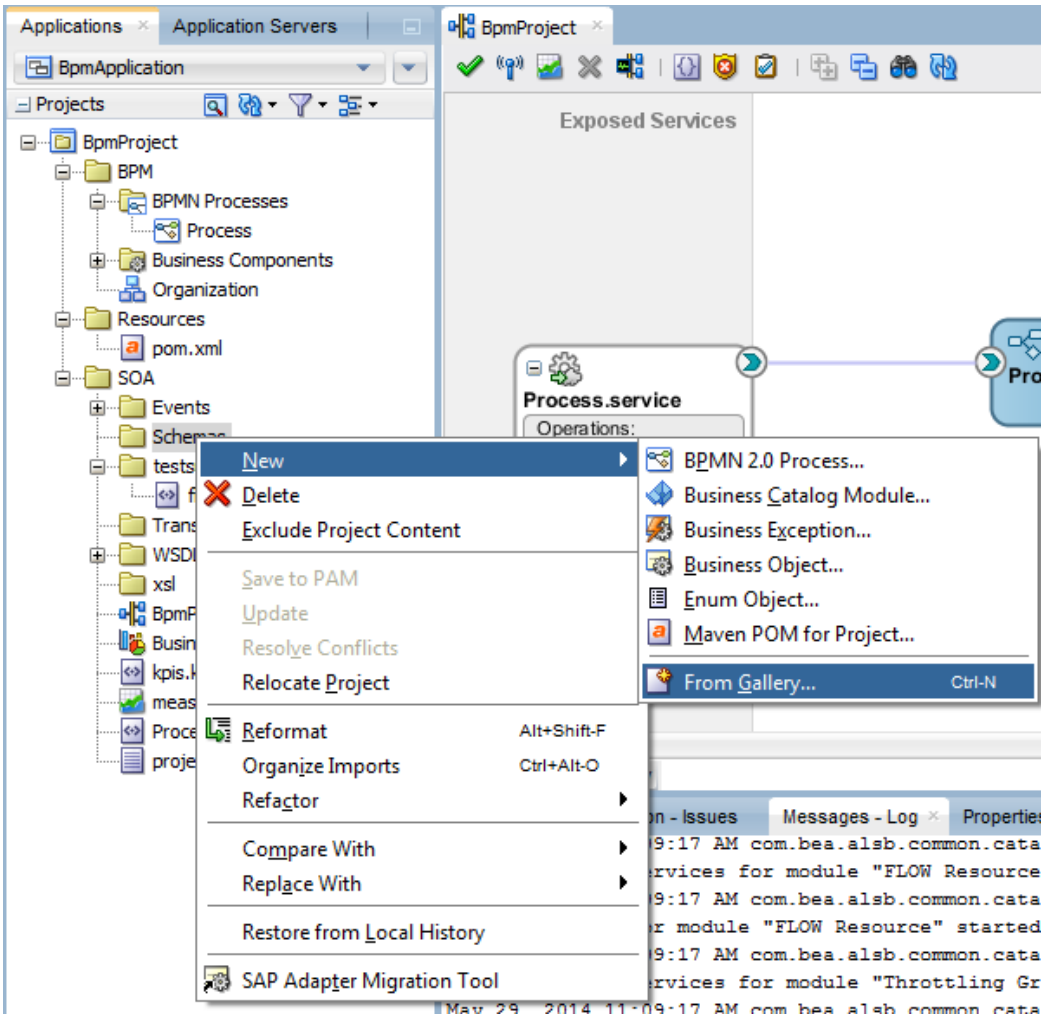
9. The composite.xml will look like the one displayed in the Figure below.

Figure 7- 7 Composite.xml



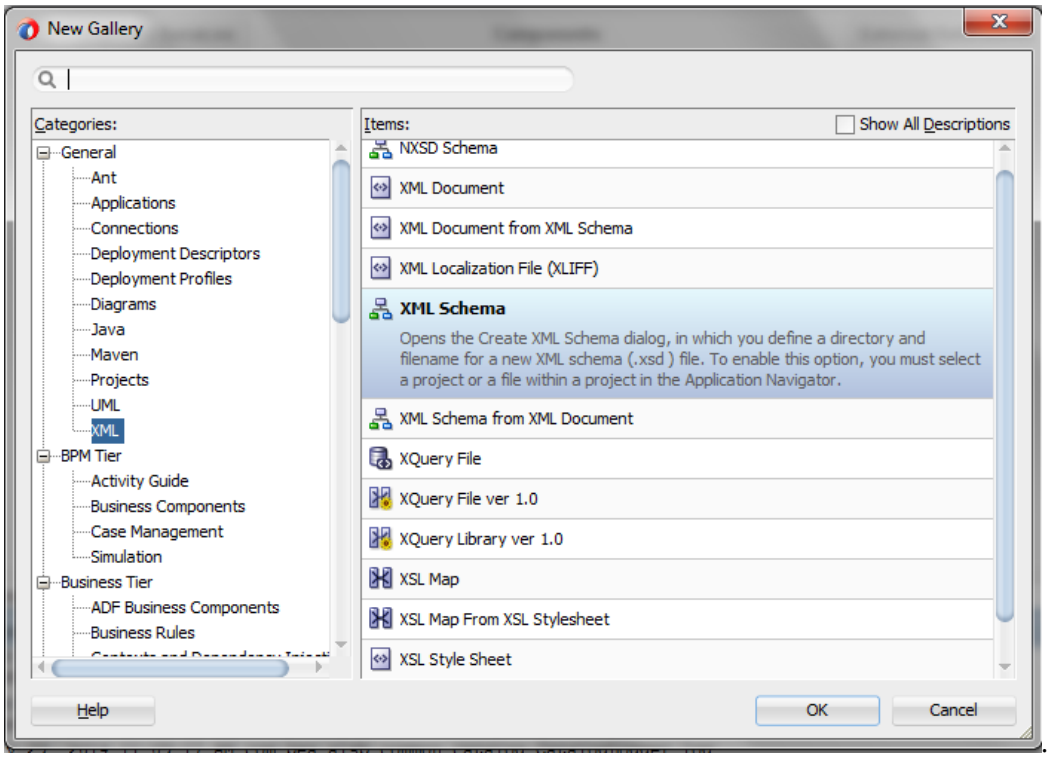
10. Next we will create the schema for our process. In the **Application Navigator**, under your project folder, right click on the schema folder and select **New** and then select **From Gallery** as shown in the Figure below.

Figure 7- 8 Create New Schema from Gallery



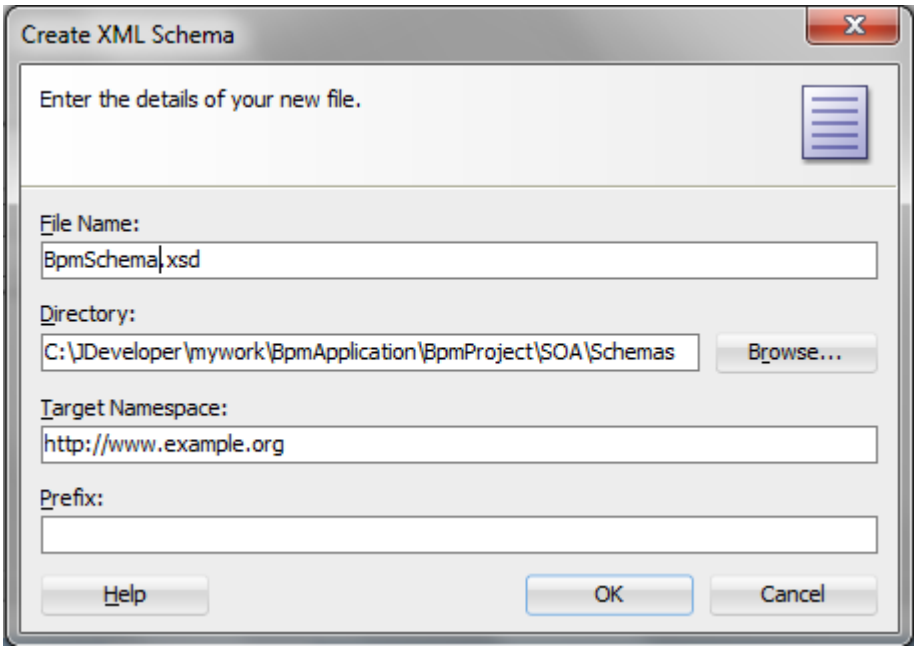
11. The **New Gallery** page is displayed. Select **XML Schema** from the **Items** list, and click **OK** as shown in the Figure below.

Figure 7- 9 Select XML Schema



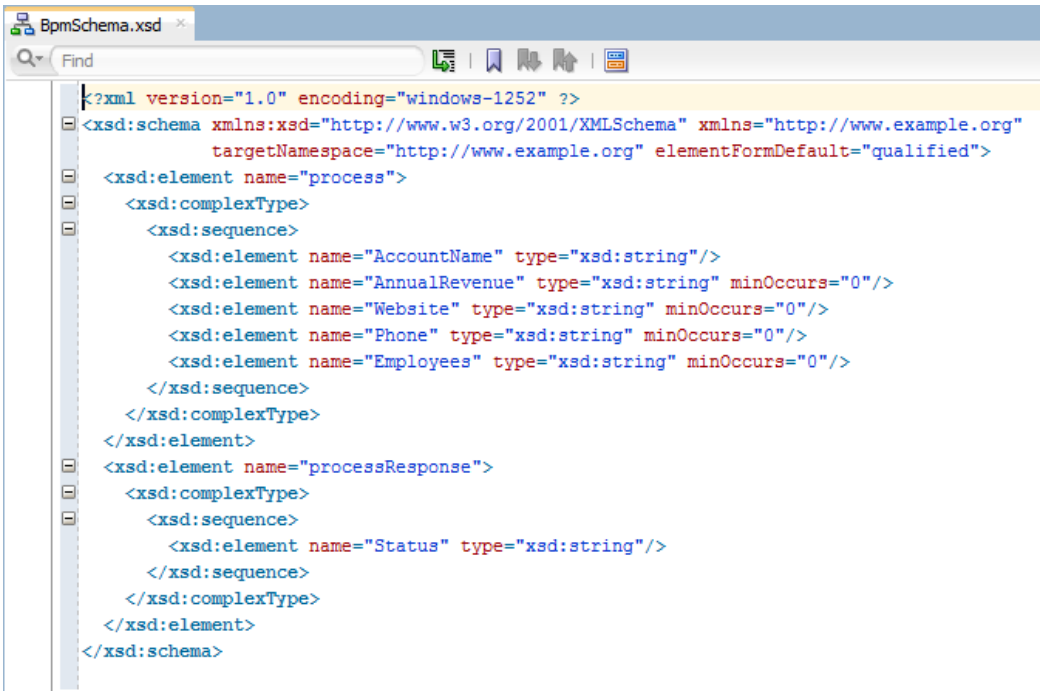
12. The **Create XML Schema** page is displayed, provide a suitable name to your schema and click **OK** as shown in the Figure below.

Figure 7- 10 Create XML Schema



13. Edit the schema file as per your business requirement. It is the responsibility of the front-end application to impose the data validations and to ensure that the input sent to SFDC via SOA is correct. The structure of the schema used in this use case is shown in the Figure below.

Figure 7- 11 Edit XML Schema

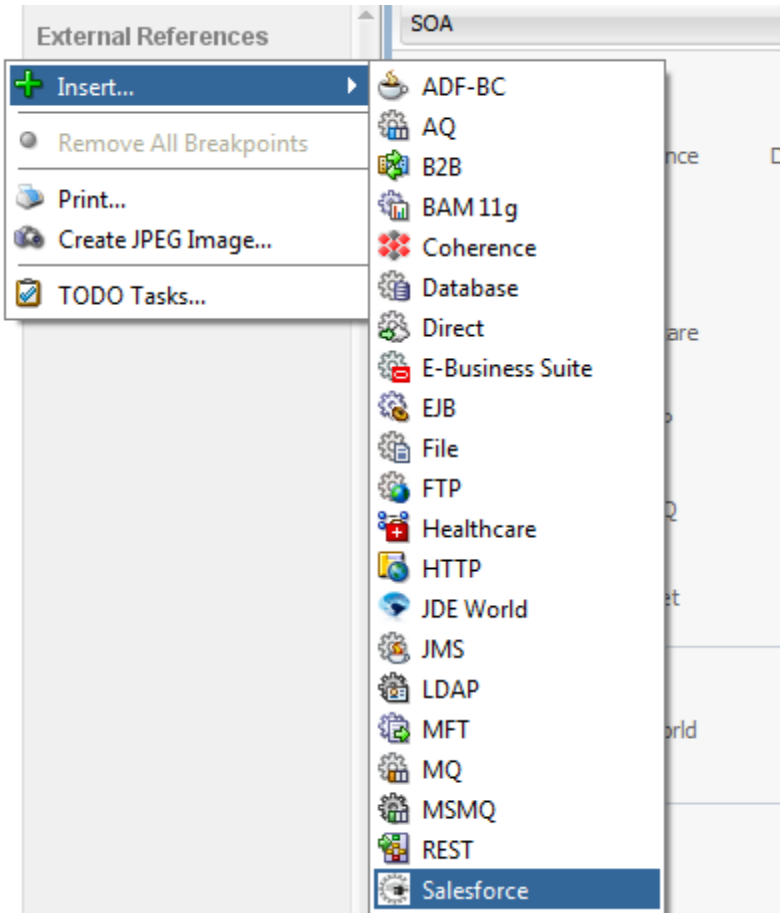


7.4 Configure Oracle Cloud Adapter for Salesforce.com

Perform the following steps to configure a New Oracle Cloud Adapter for Salesforce.com:

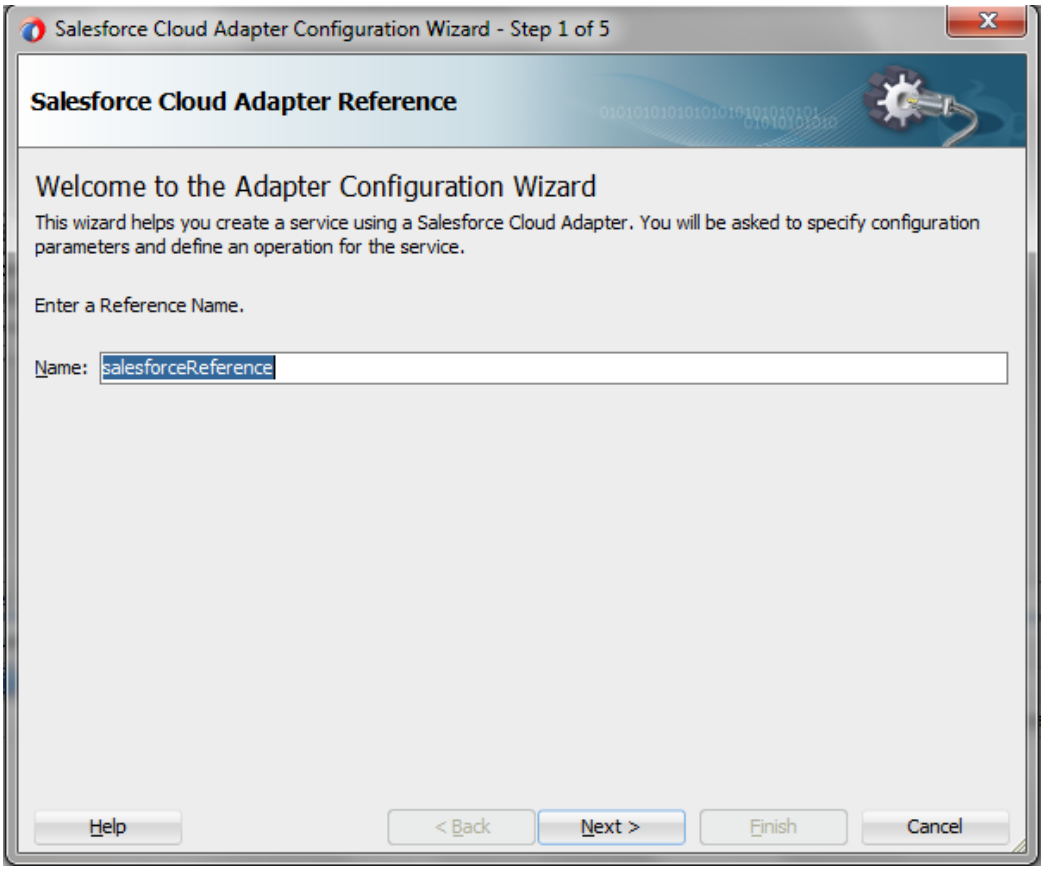
1. In **External References** swim-lane of the composite.xml file, right-click and select **Salesforce** adapter, as shown in the Figure below.

Figure 7- 12 Salesforce Adapter



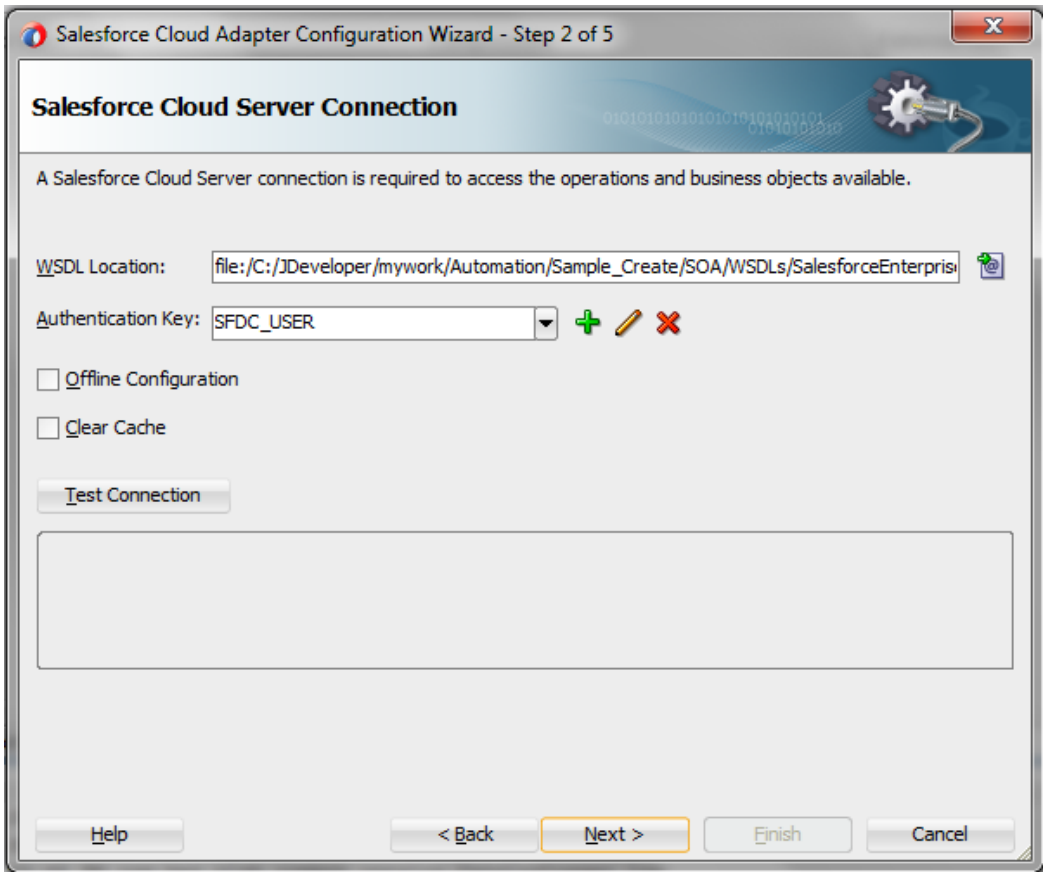
2. The Salesforce Cloud Adapter Configuration Wizard - Welcome page is displayed, as shown in the Figure below.

Figure 7- 13 Welcome Page



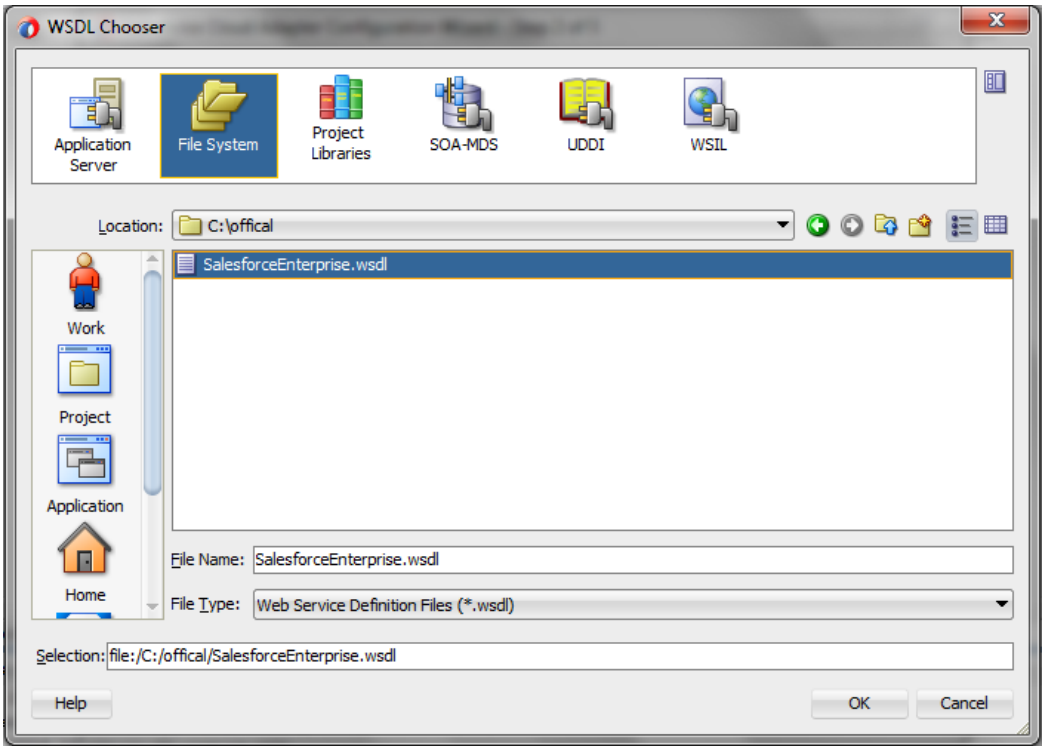
3. Click **Next**.
4. The **Salesforce Cloud Server Connection** page is displayed. The **WSDL Location** and **Authentication Key** text boxes are already populated. It picks up these values from the cache. You can re-enter these values. If you want to use a different value, click the **Find existing WSDLs** icon, which is located to the right of the **WSDL Location** field, as shown in the Figure below.

Figure 7- 14 Salesforce Cloud Server Connection Page



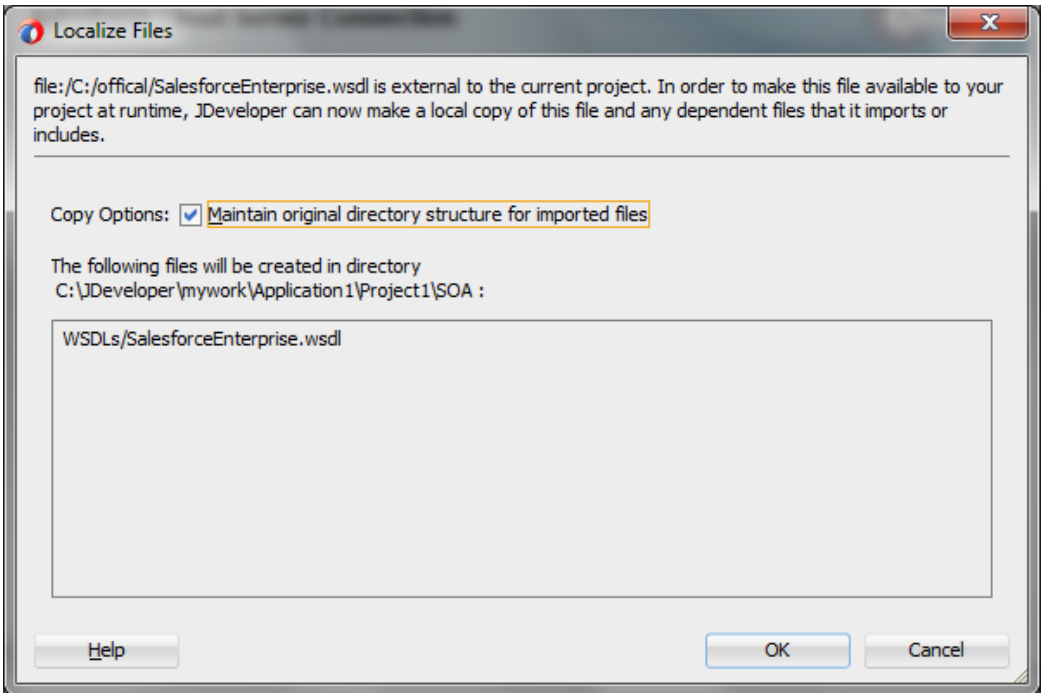
5. The **WSDL Chooser** dialog is displayed, browse and select the downloaded Enterprise WSDL and click **OK**, as shown in the Figure below.

Figure 7- 15 SOA Resource Browser



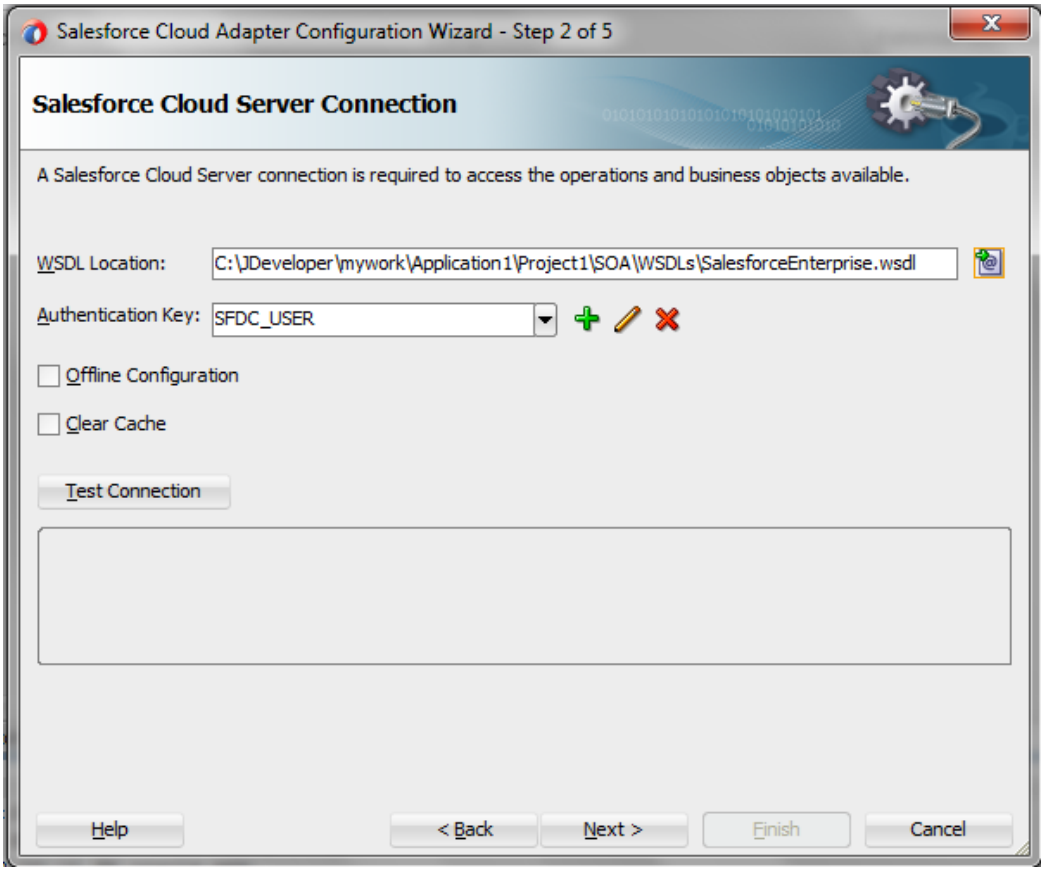
6. Click **OK**. The **Localize Files Dialog** appears, as shown in the Figure below.

Figure 7- 16 Localize Files Dialog



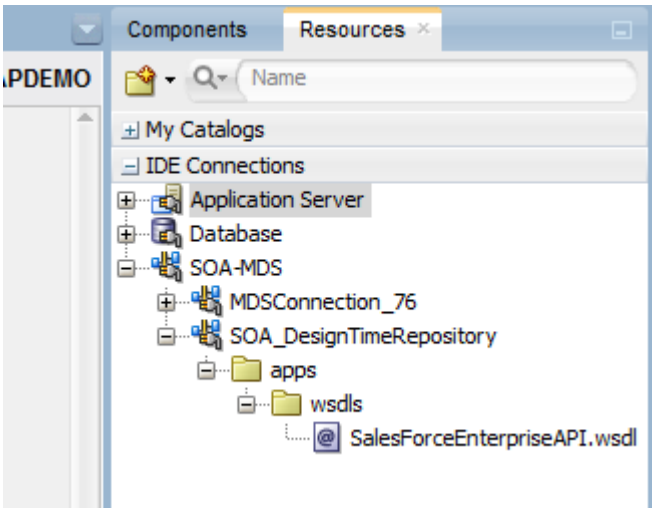
7. Click **OK**. You will be returned to the Salesforce Cloud Server Connection page.

Figure 7- 17 Salesforce Cloud Server Connection Page



Note that as an alternative, you can store WSDL at an MDS location and access it, as shown in the Figure below.

Figure 7- 18 SOA Resource Browser

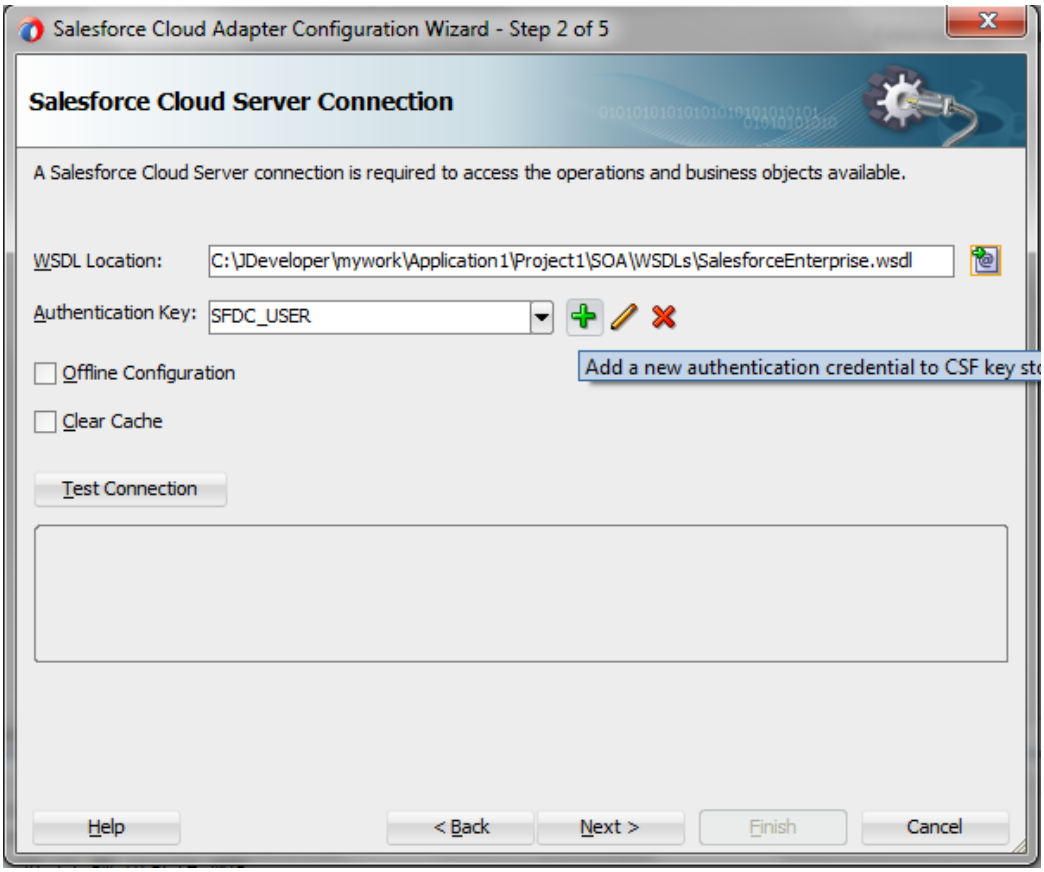


8. Traverse to IDE Connections → SOA-MDS. Select the appropriate SOA-MDS connection where you placed the Enterprise WSDL. Select the WSDL file to be used in the Adapter configuration and click **OK**.
9. The WSDL location should be of the form 'oramds:/apps/SOA/WSDLs/Integration/SalesforceReference.wsdl', as shown in the Figure below.

Figure 7- 19 WSDL location

10. Click **OK**.
11. Click **"+"** button to create a new Authentication Key, as shown in the Figure below.

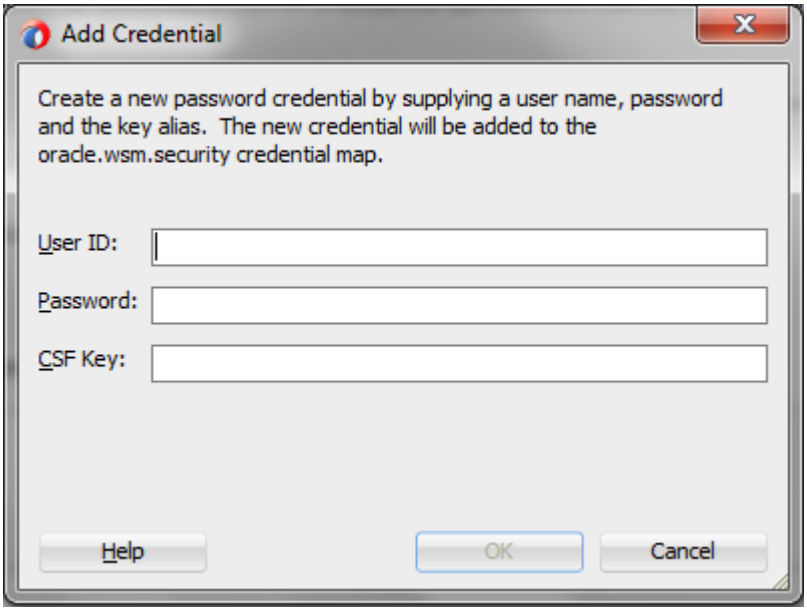
Figure 7- 20 Create a New Authentication Key



12. The **Add Credential** dialog box is displayed, as shown in the Figure below. Provide a suitable name and the Salesforce.com credentials and click **OK**.

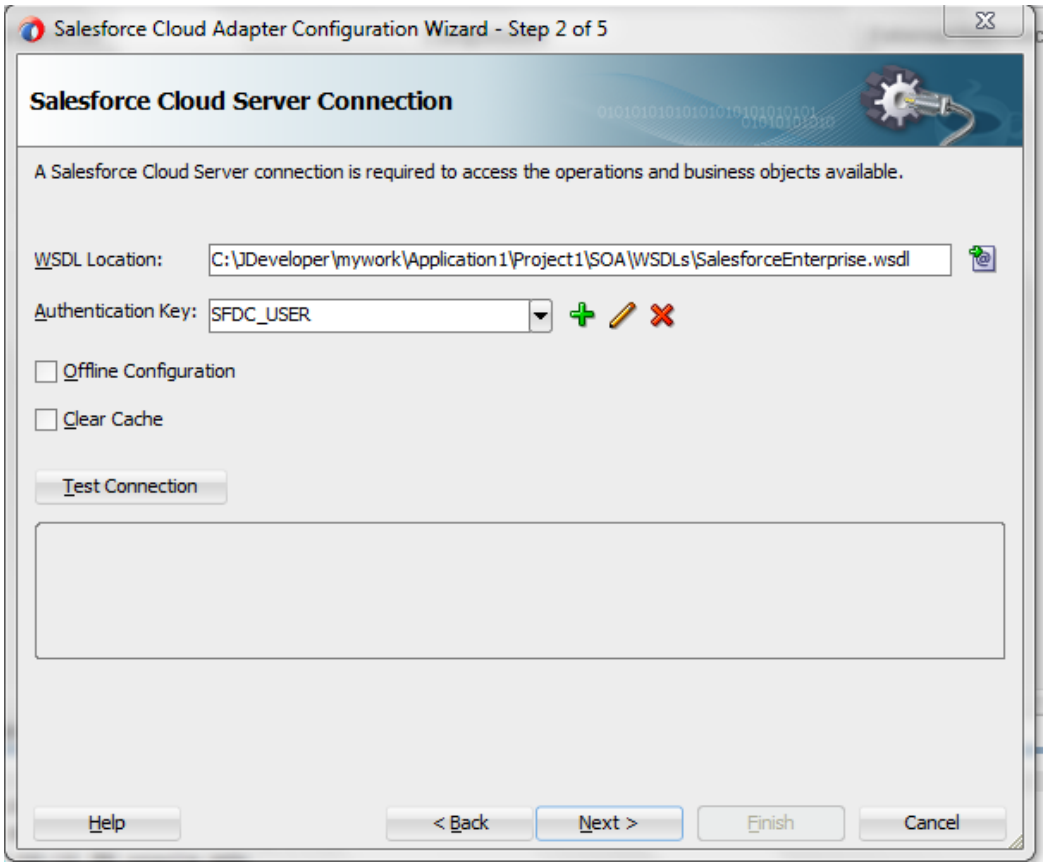
Note: The password should be a combination of Salesforce.com password and Salesforce.com Security Token.

Figure 7- 21 Add Credential



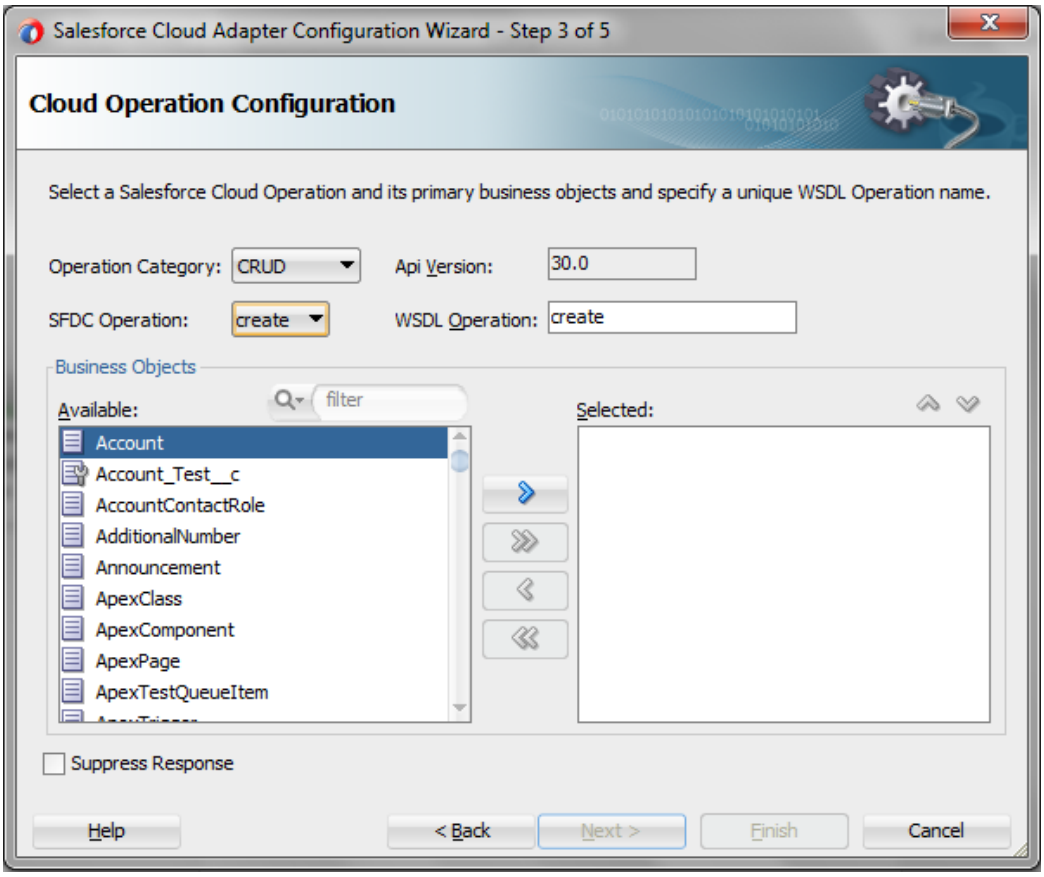
13. Click **Test Connection** button to validate the Authentication Key, as shown in the Figure below.

Figure 7- 22 Test Connection



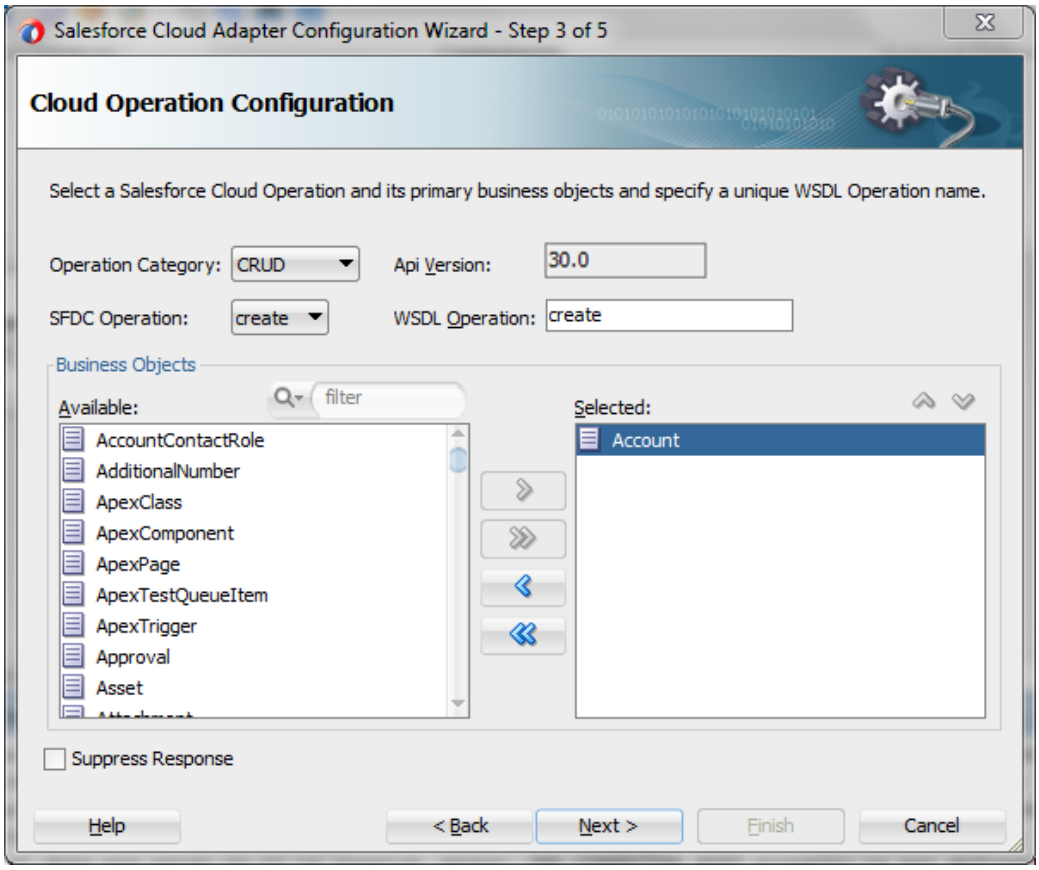
14. Click **Next**.
15. The **Cloud Operation Configuration** page is displayed, as shown in the Figure below.

Figure 7- 23 Cloud Operation Configuration Page



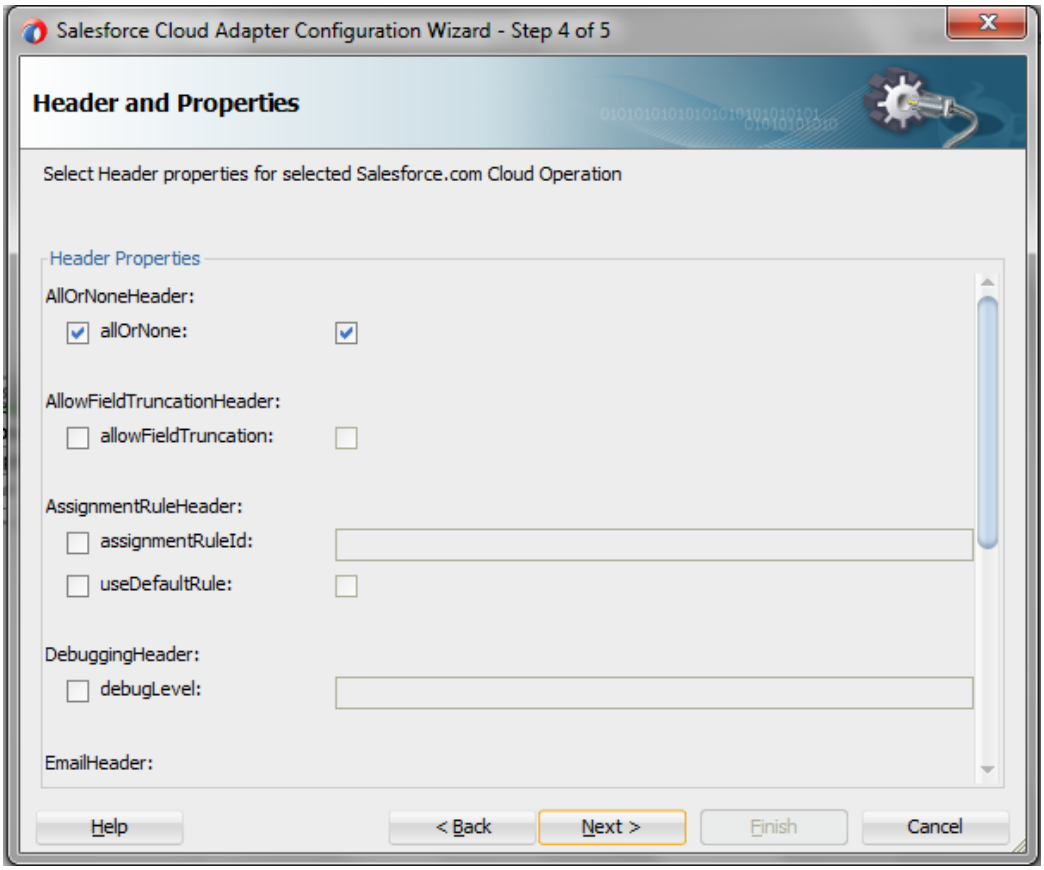
16. Now move **Account** from the list of **Available** objects to the list of the **Selected** objects. The **WSDL Operation** by default is **create** (same as SFDC Operation). You can edit the same by providing an operation name suitable to your business requirement, as shown in the Figure below.

Figure 7- 24 Cloud Operation Configuration Page



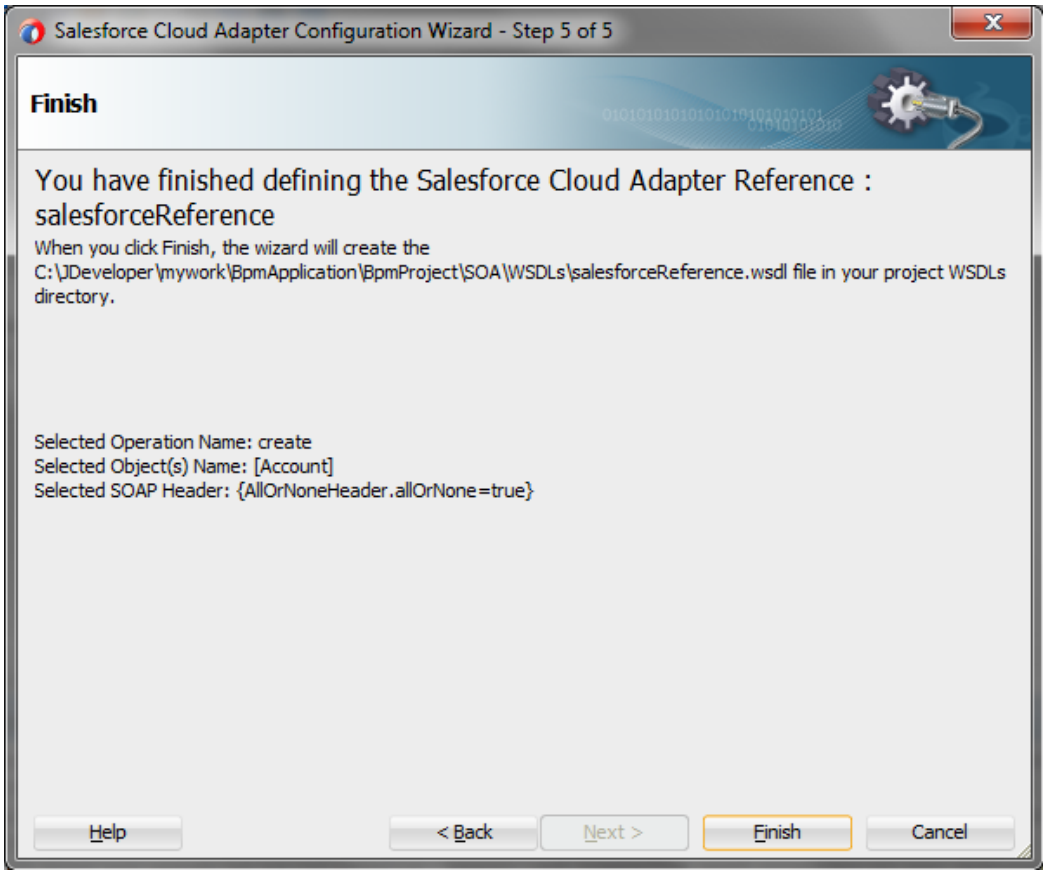
17. Click **Next**. The **Header and Properties** page is displayed, as shown in the Figure below.

Figure 7- 25 Header and Properties Page



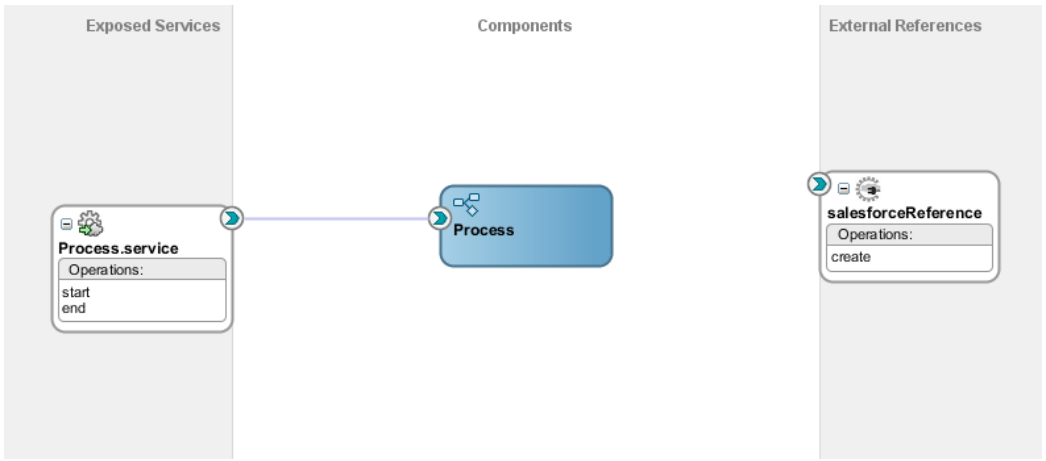
18. Select the appropriate header according to your requirement. Headers displayed in this page depend on the operation selected in the previous page.
19. Click **Next**.
20. The finish page is displayed. It provides a complete summary of the operation selected, object on which the operation will operate and the headers selected for that operation, as shown in the Figure below.

Figure 7- 26 Finish Page



21. Click the **Finish** button to complete the Adapter Configuration Wizard.
22. After clicking on **Finish** button, the following screen appears, as shown in the Figure below.

Figure 7- 27 SFDC Cloud Account

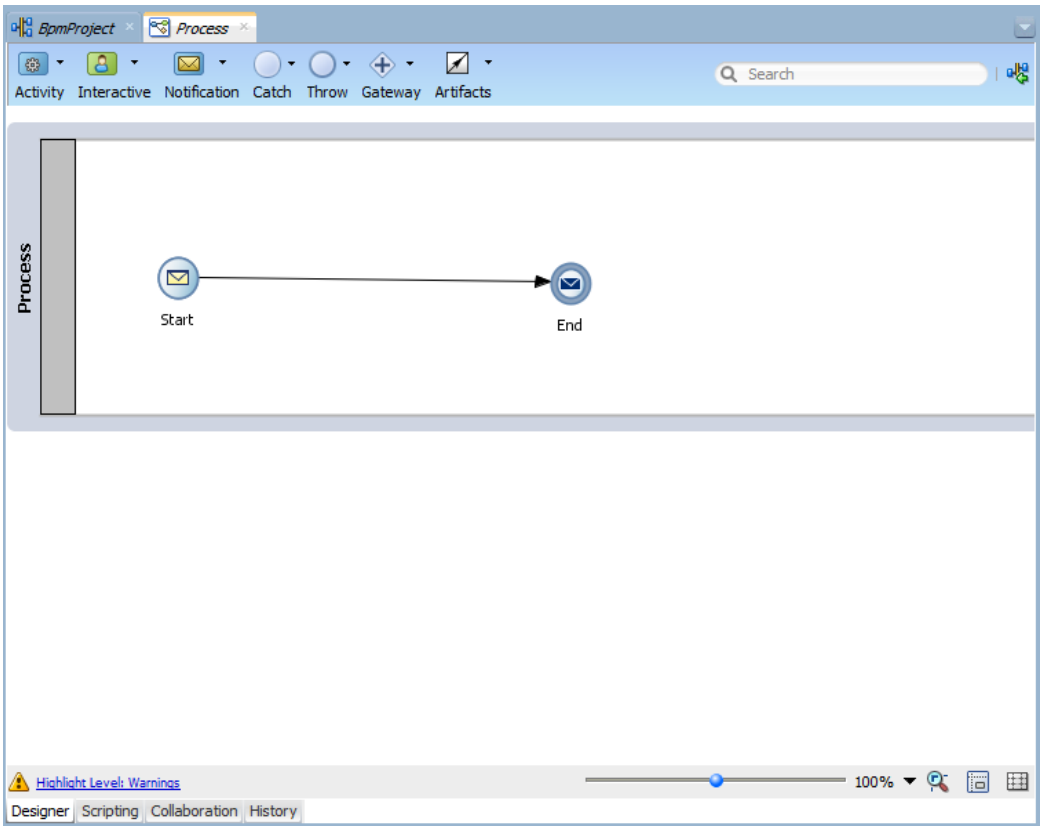


7.5 Integration with BPM

Perform the following steps to integrate with BPM:

1. In the composite.xml file of your project, a component with the name **Process** exists. Double-click and open the same. **Process.bpm** page opens, as shown in the Figure below.

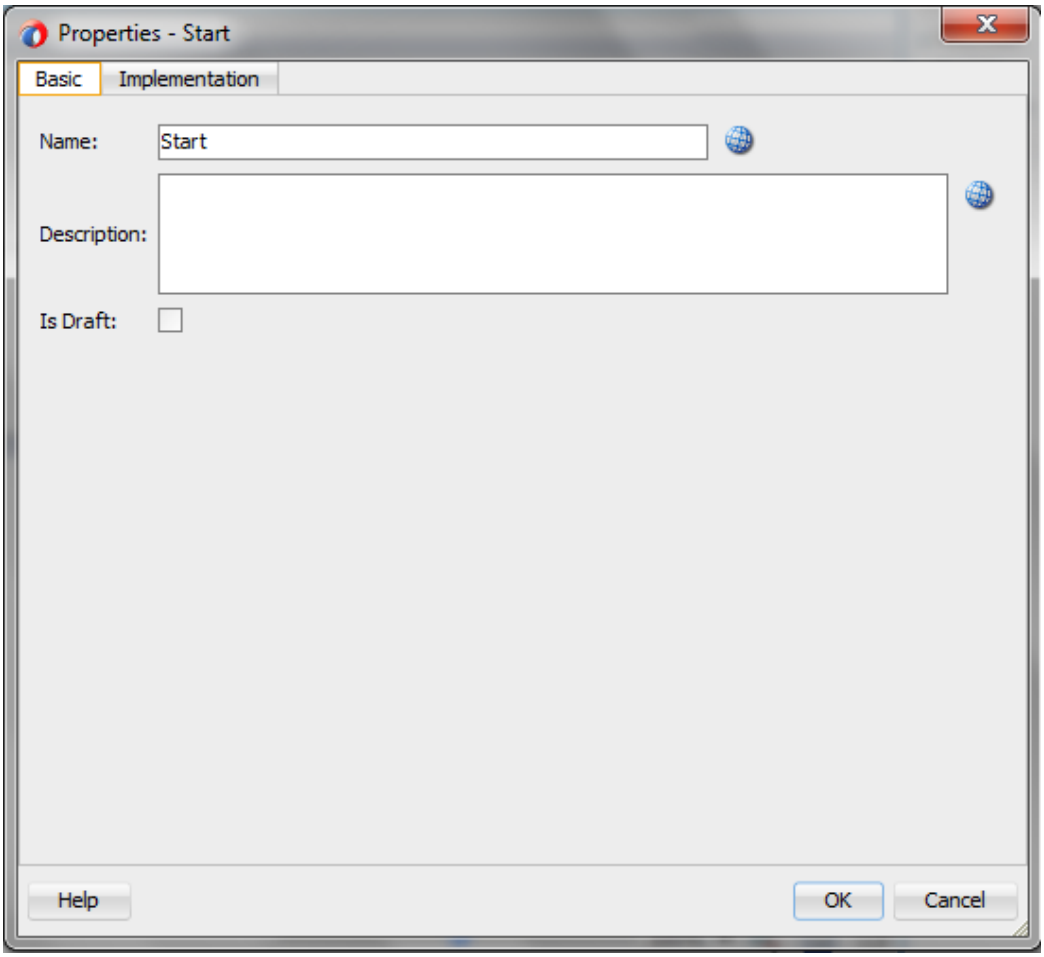
Figure 7- 28 Process.bpm in composite.xml



2. Double-click on the **Start** activity. **Properties-Start** page is opened as shown in the Figure below.

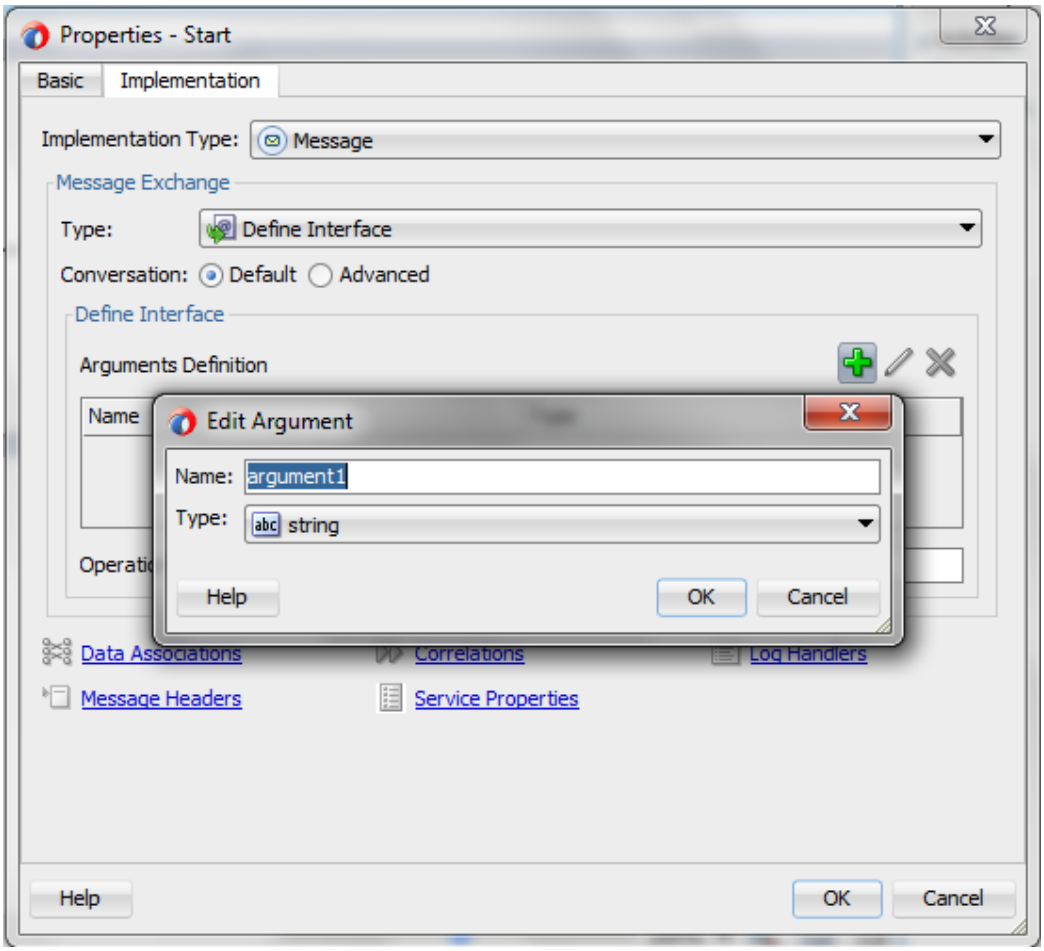
Note: The Model Starts and Ends with two circular icons. The Left circle is a **Start Activity** and the circle which is on the right side is the **End Activity.** These two activities are joined with the help of a line that represents the flow of activities through the process known as Sequence Flow.

Figure 7- 29 Start Activity



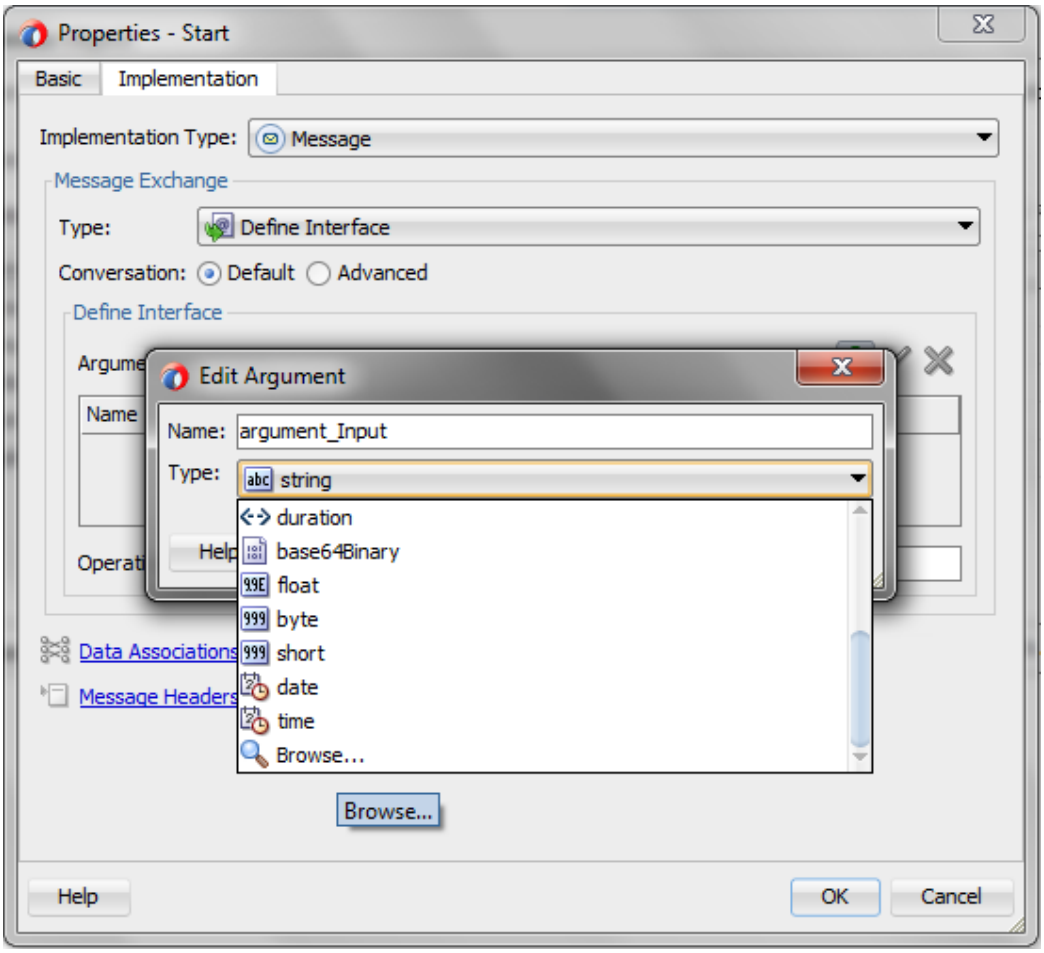
3. Left-click on **Implementation** Tab. Now in the **Argument Definition,** click on the '+' sign to create the input argument/s.

Figure 7- 30 Start – Implementation Tab – Add Argument



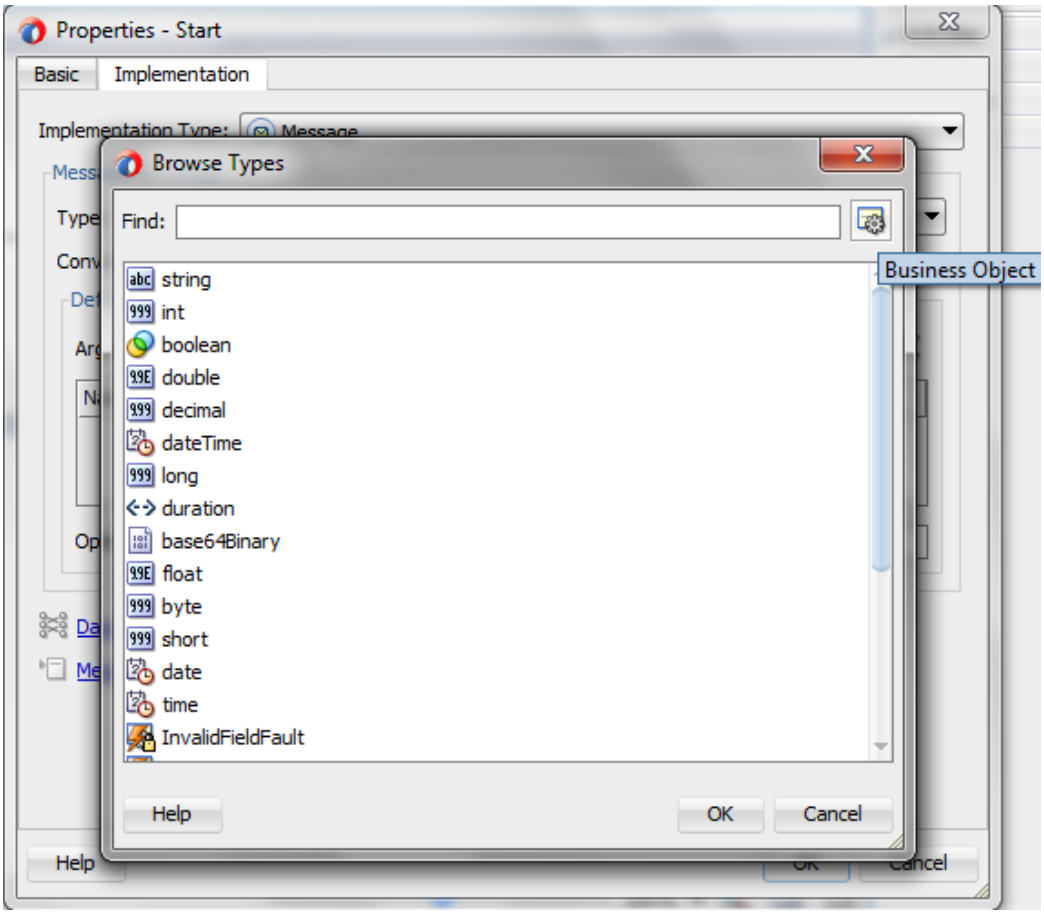
4. In the **Edit Argument** page, provide a name to your argument in the **Name** section and in the **Type** section, scroll down and select the **Browse** option.

Figure 7- 31 Edit Argument



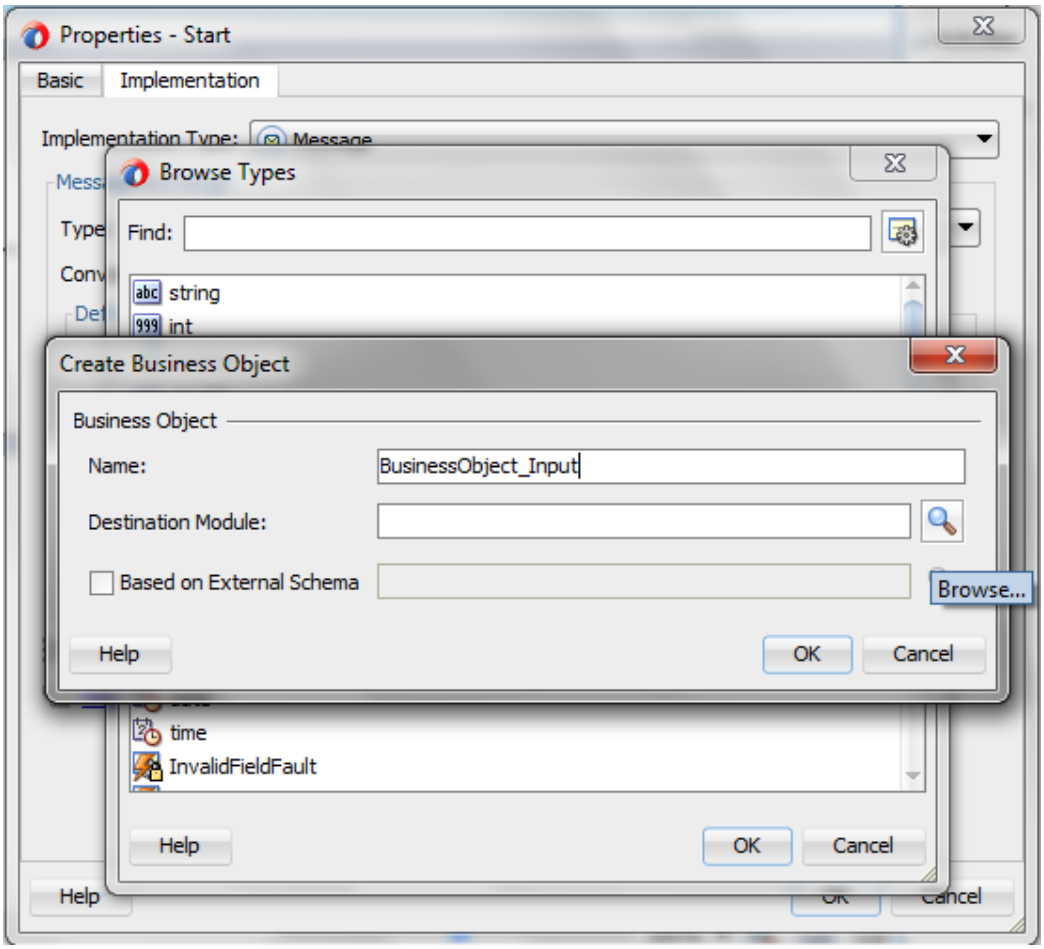
5. Next, a new **Browse Type** page opens. Click on the **Business Object** button to create one.

Figure 7- 32 Browse Types



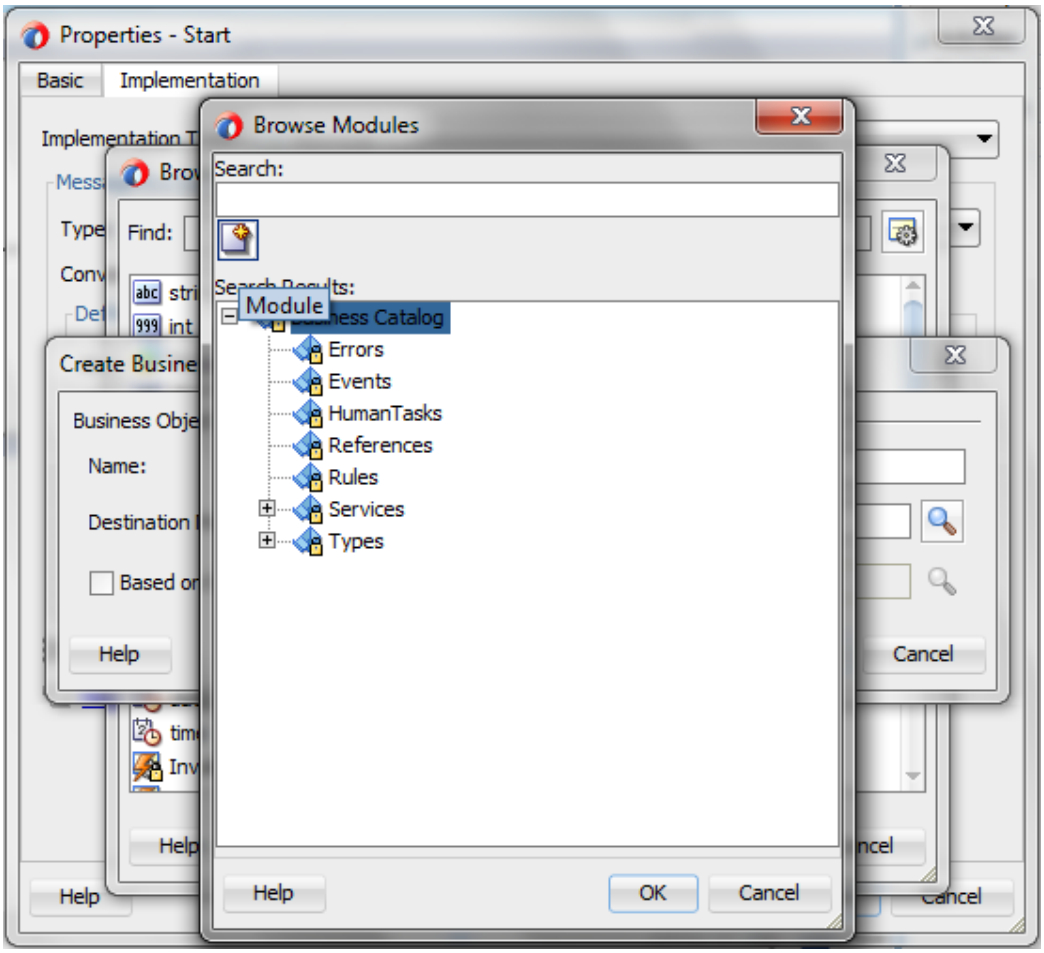
6. In the **create business object** page, provide a **Name** to it and then click on the **Magnifying** icon in front of the **Destination Module** to browse for a module in which the business object exists.

Figure 7- 33 Create Business Object



7. In the **Browse Module** Page, create a new module there by clicking on the module button as shown in the Figure below.

Figure 7- 34 Select Module



8. In the create module page give a suitable **Name** to the module and click on **OK** button. Select the module you just created in the Browse module page and click **OK** as shown in the Figures below.

Figure 7- 35 Create Module

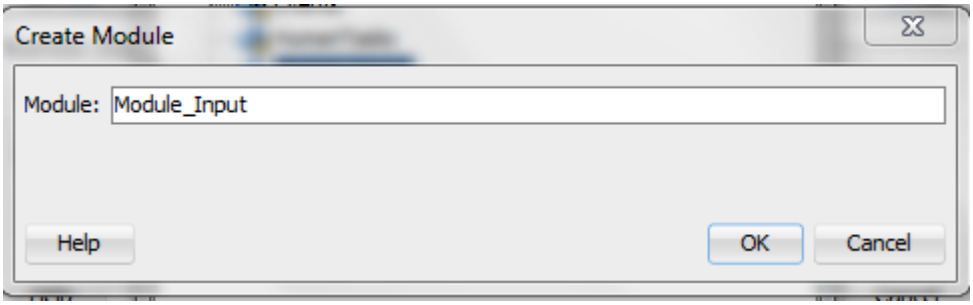
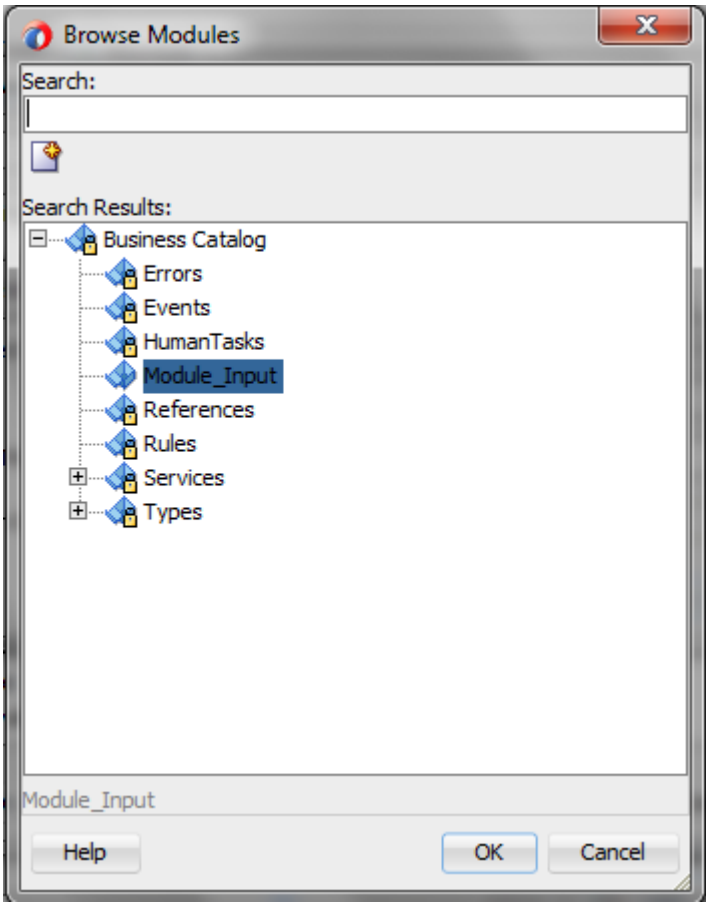
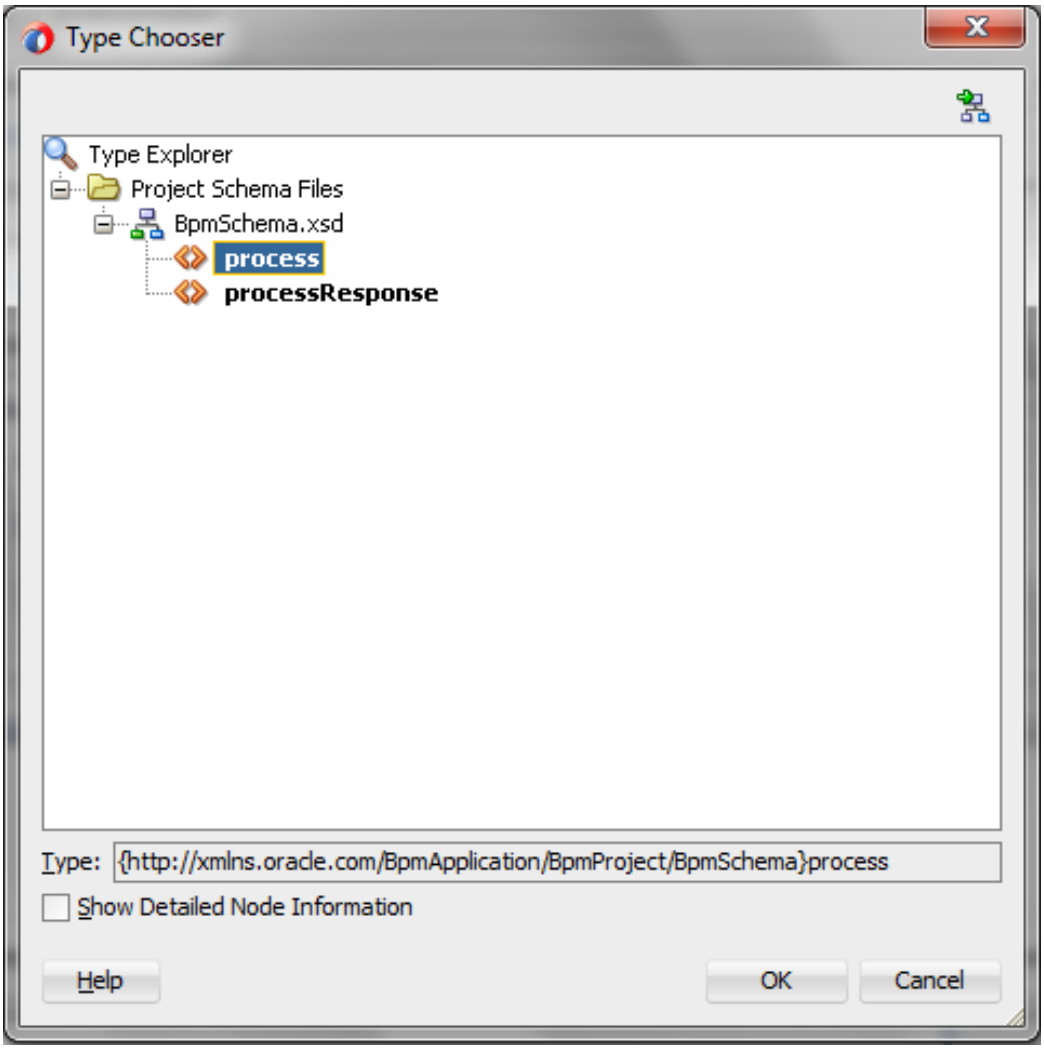


Figure 7- 36 Select created module



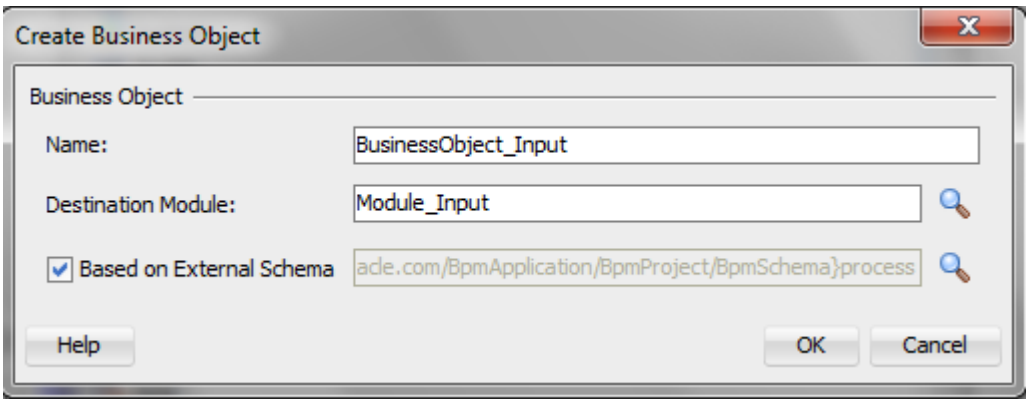
9. In the **Create Business Object** page, click on the checkbox **Based On External Schema** and browse to the respective schema file and select the input element (**Process** in this case) from the Schema and click on **OK**.

Figure 7- 37 Select input element from external schema



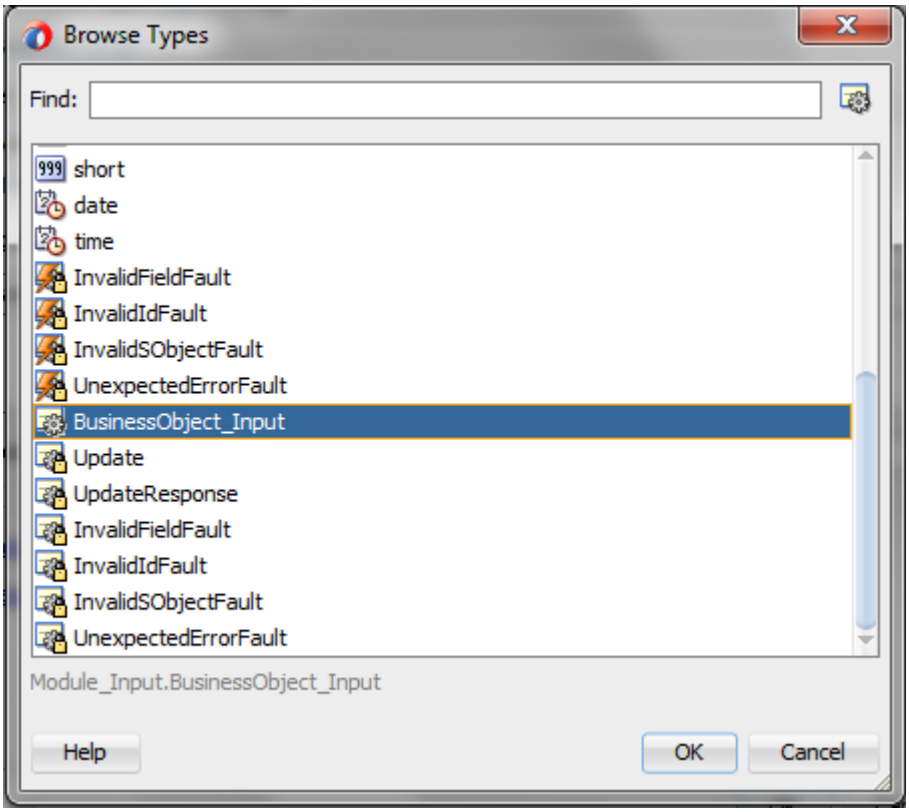
10. The **create Business object** page appears. Click **OK**.

Figure 7- 38 Business object created



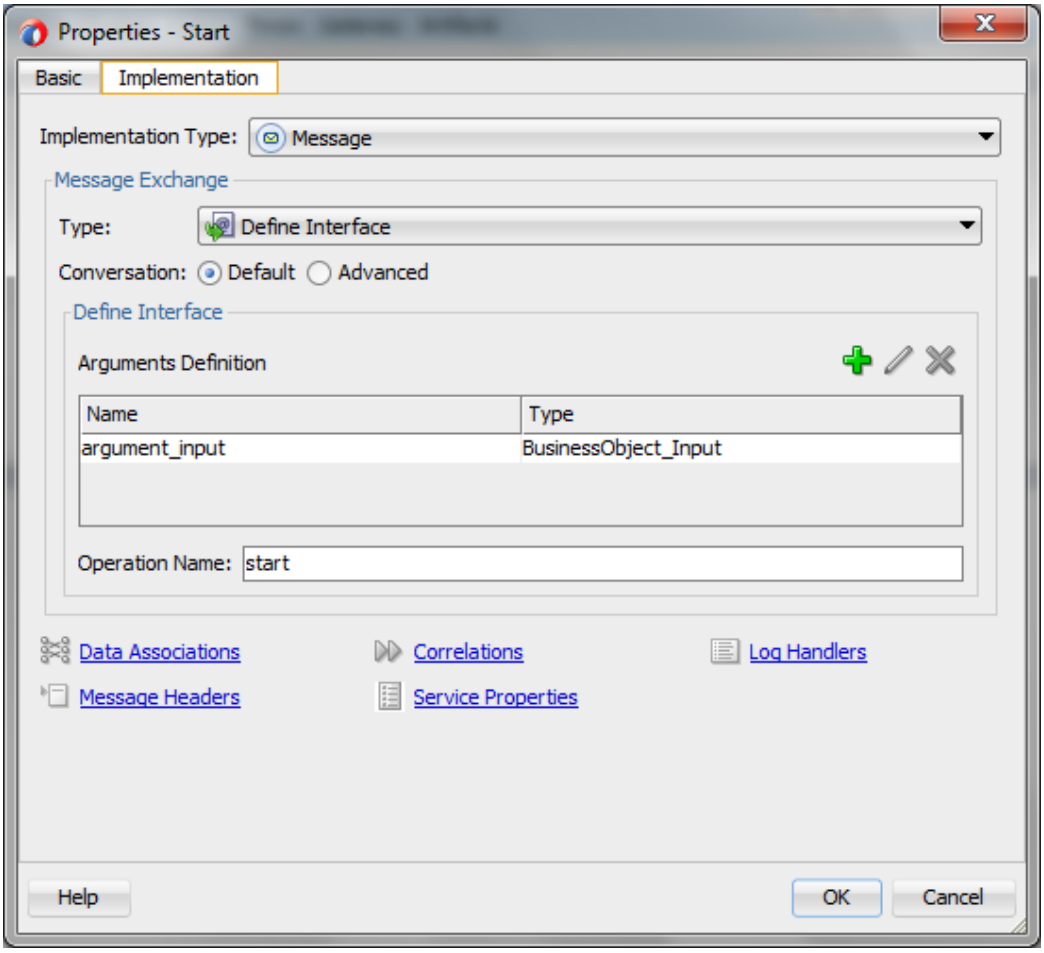
11. Now a Business Object is created. Select the business Object you have created for the input and click **OK,** as shown in the Figure below.

Figure 7- 39 Select the created Business Object



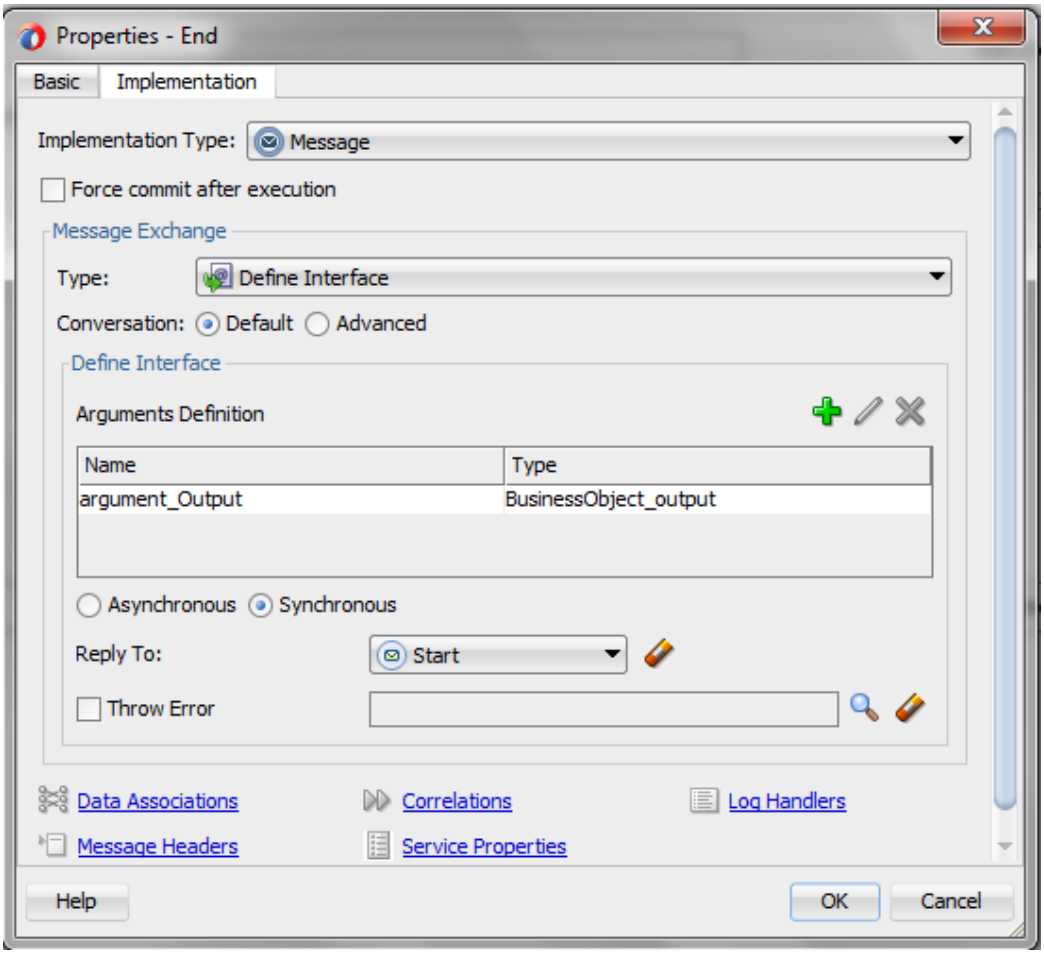
12. Properties-Start page will contain an argument for the input.

Figure 7- 40 Input Argument added



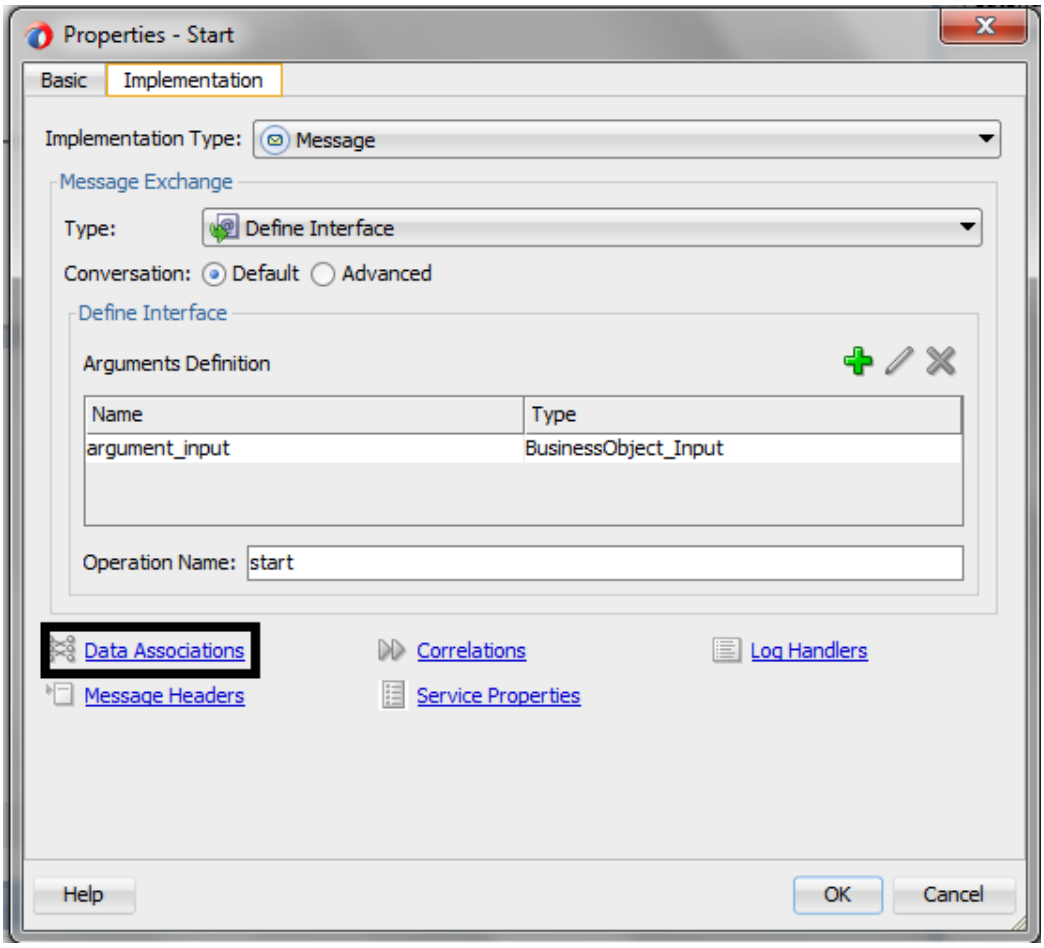
13. Follow the same steps to create **Business object** for output in the **End** activity of your process. But choose the respective schema file and select the Output element (**processResponse** in this case) from the Schema in **Create Business Object** page.

Figure 7- 41 Output argument added



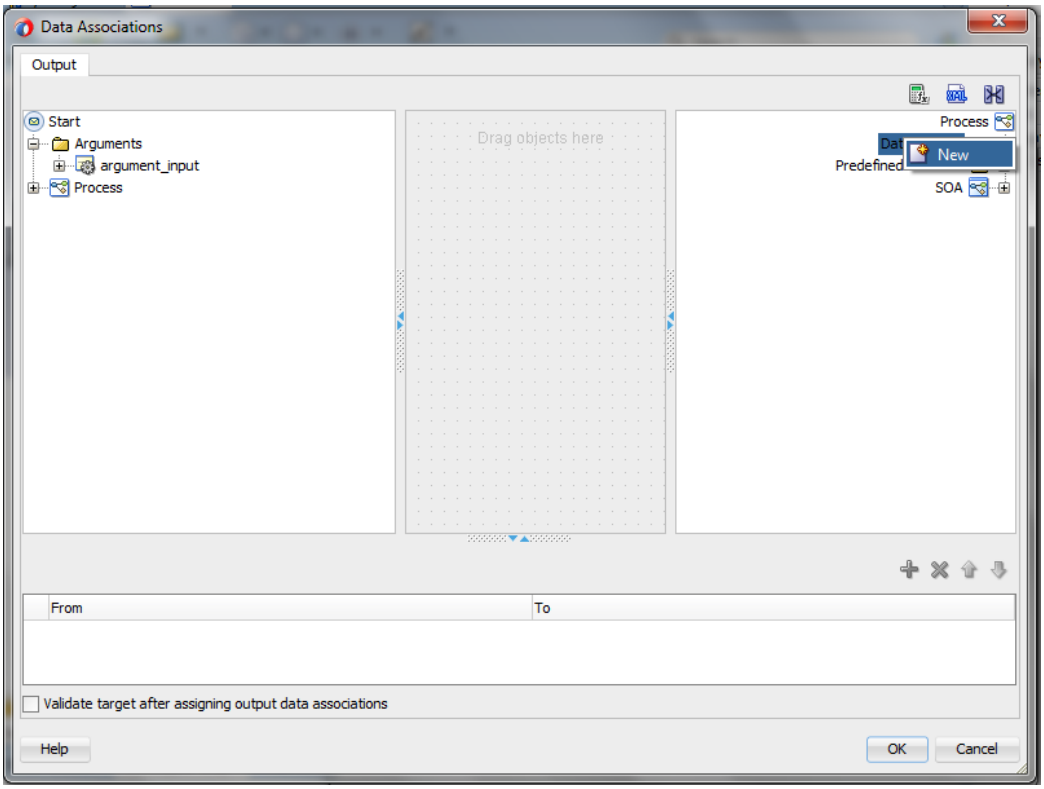
14. Again go to the **Start** activity and click on **Implementation** tab. Next, click on **Data Association** as shown in the Figure below.

Figure 7- 42 Select Data Associations



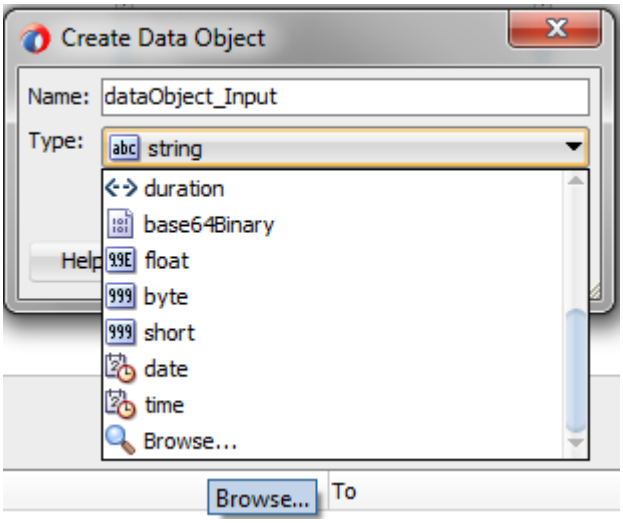
15. New Data Association page opens. In the data objects under **process,** create **Data objects** for Input, Output and Salesforce Adapter Response.
16. Right-click on the data object and click **New** as shown in the Figure below.

Figure 7- 43 Add new data object



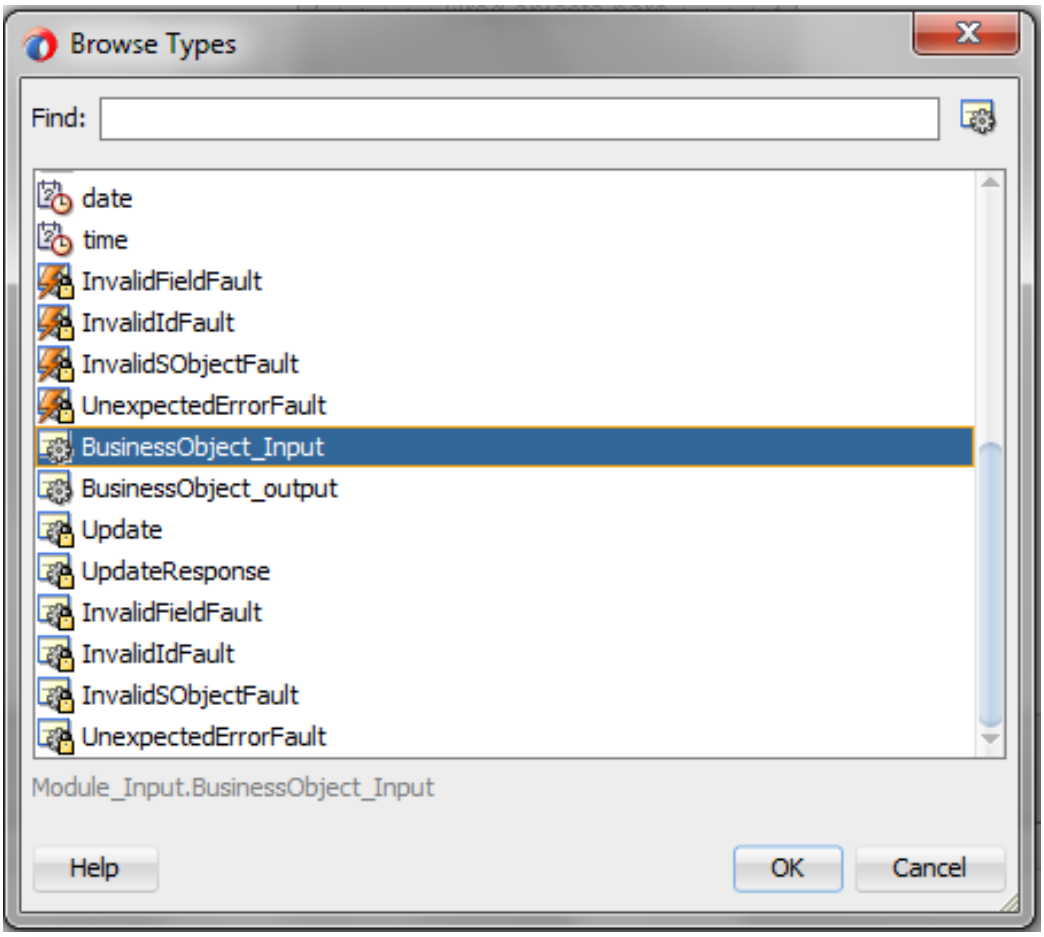
17. Provide a suitable **Name** for the data object (Input in this case) and in the **Type** section, scroll down and select the **Browse** option.

Figure 7- 44 Create Data Object



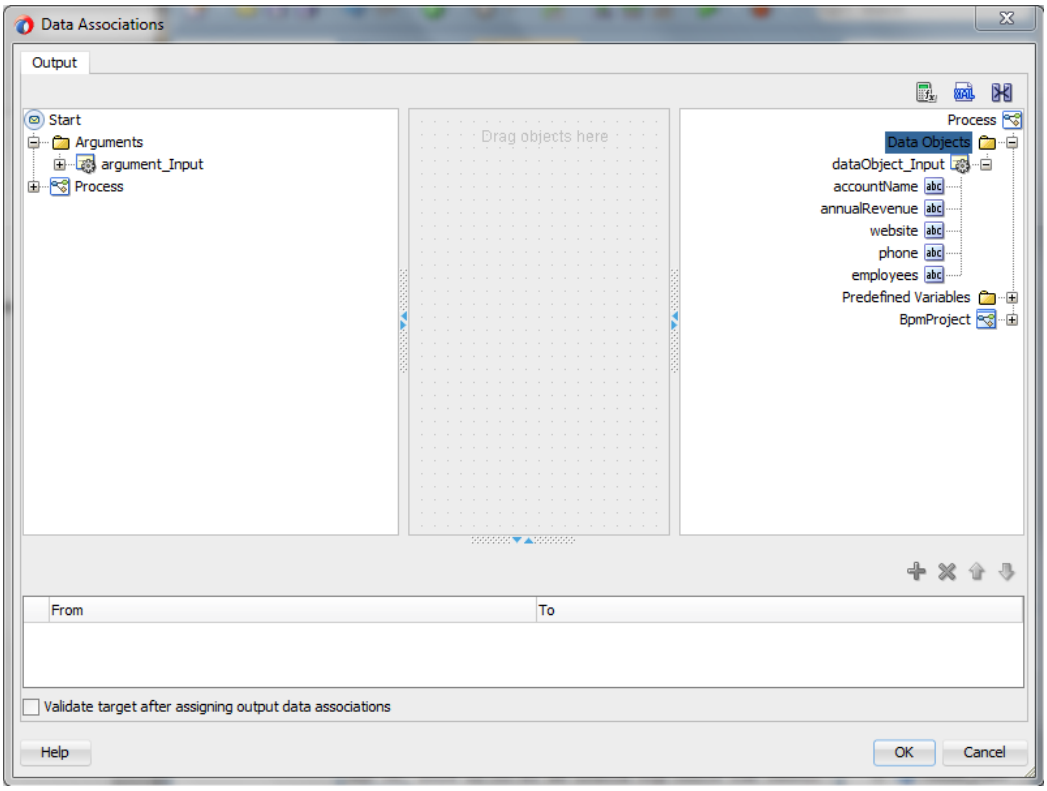
18. **Browse Type** page opens. Select the business object you created for Input argument and click **OK** button as shown in the Figure below.

Figure 7- 45 Browse to input business object



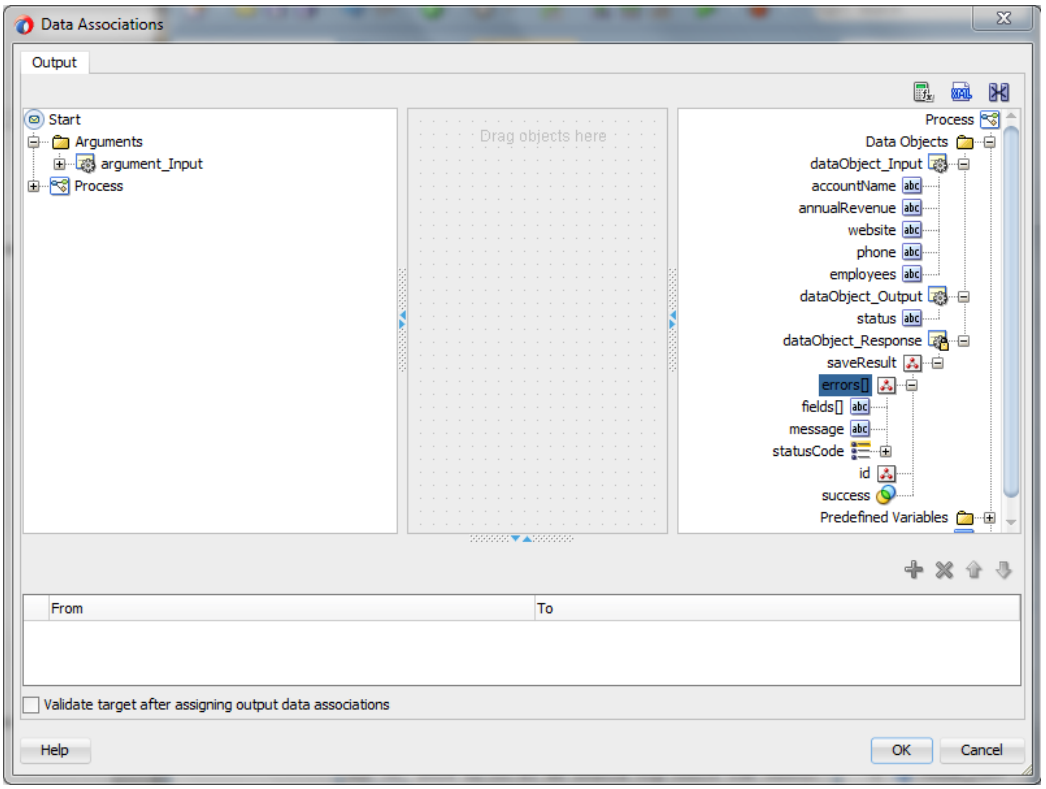
19. Click **OK** on **Create Data Object** page.
20. Data object is created for Input which can be seen under **Data Objects** of **Process,** as shown in the Figure below.

Figure 7- 46 Input data object added



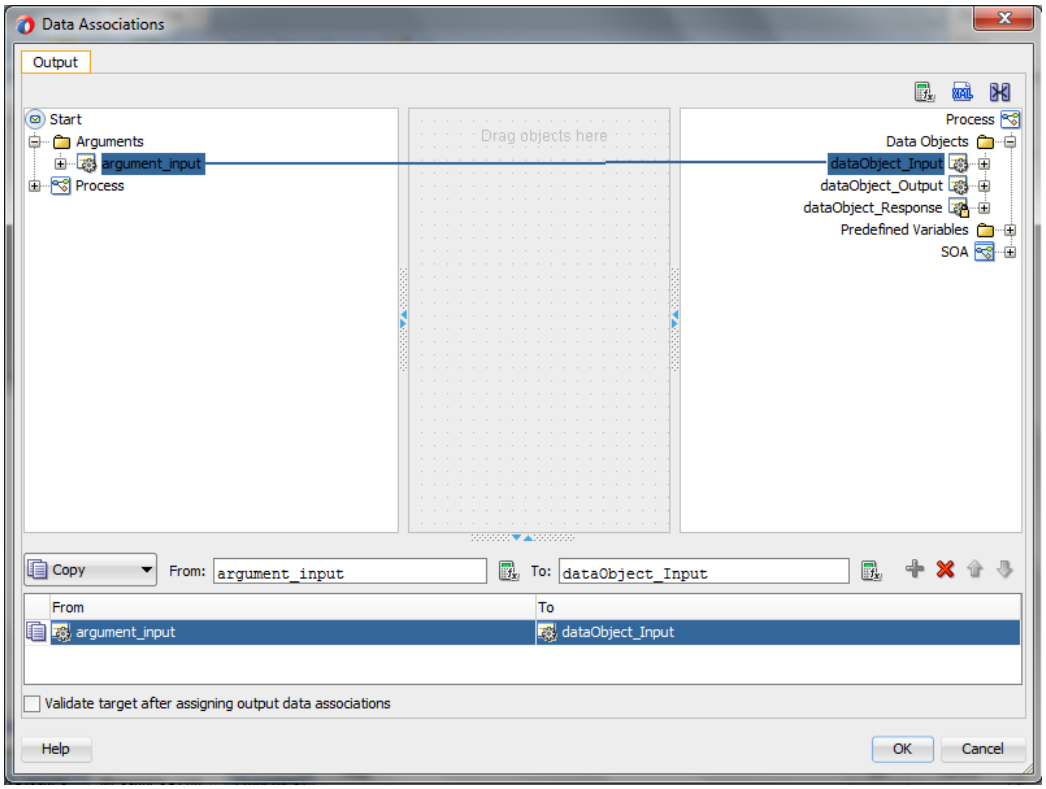
21. Follow the same steps for creating **Data Objects** for Output and Response (createResponse in this case).

Figure 7- 47 Output data object added



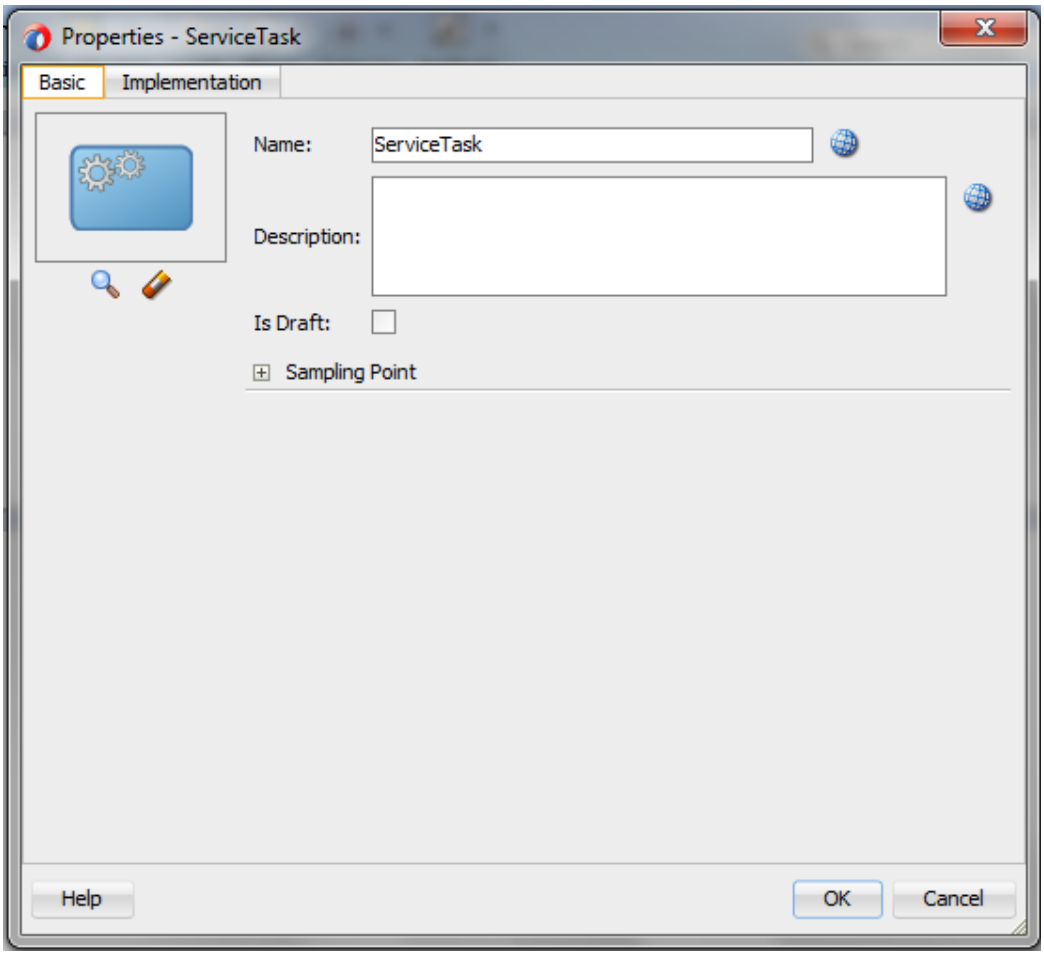
22. Map the **argument_input** under **Start** to **dataObject_Input** and click **OK** button as shown in the Figure below.

Figure 7- 48 Map Data Associations



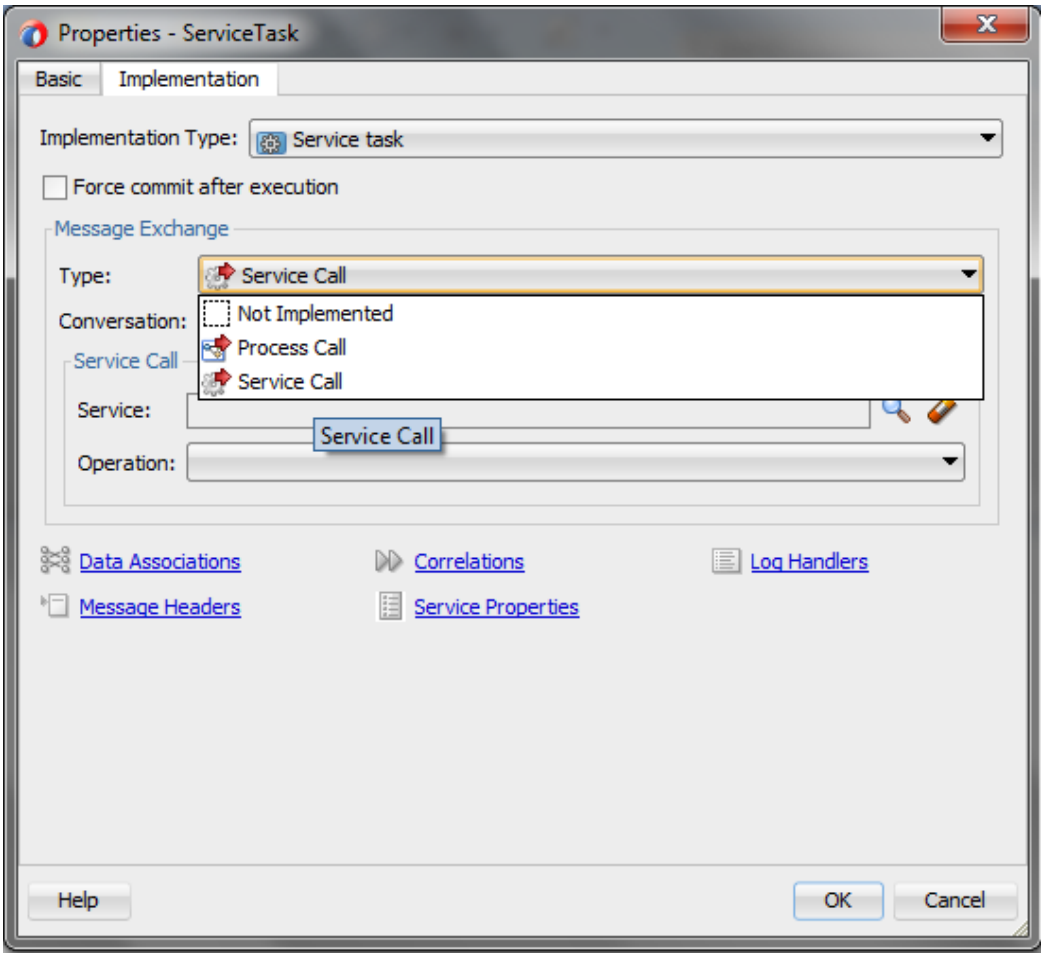
14. Drag the **Service** activity from the **Activity** tab and place it in between **Start** and **End** Activity. A new page **Properties – ServiceTask** opens as shown in the Figure below.

Figure 7- 49 ServiceTask



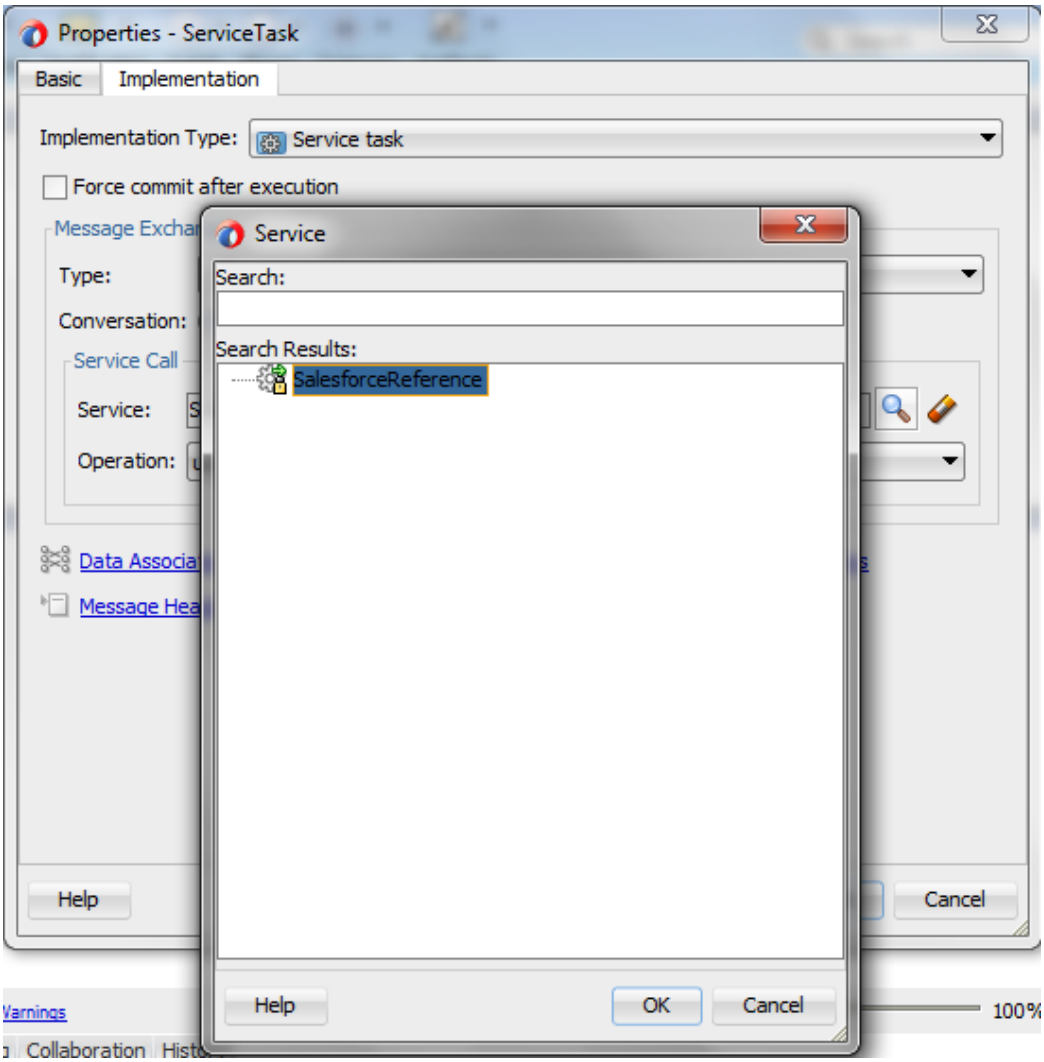
15. Click on **Implementation** tab and select **Service call** in **Type** Section.

Figure 7- 50 Select Service Call for Type



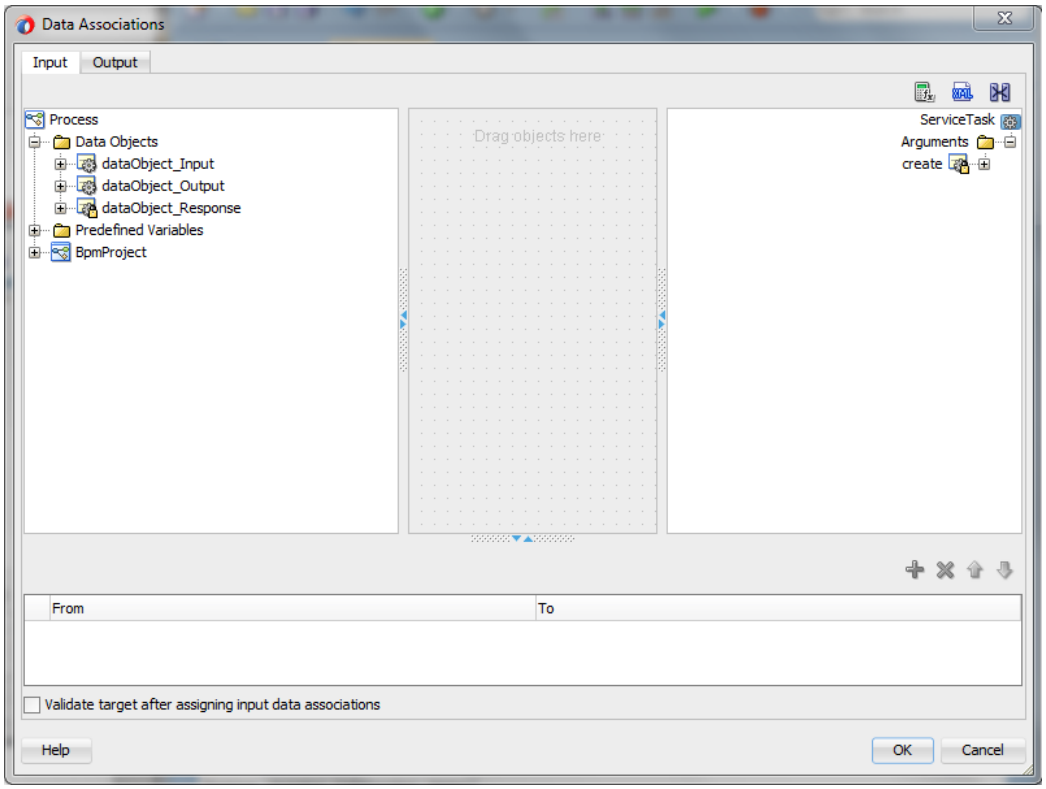
16. In **Service** section, click on **Magnifier** button and select your service for Salesforce as shown in the Figure below.

Figure 7- 51 Choose Salesforce Adapter Service



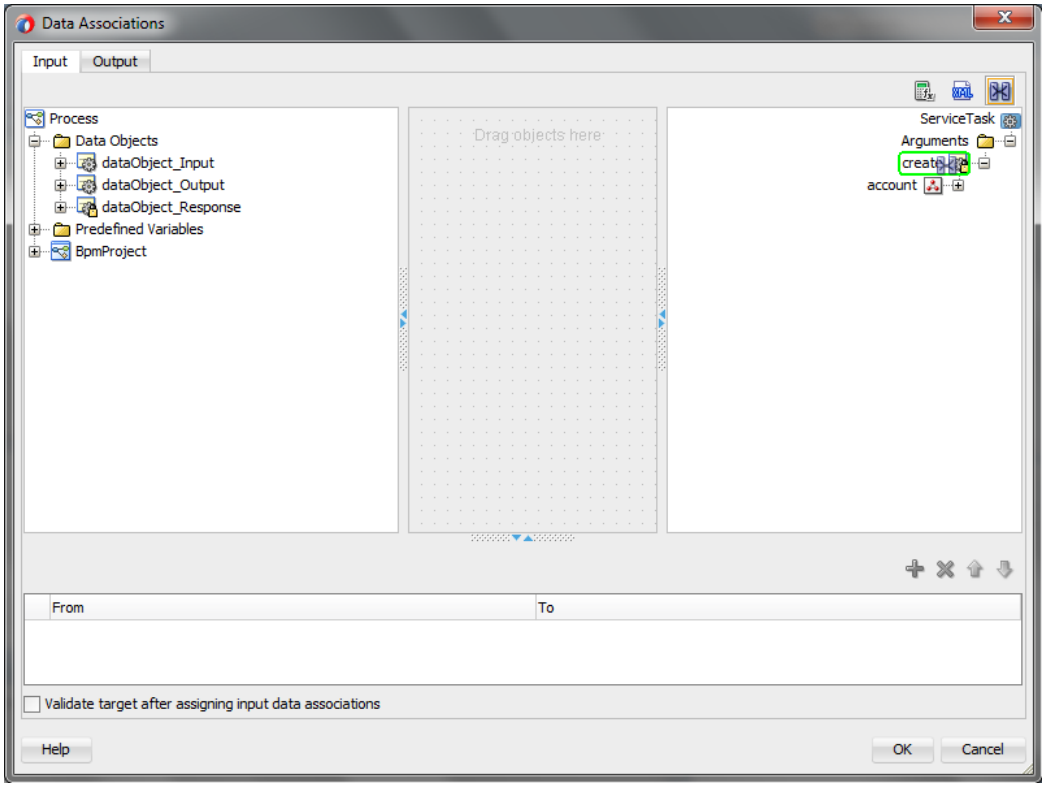
17. Click on **Data Association** and a new **Data Association** page will open as shown in the Figure below.

Figure 7- 52 Data Associations



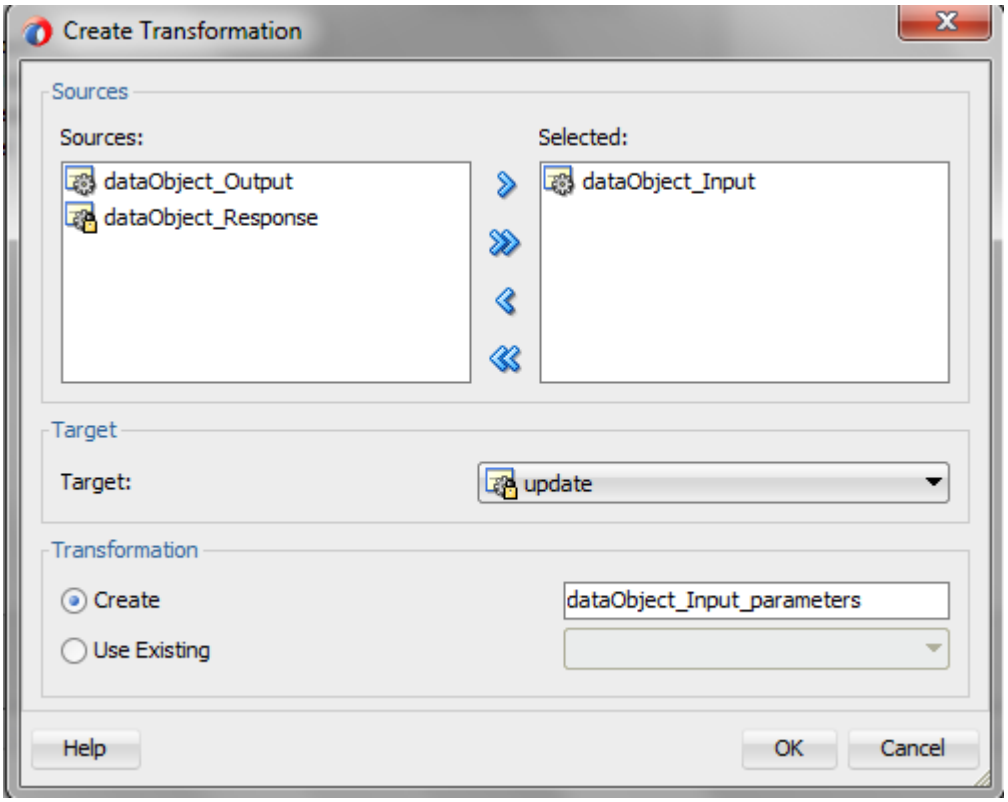
18. Now Click on **Transformation** button and drag it to **create** under Arguments in **ServiceTask**.

Figure 7- 53 Transform in Data Associations



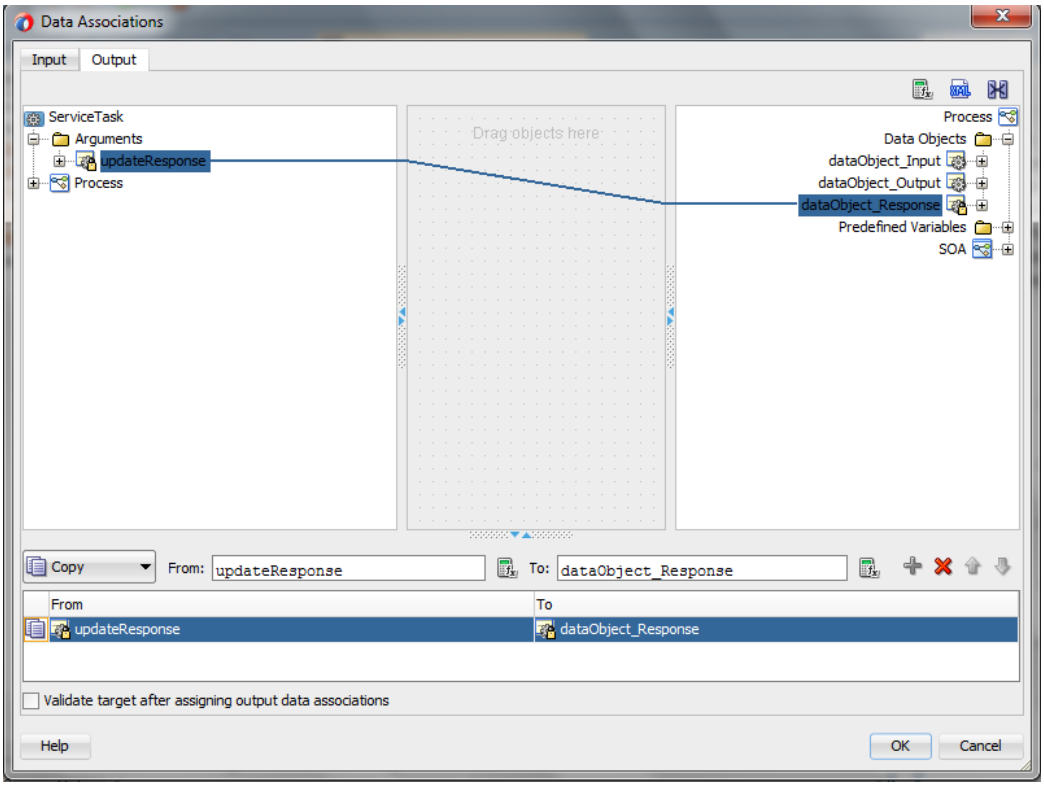
19. A new page **Create Transformation** opens. Now, move **dataObject_Input** from the list of **Source** side to the list of **Selected** and click **OK** Button as shown in the Figure below.

Figure 7- 54 Create Transformation



20. Click on the **Output** tab and map the **updateResponse** from the **serviceTask** to the **dataObject_Response** in **Process** and click **OK** button, as shown in the Figure below.

Figure 7- 55 Map Data Associations

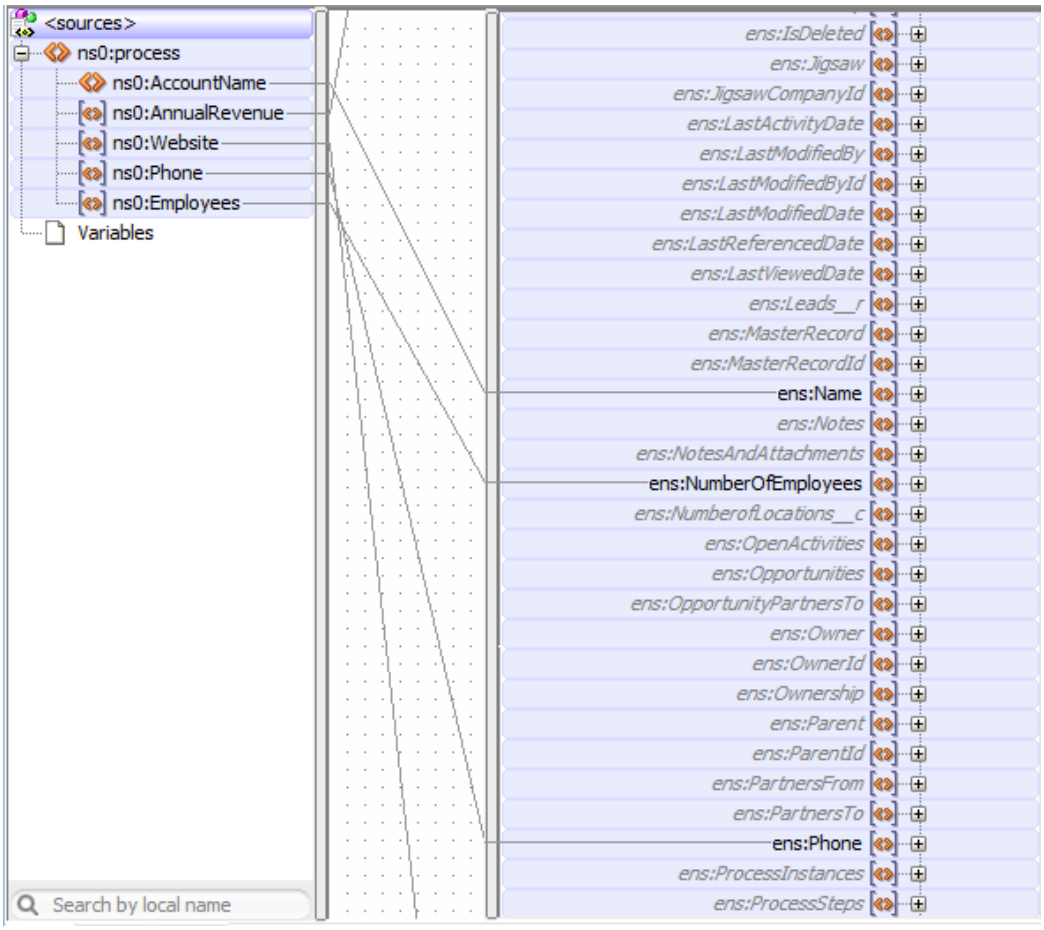


21. Perform the mapping from Source side to the Target side:

- Map AccountName to Name.
- Map AnnualRevenue to AnnualRevenue

- Map Website to Website.
- Map Phone to Phone.
- Map Employees to NumberOfEmployees.

Figure 7- 56 Mappings for input variable



22. From the **Activity** Tab, click and drag the **Script** Activity in between **ServiceTask** and **End** Activity, as shown in the Figures below.

Figure 7- 57 Activity Tab

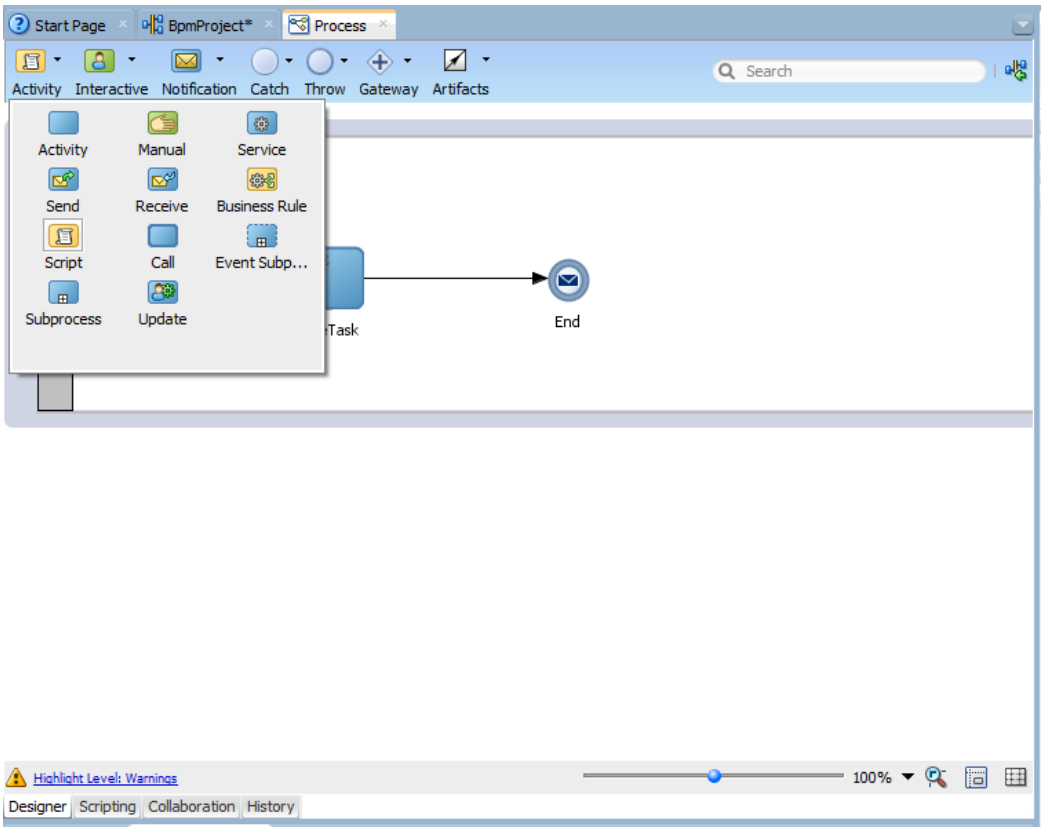
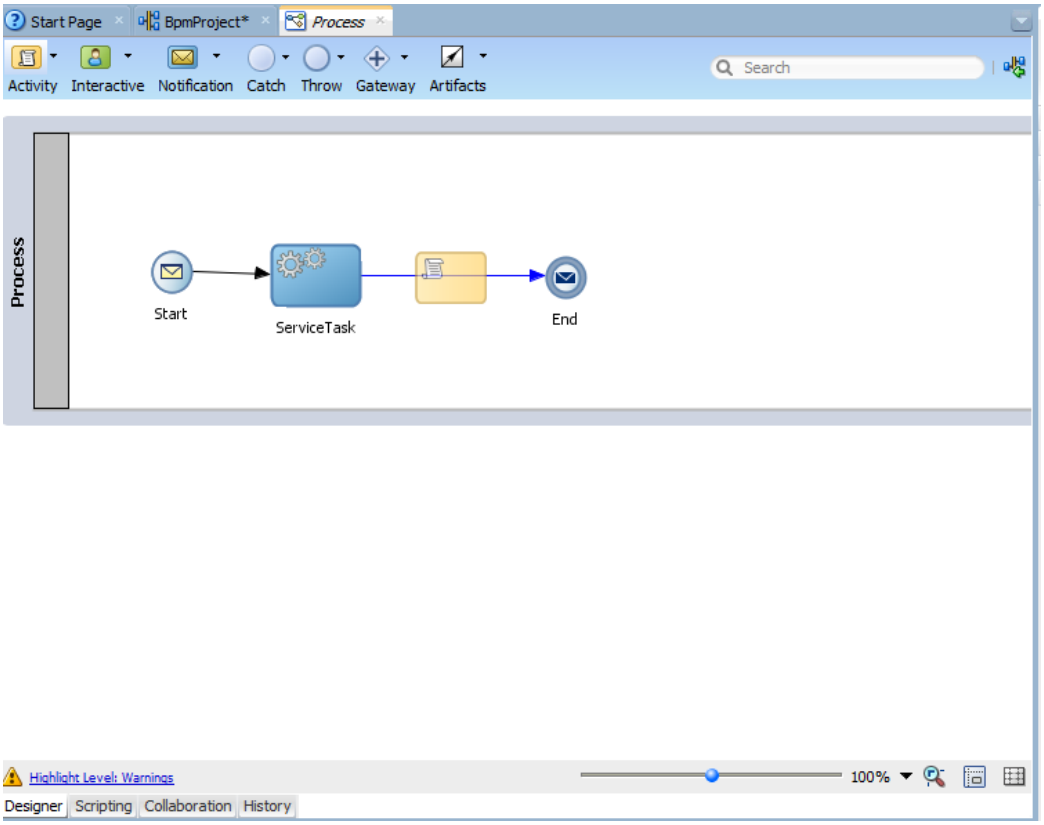
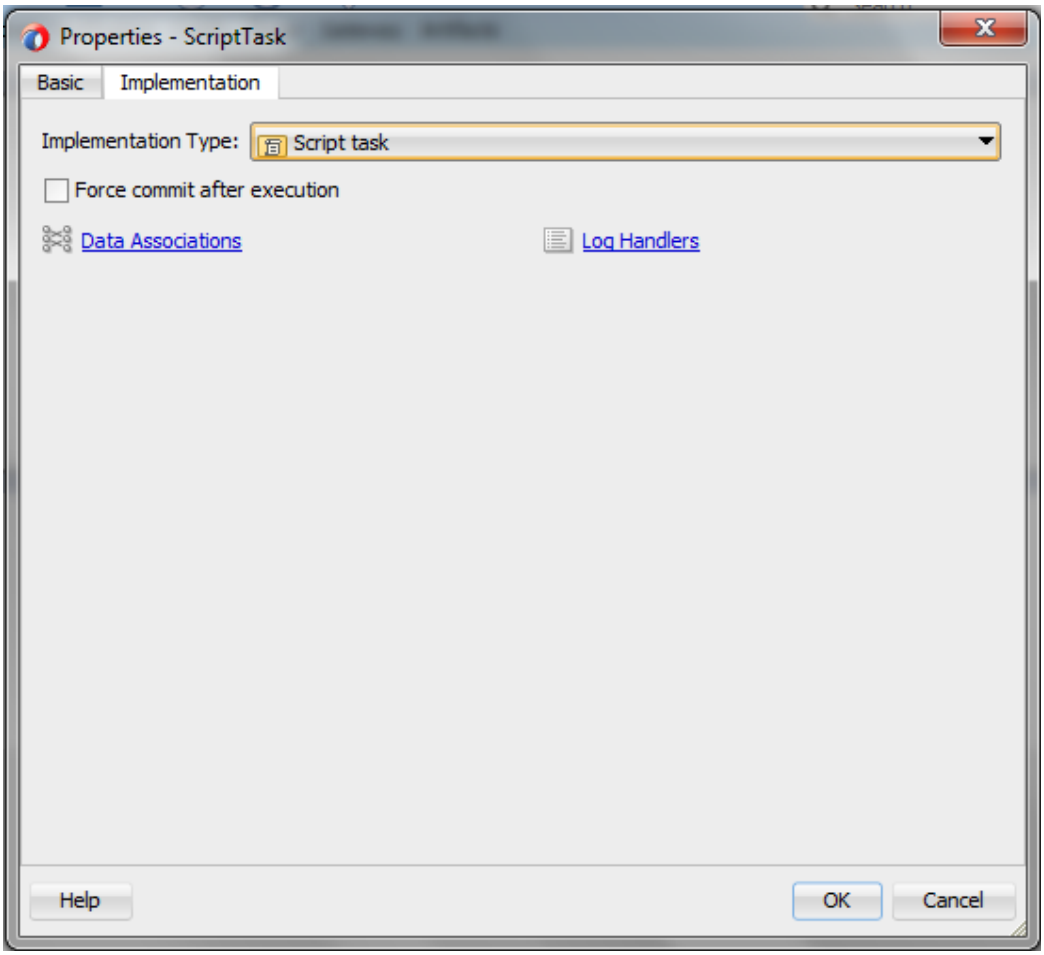


Figure 7- 58 Script Activity



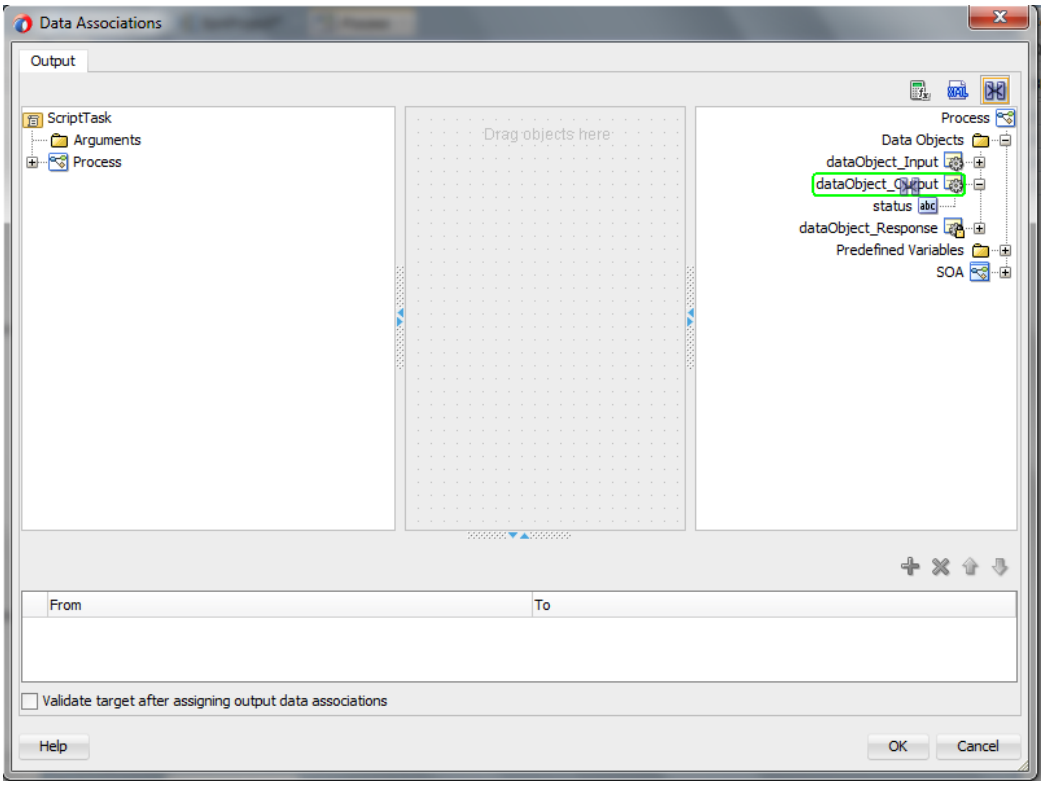
23. A new page **Properties-ScriptTask** opens. Click on **Implementation** tab and then click on **Data Association** as shown in the Figure below**.**

Figure 7- 59 Select Data Associations



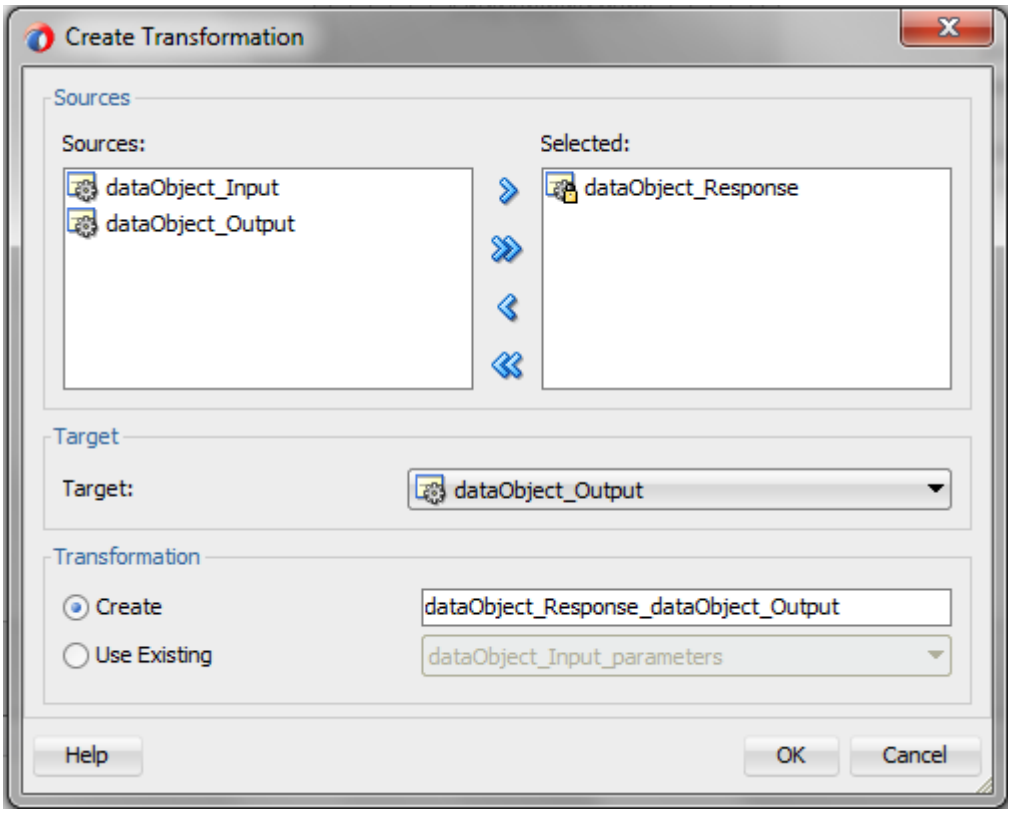
24. A new **Data Association** page opens. Click and drag the transformation icon to **dataObject_Output** under **DataObjects** of **process** as shown in the Figure below.

Figure 7- 60 Transform in Data Associations



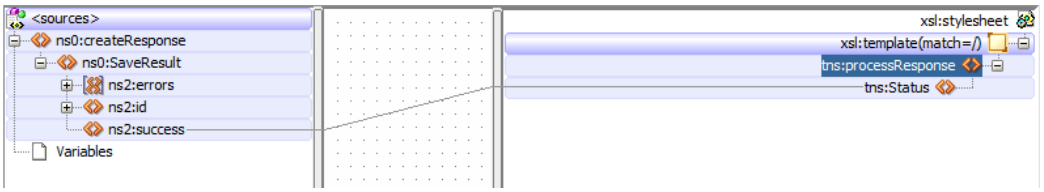
25. The **Create Transformation** page opens. Move **dataObject_Response** from the list of **Source** side to the list of **Selected** and click **OK** Button as shown in the Figure below.

Figure 7- 61 Create Transformation



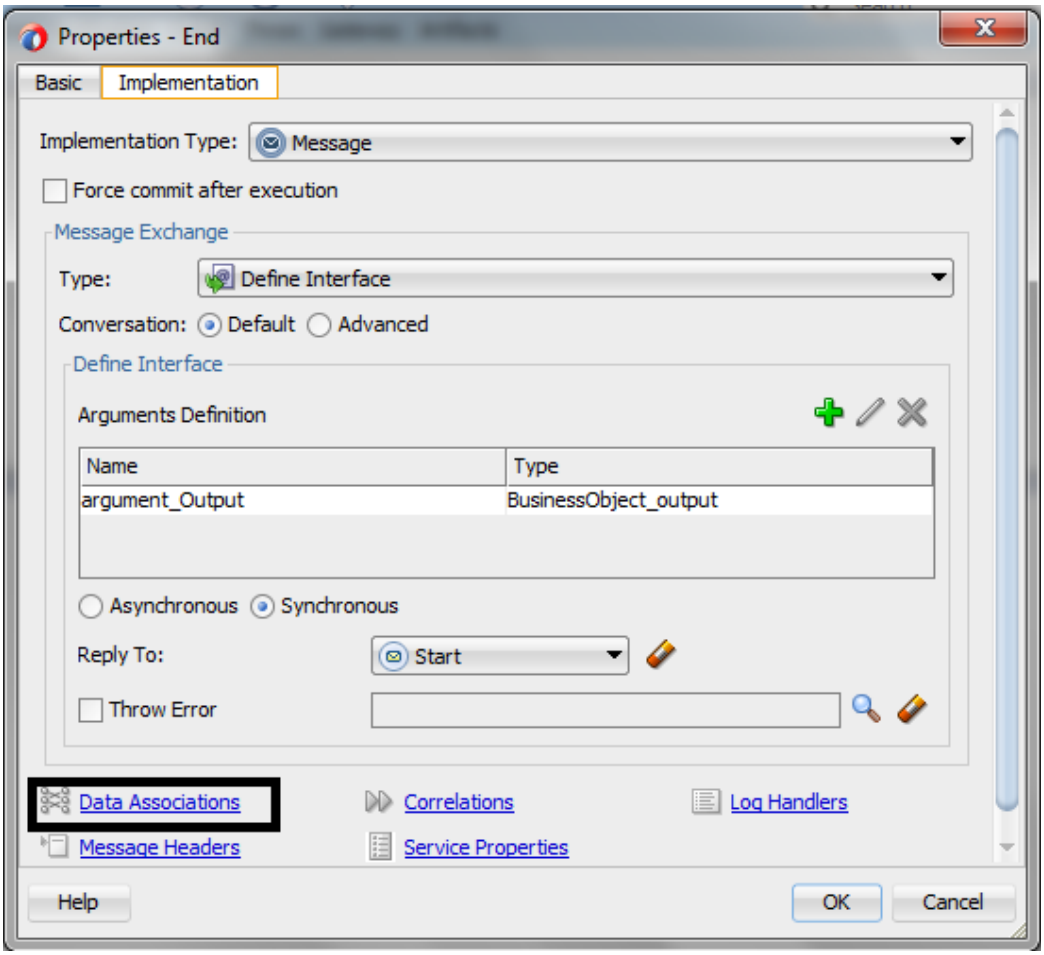
26. Now perform the mapping from Source side to the Target side.

Figure 7- 62 Mappings for Output Variable



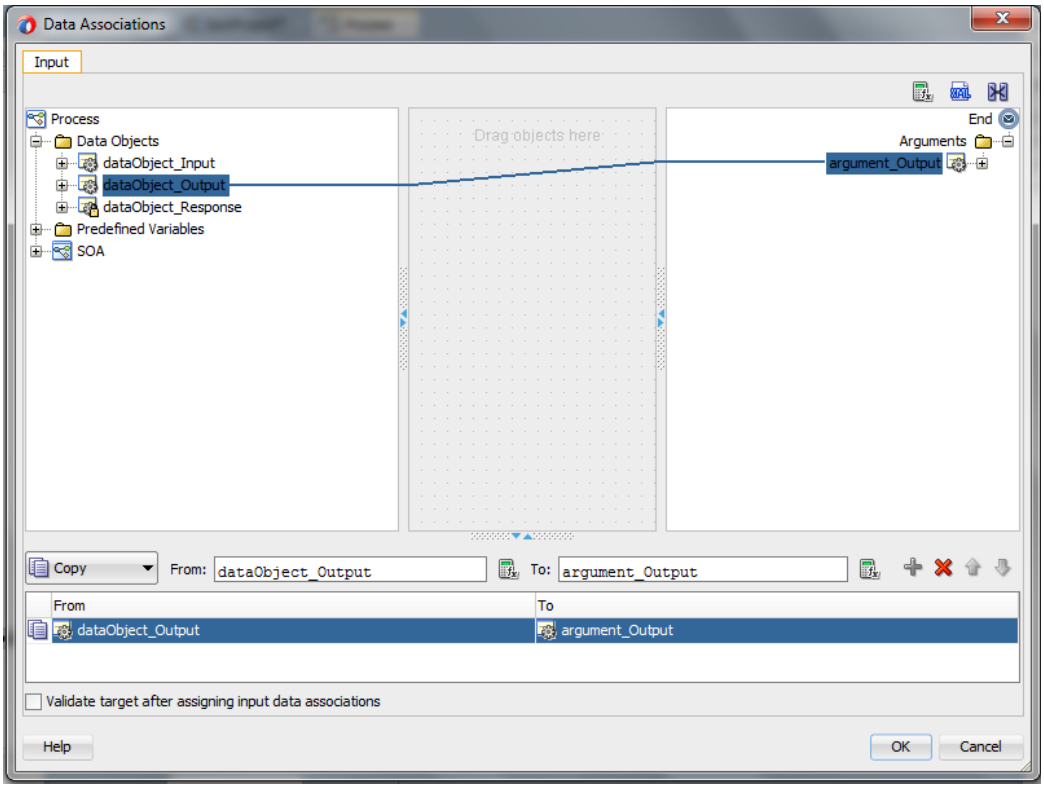
27. Now Double Click the **End** activity and then click on **Implementation** tab and then click on **Data Association** as shown in the Figure below.

Figure 7- 63 Select Data Associations



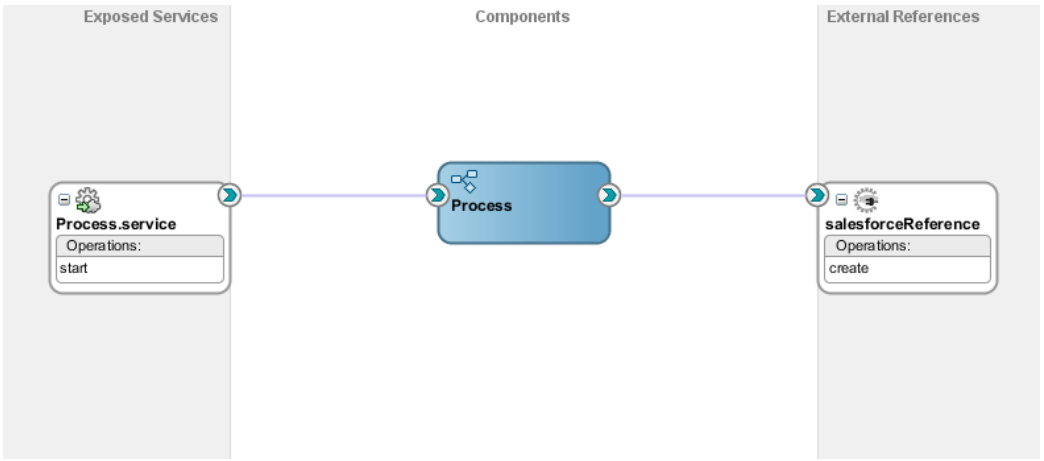
28. A new **Data Association** page will open, map the **dataObject_Output** of **process** to **argument_Output** of **End** and click **OK** as shown in the Figure below.

Figure 7- 64 Map Data Associations



29. This completes the project creation. The composite.xml looks as shown in the Figure below.

Figure 7- 65 Composite.xml

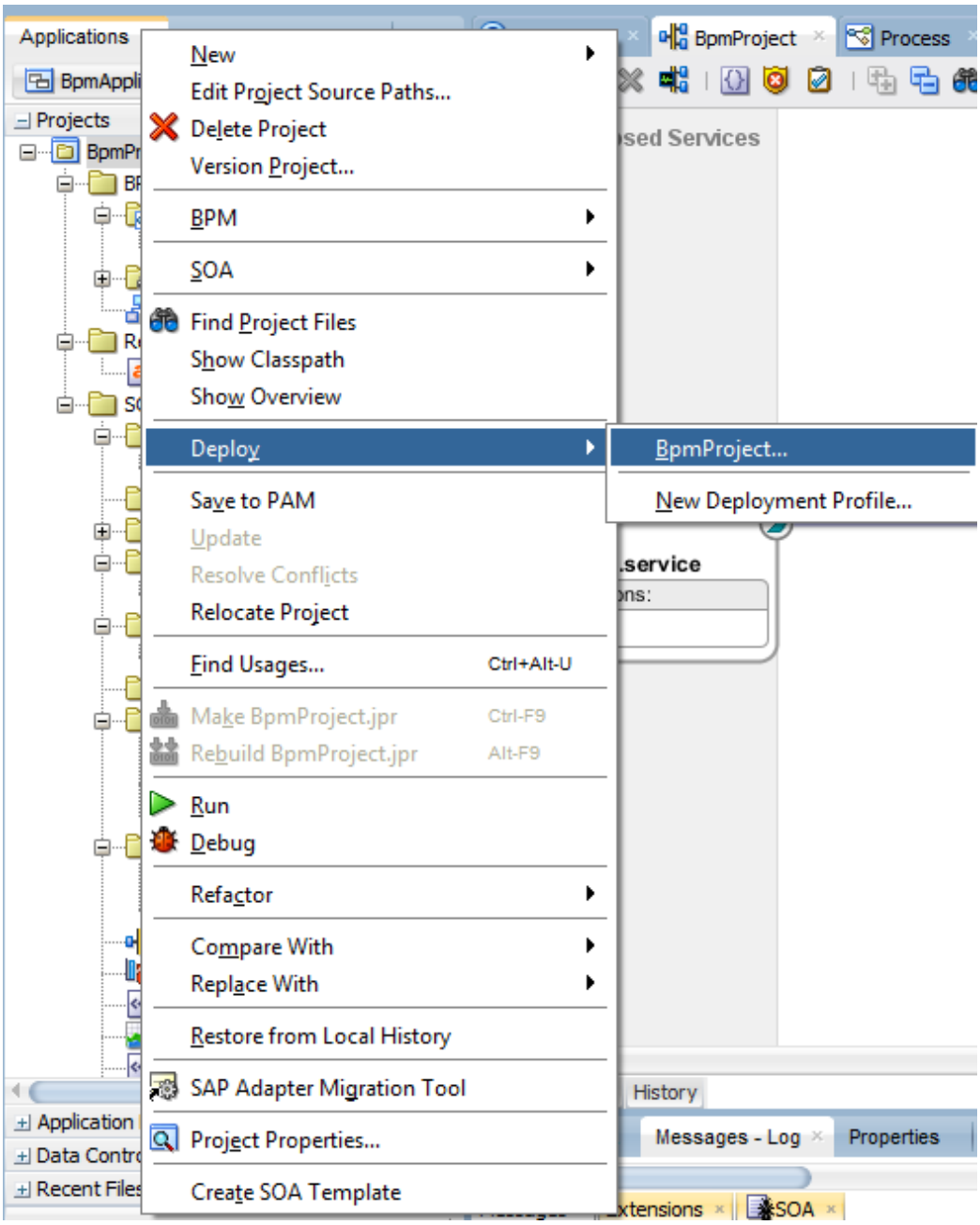


7.6 Deploy the Composite

Perform the following steps to deploy the composite

1. In the **Application Navigator** pane, right-click **BpmProject** and select Deploy → BpmProject, as shown in the Figure below.

Figure 7- 66 Deploy the composite



2. Select the **Deploy to Application Server** option and follow the instructions. Using this option, you can deploy the composite on the Application server after providing the details of the server.

7.7 Test the Composite

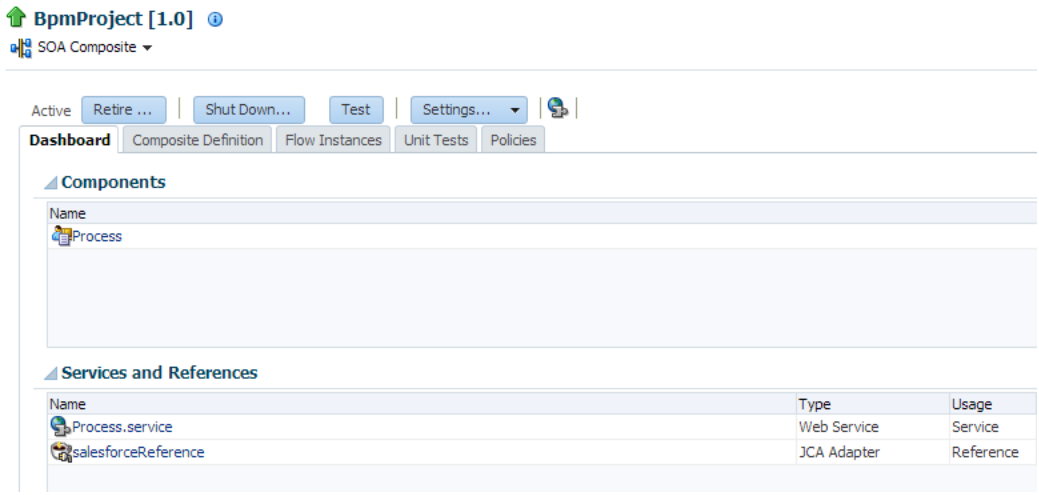
You can run and test the instances of deployed BPM composite applications from Oracle Enterprise Manager Console. This enables you to manage a composite application, initiate and track an instance of a composite and to view detailed component instance audit trails. Perform the following process to test the composite.

7.7.1 Test the Outbound Process

Perform the following steps to test the Outbound process:

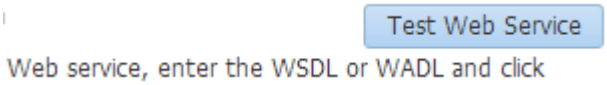
1. Login to Enterprise Manager Console for the server on which you have deployed your project.
2. Open **BpmProject** under the **Default** partition.
3. Click the **Test** button to test the Web service, as shown in the Figure below.

Figure 7- 67 Test the project



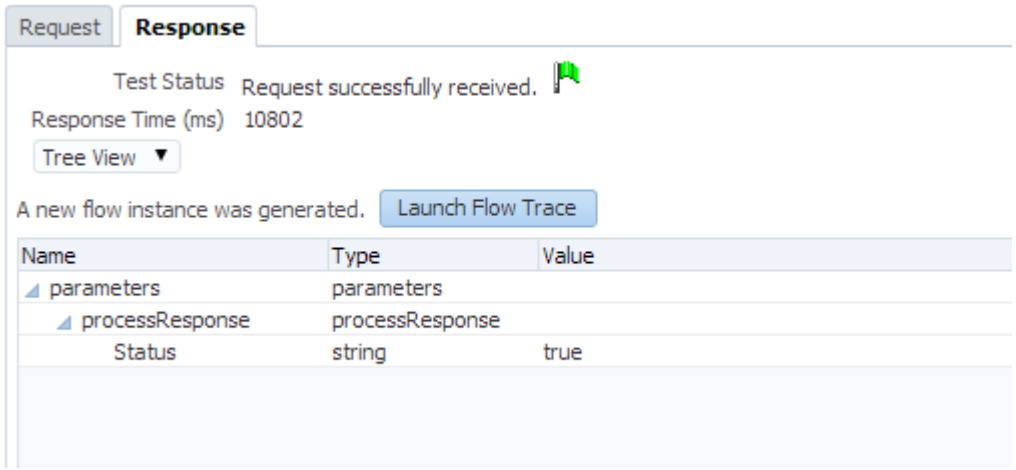
4. Provide the input payload and click the **Test Web Service** button, as shown in the Figure below.

Figure 7- 68 Test Web Service



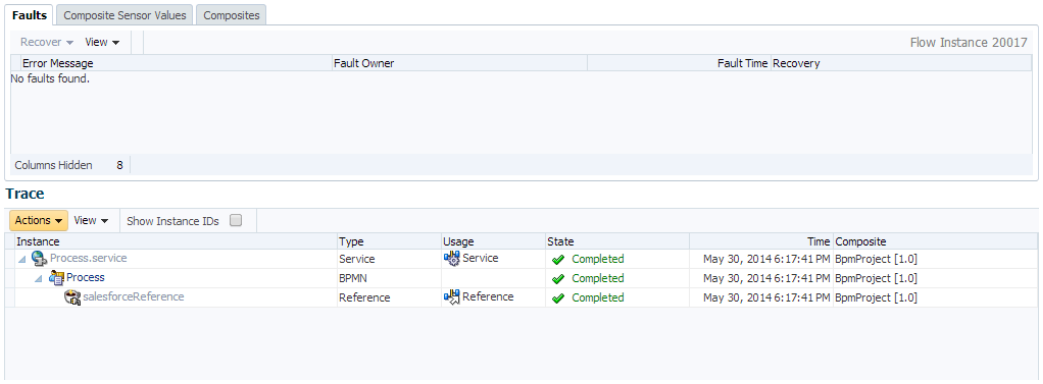
5. After successful execution, the response contains **Status** as true for Account update, as shown in the Figure below.

Figure 7- 69 Test Status



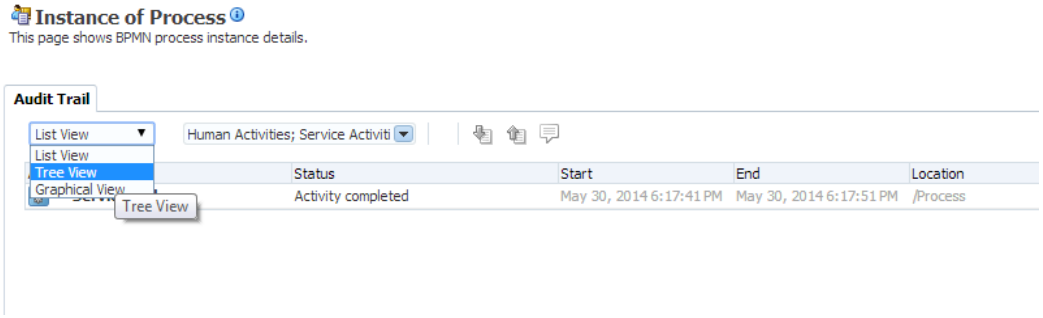
6. Click the **Launch Flow Trace** button to view the **Audit Trail**, as shown in the Figure below.

Figure 7- 70 Flow Trace



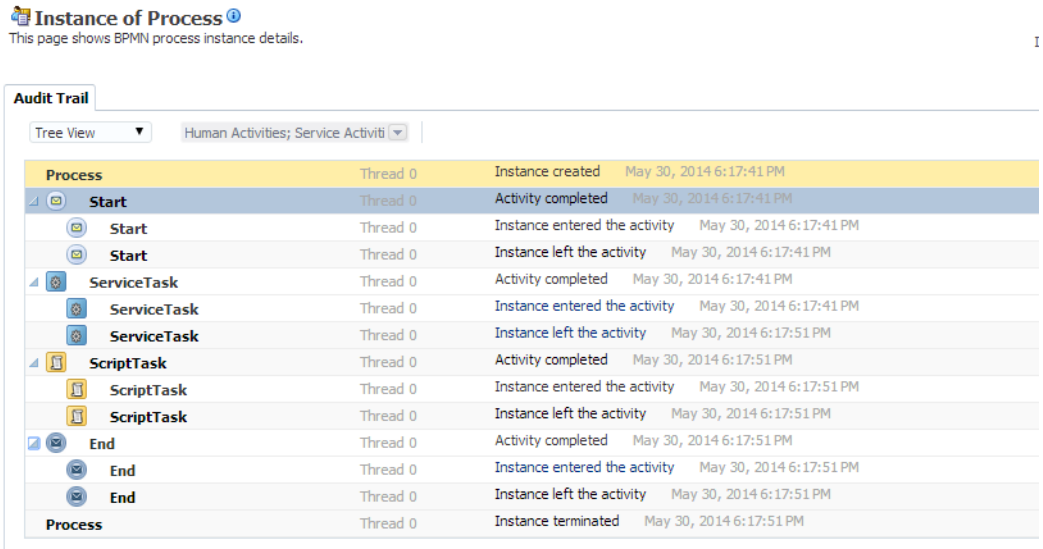
7. Click on List view and select Tree view to see the Audit trail in details, as shown in the Figure below.

Figure 7- 71 Audit Trail



8. The **Audit Trail** will look like as shown in the Figure below.

Figure 7- 72 Audit Trail – Tree View

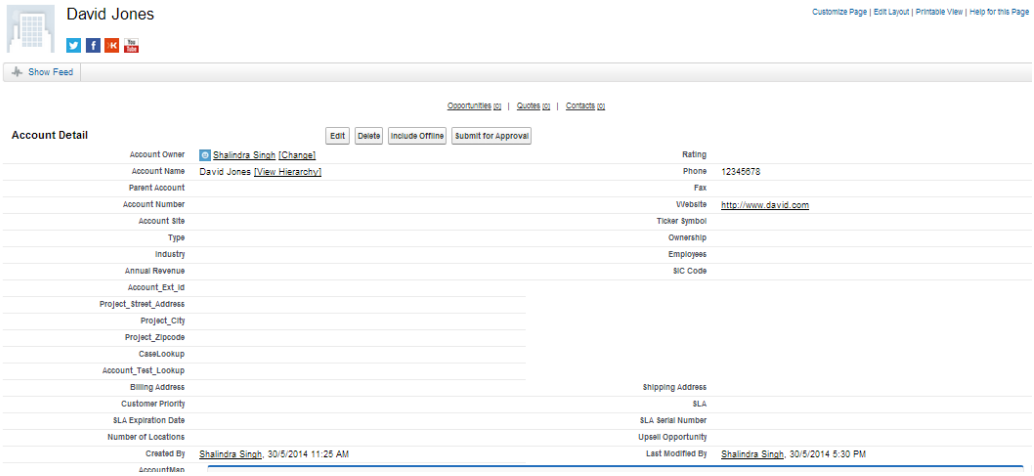


9. The Service Task of the process is shown in the Figure below.

Figure 7- 73 End Activity with Response returned

10. This completes the Create Account scenario. The created account can be seen in the Figure below.

Figure 7- 74 Created Account



8

Configuring the Oracle Cloud Adapter for Salesforce.com on Oracle WebLogic Server

This chapter gives an overview of the design-time and run-time configuration properties.

This chapter contains the following topics:

- Section 8.1, "Oracle Cloud Adapter for Salesforce.com Run-Time Properties"

8.1 Oracle Cloud Adapter for Salesforce.com Run-Time Properties

The section describes different run-time properties. This section is subdivided into multiple sections that includes:

- Section 8.1.1, “Generic Properties”
- Section 8.1.2, “Properties available in the response”
- Section 8.1.3, “Retry Properties”
- Section 8.1.4, “Precedence of Salesforce.com Property Values”

8.1.1 Generic Properties

Table 8-1 shows the Run-time properties supported by Oracle Cloud Adapter for Salesforce.com. You can provide the values for these properties while creating the BPEL process inside the invoke activity or using the Enterprise Manager console, as desired.

Some of these properties are also available in **Salesforce Cloud Adapter Configuration Wizard** for some particular operations.

Figure 1- 4 Run-time Properties

Property Name	Type	Default Value	Description	Configurable At
jca.salesforce.AllOrNoneHeader.allOrNone	Boolean	True	The AllOrNoneHeader header property allows a call to roll back all changes unless all records are processed successfully.	Oracle Cloud Adapter for Salesforce Configuration Wizard/ BPEL invoke activity
jca.salesforce.LocaleOptions.language	String	en_US	Specifies the language of the labels returned. The value must be a valid user locale (language or country), such as en_US.	BPEL invoke activity

jca.salesforce.QueryOptions.batchSize	integer	500	Specifies the batch size for queries. The default is 500; the minimum is 200, and the maximum is 2,000.	Oracle Cloud Adapter for Salesforce Configuration Wizard/ BPEL invoke activity
jca.salesforce.HttpTimeout	Integer	10 sec	Maximum value 120 sec. It specifies the timeout value for the transactions.	BPEL invoke activity
jca.salesforce.queryLocator	String	NA	The queryLocator value is returned in case of query and queryAll calls from Salesforce.com. If queryLocator value is not empty, it means there are more records to fetch for the same query. You can assign the queryLocator value to this property (jca.salesforce.queryLocator) in your BPEL process and invoke query or queryAll again. It will fetch the next set of records for the same query string. It is similar to calling queryMore operation to fetch the next set of records. Refer to the section “How to use query and queryMore Operations”	BPEL invoke activity

8.1.2 Properties available in the response

Table 8-2 shows the properties available in the response.

Table 8- 1 Properties Available in the Response

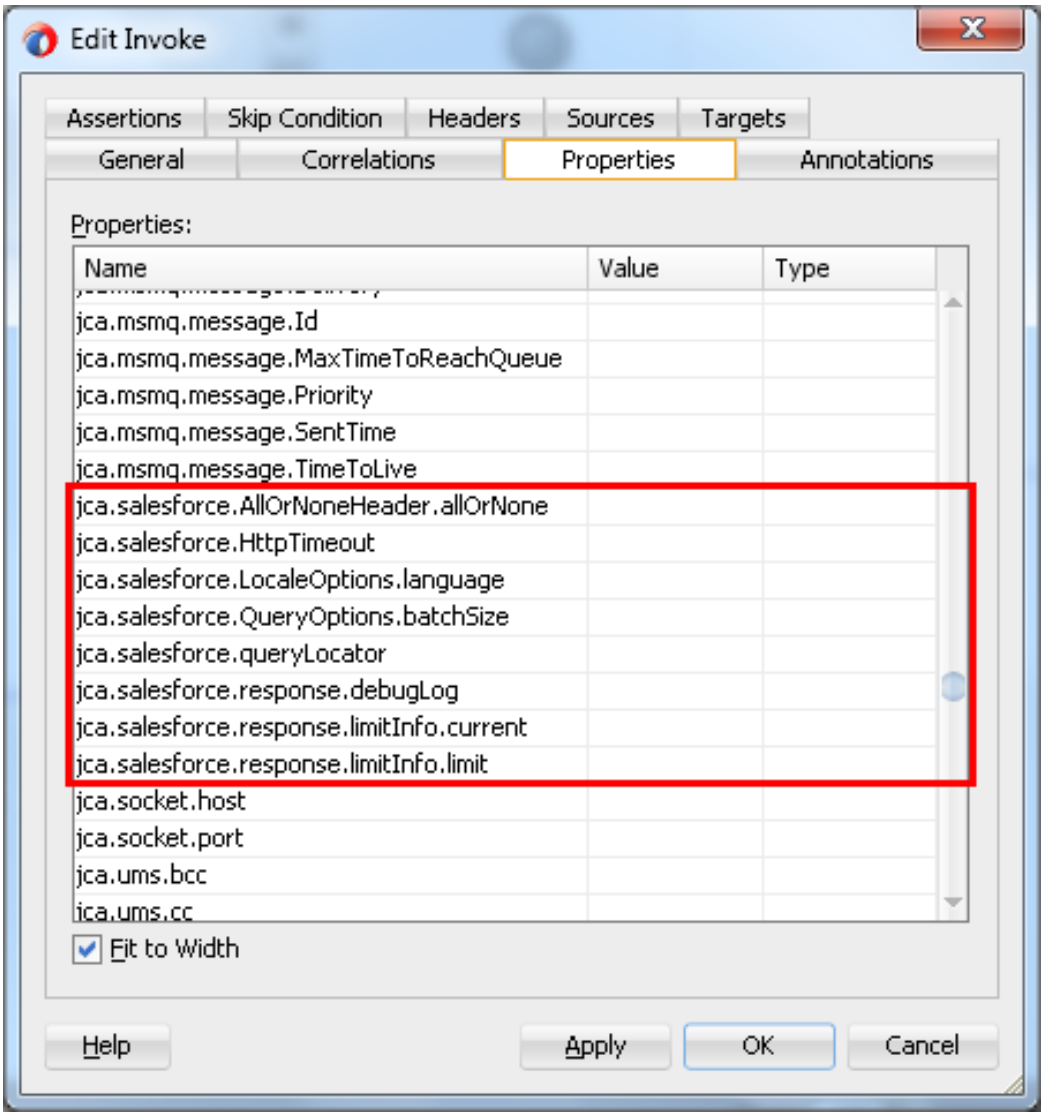
Property Name	Type	Default Value	Description	Configurable At
jca.salesforce.response.debugLog	String	NA	debugLog is returned as part of Header response, if you provide a value for debug header in the configuration wizard of the Salesforce.com adapter. To access this value you need to create a variable of type ‘string’ and extract the value of debug log into this variable in the properties section of your Invoke activity.	BPEL invoke activity

jca.salesforce.response.limitInfo.current	String	NA	This value is returned as part of Header response, specifying the number of calls that have already been used in the organization. (Supported in Salesforce.com 29.0 version or higher) To access this value you need to create a variable of type 'string' and extract the value of this property into that variable in the properties section of your Invoke activity.	BPEL invoke activity
jca.salesforce.response.limitInfo.limit	String	NA	This value is returned as part of Header response, specifying the organization's limit for the number of calls it can make. (Supported in Salesforce.com 29.0 version or higher). To access this value you need to create a variable of type 'string' and extract the value of this property into that variable in the properties section of your Invoke activity.	BPEL invoke activity

Providing the Property Values in BPEL Invoke Activity

Figure 8-1 depicts the properties available in the Invoke activity of BPEL Process. The prefix **jca.salesforce** determines that these properties are pertaining to the Salesforce.com adapter. In the value section, you can assign values to these properties using a variable or an expression if the type of property is "input". On the other hand, if the type of property is "output", its value can be extracted into a variable.

Figure 8- 1 Properties Available in the Invoke Activity of BPEL Process



8.1.3 Retry Properties

Table 8- 2 shows the configurable properties related to Oracle Cloud Adapter for Salesforce.com. These properties can be modified in the composite file or at the Enterprise Manager console.

Table 8- 3 Oracle Cloud Adapter for Salesforce.com Configurable Properties

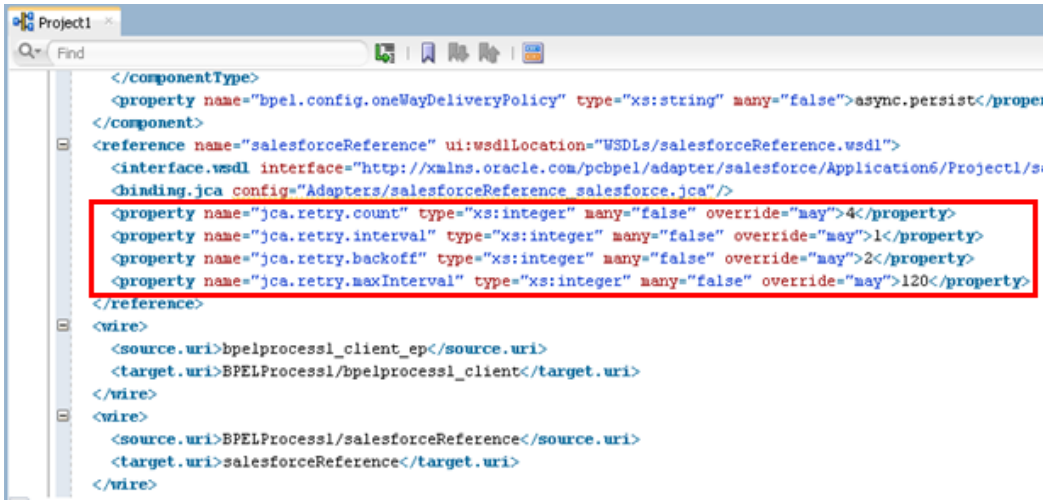
Property Name	Type	Default Value	Description	Available at
jca.retry.count	integer	4	Specifies the number of retries to post the message.	Composite.xml, EM console
jca.retry.backoff	integer	2	Specifies the retry interval growth factor.	Composite.xml, EM console
jca.retry.interval	integer	1	Specifies the time interval between two retries.	Composite.xml, EM console

jca.retry.maxInterval	integer	120	Specifies the maximum interval between two retries.	Composite.xml, EM console

8.1.3.1 Providing the Property Values in the Composite

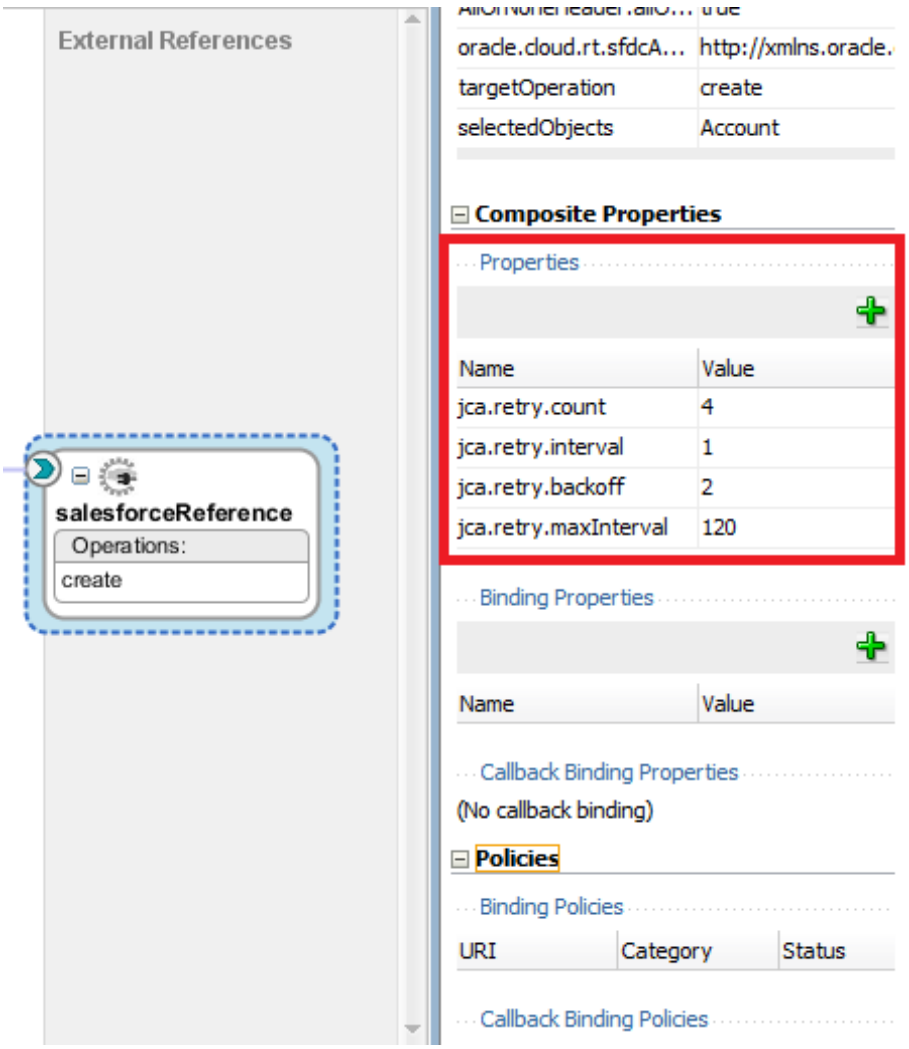
Figure 8-2 shows the properties available in the composite.xml file.

Figure 8- 2 Composite.xml File Properties



You can also edit the value of these properties in the **Properties** section of Salesforce.com Adapter. The properties section appears if you click on the Salesforce.com Adapter in the **External References** section as shown in Figure 8-3.

Figure 8- 3 Salesforce Adapter Properties



8.1.3.2 Providing the Property Values at Enterprise Manager Console

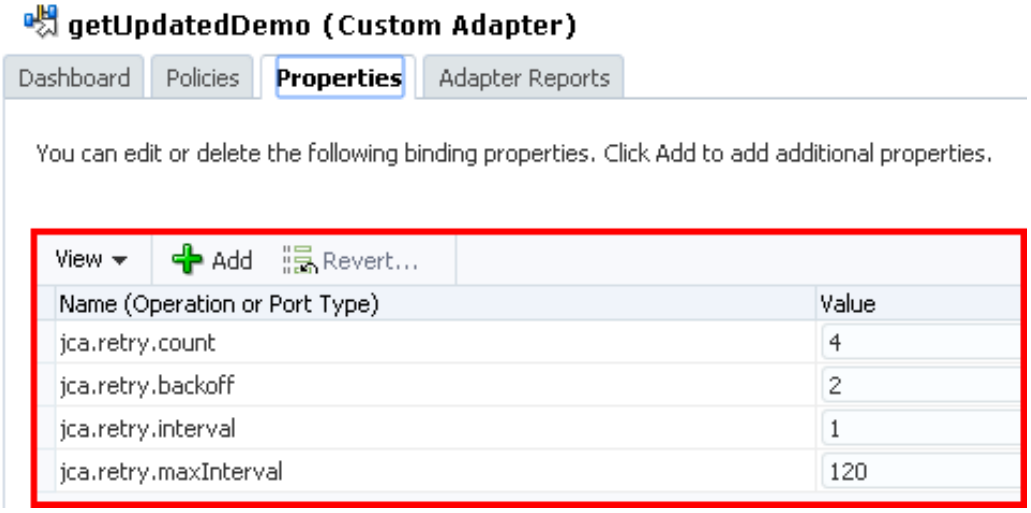
To provide the property values at the Enterprise Manager Console, follow the steps given below:

1. Proceed to **Services & References** section of your project, as shown in Figure 8-4.

Figure 8- 4 Services & References Section

2. Select the service name of your Salesforce.com Adapter from this section, and click on the **Properties** tab on the subsequent page, as shown in Figure 8-5.

Figure 8- 5 Salesforce Adapter Properties



8.1.4 Precedence of Salesforce.com Property Values

If there is provision to set a particular property at multiple positions, the highest priority would be given to the value provided at the Enterprise Manager Console, next priority would be given to the value provided while creating the composite, while the lowest priority would be given to the value provided at Salesforce Cloud Adapter Configuration Wizard.

9 Troubleshooting and Error Messages

This chapter explains the troubleshooting information and error messages that you can come across while configuring the Oracle Cloud Adapter for Salesforce.com. It contains the following topics:

- Section 9.1, "Troubleshooting and Error Messages"
- Section 9.2, "API Fault"
- Section 9.3, "Status Code"
- Section 9.4, "Known Issues"

9.1 Troubleshooting and Error Messages

The Oracle Cloud Adapter for Salesforce.com supports the adapter diagnostic framework for reporting and alerting. This provides run-time adapter diagnostic information as read-only reports in EM console.

The troubleshooting information is categorized as follows:

- Oracle Cloud Adapter for Salesforce.com Design-Time JDeveloper
- Oracle Cloud Adapter for Salesforce.com Run-Time

Log file information that can be relevant in troubleshooting can be found in the following locations based on the adapter installation.

The Oracle Cloud Adapter for Salesforce.com trace information can be found under the following directory:

For Oracle SOA Suite:

```
<ORACLE_HOME>\soa\user_projects\domains\${soa_server
domain}\servers\${soa_server name}\logs\soa-server_diagnostic.log
```

9.1.1 Oracle SFDC Cloud Adapter Design-Time JDeveloper

Table 9-1 shows the common errors faced while using Oracle Cloud Adapter for Salesforce.com in JDeveloper and their possible solutions.

Table 9- 1 Oracle Cloud Adapter for Salesforce.com in JDeveloper

Error	Solution
The login credentials are not valid, or the maximum number of logins has been exceeded. Contact your administrator for more information.	Provide the valid Salesforce.com credentials. Delete the CSF key and add it again providing the correct credentials.
Metadata for the enterprise WSDL is not available in the cache.	Connect to the internet and click on the offline configuration checkbox. This will download the metadata for offline usage.
Unable to connect to Salesforce.com server.	Ensure that you are connected to the internet and not blocked by or behind the firewall of your organization.
Error deploying the composite on soa_server: Composite with same revision ID already exists.	Check whether the project is already deployed on the server. Check the overwrite composites with the same revision box or change the revision number.
java.net.ConnectException: Connection refused: connect; No available router to destination.	Check whether the SOA server is up and running.

9.1.2 Oracle SFDC Cloud Adapter Run-time

Table 9-2 shows the common errors faced in the SOA server run-time.

Table 9- 2 Error in SOA server Run-time

Error	Solution
Unable to find username in credential store.	Make sure that CSF key being used in the adapter during the design-time is available in the credential store in the WebLogic server.
Invalid ID. ID does not belong to the selected object.	Ensure that you are providing only the ID's corresponding to the objects selected at the design-time for the operation.
Bind Parameter's values are missing.	Ensure that you have provided the values of all the bind parameters defined in the query or search string at design-time.
Bind Parameter's value is empty.	Ensure that the value of bind variable is not empty.
FileNotFoundException.	Ensure that the Enterprise WSDL file you have provided is localized into your composite, or, in case you are using MDS, make sure that it is available in the MDS.
Unknown exception while transforming request message.	Please make sure that parameter Hostname Verification field (Under the tab SSL) is set to NONE on the server side as this is the most common reason.

Set Hostname Verification to None

You might come across the Exception: *javax.net.ssl.SSLKeyException* due to failed hostname verification check on the server side when you test your composite. This error comes when Hostname Verification parameter on the server side is set to a value other than "*None*". To overcome this you can follow the below mentioned steps:

1. Login into WebLogic console.
2. Browse to the servers and select manage server on which Salesforce adapter is running (e.g., soa_server1).
3. Go to Configuration →SSL.
4. Expand **Advanced** section. You will see 'Hostname Verification'.
5. Select the "*None*" value from the dropdown and save the changes.

9.2 API Fault

You can refer the link below to get more information about the API Fault.

http://www.salesforce.com/us/developer/docs/api/Content/sforce_api_calls_concepts_core_data_objects.htm#i1421192

9.3 Status Code

You can refer the link below to get more information about the Status Code.

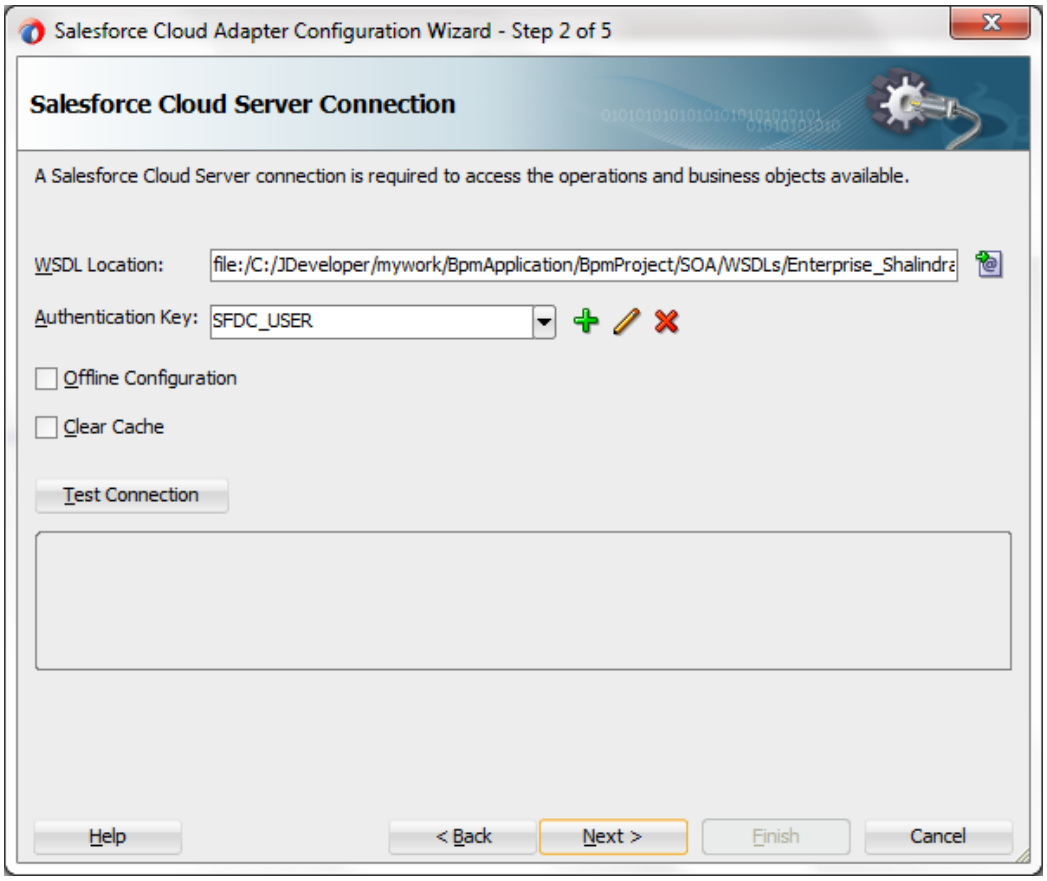
http://www.salesforce.com/us/developer/docs/api/Content/sforce_api_calls_concepts_core_data_objects.htm#i1421521

9.4 Known Issues

1. **Run-Time execution failure due to absence of the local copy of Enterprise WSDL in JDeveloper SOA Project:** While configuring Oracle Cloud Adapter for Salesforce.com, you will need to browse for the Enterprise WSDL of your Salesforce.com organization. When you point to the location of the WSDL, JDeveloper creates a copy of the WSDL in the JDeveloper SOA Project (say Project1). This copy of the WSDL is needed by the Oracle Cloud Adapter for Salesforce.com during Run-Time execution.

   The next time you create the adapter in a different project (say Project2), the adapter (by default) picks the location of the Enterprise WSDL localized in the previous project (Project1) and the previously CSF key from the cache, as shown in Figure 9-1.

Figure 9- 1 WSDL Location



You can click on **Next** button without browsing for the WSDL again and the adapter will be configured, but the execution will fail during Run-Time as the Enterprise WSDL has not been localized in the folder of Project2.

Temporary Fix: To avoid this, always browse to the location of the Enterprise WSDL (either in the file system or MDS) to localize it in the project folder.

2. **Package Version Header does not support multiple package versions:** Salesforce.com supports multiple package versions per operation in its SOAP calls. In Cloud Adapter for Salesforce.com, currently only one Package Version can be configured per operation, using the Header page of the Adapter Configuration Wizard.
3. **Relationship queries with custom object in which the child object is not suffixed with ‘s’ is not working for Oracle Cloud Adapter for Salesforce.com:** In relationship queries for standard objects, the child object ends with "s".
   For the custom objects, the child object may or may not have the "s" appended at the end depending on how the Salesforce admin has configured the custom objects' relationship.
   For instance, consider the following queries:
   1) In the following parent-to-child query for standard object, the child object ‘Contact’ is suffixed with "s" and hence the query is working for the Adapter.
   *SELECT ID, Name, (SELECT ID, FirstName, LastName FROM **Contacts**) FROM Account*
   2) In the following parent-to-child query for custom objects, the child object may or may not be suffixed with "s" depending on how the relationship was configured by the Salesforce admin. Such queries (Daughters_r and Daughter_r) which are not suffixed with ‘s’ are currently not working for the Adapter.

*SELECT ID, Name, Anniversary__c, (SELECT Id, Name, Birthday__c FROM Mother__c.**Daughters__r**) FROM Mother__c*

OR

*SELECT Id, Name, Anniversary__c, (SELECT Id, Name, Birthday__c FROM Mother__c.**Daughter__r**) FROM Mother__c*

4. **In BPEL orchestration of Oracle Cloud Adapter for Salesforce.com, "Transform" Oracle Extension with multiple sources gives error at Run-Time:** If you use a "Transform" Oracle Extension with single source in your BPEL orchestration, it works as expected. But, if you use a "Transform" Oracle Extension with multiple sources then it gives an error at Run-Time.

   Temporary Fix: Map the output of the Adapter (for e.g., to create operation) to an intermediate variable (referring to createResponse from the adapter WSDL i.e. a replica of response returned from Salesforce). Similarly, map the Output of the salesforce adapter for another operation (for e.g., to update operation) to another intermediate variable (referring to updateResponse from the adapter WSDL). Next a transform activity is introduced which has two sources -first, the intermediate variable that contained createResponse and second, the intermediate variable that contained updateResponse. This transform activity works with multiple sources of Salesforce Adapter responses

5. **In BPEL 2.0, response cannot be mapped using Assign activity:** While using BPEL 2.0 process, if you try to assign the value of response returned from Salesforce adapter to a BPEL variable, it fails with selection failure the exception message being - Exception is thrown because the "from – spec" is evaluated to be @ empty.

   Temporary Fix: In your BPEL process, create a variable of message type using the response message of adapter generated WSDL. First copy the response of Invoke activity's output variable (the one that calls the salesforce adapter) to the intermediate variable created above. Thereafter, using the intermediate variable, perform any mapping / assign to any other variable in your BPEL orchestration.

6. **Mobile Filter field on operation page:** It is possible to drag the filter field on operation page which leads to uncaught exception in JDeveloper.

   Temporary Fix: Do not try to drag the filter field on operation page of Oracle Cloud Adapter for Salesforce.com

7. Help content is not available on the Query/Search Test Window

8. **Double-click does not select WSDL in WSDL Chooser dialog:** You cannot select the Enterprise WSDL of your Salesforce organization in the WSDL chooser dialog of connection page, by double-clicking on it. You need to select the WSDL at the appropriate location, and then click **OK**.

9. **Double-click to select Salesforce Object doesn't work:** On the 'Cloud Operation Configuration' Page in the Business Objects sections, if you double-click on an object in the 'Available' section, it does not move to the 'Selected' section. Vice versa is also true, when you double-clicking on an object in the 'Selected' section, it does not move to the 'Available' section.

Temporary Fix: Use the arrow buttons to select or unselect objects.

10 **In BPMN process, Transformation of Salesforce adapter response is not working:** If in the data association tab of a BPMN process, you introduce a transformation, wherein Adapter response is the source element; it will fail at the run-time.

Temporary Fix: Output of salesforce adapter is being mapped to a data object2 (referring to createResponse from the adapter WSDL – i.e., is a replica of response returned from salesforce). Next, a Script task is introduced (which is just like assign/transform activity in BPEL). Using this script activity, the value of dataobject2 is mapped to another dataobject1 (referring to an element in our custom XSD) using a transformation. It is working till the end point i.e. multiple outputs are being returned.

9.5 Limitations

It is important to note that Salesforce API either supports Unicode characters or ISO-8859-1 character. The character set for a user depends on the Salesforce instance it uses. For the instance SSL, the encoding is ISO-8859-1 and for the rest, it is UTF-8. Moreover, if ISO characters are used in Salesforce organization in an instance which is non-SSL instance, can result in an error.

The Adapter for Salesforce.com is built to work on a single character encoding at a time that could either be ISO-8859-1 or UTF-8. Hence, if the organization use a character encoding different than the encoding assigned as per the Instance, the adapter may not recognize it and return an error. To learn more about the supported character sets by Salesforce, check ‘Internationalization and Character Sets’ section here.

Note: The instance of an organization is indicated as a prefix in the URL of the Salesforce login. For example, https://na2.salesforce.com, here NA2 is the instance. For more information on list of available Salesforce Instance, Click here.

---

10 Migration Support

This chapter provides information about the migration support in 12c version of Oracle Cloud Adapter for Salesforce.com for the 11g version of Oracle Cloud Adapter for Salesforce.com. This can be used as a reference by the organizations currently using the Oracle Cloud Adapter for Salesforce.com on Oracle SOA Suite 11g and migrating to the Oracle SOA Suite 12c.

This chapter contains the following topics:

- Section 10.1, "Complete Backward Compatibility"
- Section 10.2, "Migrating 11g Application and Projects to 12c"
- Section 10.3, "Points to Remember"

10.1 Complete Backward Compatibility

The Oracle Cloud Adapter for Salesforce.com is completely compatible with the previous release for Oracle SOA Suite 11g. All your composites containing the Oracle Cloud Adapter for Salesforce.com created in Oracle SOA Suite 11g can be ported to Oracle SOA Suite 12c by a simple process outlined in the next section of this chapter.

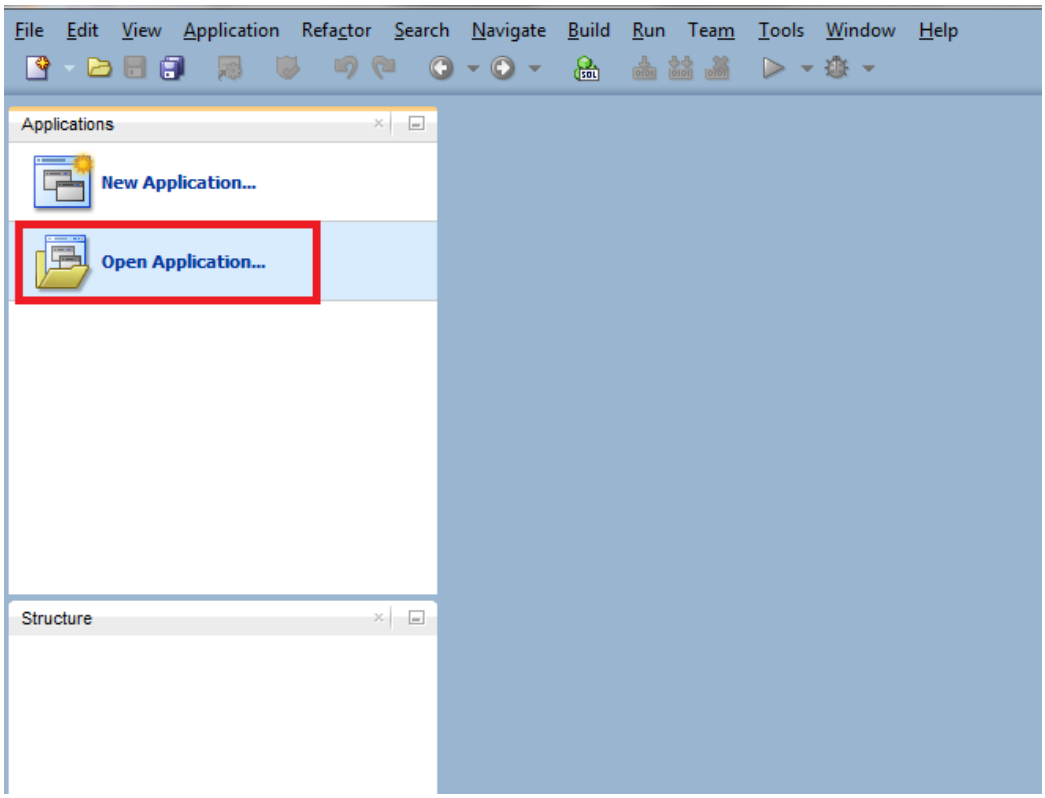
This will allow you to seamlessly upgrade to the latest version of Oracle SOA Suite and enjoy a host of new features. Once the migration of projects and application is done in the JDeveloper of 12c, you can upgrade your existing Oracle Cloud Adapter for Salesforce.com to the latest version by either double-clicking the adapter in External References swim lane or by right-clicking on it and selecting “Edit” as shown in the following section.

10.2 Migrating 11g Application and Projects to 12c

This section will outline the steps to migrate your applications and projects created in Oracle SOA Suite 11g to the latest version of Oracle SOA Suite 12c.
Follow the steps as shown:

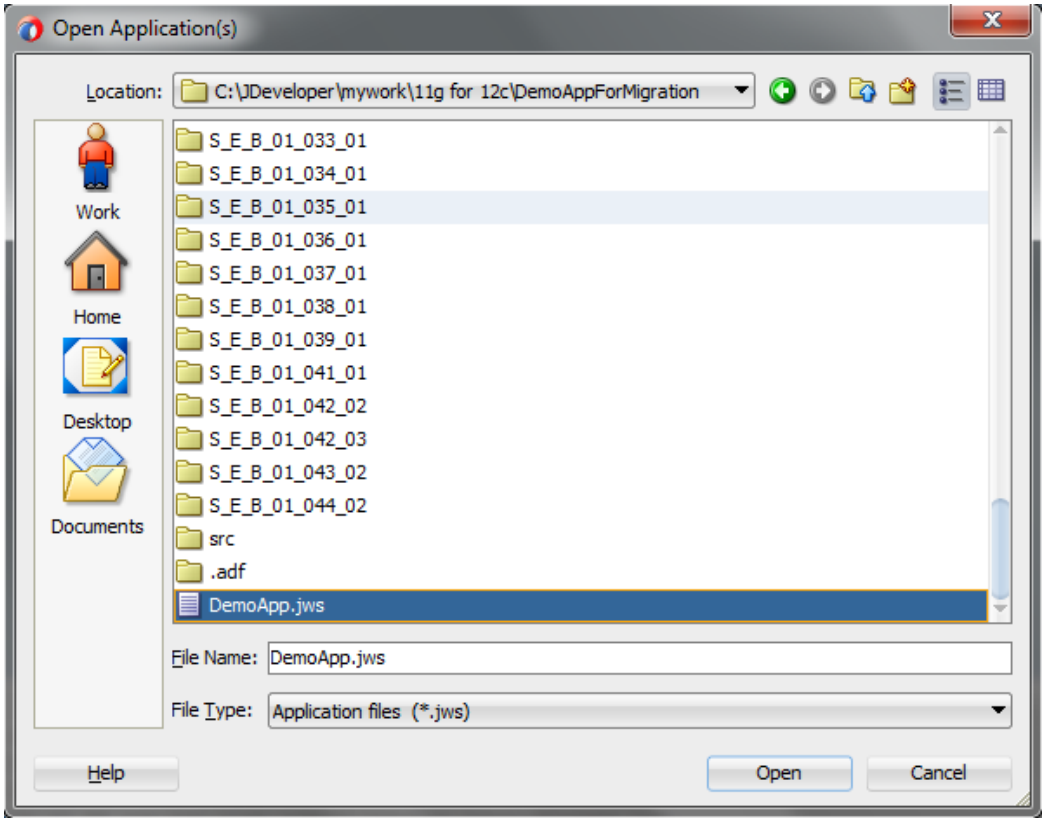
1. Create a copy of your existing 11g applications in a new directory inside your “mywork” folder, for example – C:\JDeveloper\mywork\11g-12c\
2. Open JDeveloper 12c.
3. Click on Open Application

Figure 10- 1 Open Existing Application



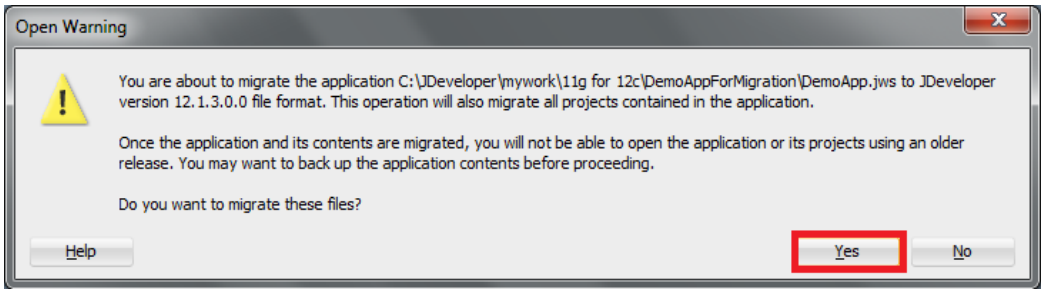
4. Select the .jws file of the 11g application you had copied into the location mentioned in step 1. If you only wish to migrate a single project, you can select the .jpr file of that project.

Figure 10- 2 Choose .jws of 11g application



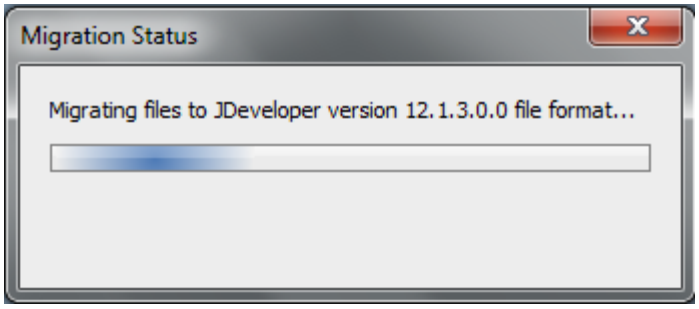
5. You will see a prompt warning stating that you are about to migrate your 11g files to the 12c format. Click “Yes” to continue.

Figure 10- 3 Confirm Migration to 12c



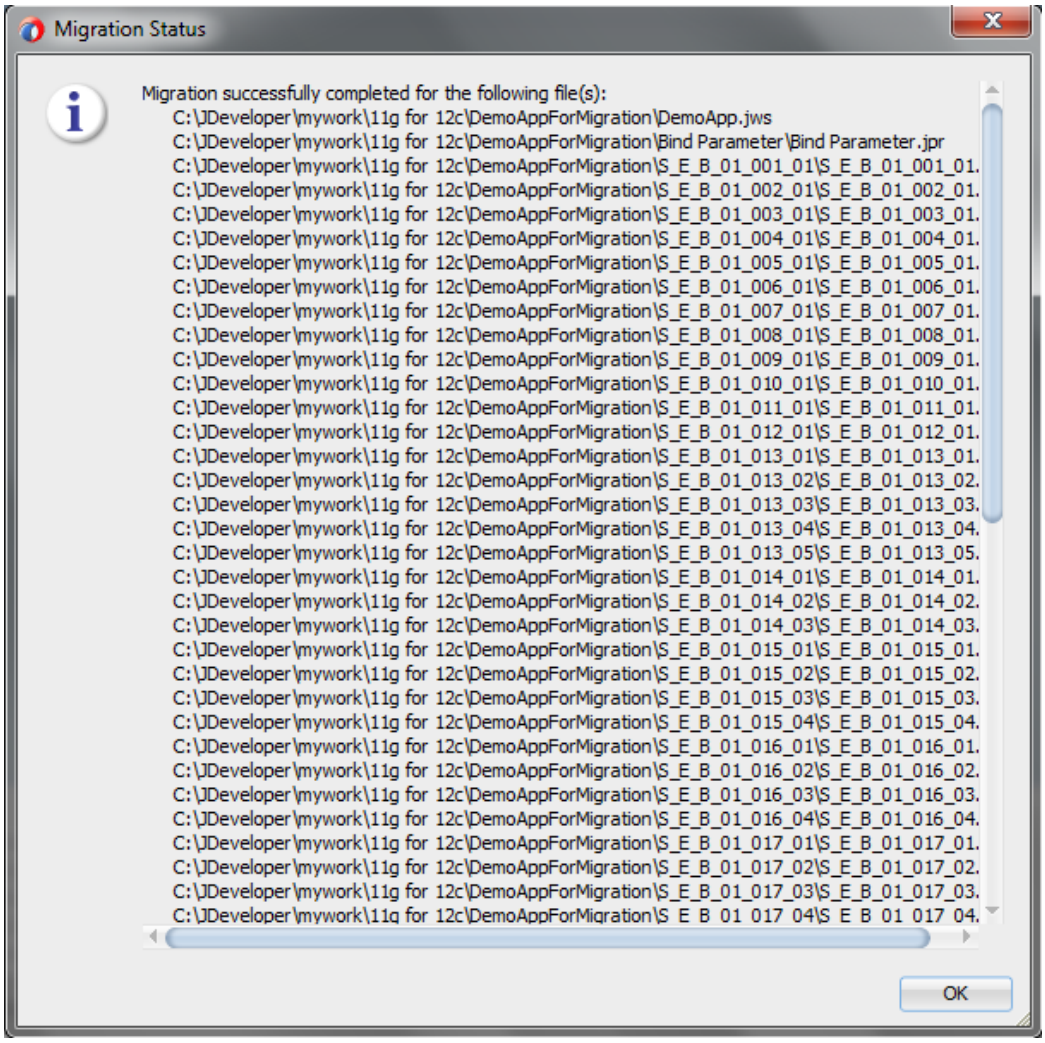
6. Wait for a few minutes while JDeveloper migrates your application to 12c format. This process can take a few minutes depending upon the size of your application. During this time you will see the following dialog.

Figure 10- 4 Wait while migration completes



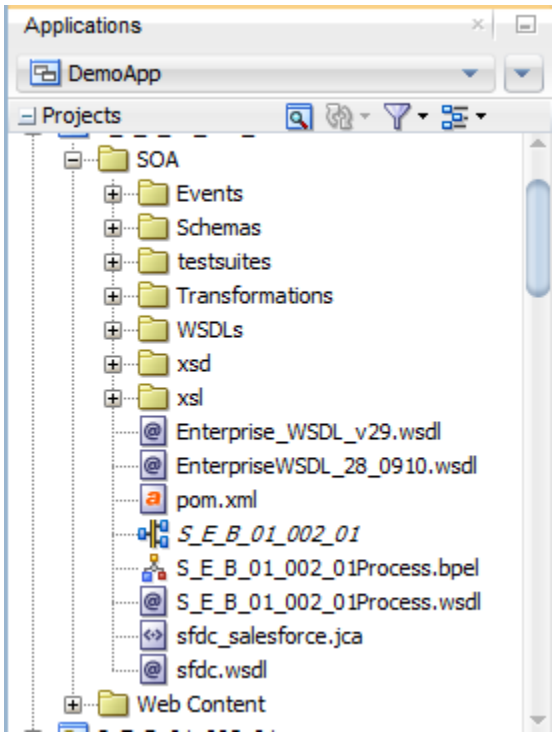
7. After a few minutes, you will see the following dialog stating that successful migration for all your project files in the application.

Figure 10- 5 Migration successfully completed



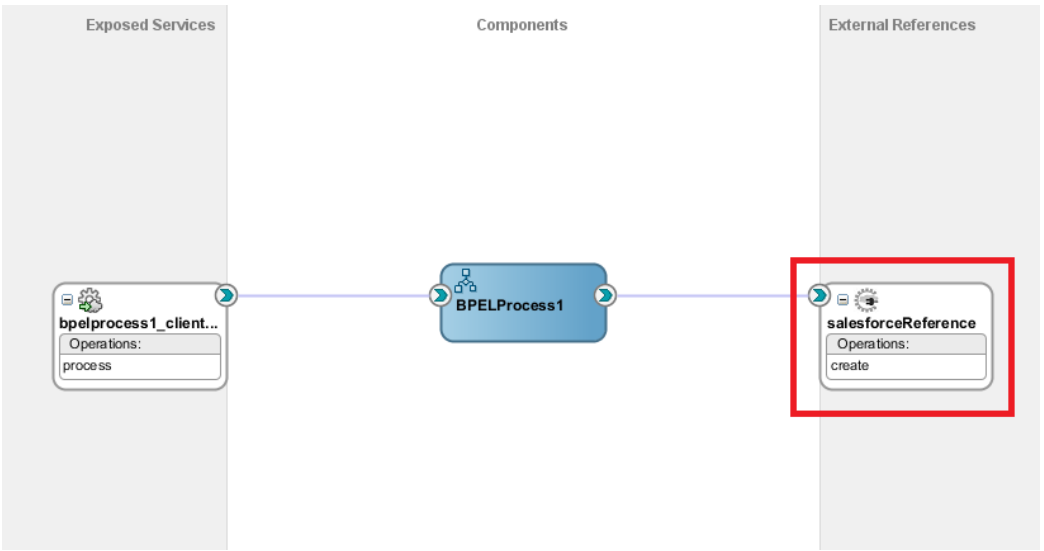
8. Click **OK** and you will see all your projects in the Projects Tab in Application. If you expand one of your projects, you will see the inner directory structure to be different than it was in 11g (this is expected behavior and will not affect the functioning of your projects).

Figure 10- 6 Migrated Project Structure



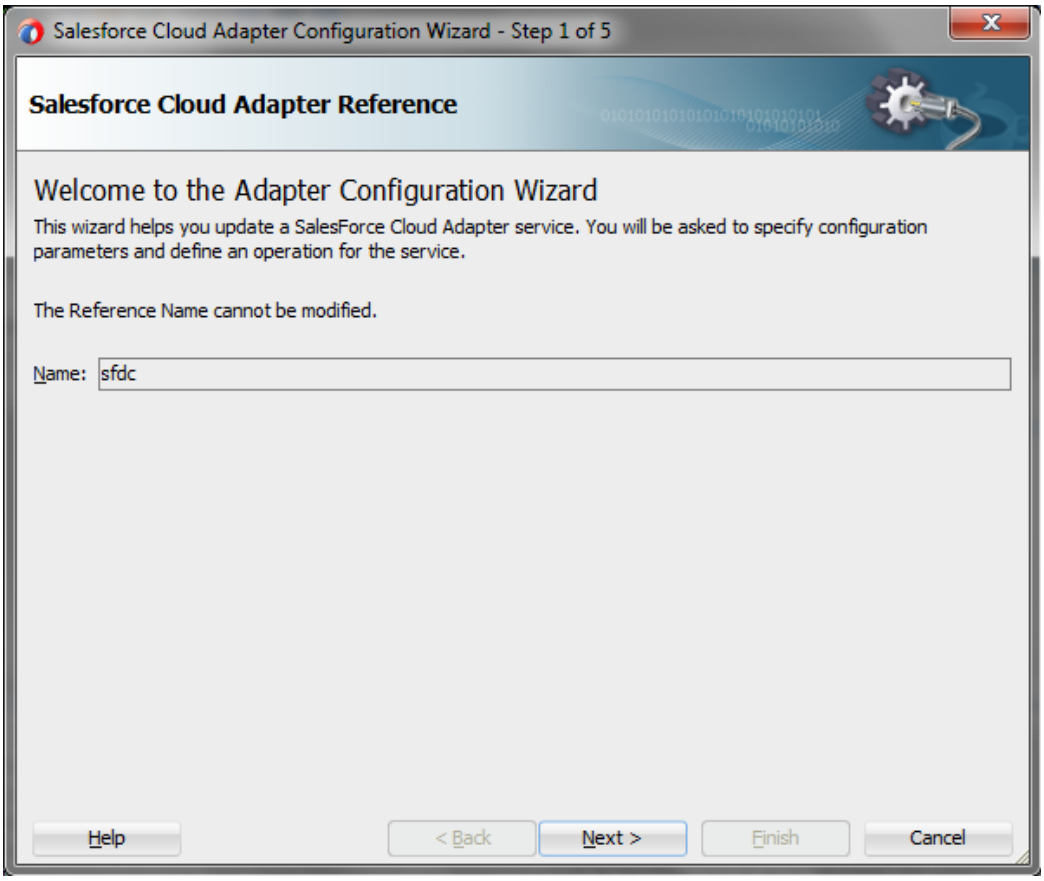
9. This completes the migration of your 11g application/projects to 12c. You can now proceed to update your Oracle Cloud Adapter for Salesforce.com to the latest version.
10. Open the composite.xml of the project you wish to update the adapter in, and edit the Oracle Cloud Adapter for Salesforce.com in External References swim-lane. (This project shown contains only one instance of the adapter, your projects may contain more and each instance will require the Steps 10.)

Figure 10- 7 Edit Oracle Cloud Adapter for Salesforce.com



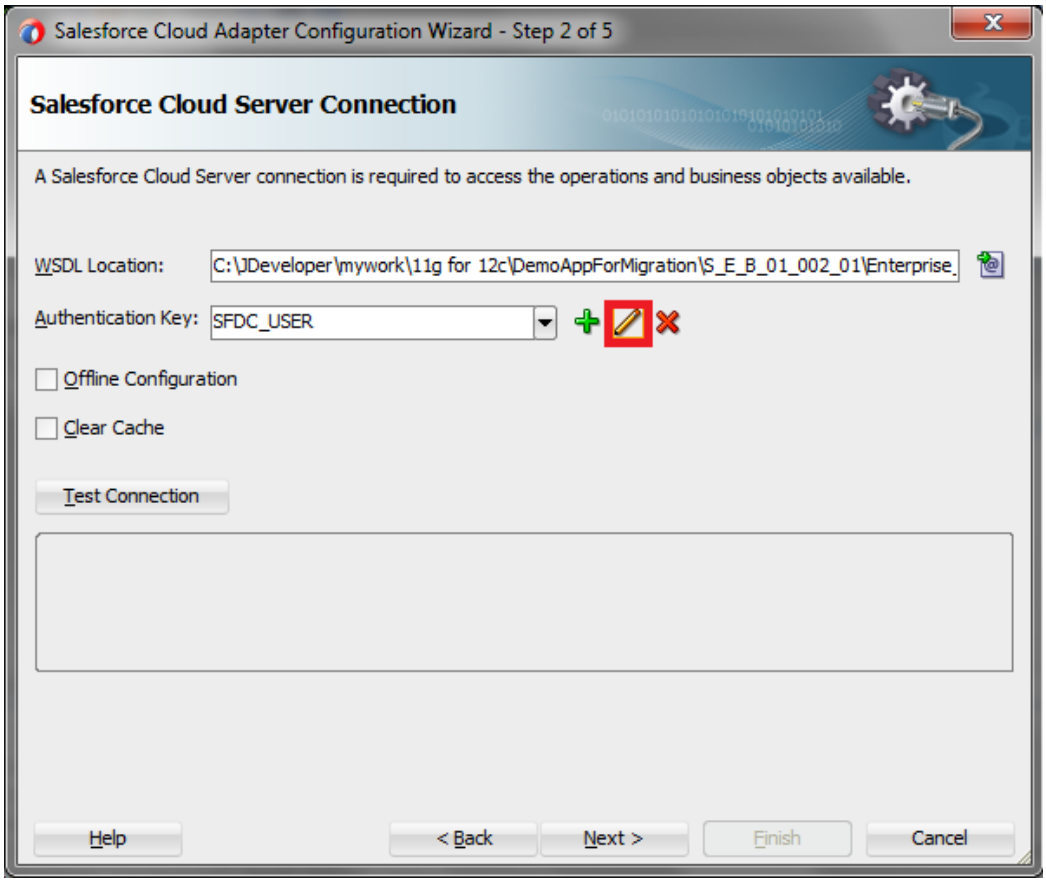
11. On editing the adapter, you will see the Adapter Configuration Wizard – Page 1, you cannot make any changes here to maintain the sanctity of your mappings in the BPEL process. Click "**Next**".

Figure 10- 8 Oracle Cloud Adapter for Salesforce.com Welcome Screen



12. On the page that follows, you will see the Authentication Key field already populated with the key you had created during 11g project creation for the adapter. You will need to enter the username and password for this key again by clicking on the Edit button. (Please note that this step has to be followed only the first time you edit the adapter in a newly migrated application. Once you have configured the same key again in this application, it will show up in all future adapter edits within this application and you can skip the steps 13-14)

Figure 10- 9 Oracle Cloud Adapter for Salesforce.com Connection Page - Edit CSF Key



13. Configure the Authentication key as shown in section "4.2.2.2 Authentic Key".
14. Click "**Next**" on all the subsequent screens without making any changes and Finish the configuration of the adapter on the Finish page.
15. This completes the migration of your Oracle Cloud Adapter for Salesforce.com from 11g version to the latest 12c version.
16. You can now proceed with deploying the composites to the WebLogic Server.

10.3 Points to Remember

There are certain points you must remember while migrating your projects from 11g to 12c.

1. Take a backup of your 11g applications/projects in a separate location before you migrate.
2. Once an application has been migrated, it will no longer be usable in Oracle SOA Suite 11g. This will not affect your already deployed composites, but you will not be able to make any changes to them after migration.
3. It is better to make a copy of your 11g applications and then open it in 12c, this way if you face any issues during your migration, your original composites will still remain usable and editable.

11

Oracle Cloud Adapter for Salesforce.com Use Cases

This section provides details of a few Use Cases to illustrate the end-to-end use of Oracle Cloud Adapter for Salesforce.com. These can be used as a reference by an organization planning to integrate with Salesforce.com using the Oracle Cloud Adapter for Salesforce.com on Oracle Fusion Middleware.

This chapter aims to introduce the user to different integration scenarios to Salesforce.com using various components of Oracle SOA Suite. The chapter focuses on guiding the user on various functionalities of the Oracle Cloud adapter for Salesforce.com – queryMore operation, Offline Configuration, MDS, Salesforce headers, etc. by means of few examples. These examples would be beneficial in resolving any complexities associated with the integration to Salesforce.

This chapter is divided into two parts:

- Part I- BPEL Use Cases
- Part II- BPM Use Cases

Part I

BPEL Use Cases

This section provides details of a few BPEL Use Cases to illustrate the end-to-end use of Oracle Cloud Adapter for Salesforce.com. These can be used as a reference by an organization planning to integrate with Salesforce.com using the Oracle Cloud Adapter for Salesforce.com on Oracle Fusion Middleware.

This chapter contains the following topics:

- Section 11.1, "How can I build integration in which the SOA client is not interested in the response?"
- Section 11.2, "How to use query and queryMore Operations?"
- Section 11.3, "I do not have a net connection right now or Salesforce.com is down, how I create my Integration using Cloud Adapter for Salesforce.com?"
- Section 11.4, "How to use MDS for importing WSDL into JDeveloper?"
- Section 11.5, "How to use Debug Header (Response header)?"

11.1. How can I build integration in which the SOA client is not interested in the response?

Overview: This use case is an extension to the integration explained in Chapter 5 " Integration with Different Service Components (BPEL/Mediator) in Oracle SOA Suite". It demonstrates a scenario in which the response returned from Salesforce.com holds no significance to the user and to get rid of the unwanted load on the server (due to large response returned from Salesforce.com) the user can opt for the 'Suppress Response' functionality of Oracle Cloud Adapter for Salesforce.com. To sum up, this use case demonstrates an account creation on Salesforce.com and also explains how to suppress the response returned from Salesforce.com.

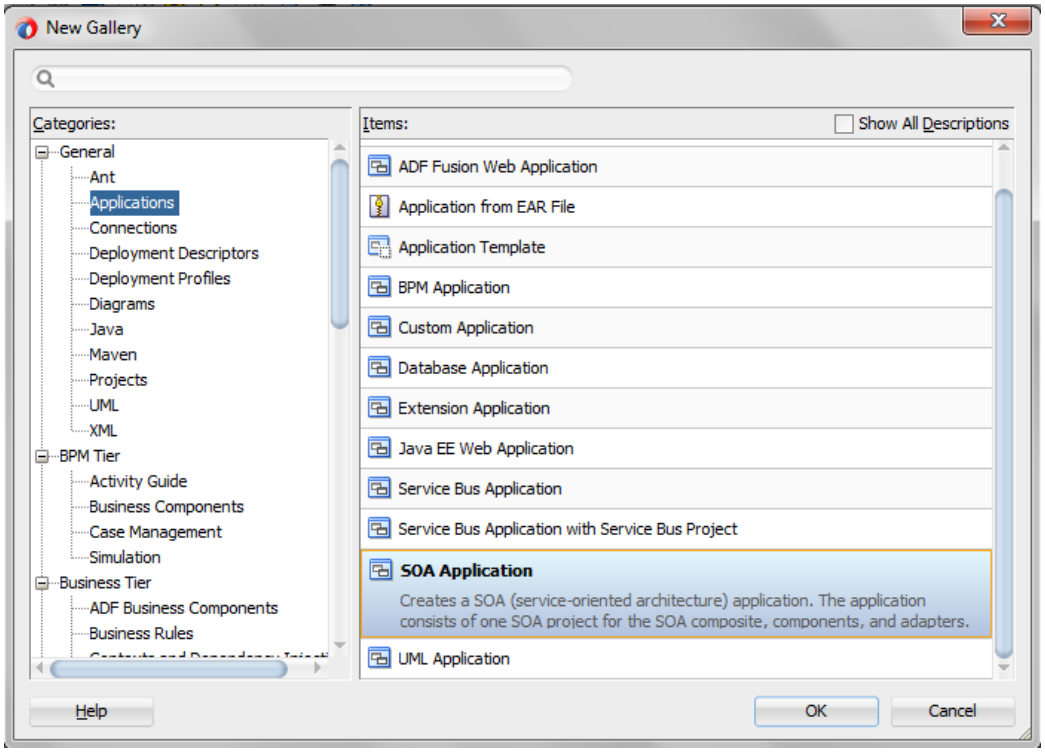
11.1.1. Creating the BPEL Process

Creating a New Application

Perform the following steps to create a New Application:

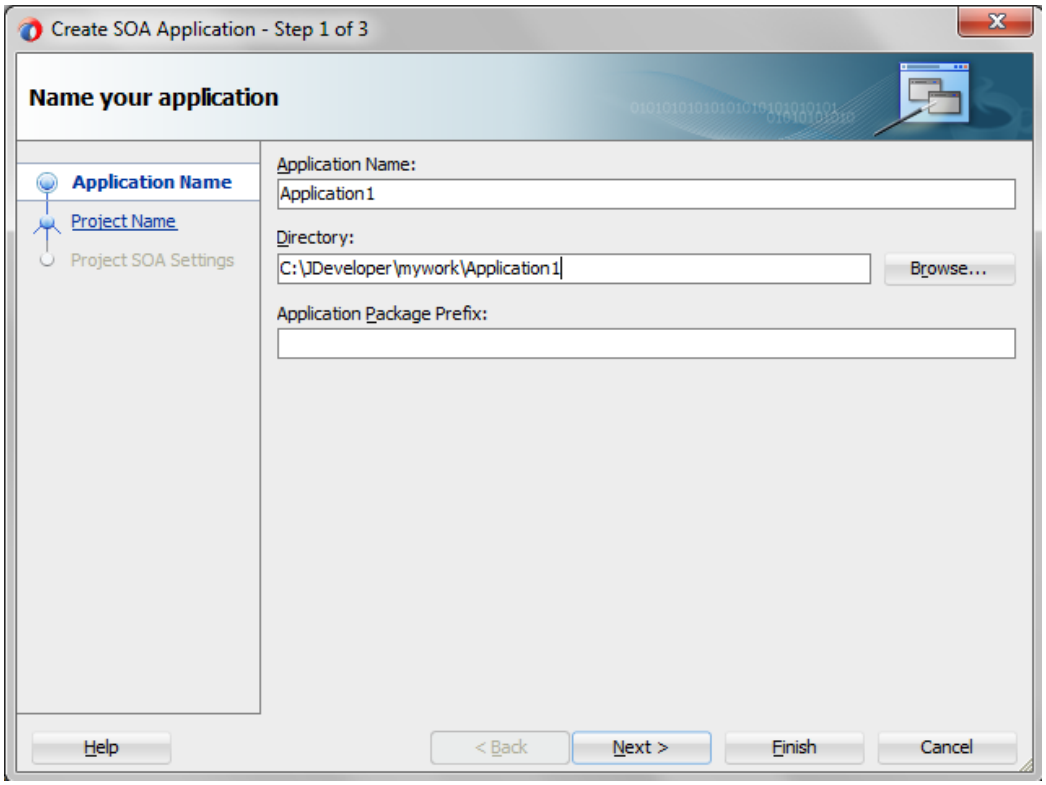
1. In the **File** menu of JDeveloper, click **New** and select **Application**.
2. The **New Gallery** page is displayed, select **SOA Application** from the **Items** list and click **OK**, as shown in Figure, as shown in Figure 11- 1.

Figure 11 - 1 Create SOA Application



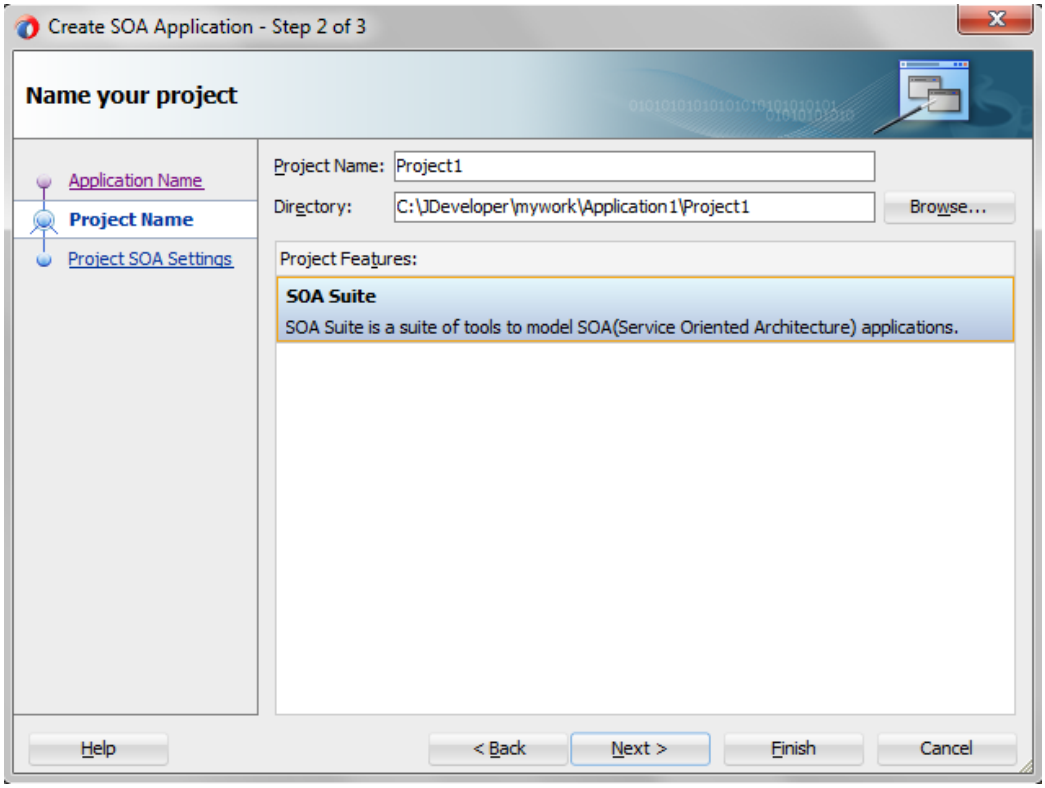
3. Provide a suitable name to your application, as shown in Figure 11- 2.

Figure 11 - 2 Name your application



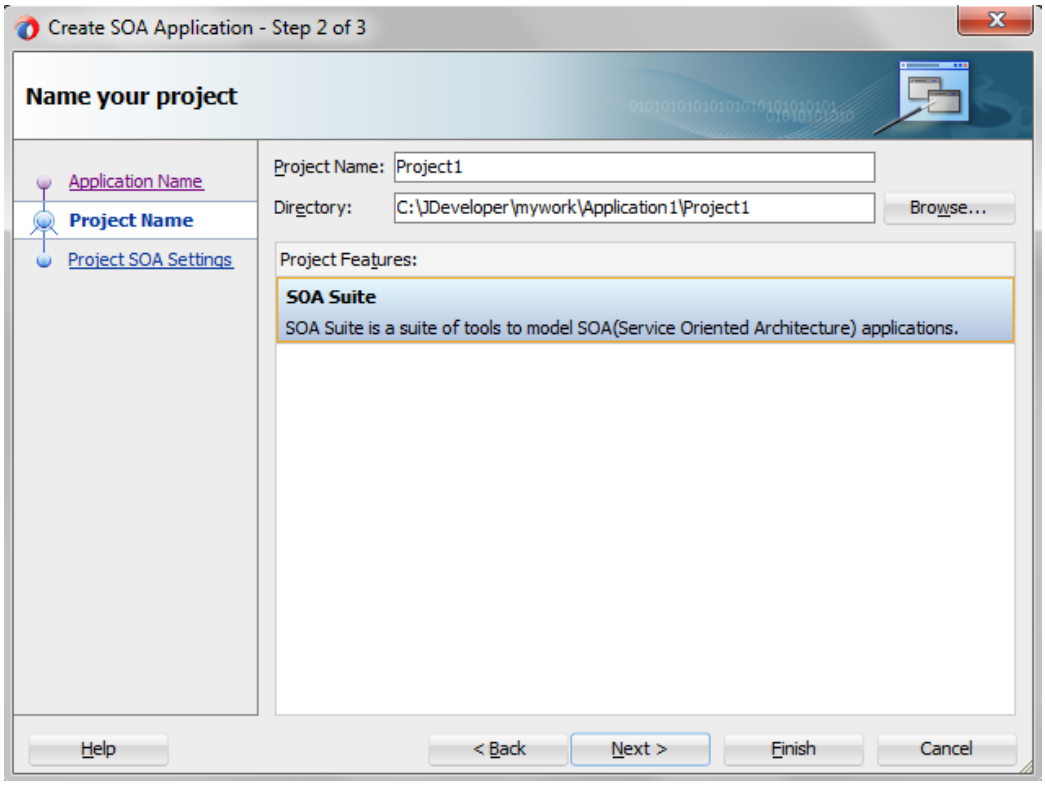
4. Click **Next** and provide a suitable name to your project, as shown in Figure 11- 3.

Figure 11 - 3 Name your project



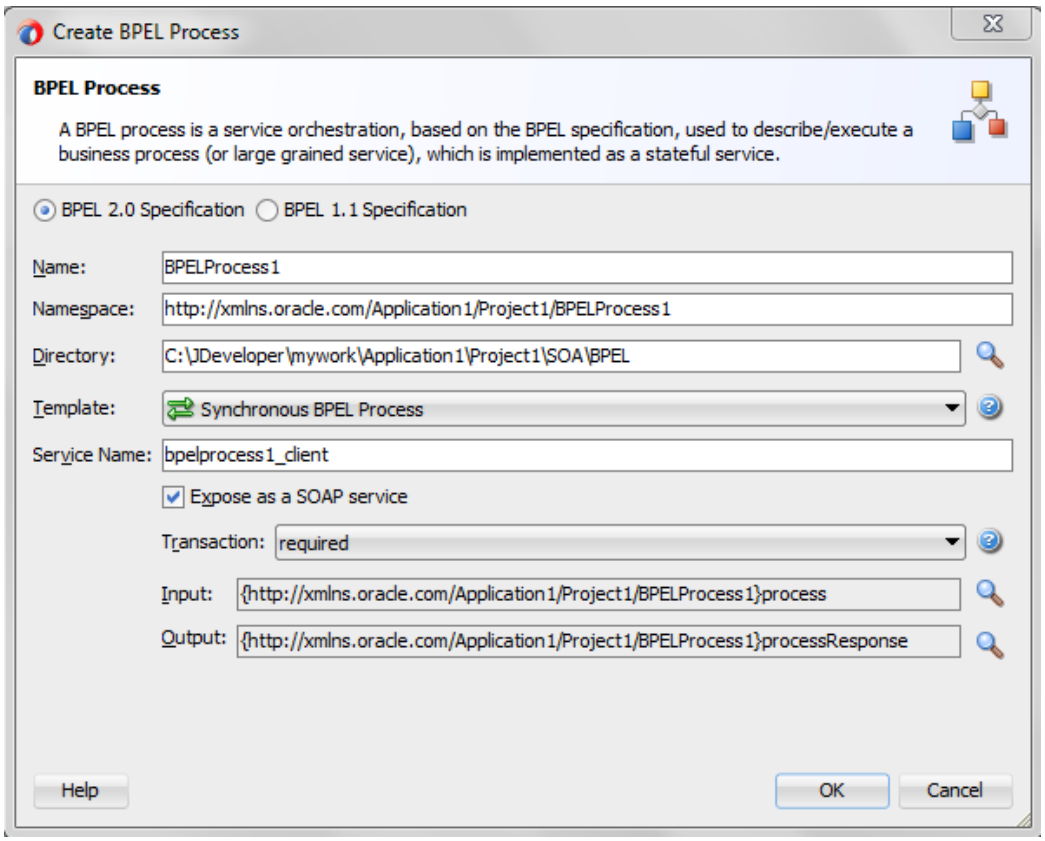
5. Click **Next** and select **Composite with BPEL Process** from **Standard Composite** list, as shown in Figure 11- 4.

Figure 10 - 4 Configure SOA Setting



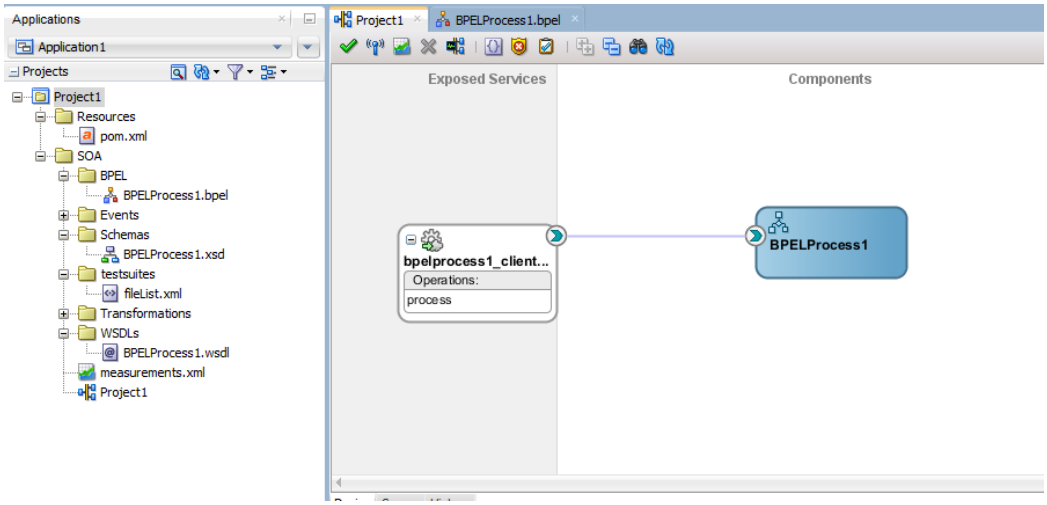
6. Click **Finish**.

7. Select the **Synchronous BPEL Process** from **Template** drop-down and click **OK**.



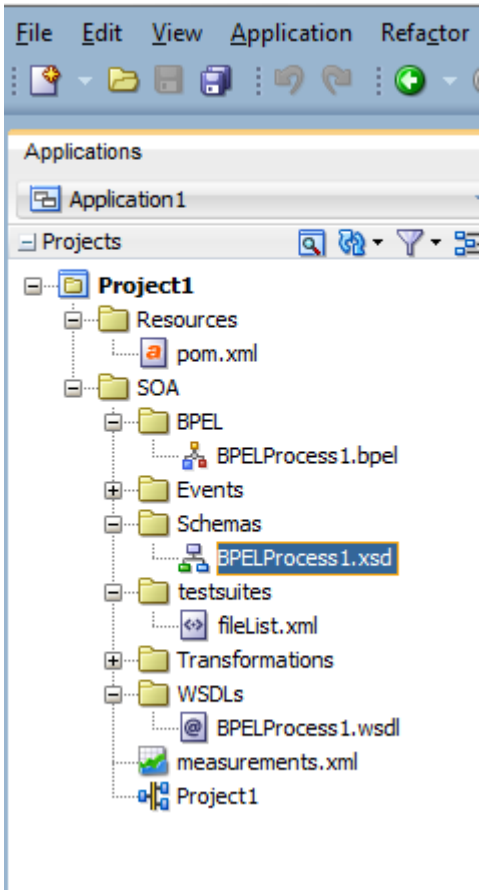
8. The composite.xml looks as shown in Figure 11-5.

Figure 11 - 5 Composite.xml



9. In the **Application Navigator**, under the **xsd** folder, an XML schema file with the name **BPELProcess1.xsd** is displayed after BPEL process creation, as shown in Figure 11- 6.

Figure 11 - 6 XML Schema File



10. Edit this schema file as per your business requirement. It is the responsibility of the front-end application to enforce data validations and to ensure that the input sent to SFDC via SOA is error free. The structure of the schema used in this use case is shown in Figure 11- 7.

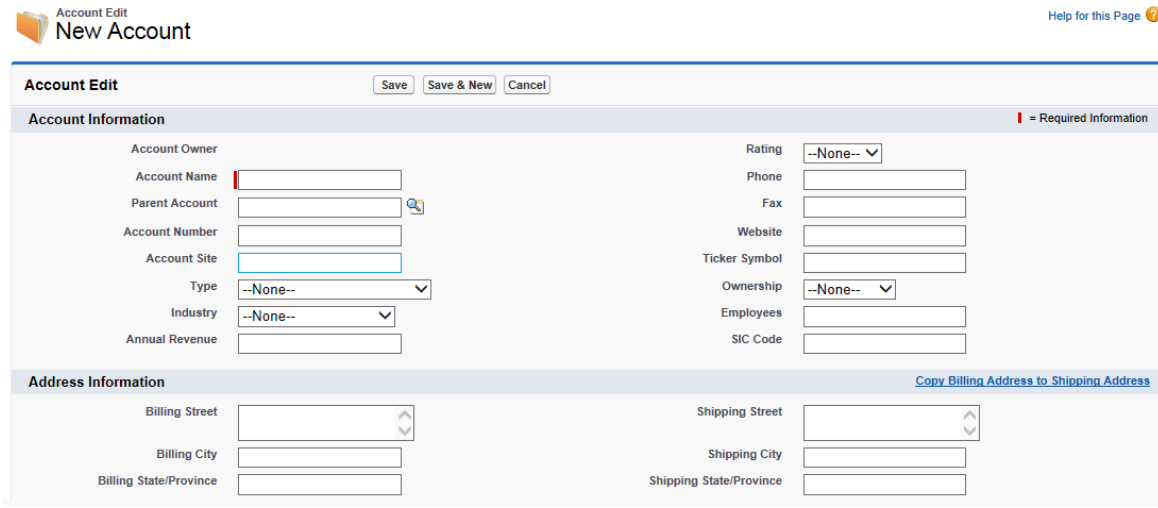
Figure 11 - 7 Edit XML Schema File

```
<?xml version="1.0" encoding="UTF-8"?>
<schema attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="http://xmlns.oracle.com/Final/Project1/BPELProcess1"
        xmlns="http://www.w3.org/2001/XMLSchema">
        <element name="process">
                <complexType>
                        <sequence>
    <element name="ProcessMultiple" maxOccurs="unbounded">
     <complexType>
      <sequence>
       <element name="AccountName" type="string"/>
       <element name="Acc_Ext_Id_c__c" type="string"/>
       <element name="AnnualRevenue" type="string" minOccurs="0"/>
       <element name="BillingCountry" type="string" minOccurs="0"/>
      </sequence>
     </complexType>
    </element>
   </sequence>
                </complexType>
        </element>
        <element name="processResponse">
                <complexType>
                        <sequence>
                                <element name="result" type="string"/>
                        </sequence>
                </complexType>
        </element>
</schema>
```

Source | History

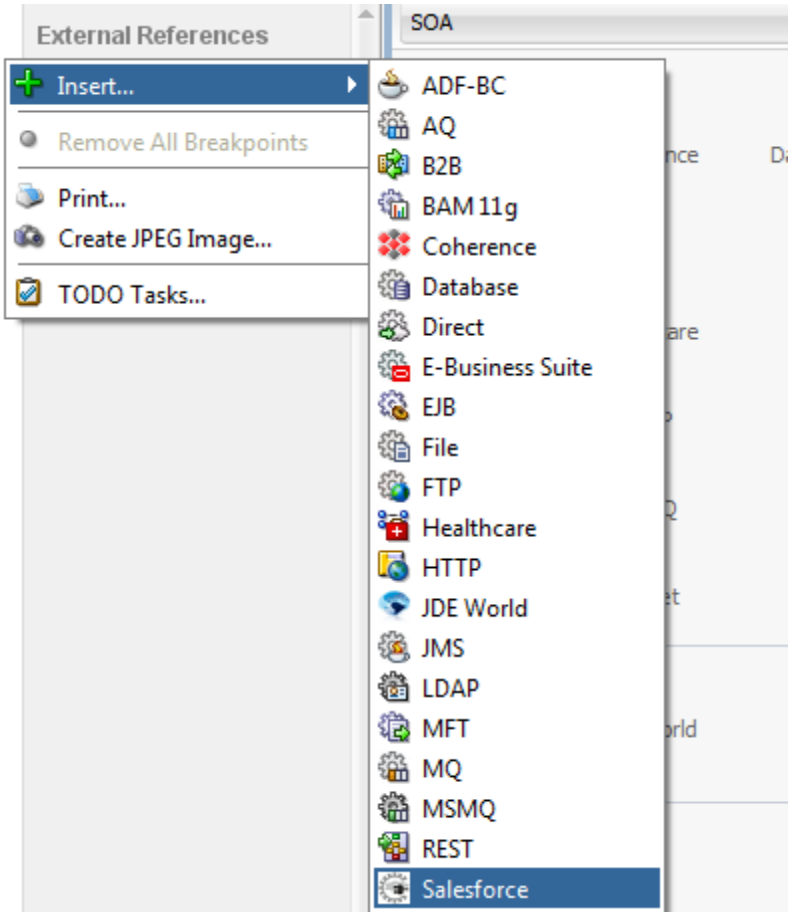
11. Figure 11- 8 shows create account page on Salesforce.com. The fields with a red mark are mandatory fields. This structure may vary for different organizations.

Figure 11 - 8 Create Account Page on Salesforce



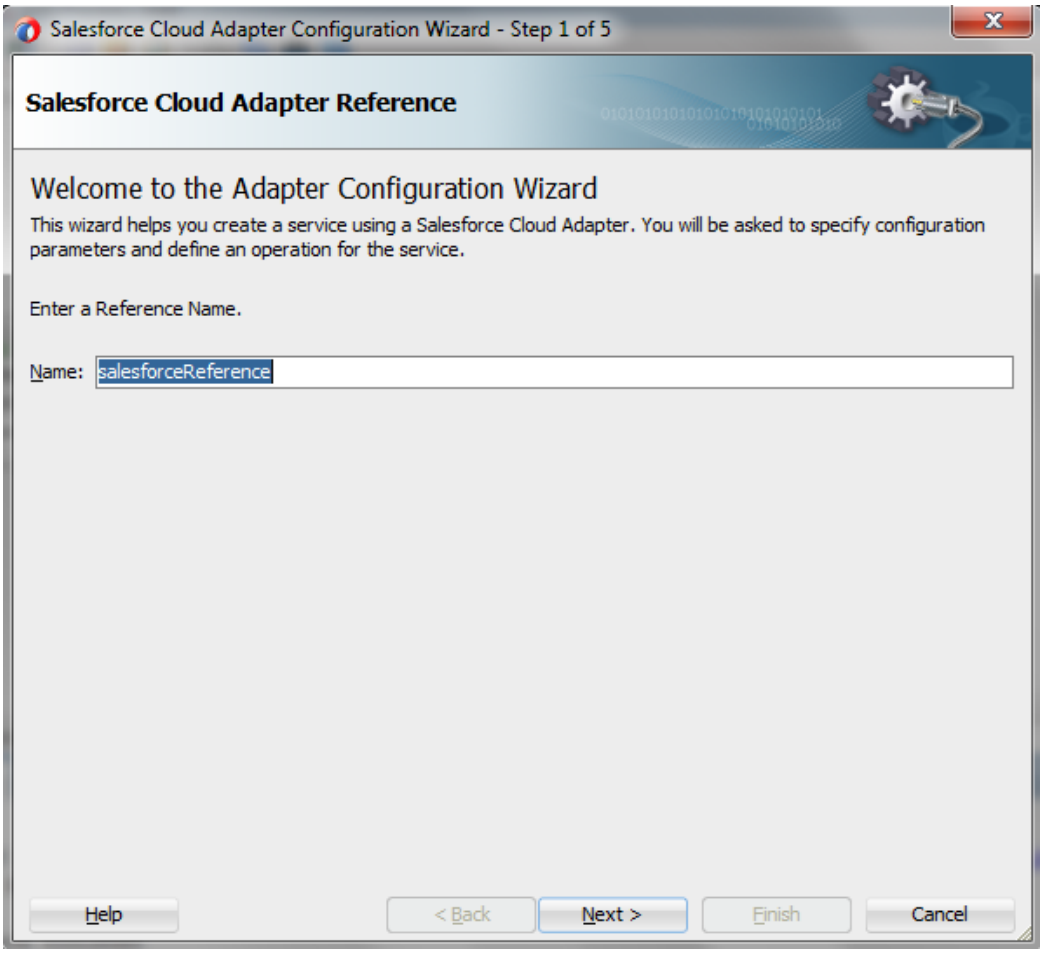
12. In **External References** swim lane of the composite.xml file, right-click and select **Salesforce** adapter, as shown in Figure 11- 9.

Figure 11 - 9 Oracle Cloud Adapter for Salesforce Adapter



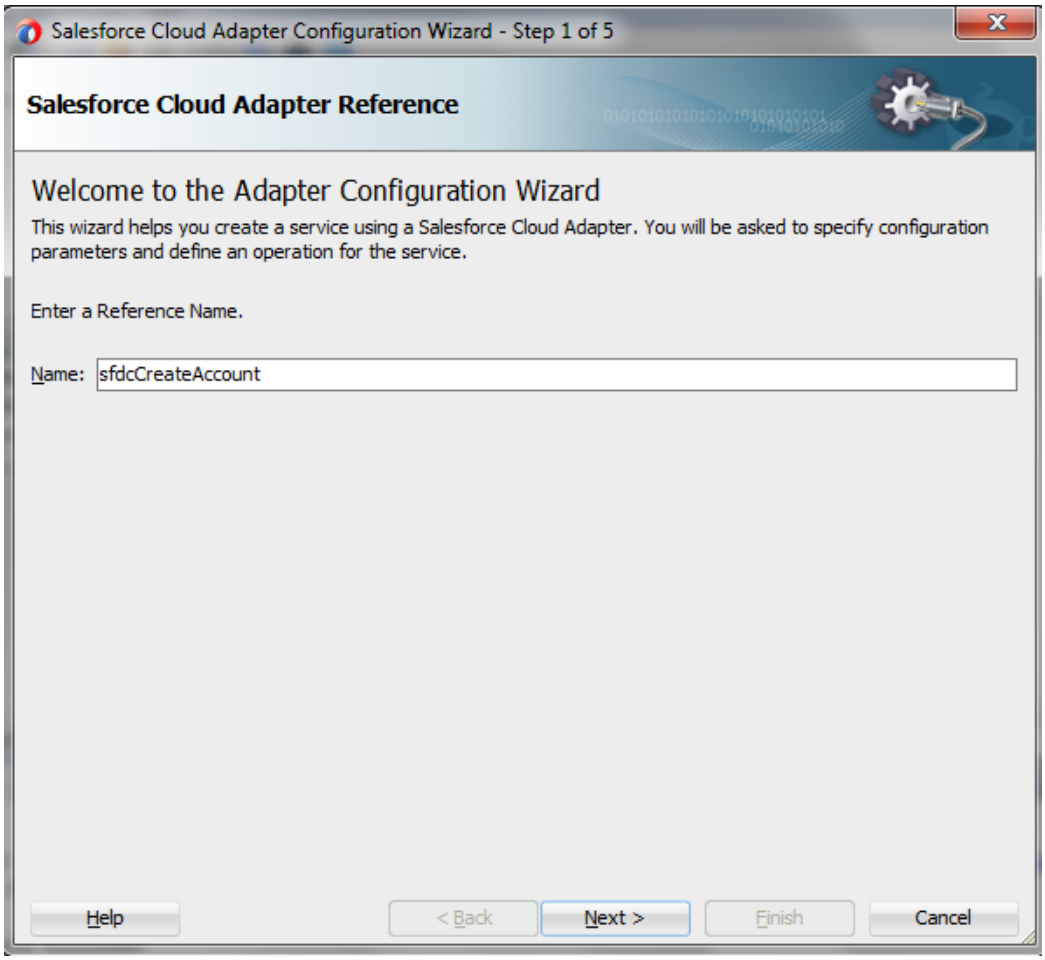
13. The Salesforce Cloud Adapter Configuration Wizard -Welcome page is displayed, as shown in Figure 11- 10.

Figure 11 - 10 Salesforce Cloud Adapter Configuration Wizard -Welcome Page



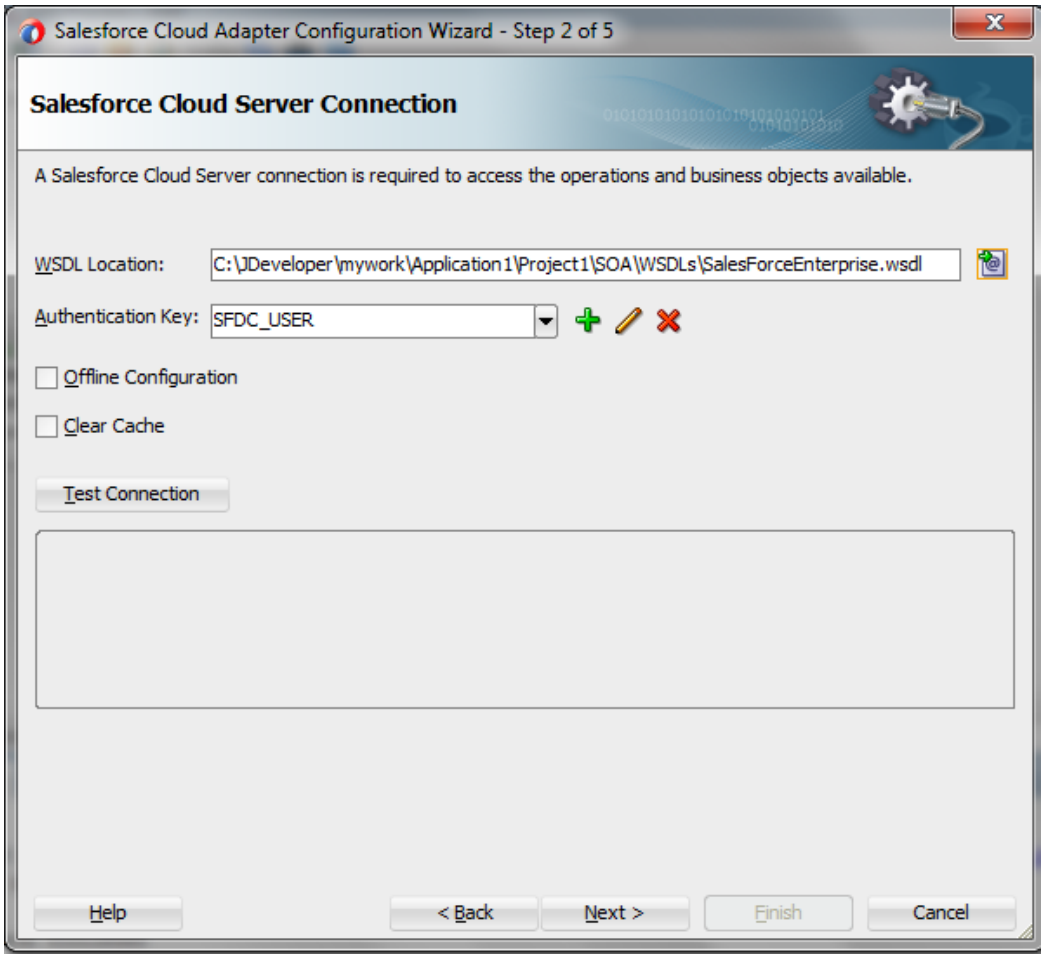
14. The **Name** box is already populated. It picks up these values from the cache. You can re-enter **Reference Name** if you want to use a different Reference Name and click **Next**, as shown in Figure 11- 11.

Figure 11 - 11 Enter Reference Name



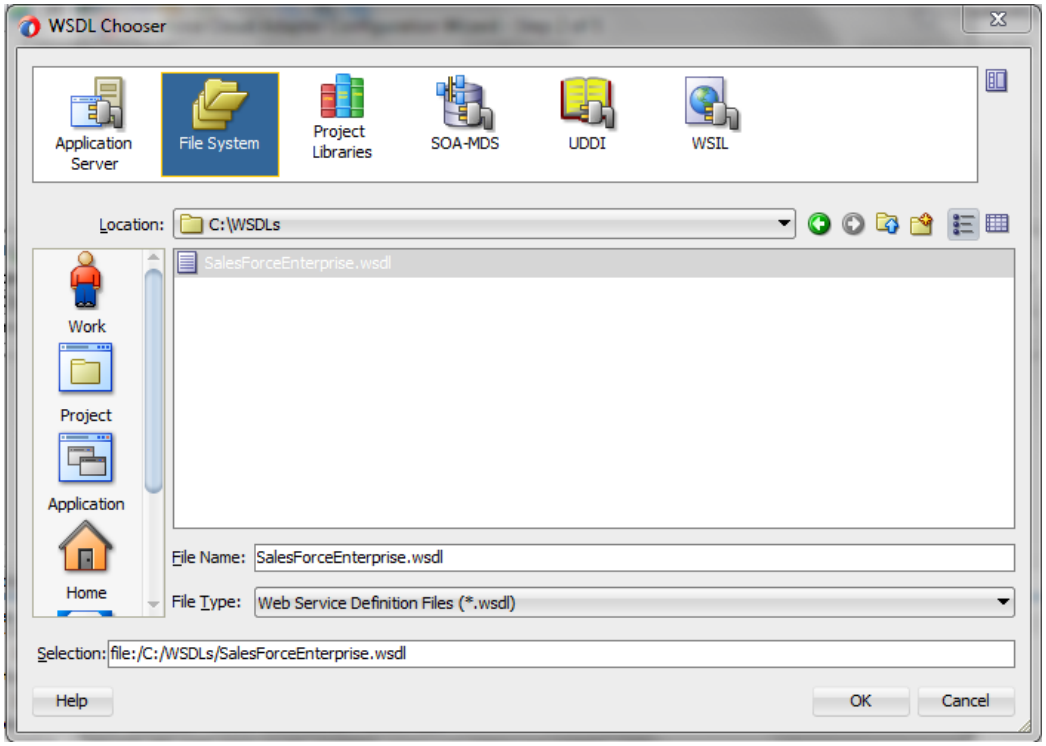
15. The **Salesforce Cloud Server Connection** page is displayed. The **WSDL Location** and **Authentication Key** text boxes are already populated. It picks up these values from the cache. You can re-enter these values. If you want to use a different value, click the **Find existing WSDLs** icon, which is located to the right of the **WSDL Location** field, as shown in Figure 11- 12.

Figure 11 - 12 Find existing WSDLs



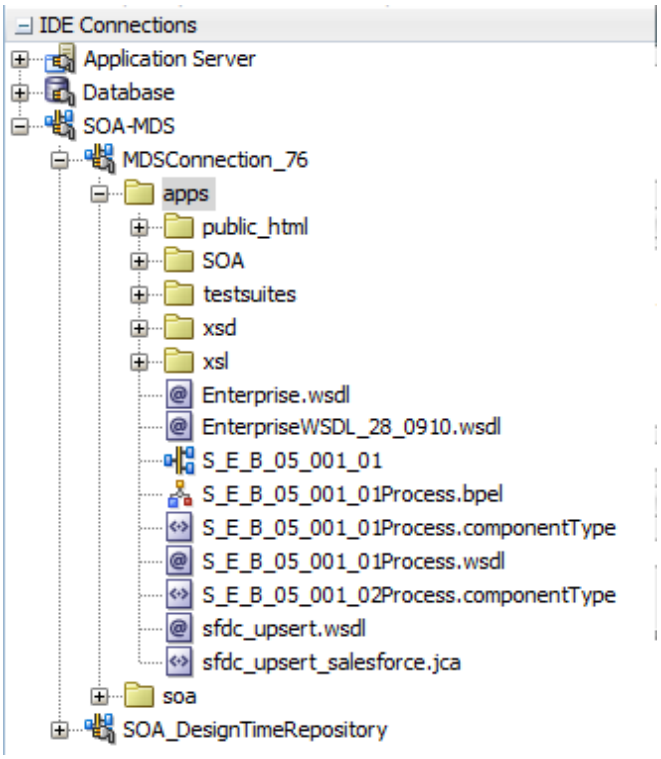
16. The **WSDL Chooser** dialog is displayed, browse and select the downloaded Enterprise WSDL and click **OK**, as shown in Figure 11- 13.

Figure 11 - 13 SOA Resource Browser



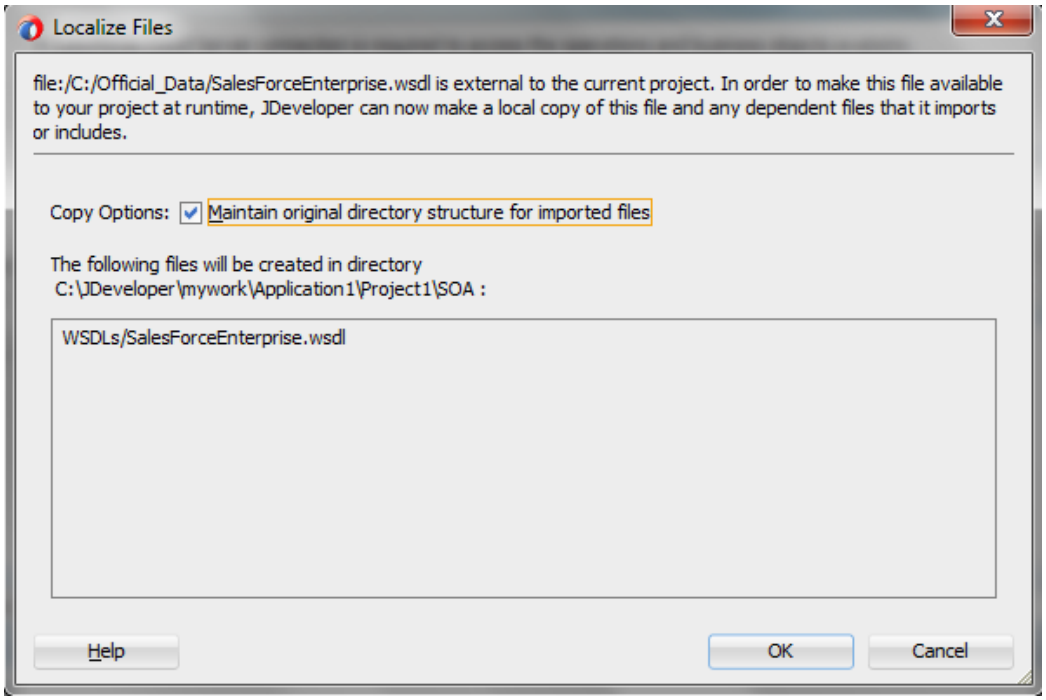
Note that as an alternative, you can store WSDL at an MDS location and access it, as shown in Figure 11- 14.

Figure 11 - 14 SOA Resource Browser



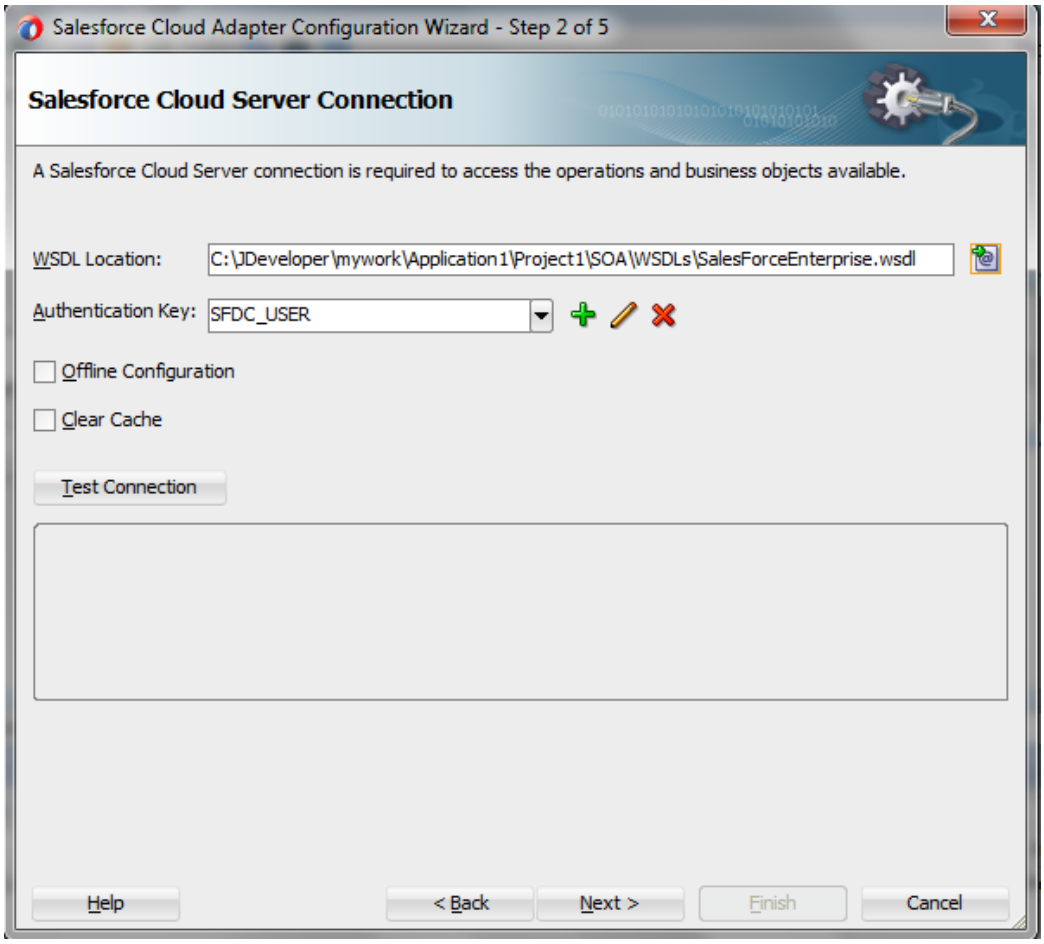
17. Copy the WSDL to your project folder, as shown in Figure 11- 15.

Figure 11 - 15 Copy WSDL to Your Project Folder



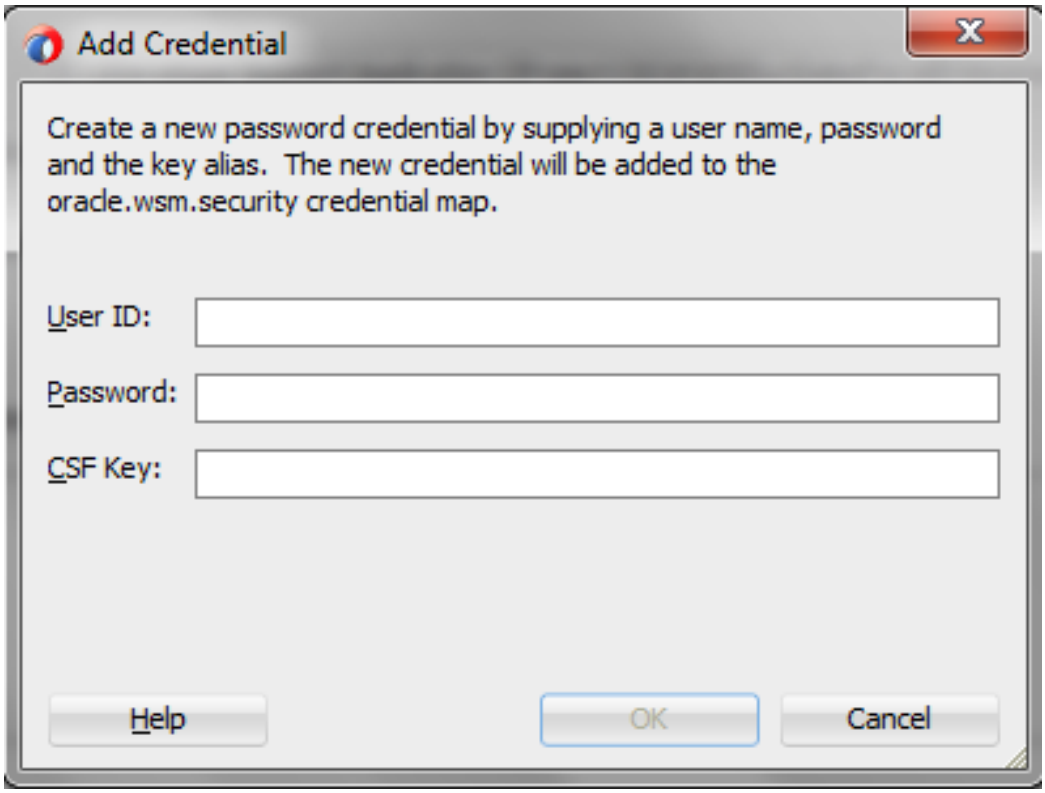
18. Click **OK**. You are returned to the Salesforce Cloud Server Connection page.
19. Click **"+"** button to create a new **Authentication Key**, as shown in Figure 11- 16.

Figure 11 - 16 Create a New Authentication Key



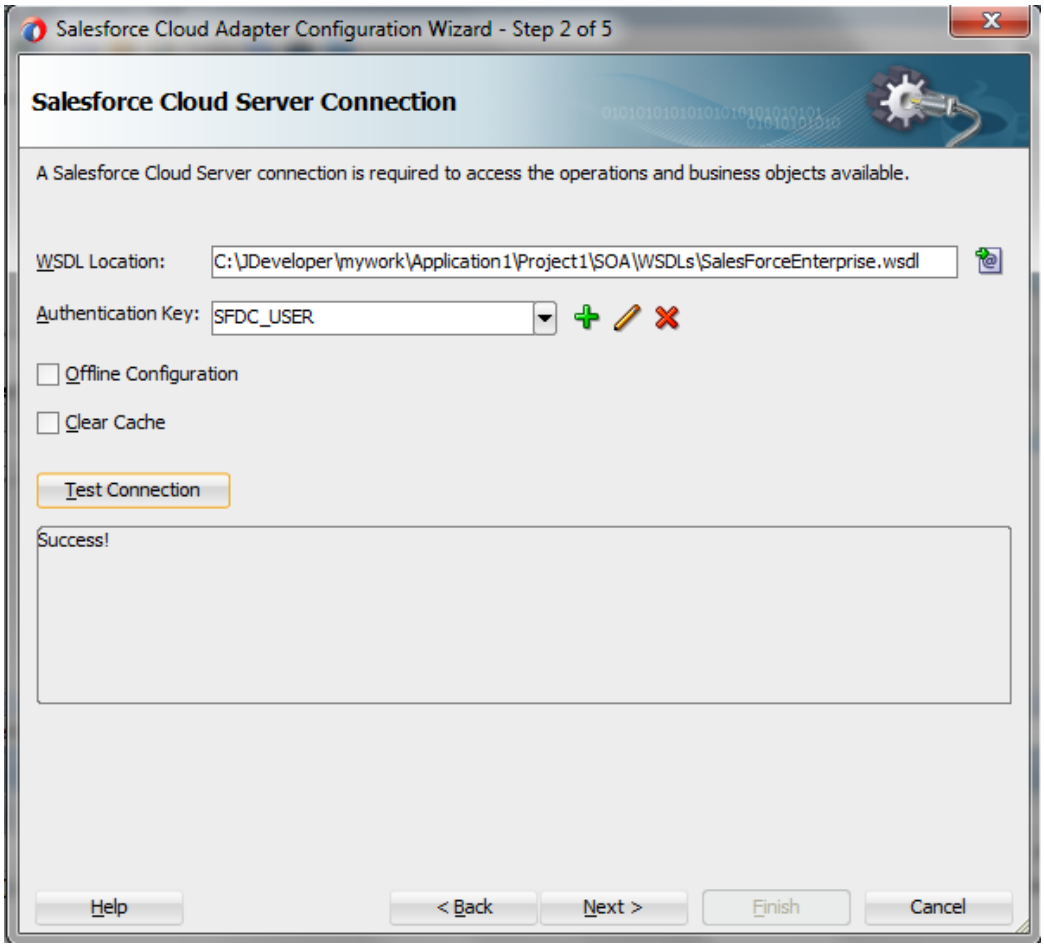
20. The **Add Credential** page is displayed, as shown in Figure 11- 17. Provide a suitable name and the Salesforce.com credentials.

Figure 11 - 17 Add Credential



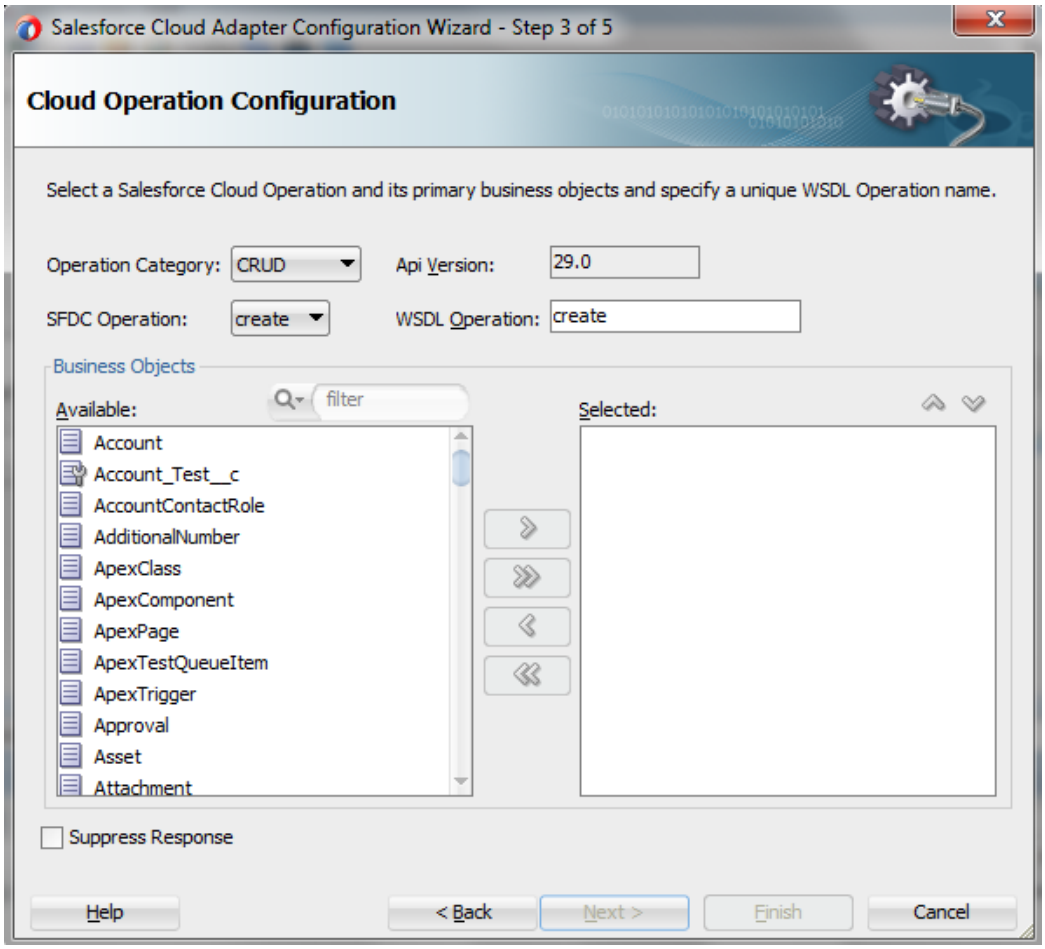
21. Click **Test Connection** button to validate the Authentication Key, as shown in Figure 11- 18.

Figure 11 - 18 Test Connection



22. Click **Next**.
23. The **Cloud Operation Configuration** page is displayed, as shown in Figure 11- 19.

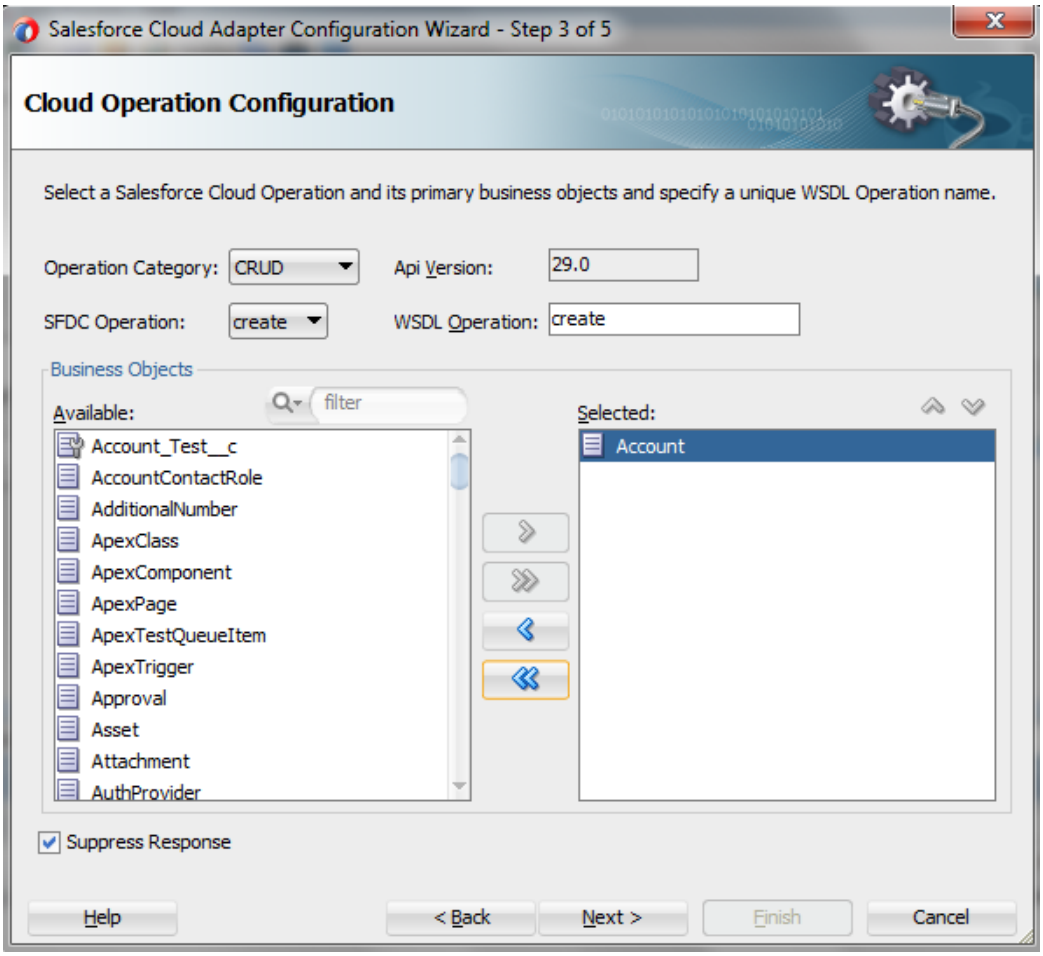
Figure 11 - 19 Cloud Operation Configuration Page



24. Since the scenario is to create an Account on Salesforce.com, select **Operation Category** as **CRUD** and **SFDC Operation** as **Create**. Now, move **Account** from the list of **Available** objects to list of the **Selected** objects. The **WSDL Operation** by default is **create** (same as SFDC Operation). You can edit the same by providing an operation name suitable to your business requirement, as shown in Figure 11- 20.

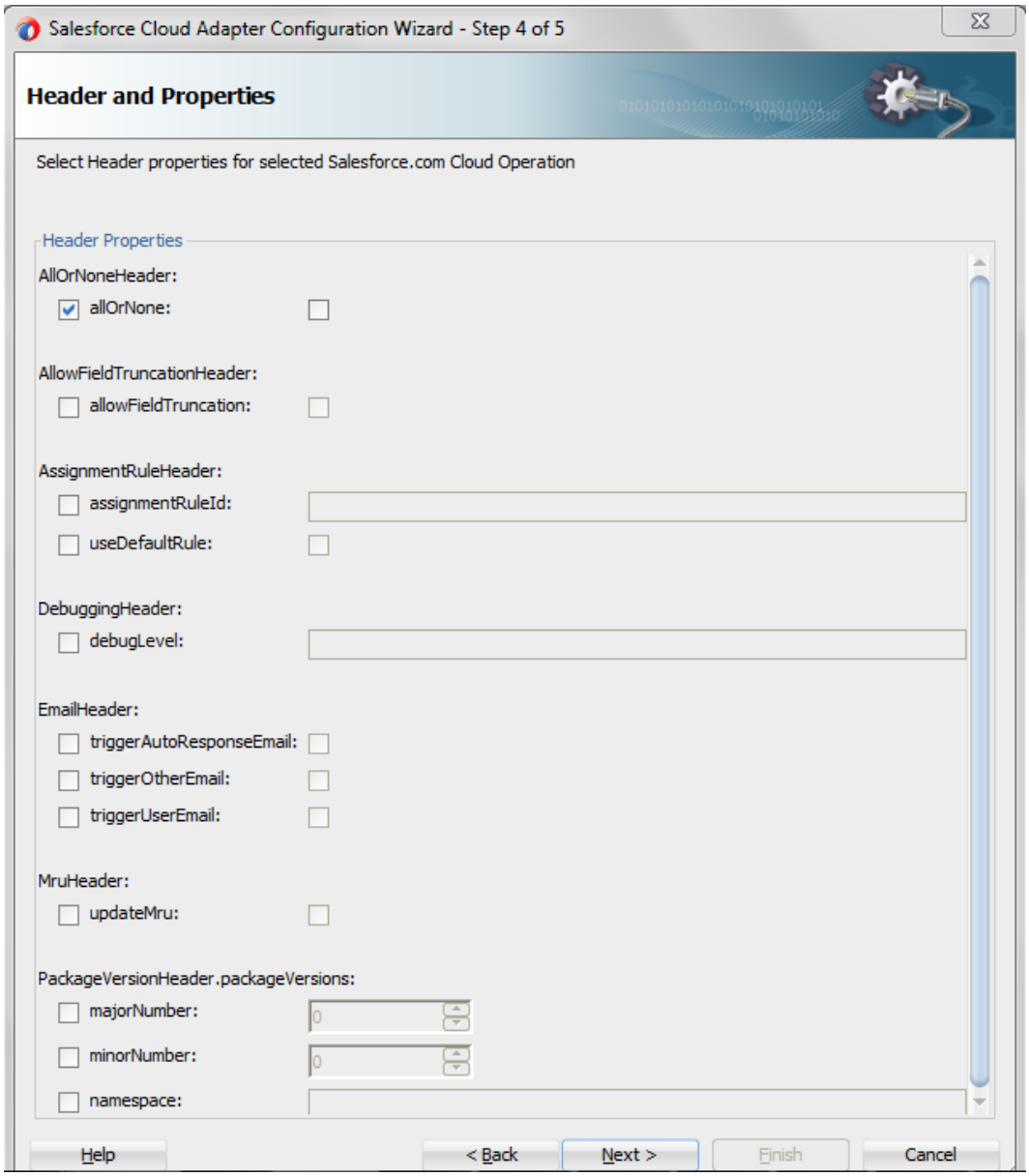
25. Check the **Suppress Response** option, because we do not have any use for the response returned by salesforce.com.

Figure 11 - 20 Cloud Operation Configuration Page



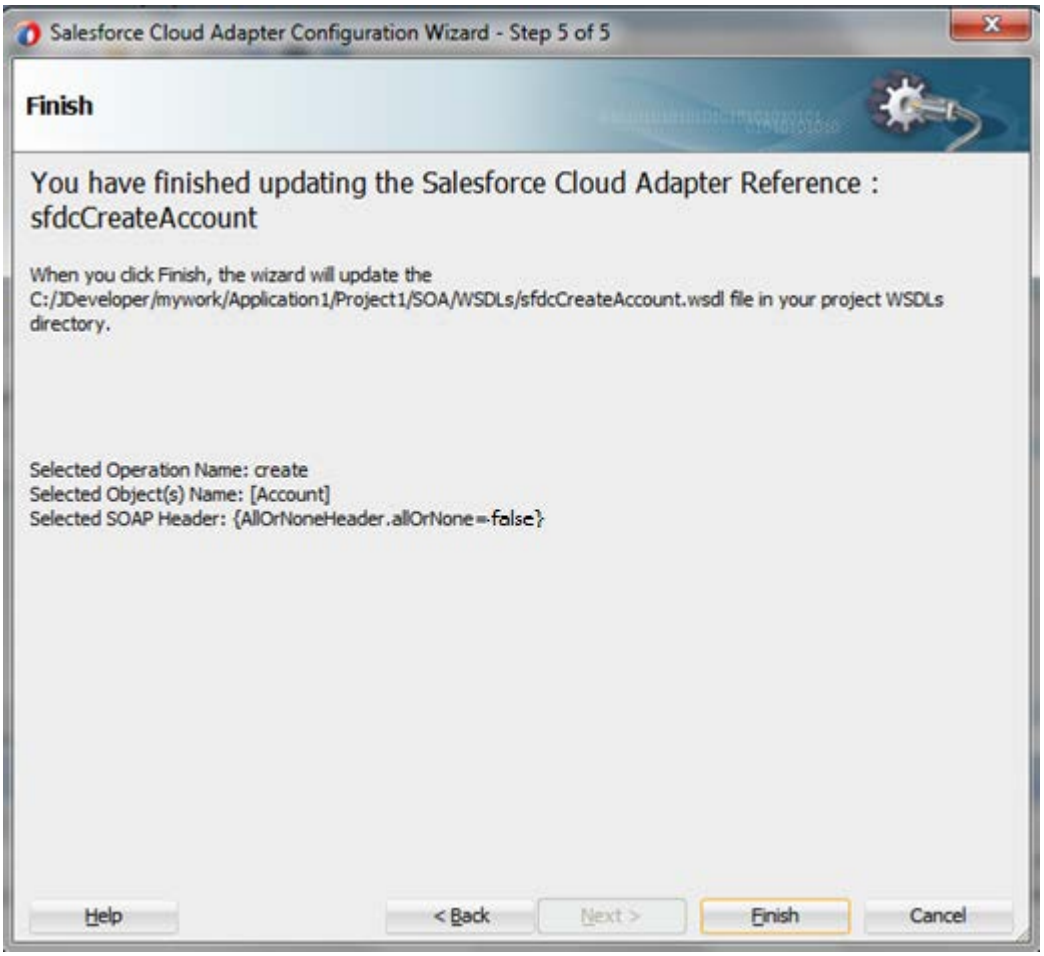
26. Click **Next**. The **Header and Properties** page is displayed, as shown in Figure 11-21.

Figure 11 - 21 Header and Properties Page



27. Select the appropriate header to your requirement. Headers displayed on this page depend on the operation selected in the previous page.
28. Click **Next**.
29. The finish page is displayed. It provides a complete summary of the operation selected, the object on which the operation would operate and the headers selected for that operation, as shown in Figure 11- 22.

Figure 11 - 22 Finish Page



30. Click the **Finish** button to complete adapter configuration.
31. After clicking on the **Finish** button. The following screen appears, as shown in Figure 11-23.

Figure 11 - 23 SFDC Cloud Account

32. Two artifacts are created after the adapter creation:
    A. sfdcCreateAccount_salesforce.jca
    B. sfdcCreateAccount.wsdl

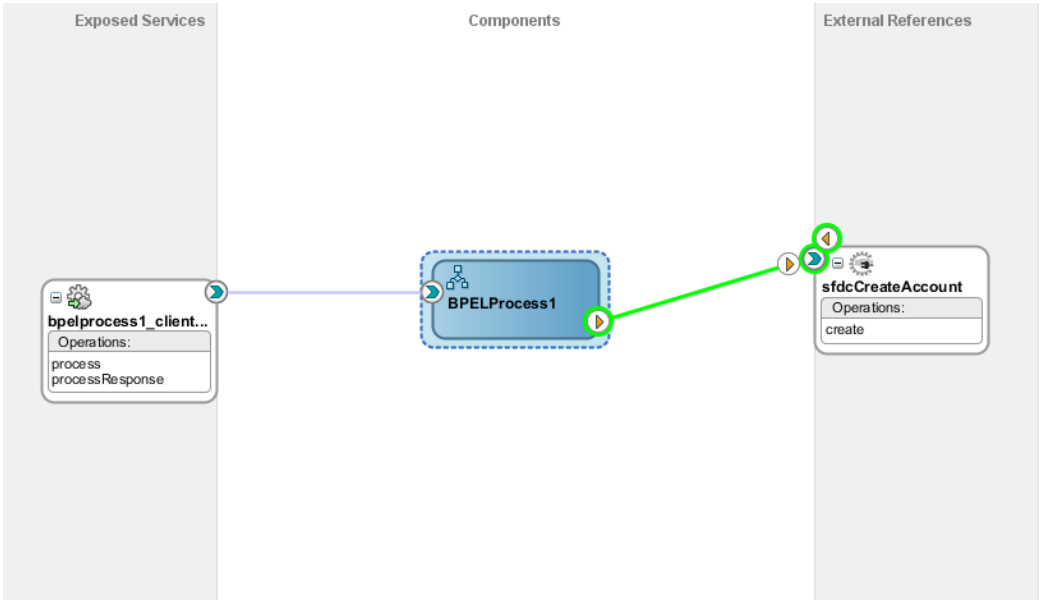
33. Double-click and open the WSDL file. The structure of ‘createResponse’ is shown in Figure 11- 24.

Figure 11 - 24 WSDL file

```
<xs:element name="createResponse" type="ns2:VoidType"/>
<xs:complexType name="VoidType"/>
```

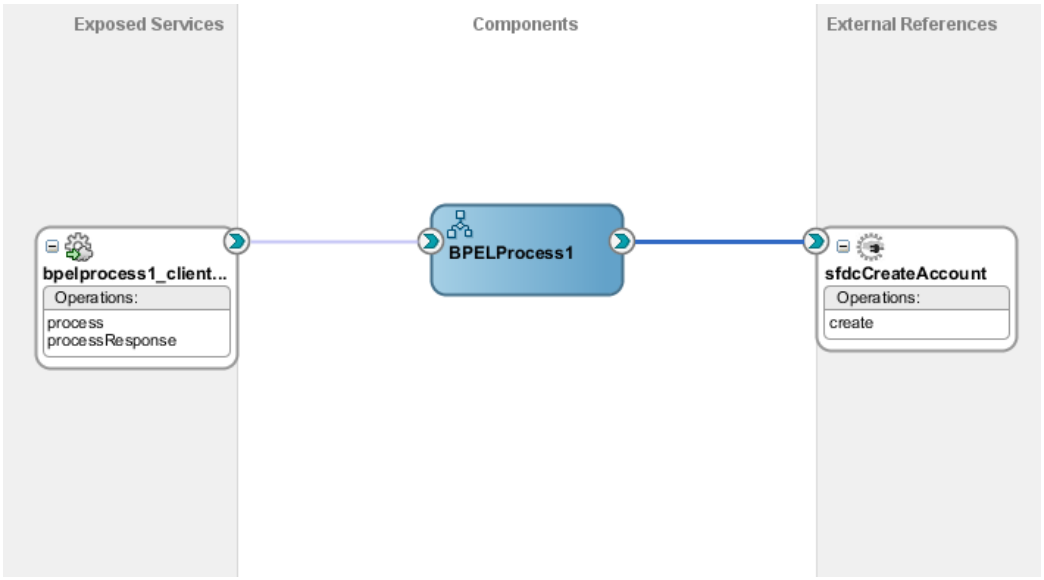
33. Connect **BPELProcess1** and **sfdcCreateAccount** via a wire, as shown in Figure 11- 25.

Figure 11 - 25 Wiring BPELProcess1 and sfdcCreateAccount



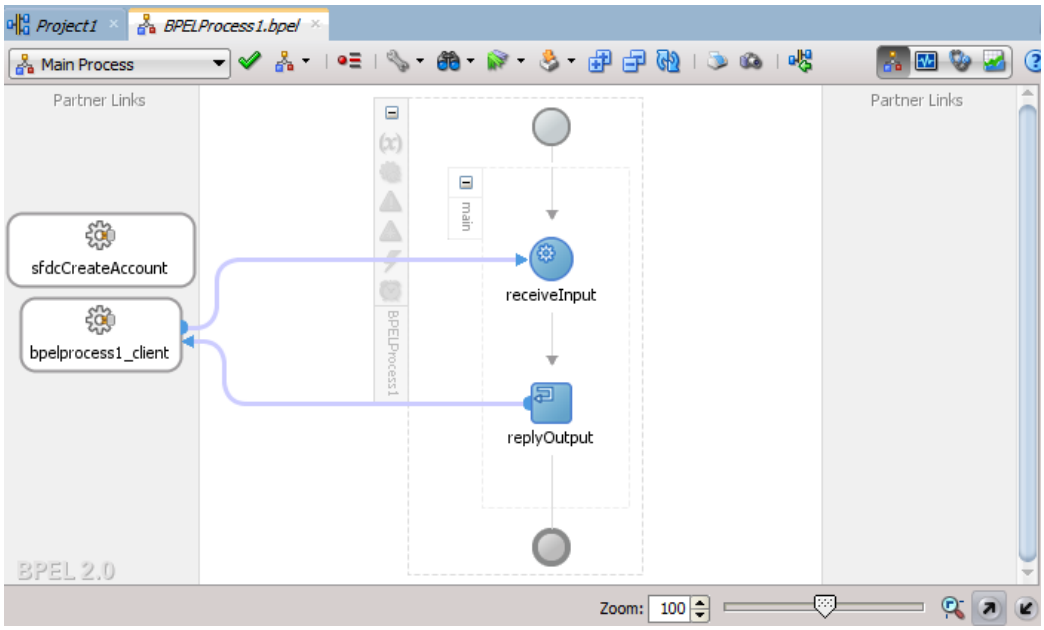
34. After wiring your composite will look like Figure 11- 26.

Figure 11 - 26 Wiring BPELProcess1 and sfdcCreateAccount



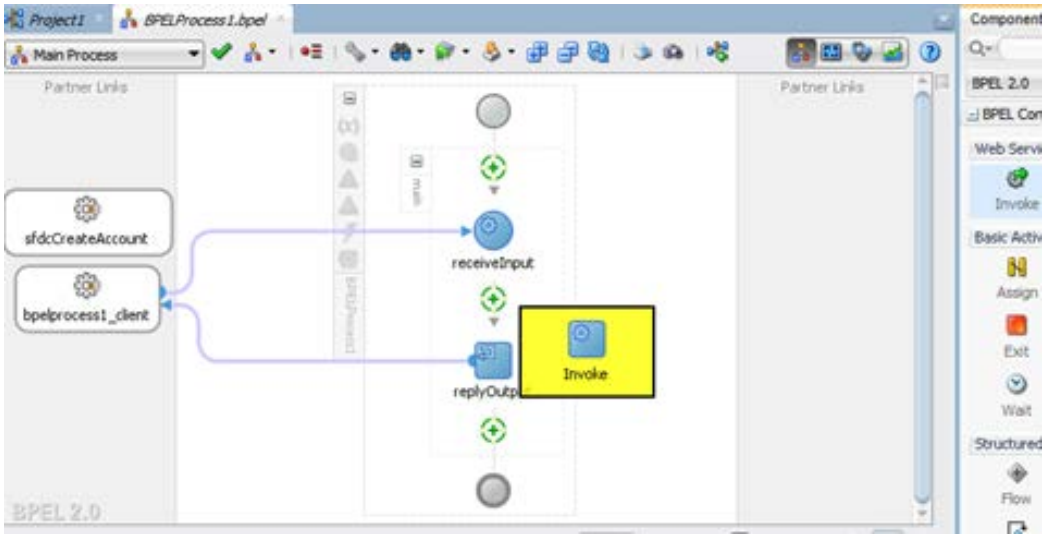
35. Double-click and open **BPELProcess1**. The **sfdcCreateAccount** adapter should be present as part of Partner Links, as shown in Figure 11- 27.

Figure 11 - 27 Open BPELProcess1

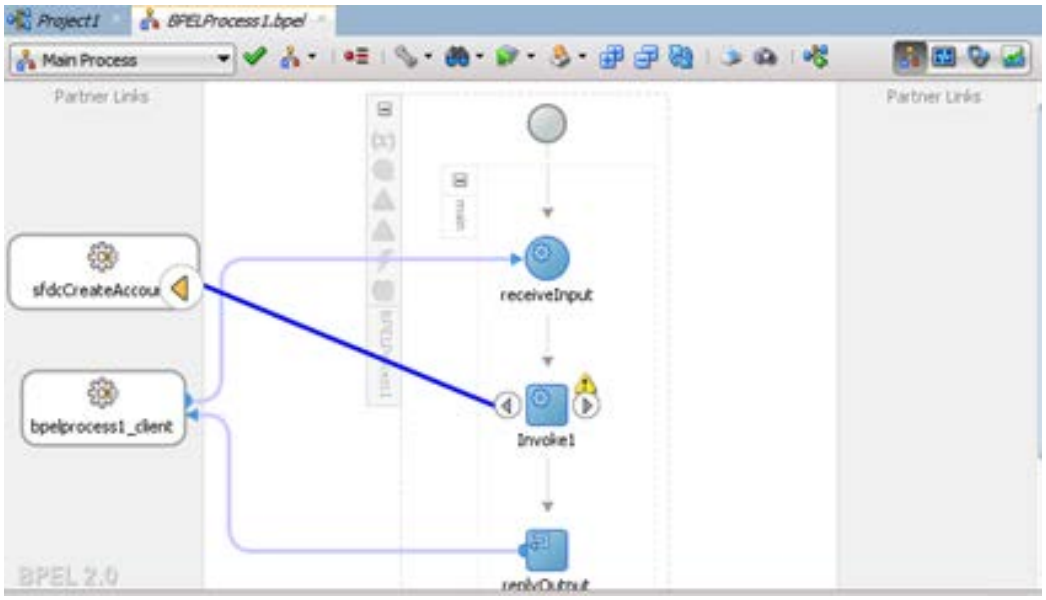


36. Invoke activity to invoke the **sfdcCreateAccount** Partner Link
37. From the Component Palette select the '**Invoke**' activity and drag and drop the same in the middle of orchestration as shown in the Figure 11-28.

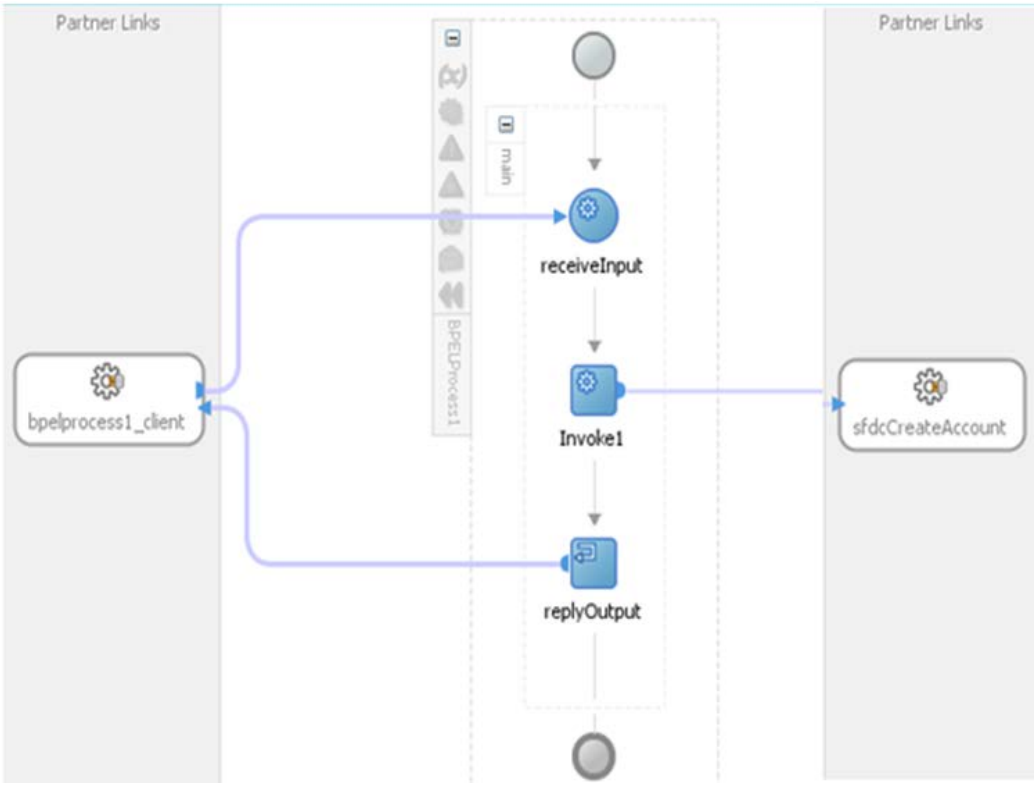
Figure 11 - 28 Add invoke activity to invoke the sfdcCreateAccount Partner Link



38. Wire the Invoke activity to partner link 'sfdcCreateAccount'

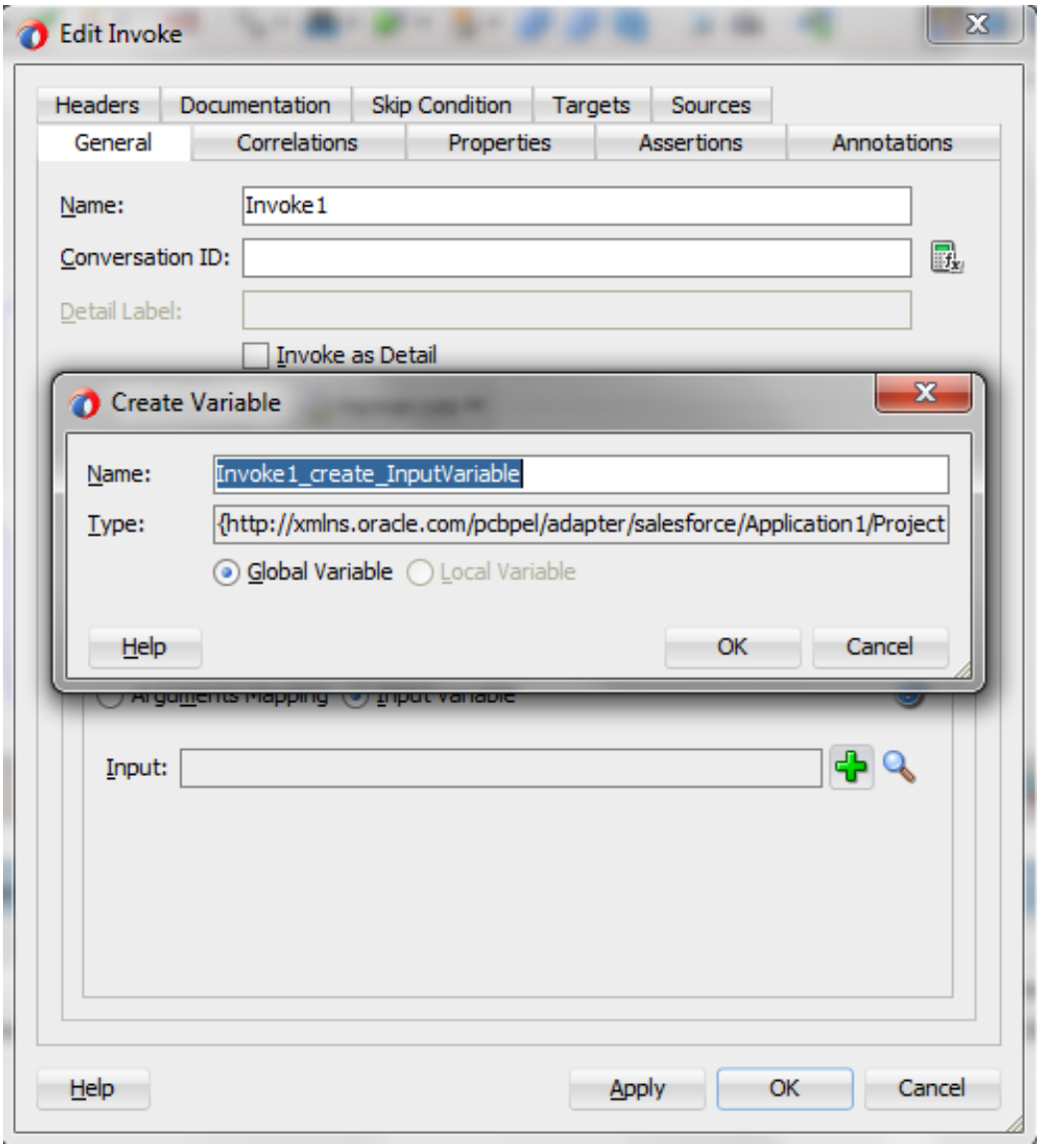


39. The BPEL orchestration should now look as shown in the screenshot below:



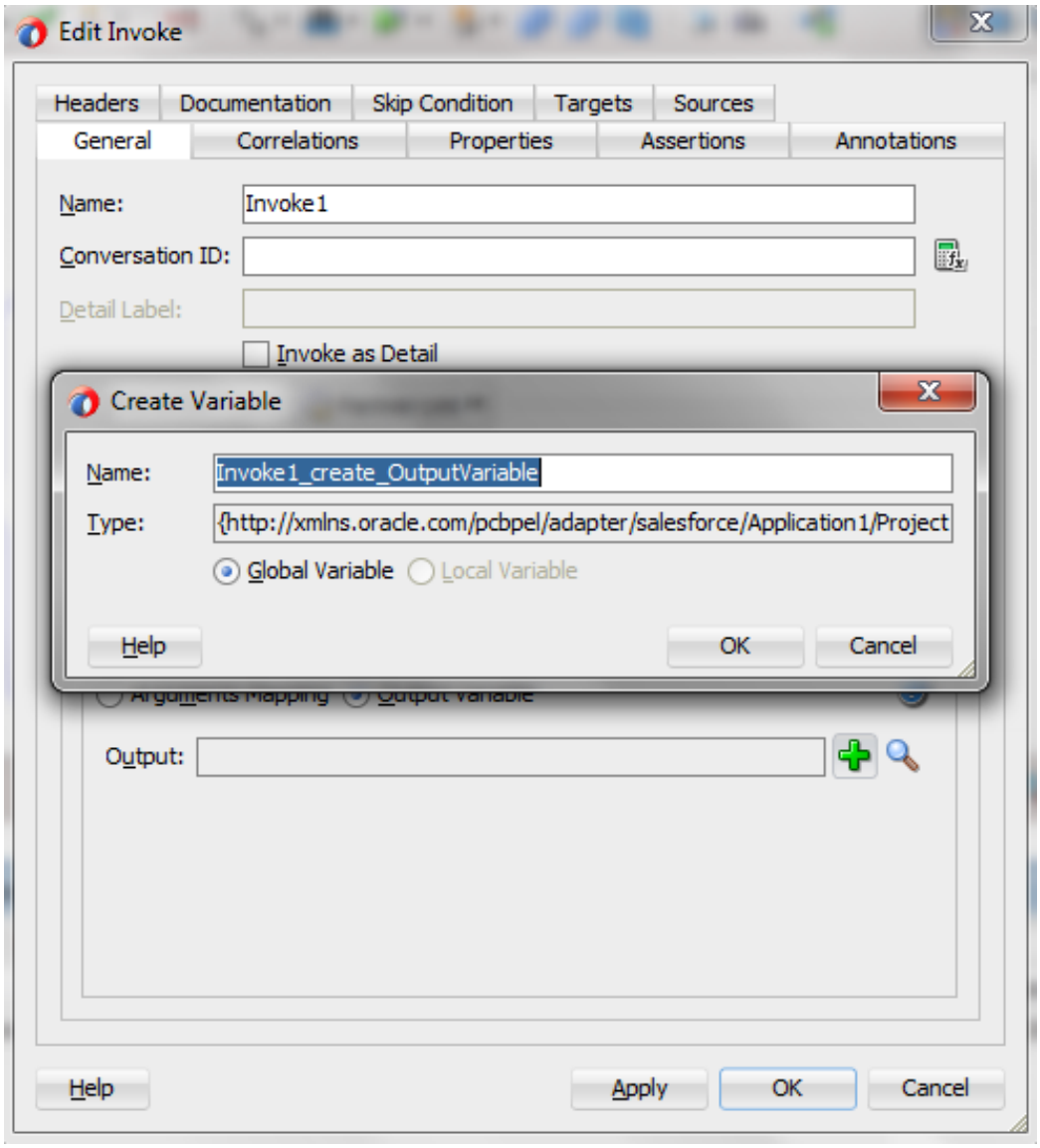
40. Create an input variable to the partner link by clicking the '+' button adjacent to **Input** text box in the **Variables** section. The **Create Variable** dialog is displayed, as shown in Figure 11-29.

Figure 11 - 29 Create an Input Variable



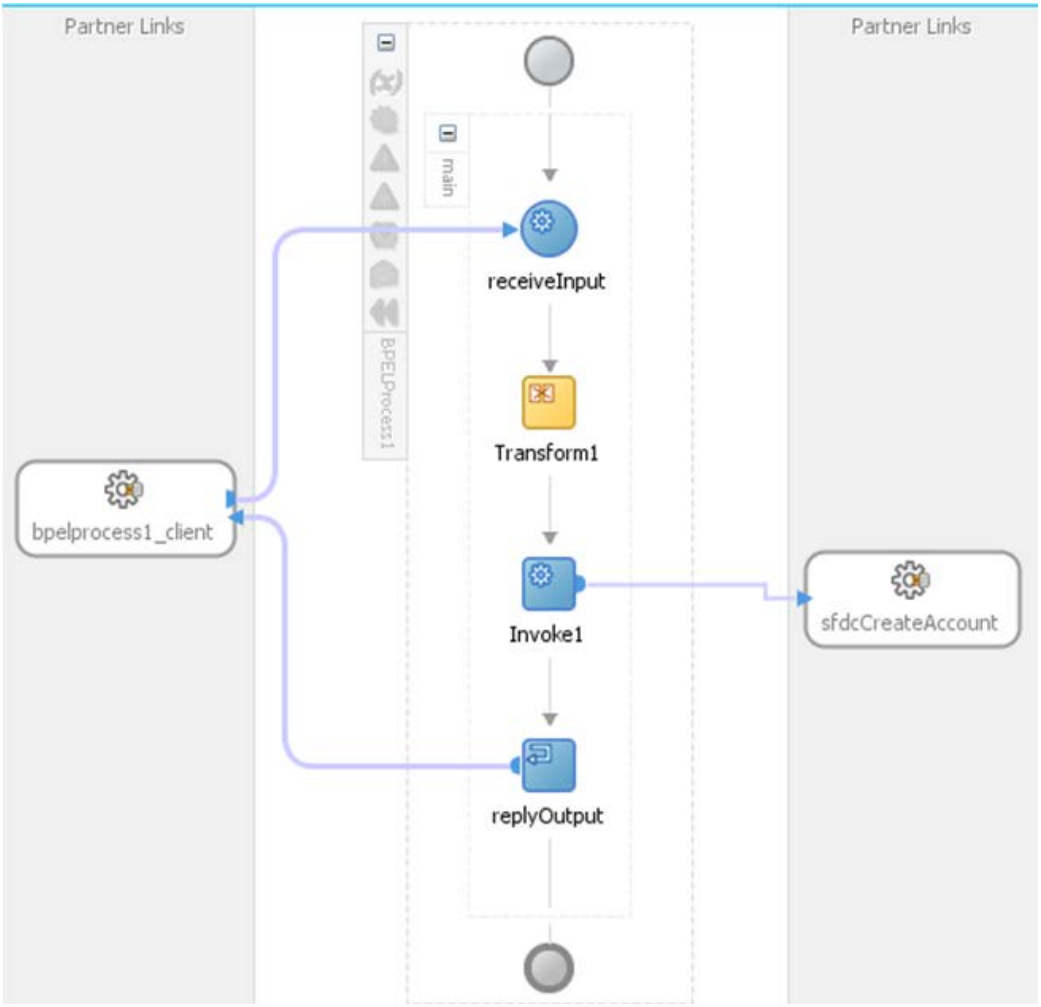
41. Create an output variable from the partner link by clicking the '+' button adjacent to **Output** text box in the **Variables** section. The **Create Variable** dialog is displayed, as shown in Figure 11- 30.

Figure 11 - 30 Create an Output Variable



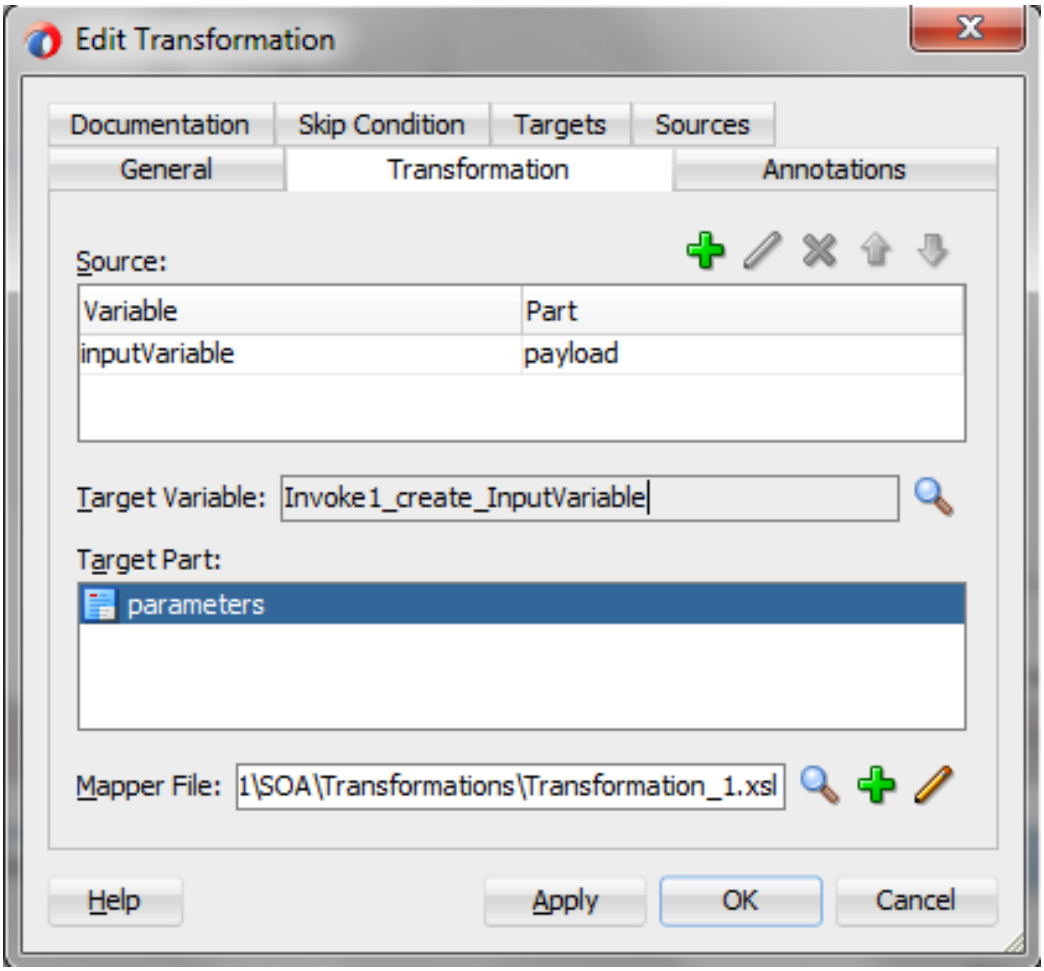
42. Introduce a transform activity, prior to the Invoke activity, as shown in Figure 11-31.

Figure 11 - 31 Transform Activities



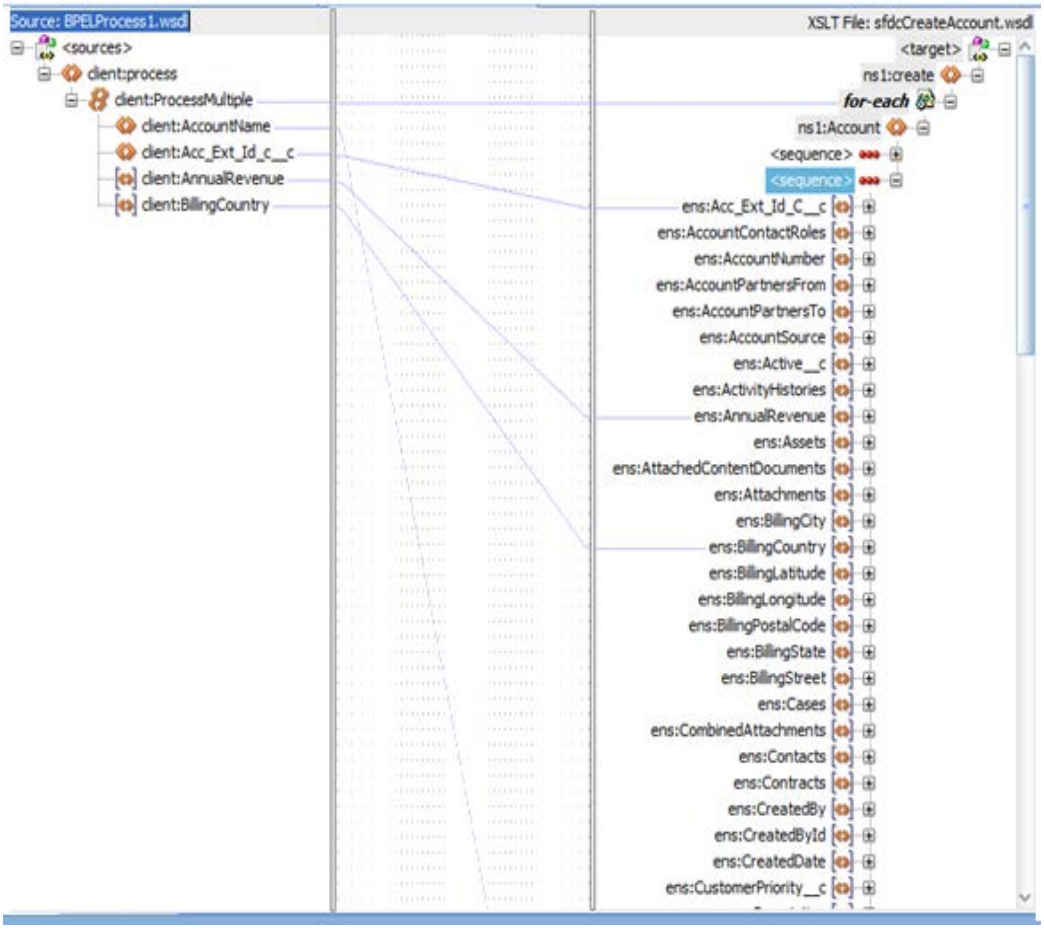
43. Map values from receive activity's input variable to invoke activity's input variable, as shown in Figure 11-32.

Figure 11 - 32 Edit Transform



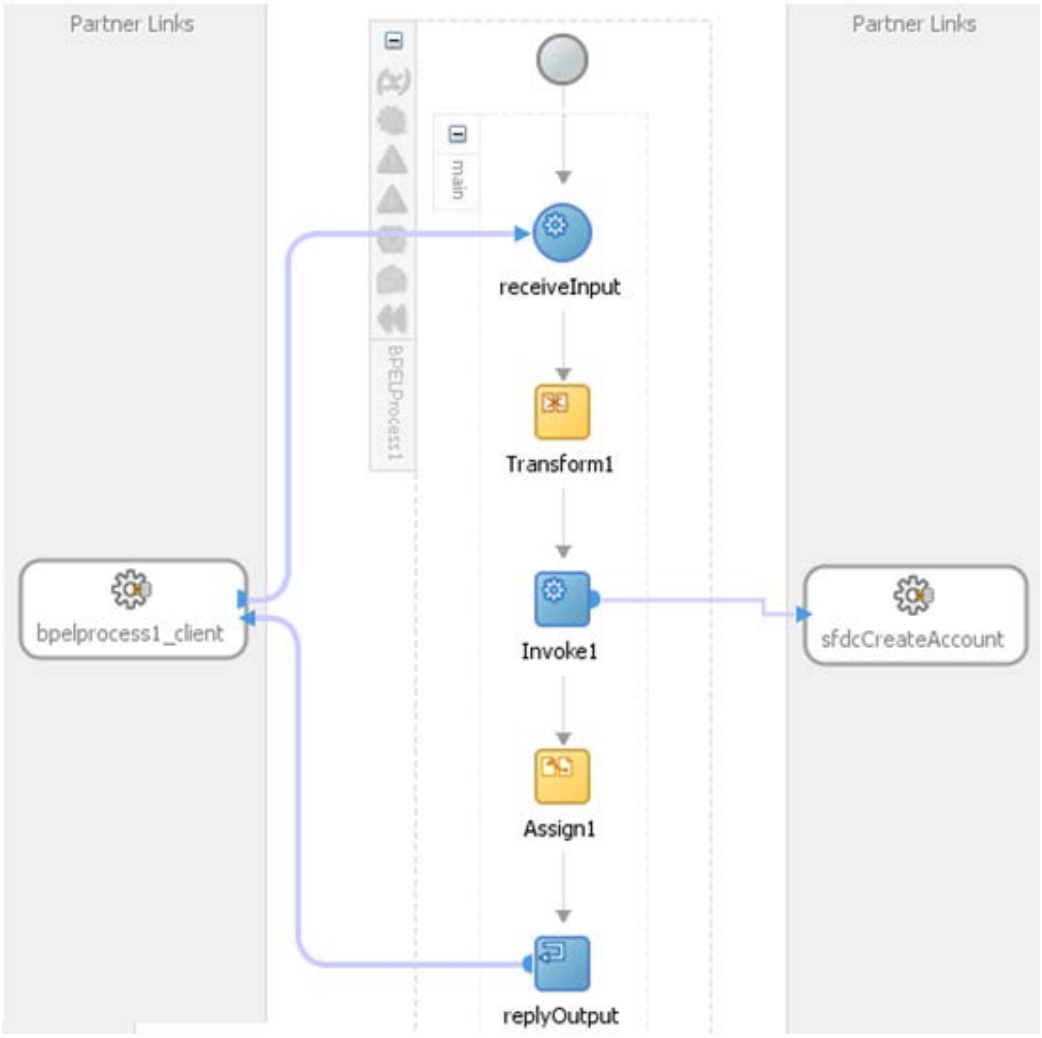
44. Click the '+' button adjacent to **Mapper File** text box to open the **Transformation_1.xsl** file.
45. Perform the mapping between **inputVariable** and **Invoke1_create_InputVariablea**:
    - Map Account_Name with Name.
    - Map Acc_Ext_Id_C__c with Acc_Ext_Id_C__c.
    - Map AnnualRevenue with AnnualRevenue.
    - Map BillingCountry with BillingCountry, as shown in Figure 11-33.

Figure 11 - 33 Mapping between inputVariable and Invoke1_create_InputVariable



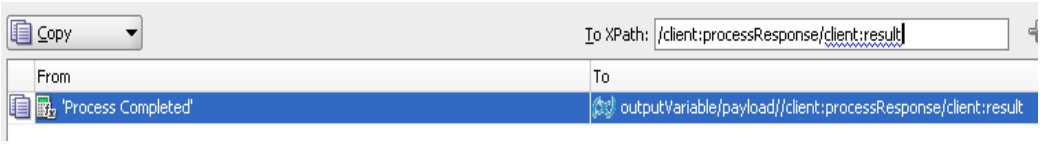
46. Since the response is suppressed, you would not map any element from **createResponse** to output variable.
47. Add an assign activity post the invoke activity, as shown in Figure 11-34.

Figure 11 - 34 Assign activity and Invoke Activity



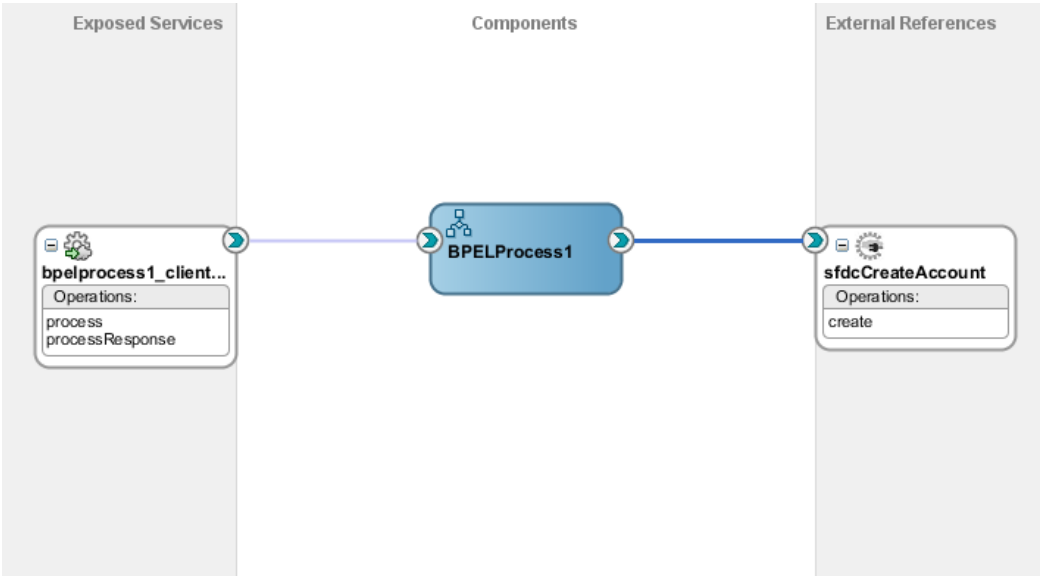
48. Provide a value to the output variable to ensure that the process is completed successfully, as shown in Figure 11-35.

Figure 11 - 35 Process Completed Screen



49. This completes the project creation. The composite.xml will look Figure 11-36.

Figure 11 - 36 Composite.xml After Project Creation



Deploy the Composite

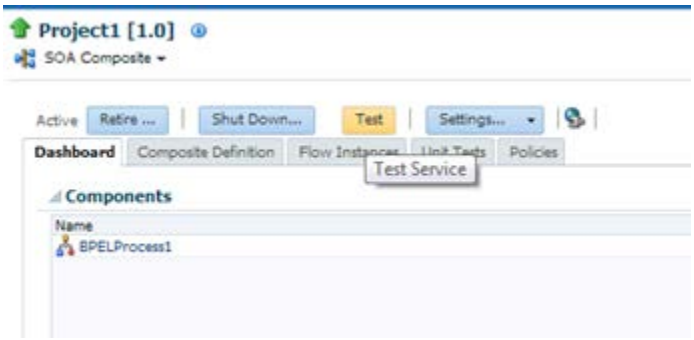
To deploy the Composite, refer to the section “Deploy the Composite”.

Testing the Process on Enterprise Manager Console

Perform the following steps to test the process on Enterprise Manager Console:

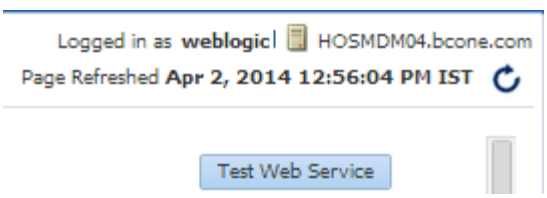
1. Login to Enterprise Manager Console for the server you deployed your project on.
2. Open **Project1** under the **default** partition.
3. Click the **Test** button to test the Web service, as shown in Figure 11-37.

Figure 11 - 37 Test the Web Service



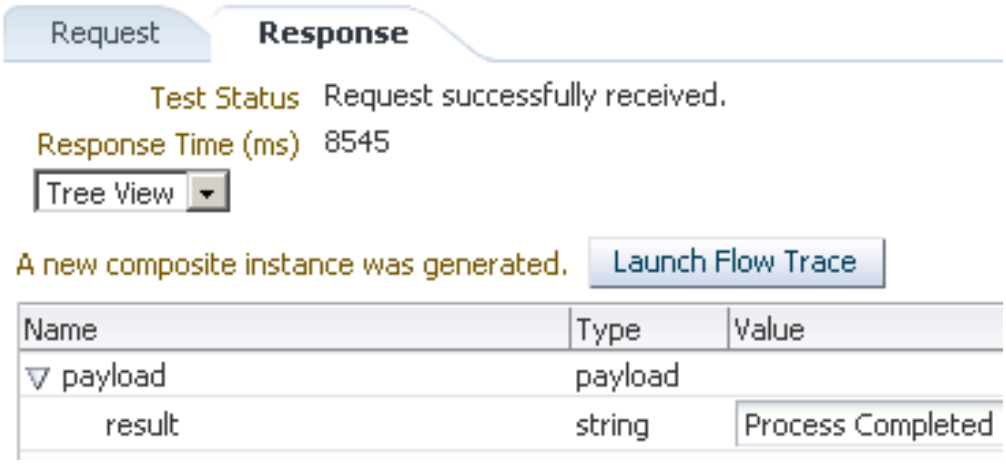
4. Provide the input payload and click the **Test Web Service** button, as shown in Figure 11-38.

Figure 11 - 38 Test Web Service



5. After successful execution the response is as shown in Figure 11-39.

Figure 11 - 39 Test Status

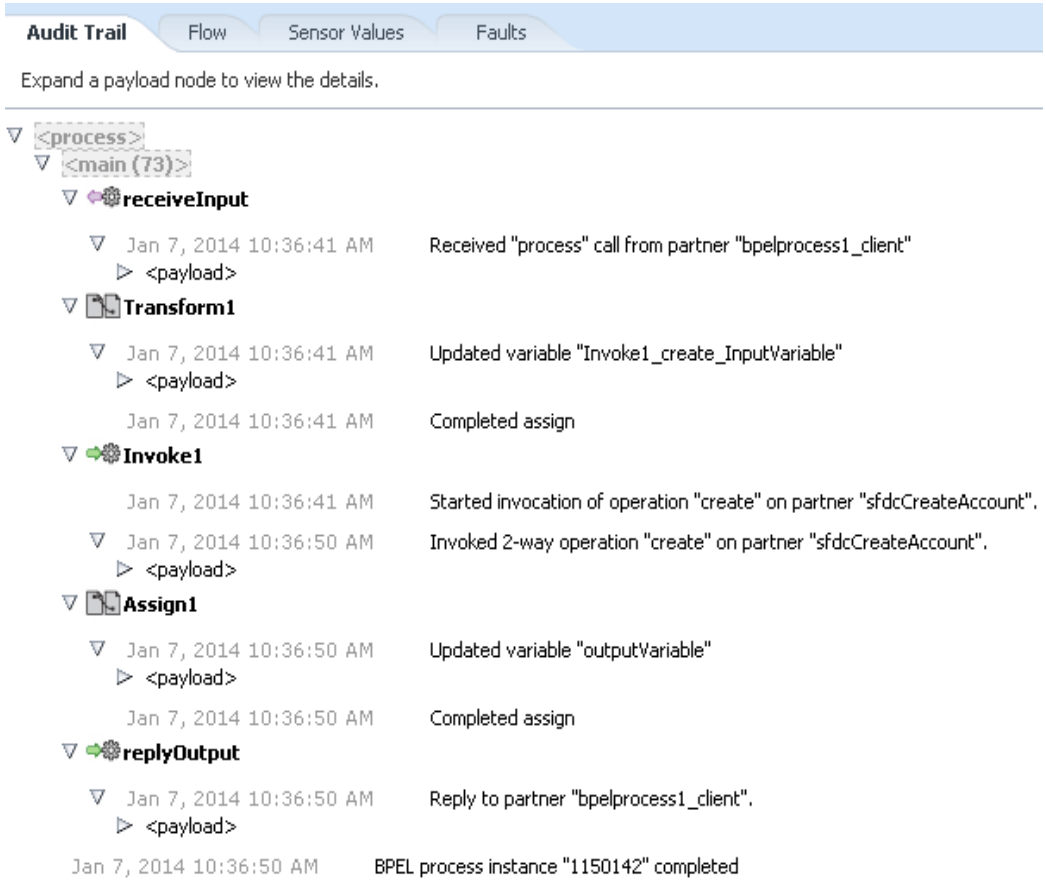


6. Click the **Launch Flow Trace** button to view the **Audit Trail**, as shown in Figure 11-40.

Figure 11 - 40 Launch Flow Trace

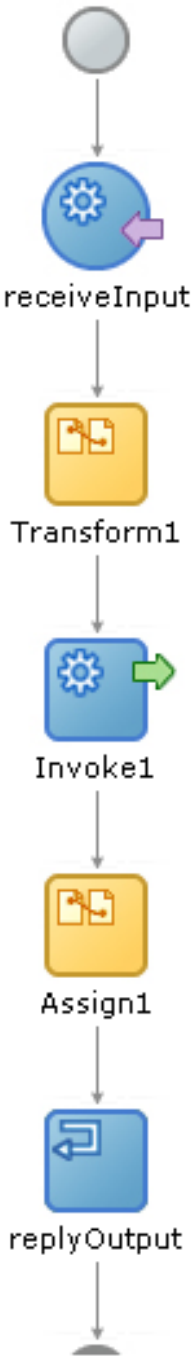
7. The **Audit Trail** will look like Figure 11-41.

Figure 11 - 41 Audit Trail



8. Click on **Flow** tab, The Flow tab is shown in Figure 11-42.

Figure 11 - 42 Flow Tab

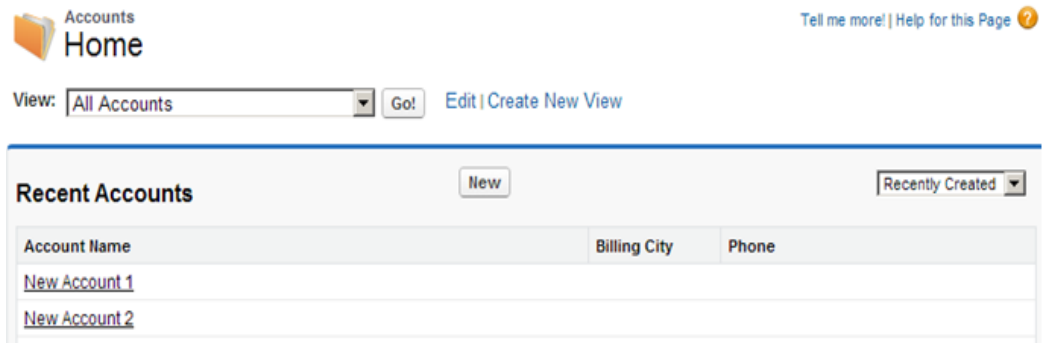


9. The Invoke activity of the process is shown in Figure 11-43.

Figure 11 - 43 Invoke Activity

10. This completes the create account scenario. The creation of account can be verified on Salesforce.com as shown in Figure 11-44 (because the response returned from salesforce.com is suppressed and not shown in the audit trail).

Figure 11 - 44 Create Account

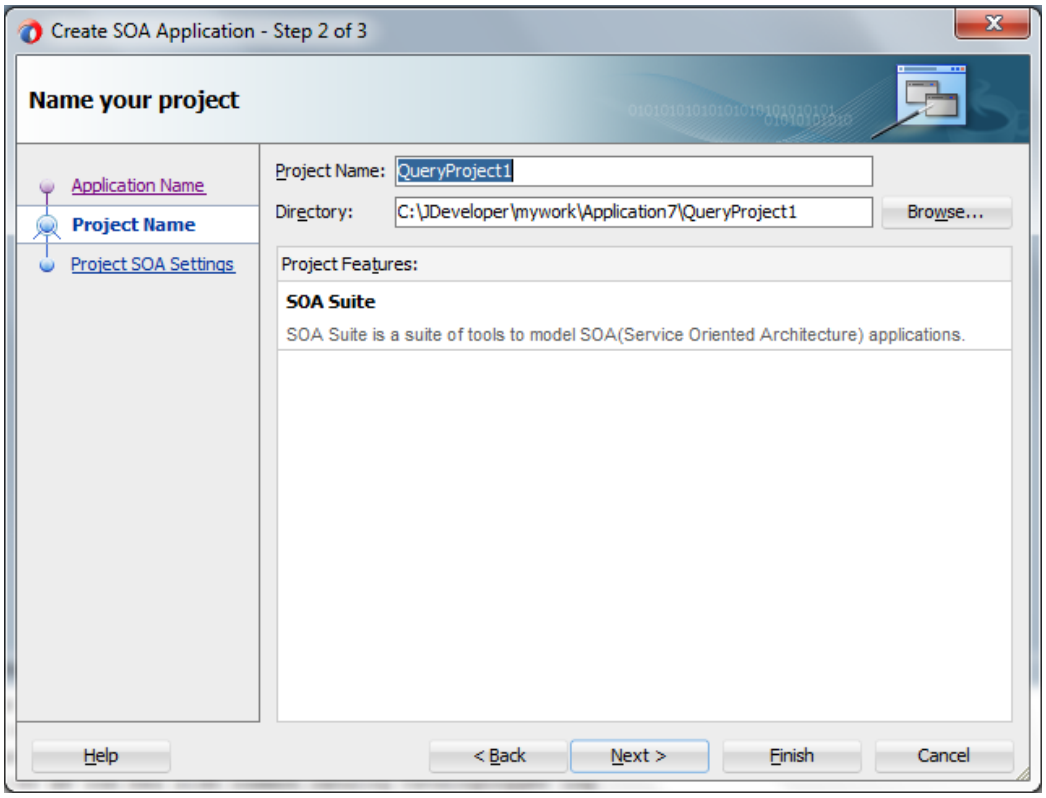


11.2. How to use query and queryMore Operations?

Perform the following steps to use query and queryMore Operations.

1. Create a New project in an existing application or in a new application.
2. Provide a suitable name to your project as shown in Figure 11-45.

Figure 11 - 45 Name your project

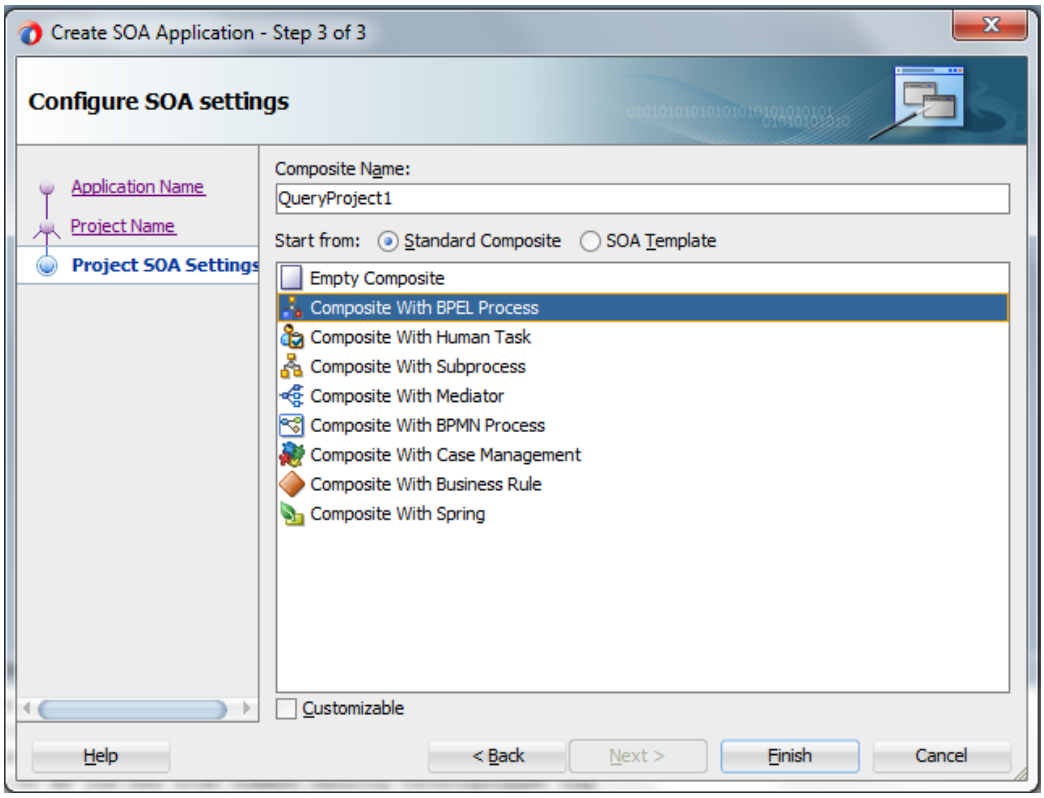


11.2.1. Create BPEL Process

Perform the following steps to create BPEL Process.

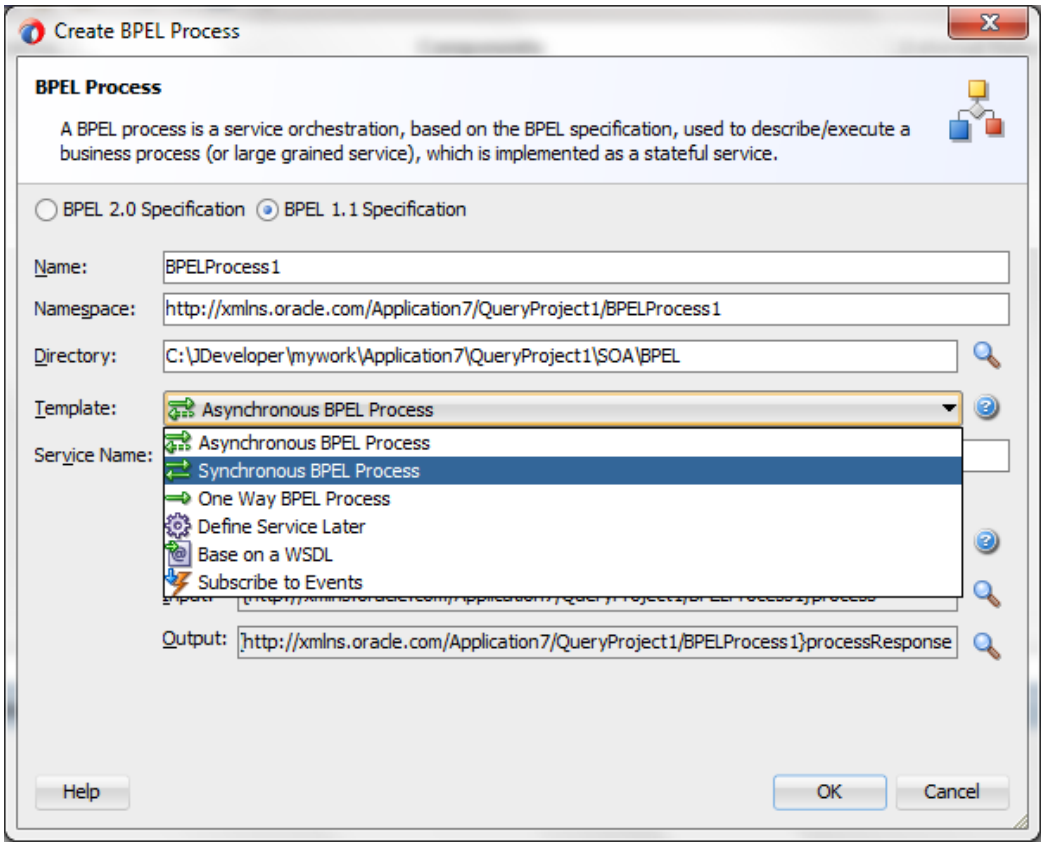
1. Click **Next** and select **Composite with BPEL Process** from **Standard Composite** list, as shown in Figure 11-46.

Figure 11 - 46 Configure SOA Setting



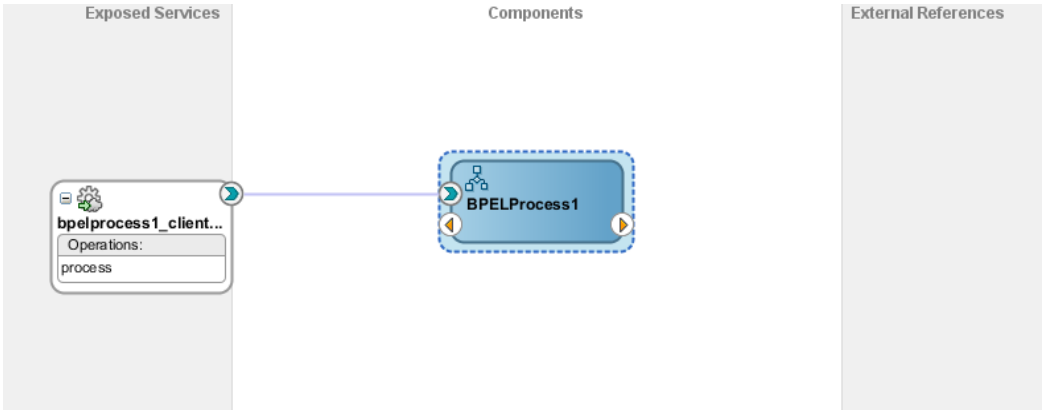
2. Click **Finish**.
3. Select the BPEL 1.1 radio button and **Synchronous BPEL Process** from **Template** drop-down and click **OK**, as shown in Figure 11-47.

Figure 11 - 47 Create BPEL Process



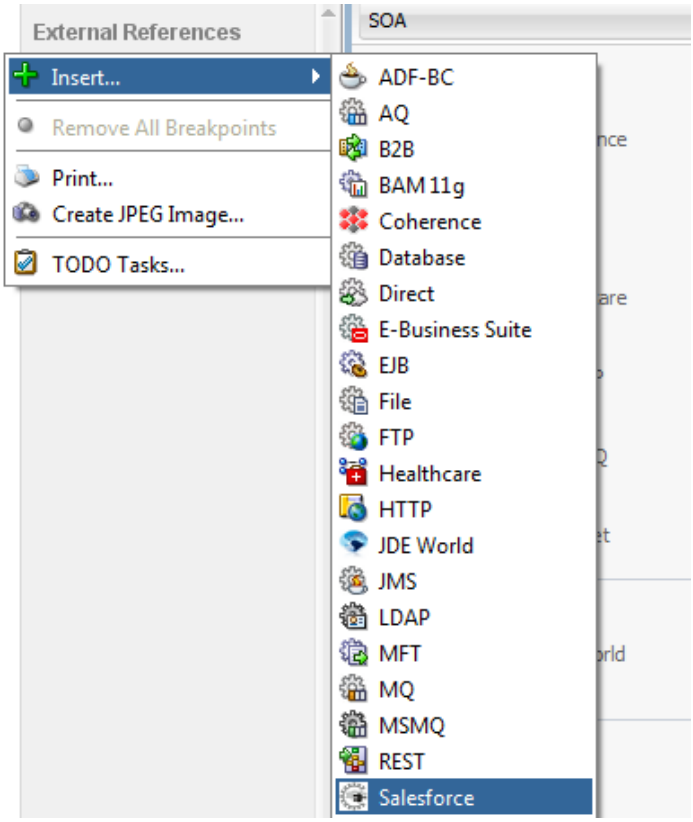
The composite.xml looks like Figure 11-48.

Figure 11 - 48 Composite.xml



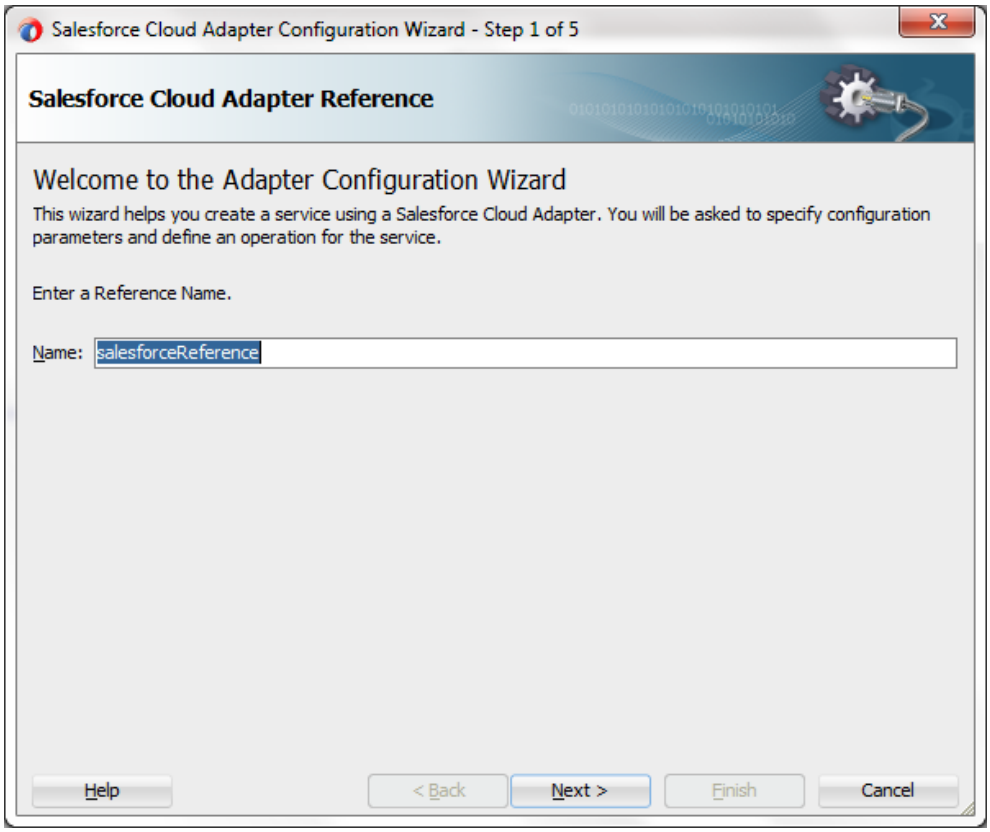
4. In **External References** swim-lane of the composite.xml file, right-click and select **Salesforce** adapter, as shown in Figure 11-49.

Figure 11 - 49 Salesforce adapter



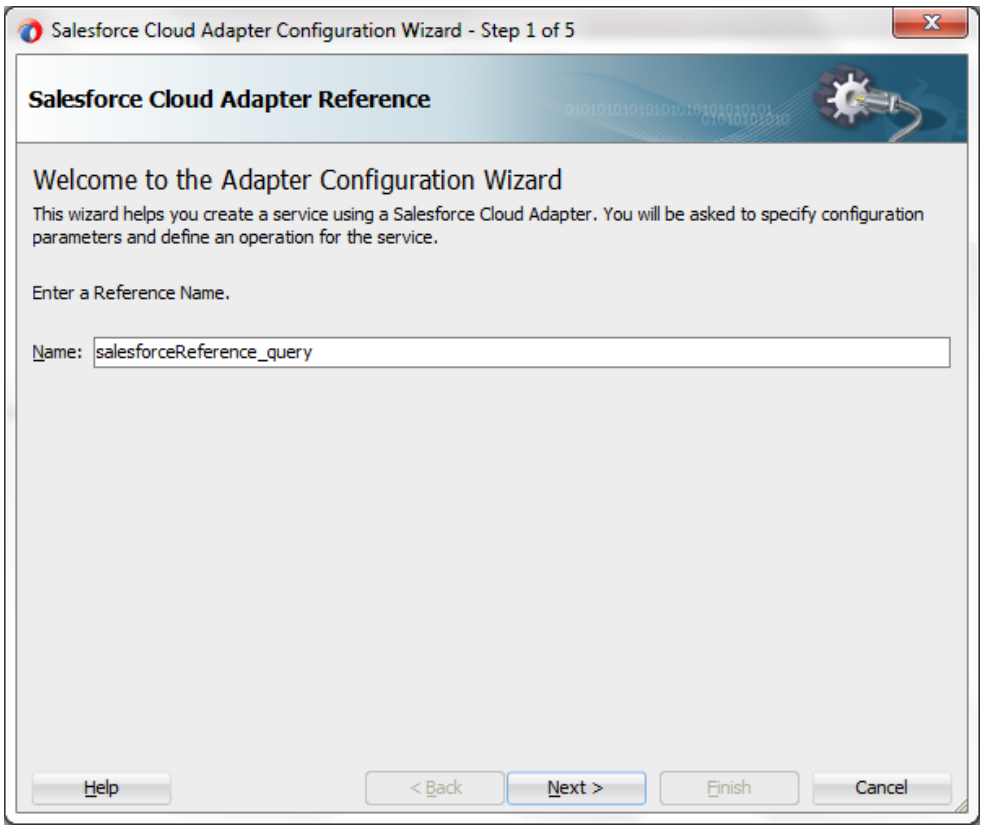
5. The Salesforce Cloud Adapter Configuration Wizard -Welcome page is displayed, as shown in Figure 11-50.

Figure 11 - 50 Welcome Page



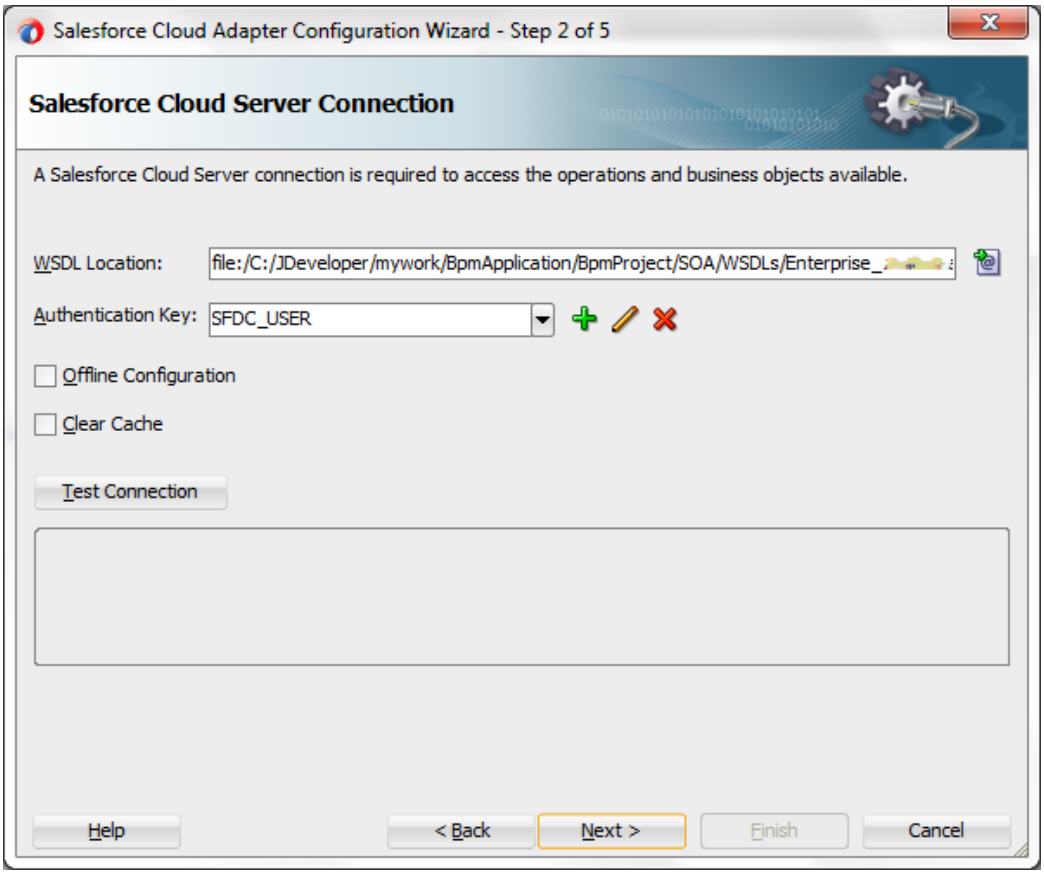
6. In the **Welcome** page, Enter a Reference Name in the **Name** field, as shown in Figure 11-51

Figure 11 - 51 Name your Service



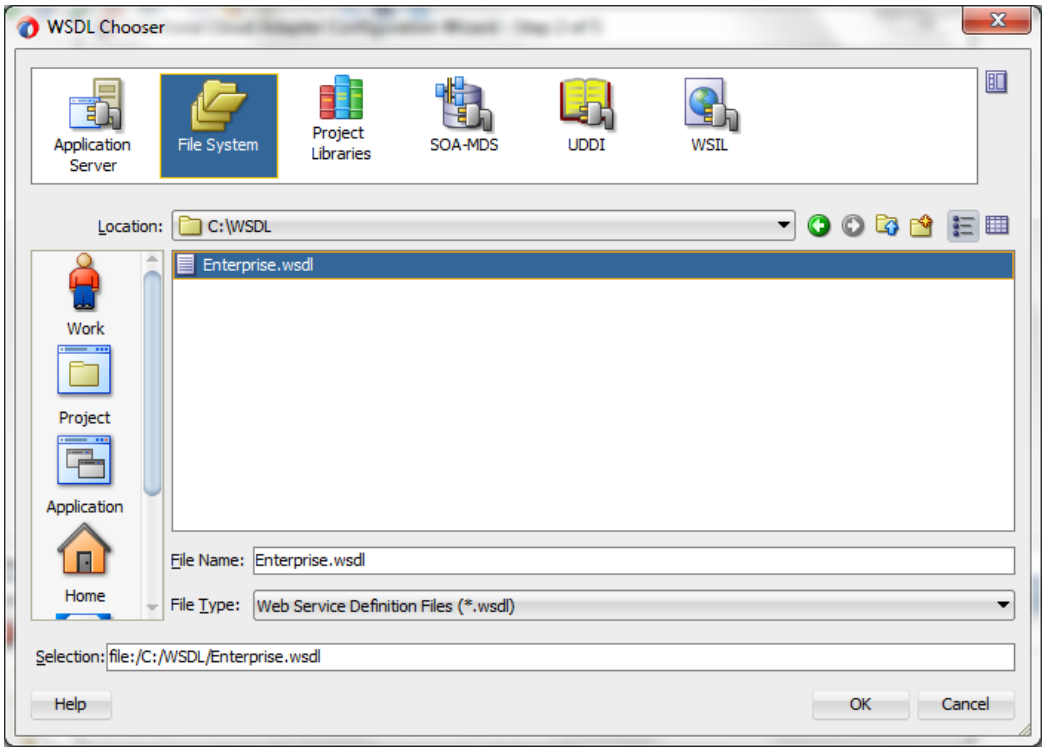
7. Click **Next**.
8. The **Salesforce Cloud Server Connection** page is displayed. The **WSDL Location** and **Authentication Key** text boxes are already populated. It picks up these values from the cache. You can re-enter these values. If you want to use a different value, click the **Find existing WSDLs** icon, which is located to the right of the **WSDL Location** field, as shown in Figure 11-52.

Figure 11 - 52 Salesforce Cloud Server Connection Page



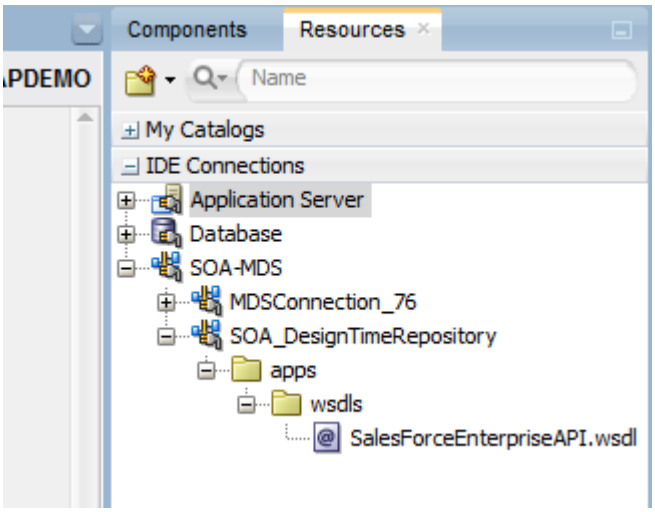
9. The **WSDL Chooser** dialog is displayed, browse and select the downloaded Enterprise WSDL and click **OK**, as shown in Figure 11-53.

Figure 11 - 53 SOA Resource Browser



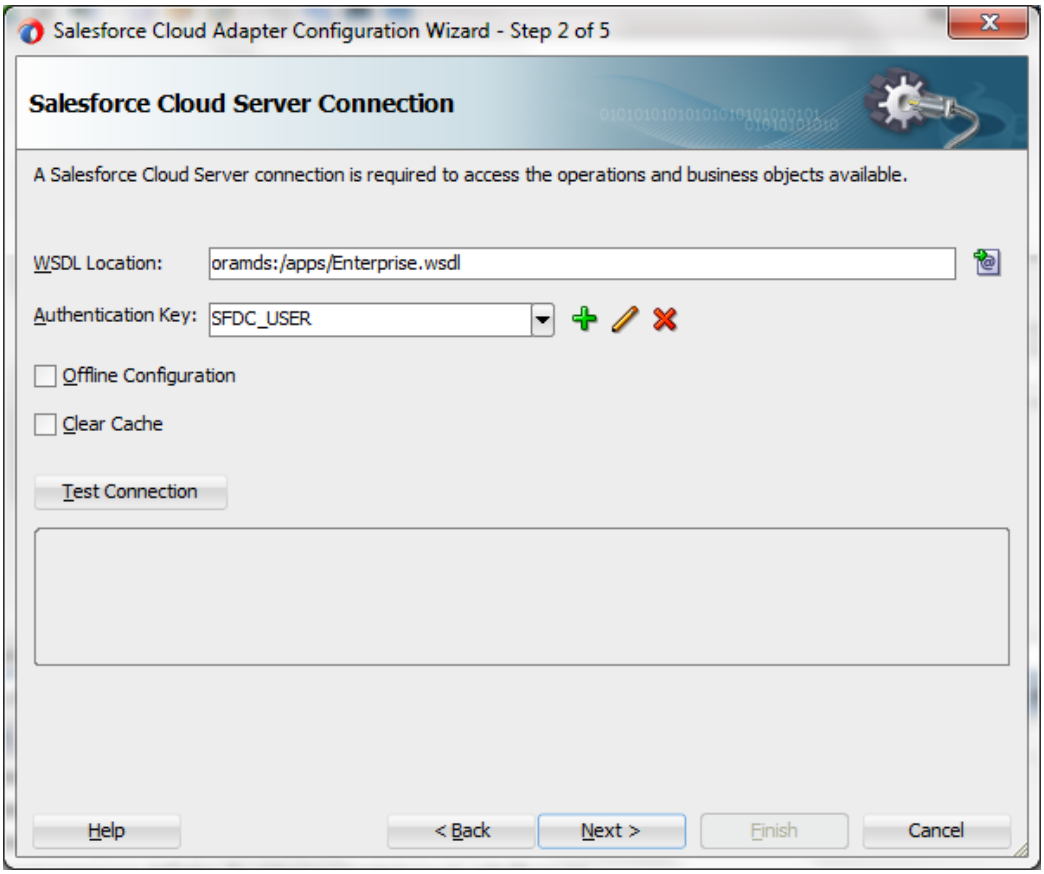
Note: Alternatively, you can store WSDL at an MDS location and access it, as shown in Figure 11-54.

Figure 11 - 54 SOA Resource Browser



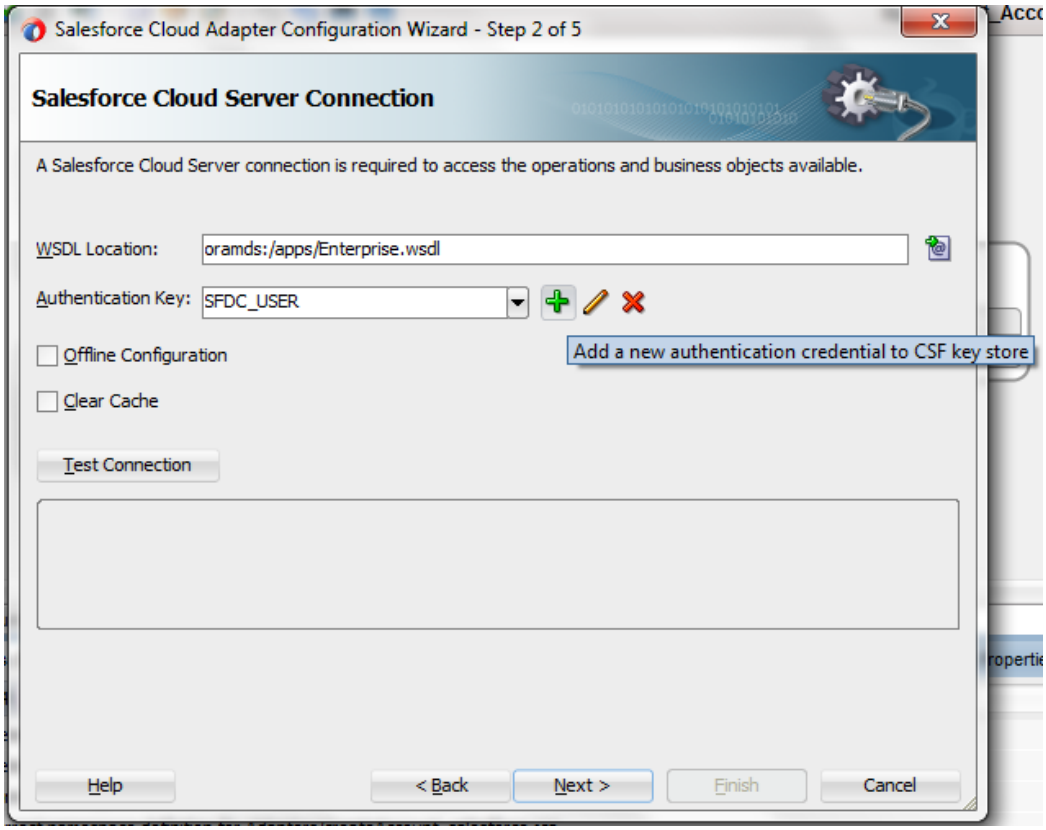
10. Click **OK**.
11. Traverse to IDE Connections → SOA-MDS. Select the appropriate SOA-MDS connection where you placed the Enterprise WSDL. Select the WSDL file to be used in the adapter configuration and click **OK**.
12. The WSDL location should be of the form ‘oramds:/apps/SOA/WSDLs/Integration/WSDLNAME.wsdl’, as shown in Figure 11-55.

Figure 11 - 55 WSDL location



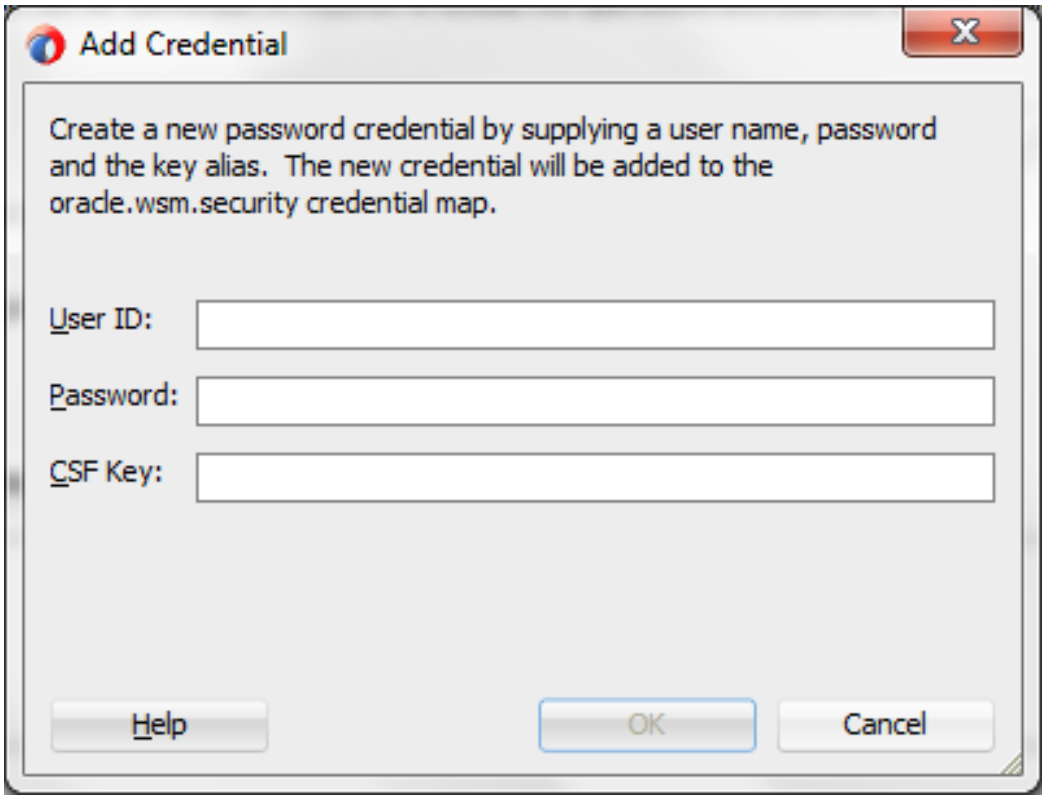
13. Click **OK**.
14. Click + button to create a new **Authentication Key**, as shown in Figure 11- 56.

Figure 11 - 56 Create a New Authentication Key



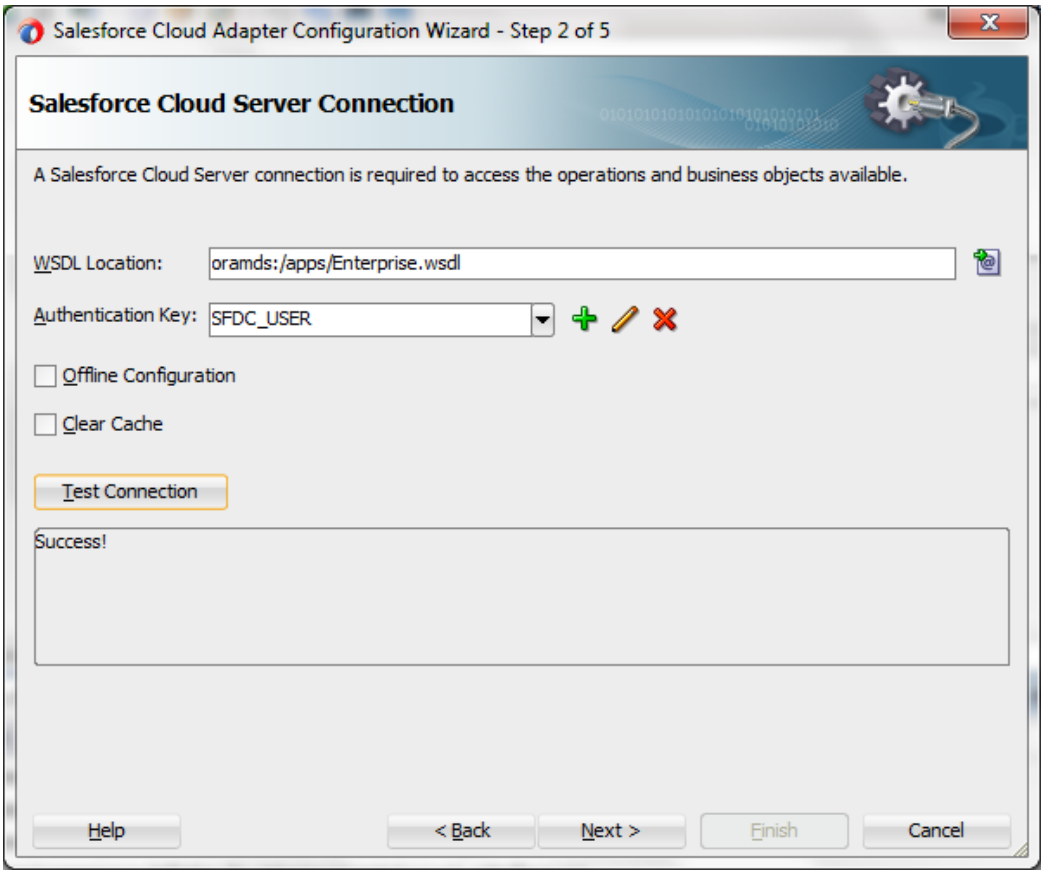
15. The **Add Credential** page is displayed, as shown in Figure 11- 57. Provide a suitable name and the Salesforce.com credentials. The password should be a combination of Salesforce.com password and Salesforce.com Security Token.

Figure 11 - 57 Add Credential



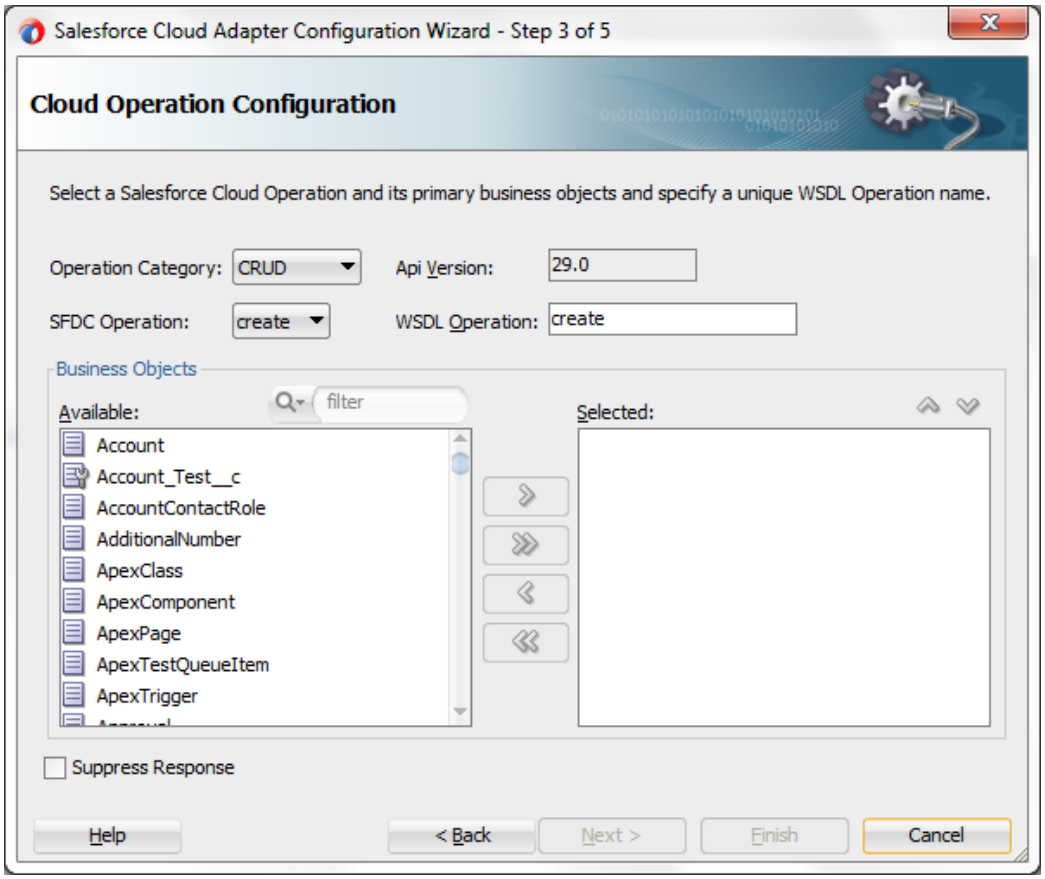
16. Click **Test Connection** button to validate the Authentication Key, as shown in Figure 11-58 .

Figure 11 - 58 Test Connection



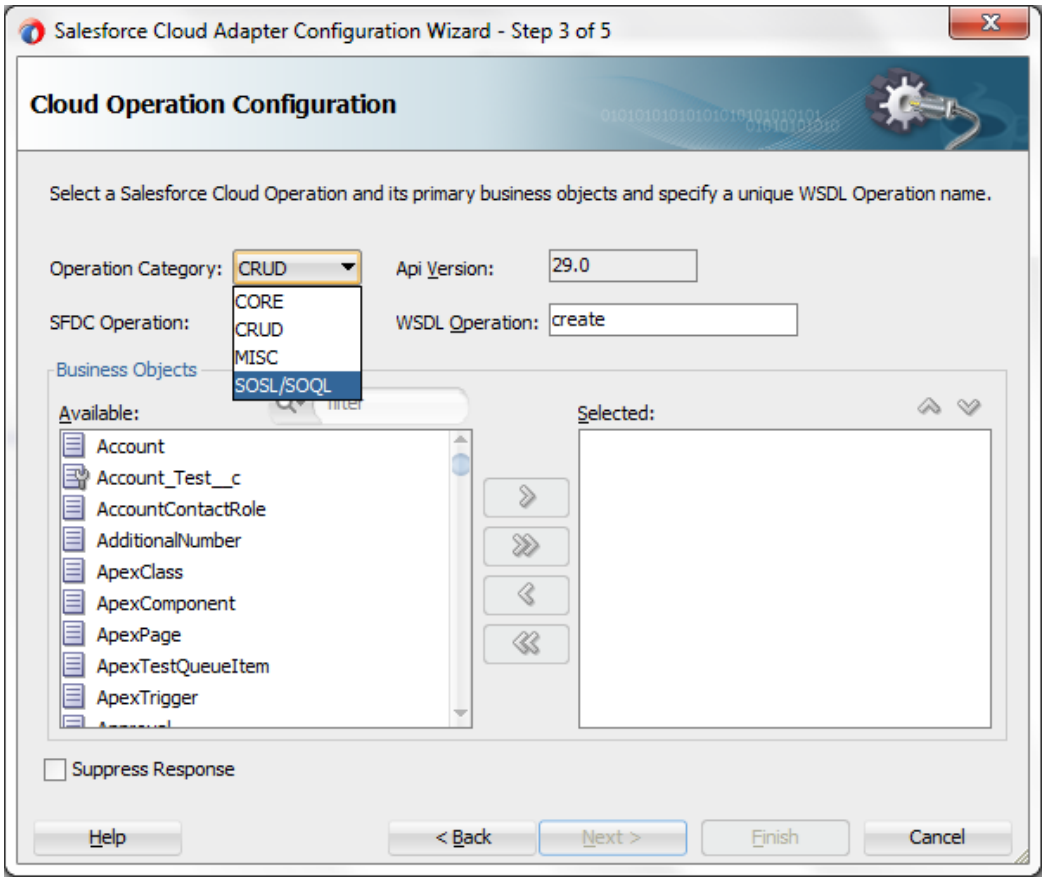
17. Click **Next**.
18. The **Cloud Operation Configuration** page is displayed, as shown in Figure 11-59.

Figure 11 - 59 Cloud Operation Configuration Page



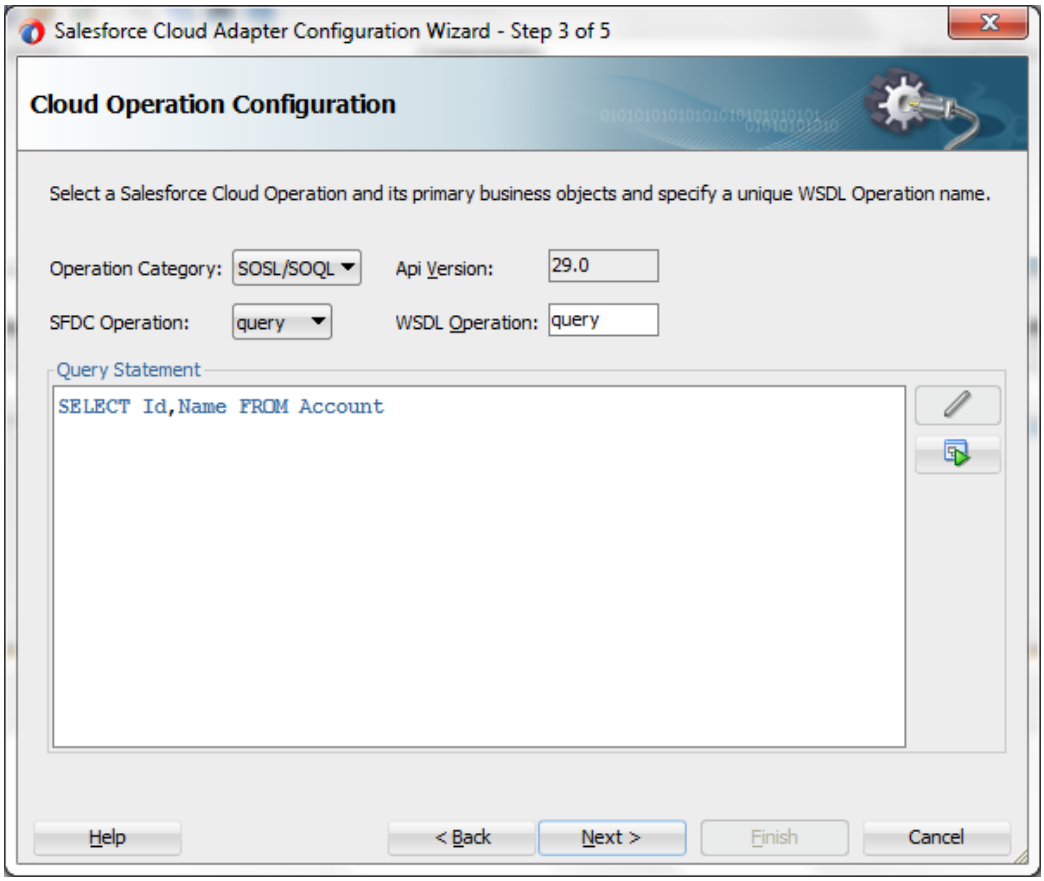
19. From the list of **Operation Category**, select **SOSL/SOQL**, as shown in the Figure 11-60.

Figure 11 - 60 Select SOSL/SOQL



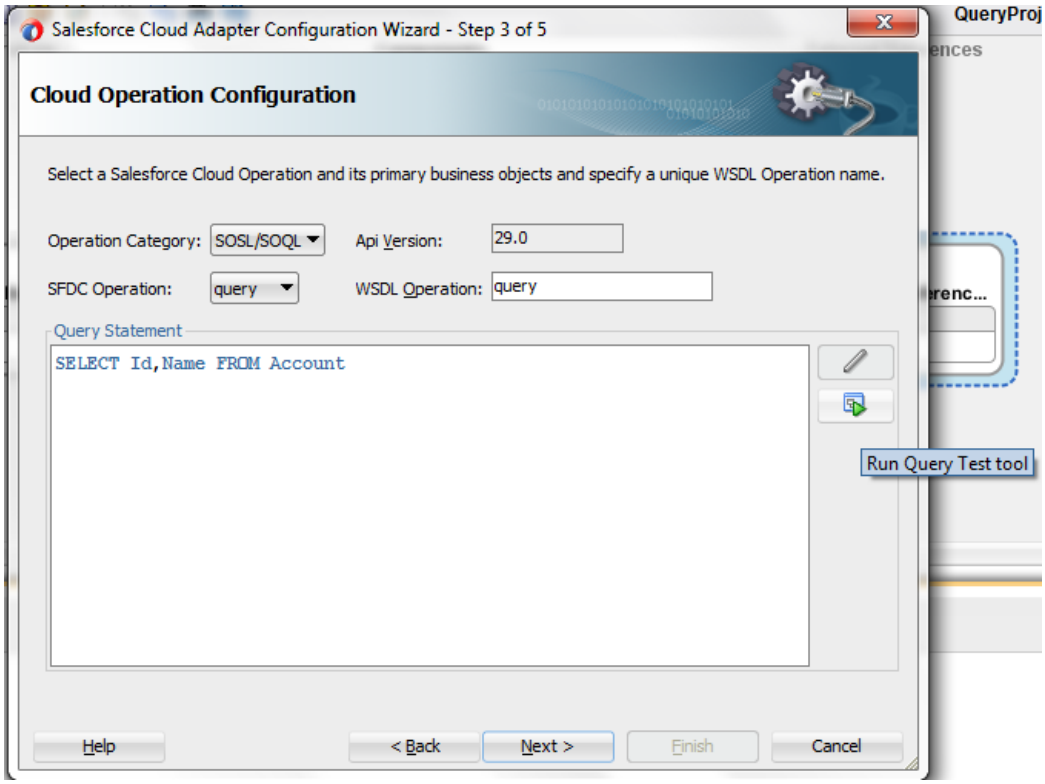
20. Now, the query operation will automatically be selected. Provide the query string in the text box, as shown in the Figure 11-61.

Figure 11 - 61 Provide the query String



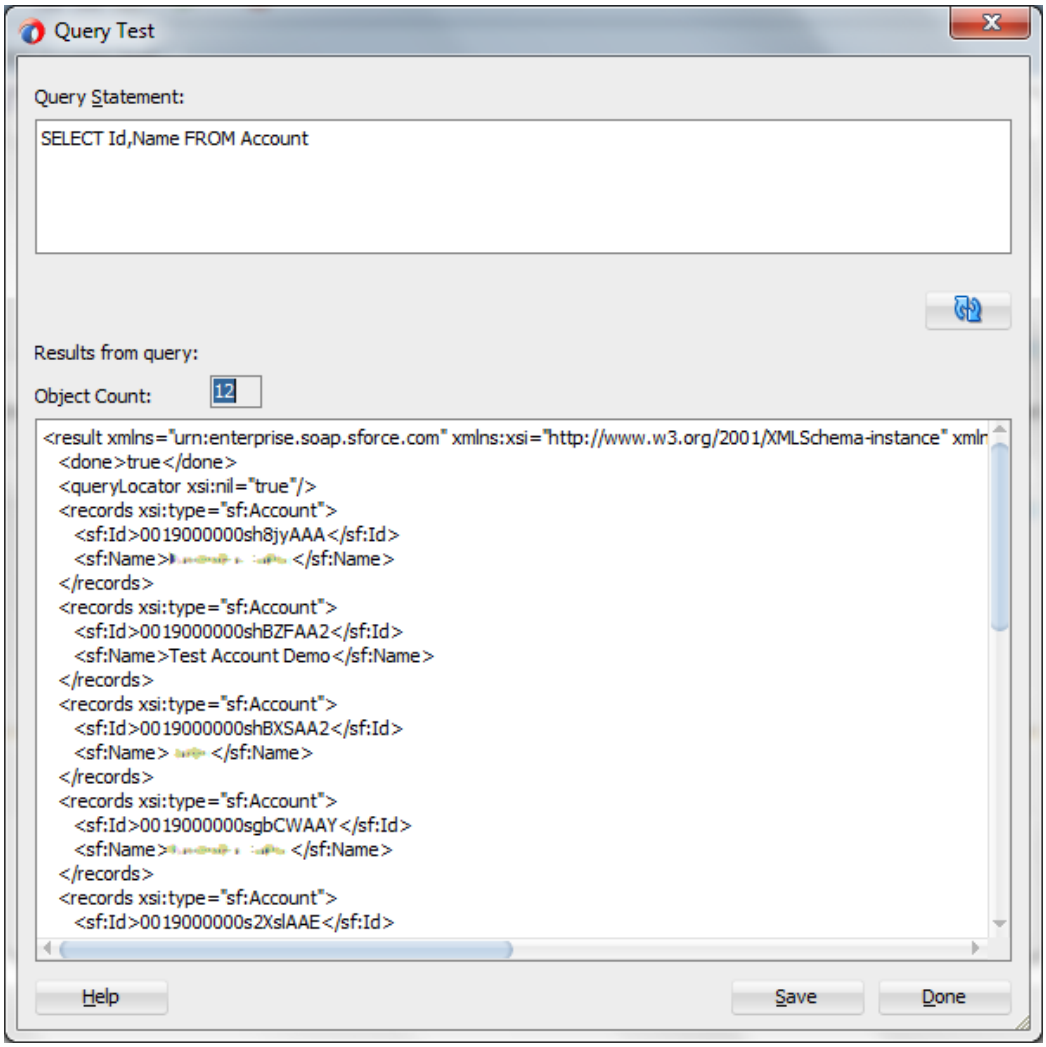
21. You can also test the query string after clicking on the **Query Test** button. As shown in Figure 11-62.

Figure 11 - 62 Query Test button



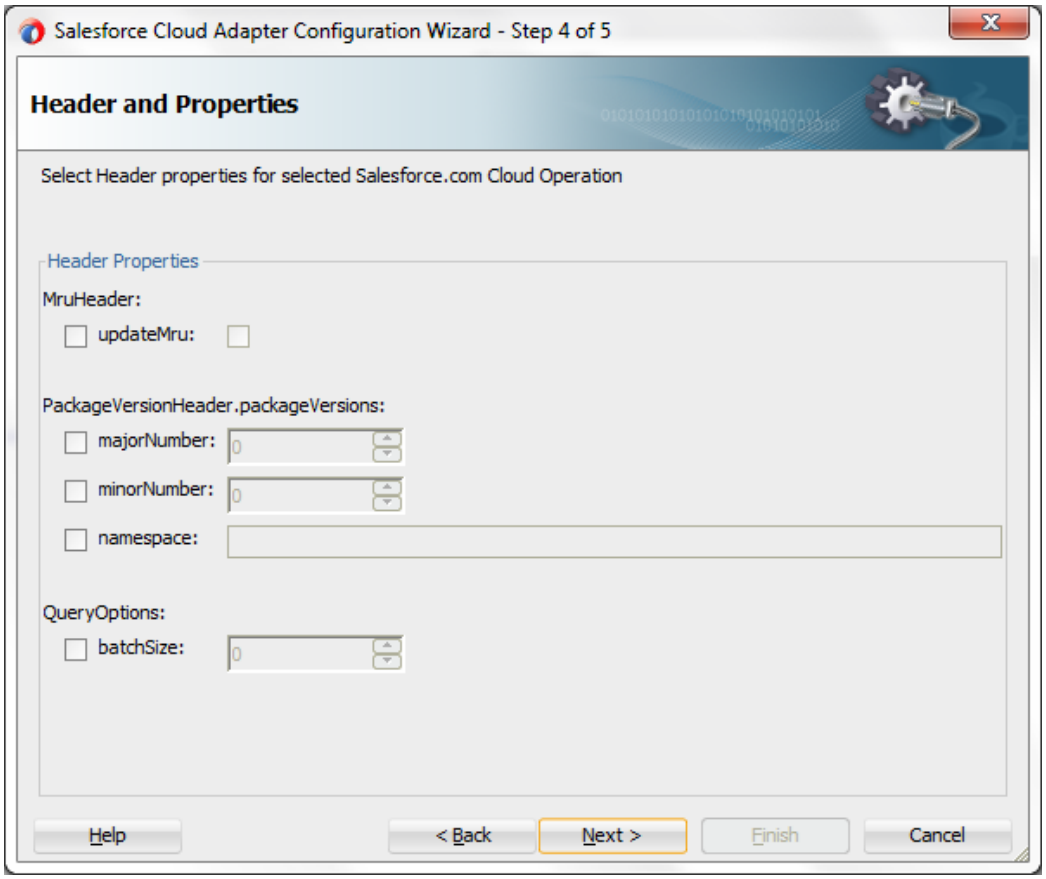
22. After clicking on query test button, a **Query Test** dialog appears, as shown in Figure 11-63.

Figure 11 - 63 Query Test



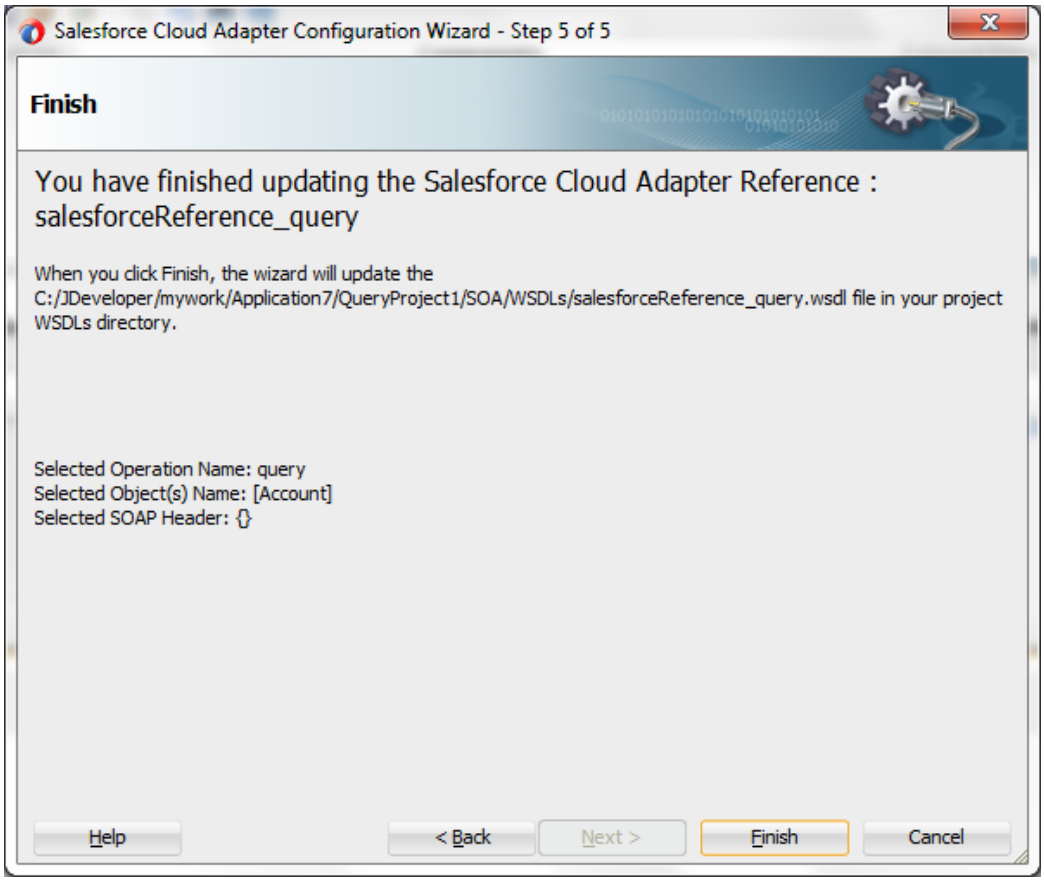
23. Click on **Save** button, if you have modified your query in the query test dialog, and wish to save the modified query, else, click on **Done** button, which would display the original query. Either of the buttons will take you back on the Cloud Operation Configuration page.
24. Click on **Next**. The **Header and Properties** page is displayed. Provide the value of headers as per your business requirement, as shown in Figure 11-64.

Figure 11 - 64 Provide the Value of Headers



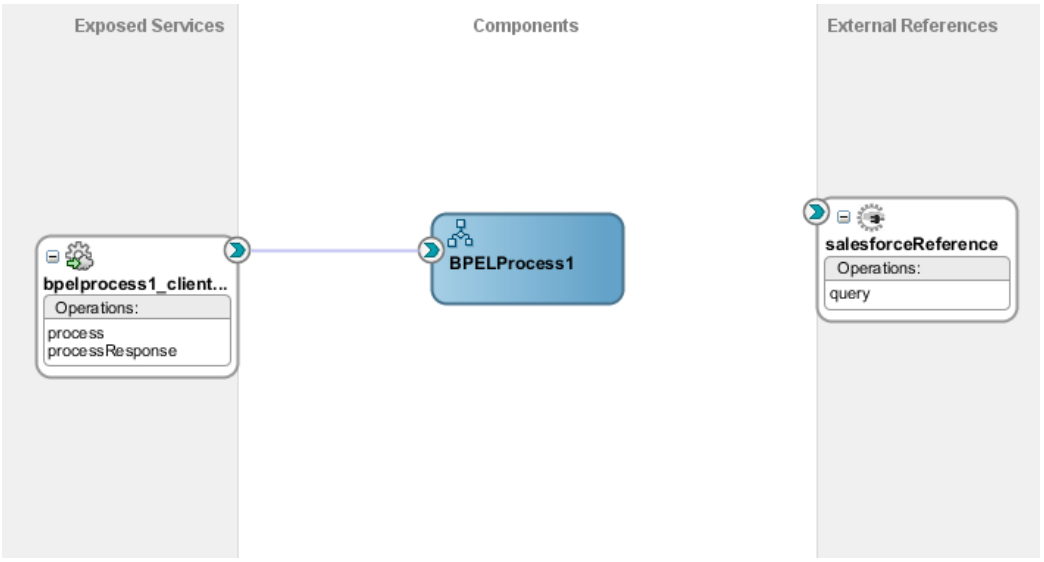
25. Click **Next**.
26. The finish page is displayed. It provides a complete summary of the operation selected, object on which the operation would operate and the headers selected for that operation, as shown in Figure 11-65.

Figure 11 - 65 Finish Page



27. Click the **Finish** button to complete adapter configuration.
28. After clicking on **Finish** button, the following screen appears, as shown in Figure 11-66.

Figure 11 - 66 Composite Screen



Understanding query Response

The response returned by query operation has the following structure:

1. **done:** It is a Boolean element that contains the following values:

- **True:** If all records are returned in a single invoke to query operation.
- **False:** If all records are not returned in a single invoke to query operation, i.e., there are more records to fetch.

2. **queryLocator:** If the value of the element **done** is false, then query operation returns a queryLocator value that is used to make queryMore call to Salesforce.com.
3. **Records:** The set of records queried by the user.
4. **Size:** The number of records for that particular query.

Figure 11-67 shows all the above mentioned values.

Figure 11 - 67 Query Operation Structure

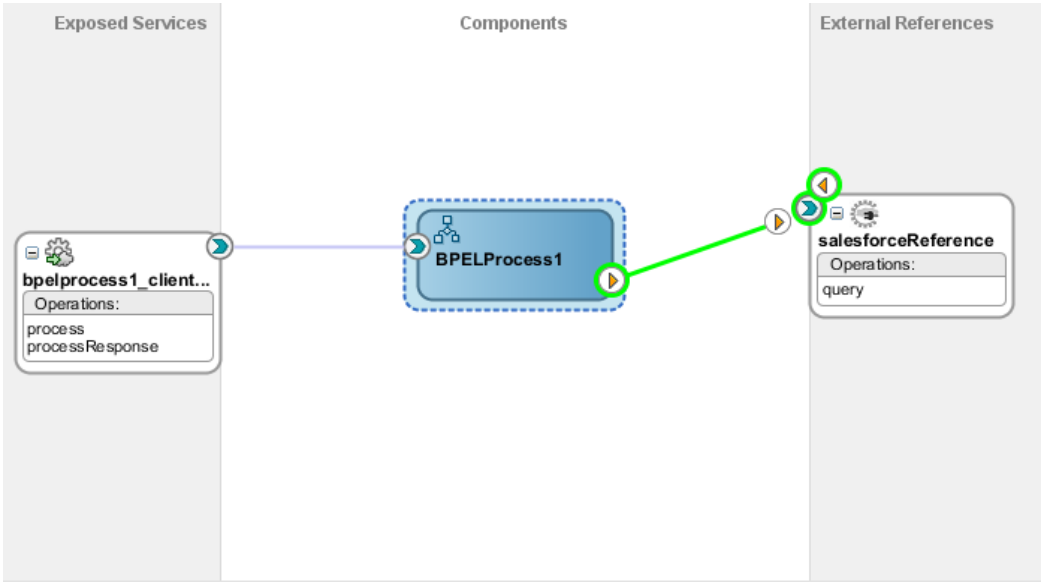
```
<complexType name="QueryResult">
    <sequence>
        <element name="done" type="xsd:boolean"/>
        <element name="queryLocator" type="tns:QueryLocator" nillable="true"/>
        <element name="records" type="ens:sObject" nillable="true" minOccurs="0" maxOccu
        <element name="size" type="xsd:int"/>
    </sequence>
</complexType>
```

Integration with BPEL

Perform the following steps for integration with BPEL:

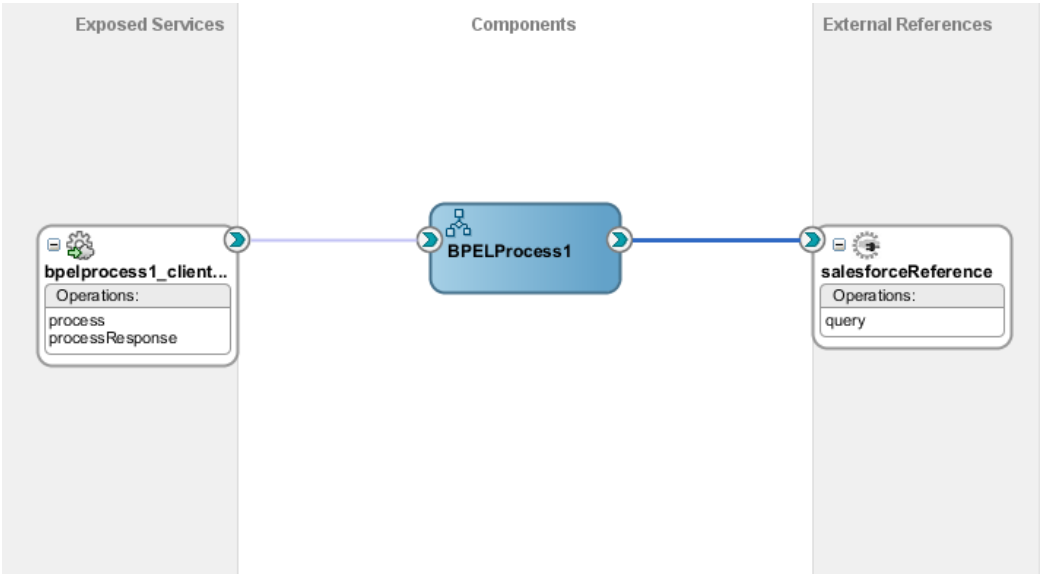
1. Connect **BPELProcess1** and **query** via a wire, as shown in Figure 11-68.

Figure 11 - 68 Wiring BPELProcess1 and query



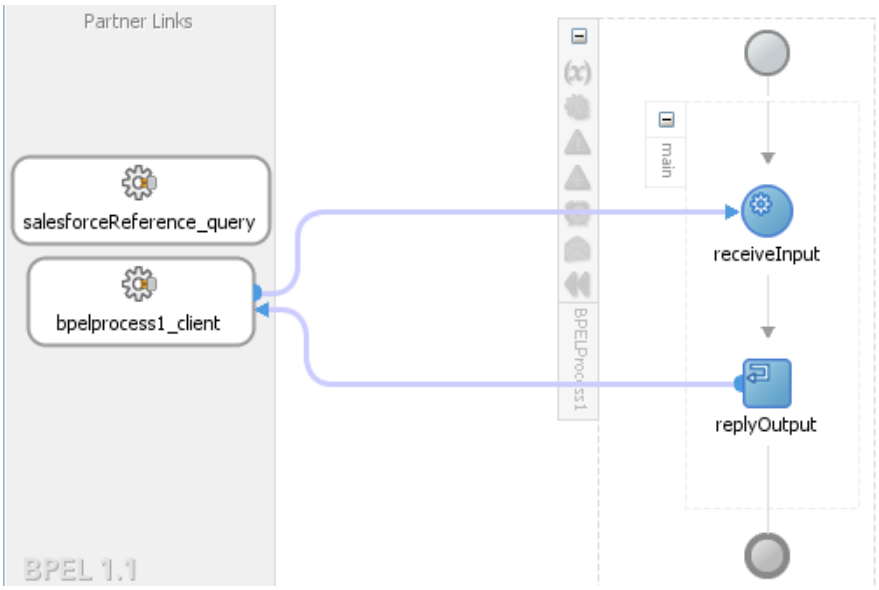
2. After wiring, your composite will look, as shown in Figure 11-69.

Figure 11 - 69 Wiring BPELProcess1 and query



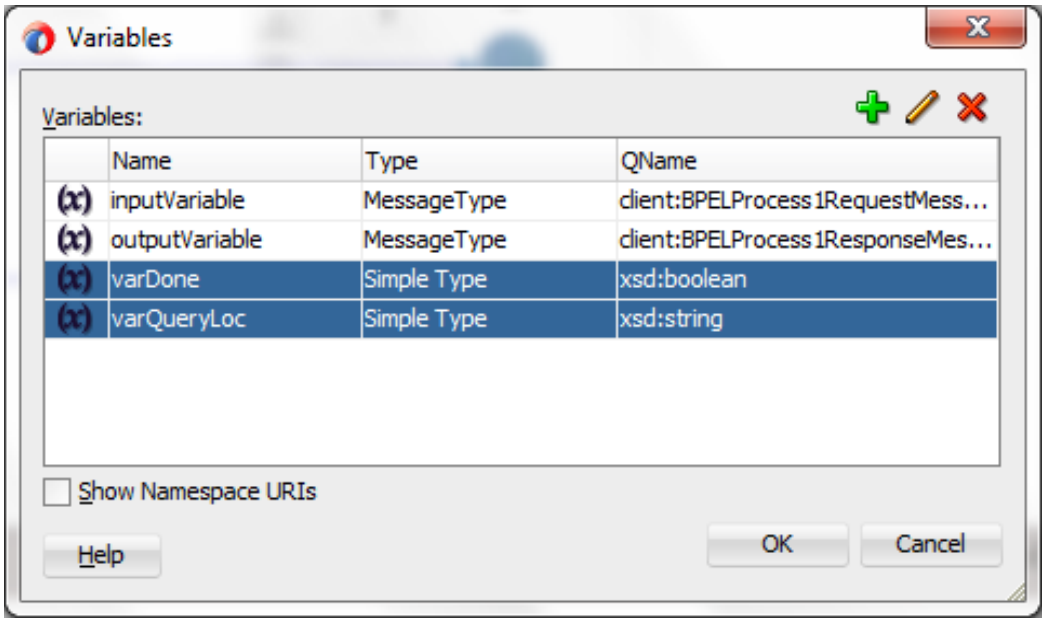
3. Double-click and open **BPELProcess1**. The **query** adapter should be present as part of Partner Links, as shown in Figure 11-70.

Figure 11 - 70 Open BPELProcess1



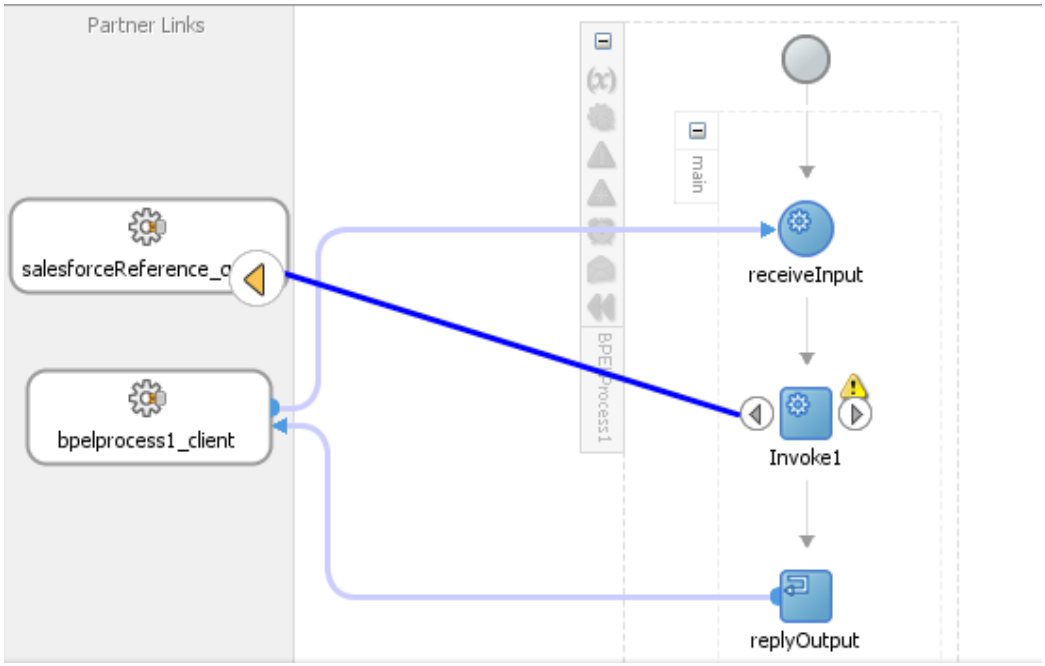
4. Create two variables **varDone** and **varQueryLoc** to track the values of "queryLocator" and "done" returned in query response from Salesforce.com, as shown in Figure 11-71.

Figure 11 - 71 Variables Screen



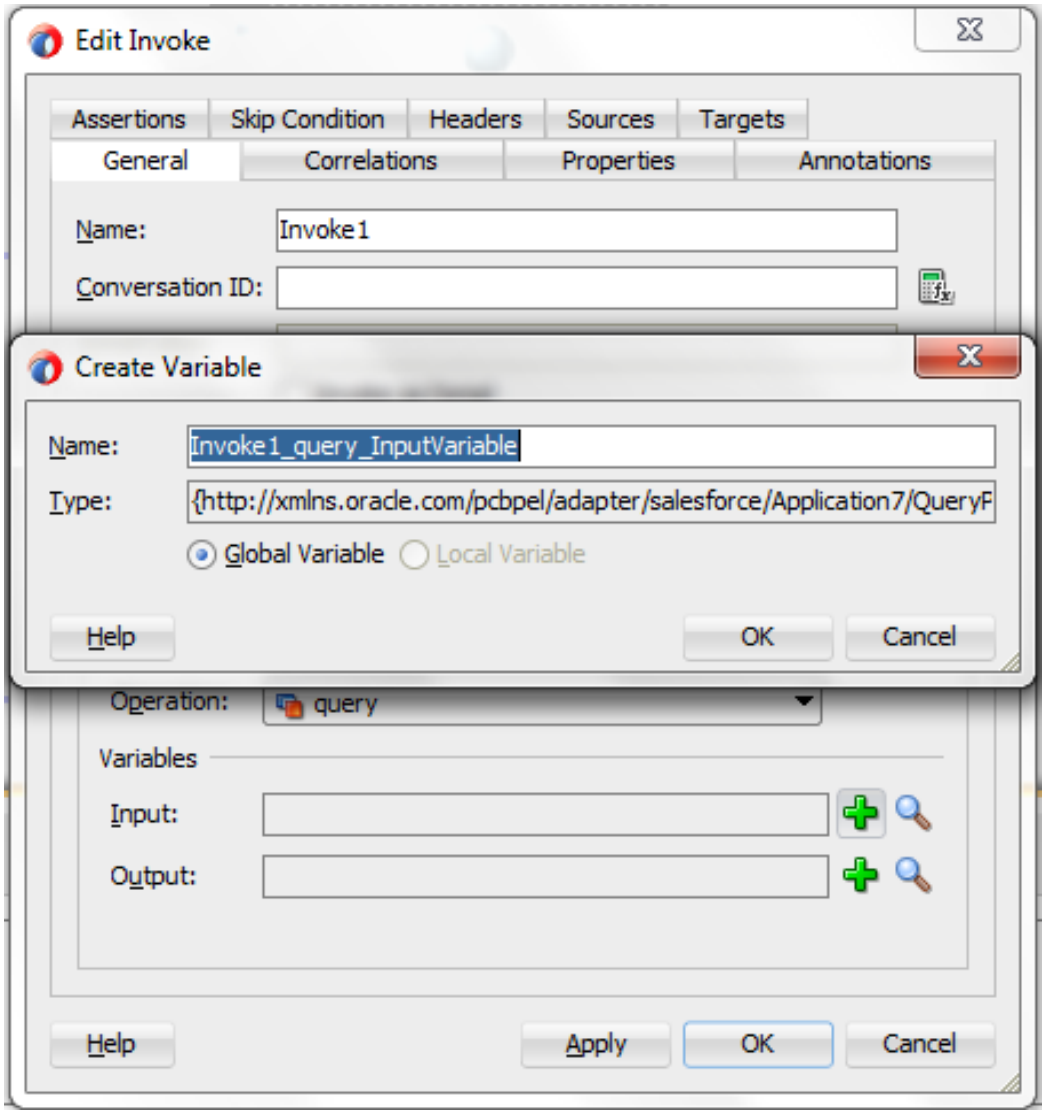
5. Add an **invoke activity** to invoke the query Partner Links, as shown in Figure 11-72.

Figure 11 - 72 Invoke activity



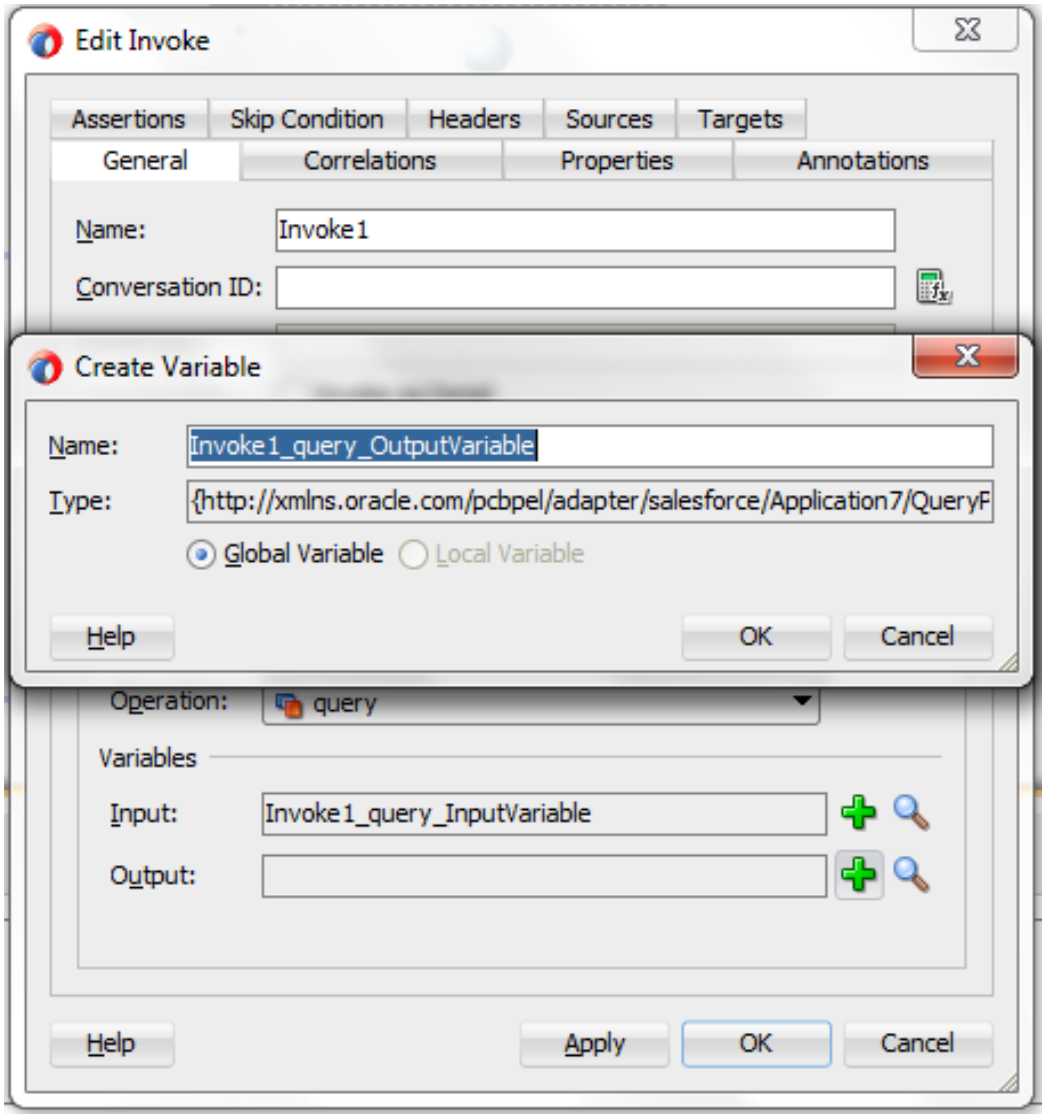
6. Create an input variable to the partner link by clicking the '+' button adjacent to **Input** text box in the **Variables** section. The **Create Variable** dialog is displayed, as shown in Figure 11-73.

Figure 11 - 73 Create Variable



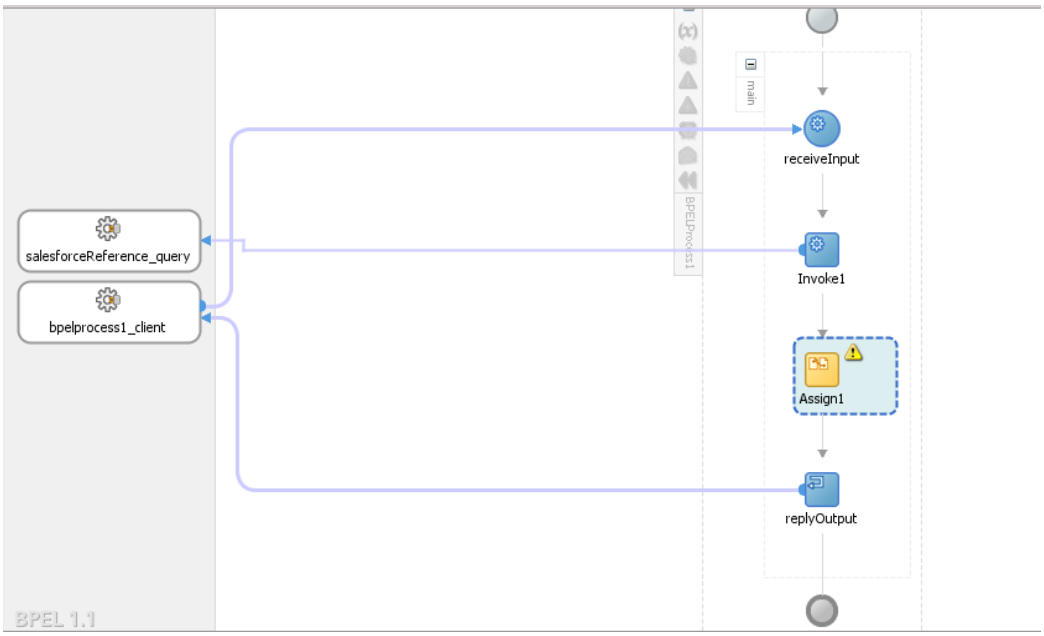
7. Create an output variable from the partner link by clicking the '+' button adjacent to **Output** text box in the **Variables** section. The **Create Variable** dialog is displayed, as shown in Figure 11-74.

Figure 11 - 74 Create Variable



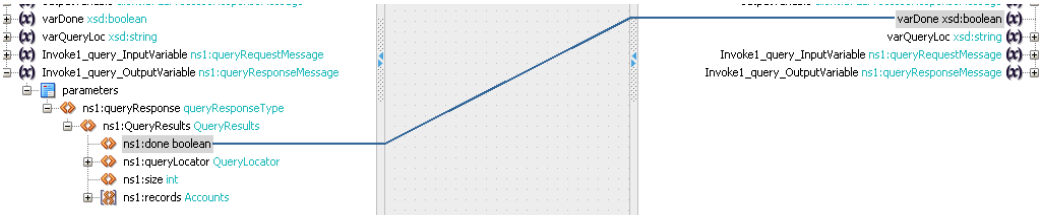
8. Introduce Assign activity right after the invoke activity, as shown in Figure 11-75.

Figure 11 - 75 Introduce Assign activity



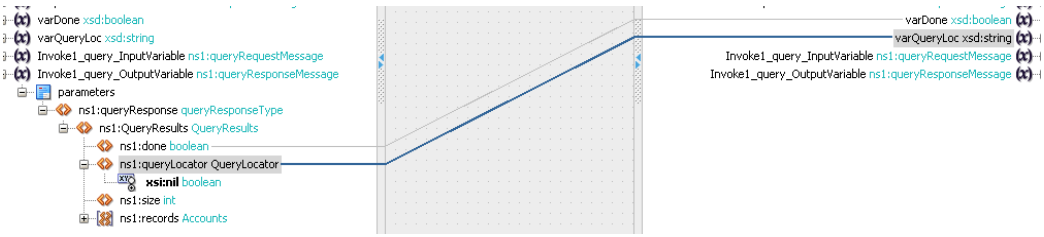
9. Inside the Assign activity, assign the value of "Invoke1_query_OutputVariable/done" variable to "varDone" variable, as shown in Figure 11-76.

Figure 11 - 76 Wire the Assign activity



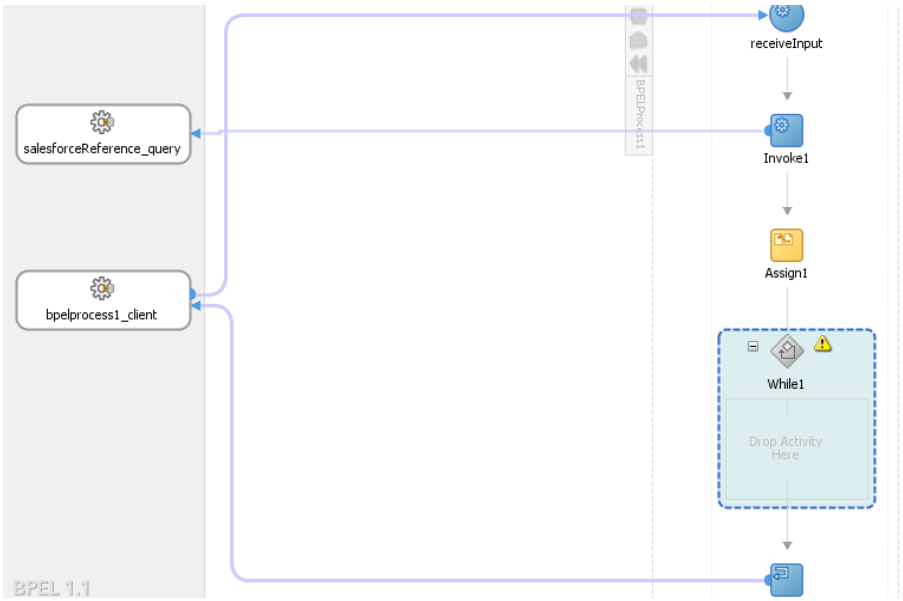
10. Next, assign the value of "Invoke1_query_OutputVariable/queryLocator" variable to "varQueryLoc" variable in the same Assign activity, as shown in Figure 11-77.
11. Click on **Apply** and then **OK**.

Figure 11 - 77 Wire the Assign activity



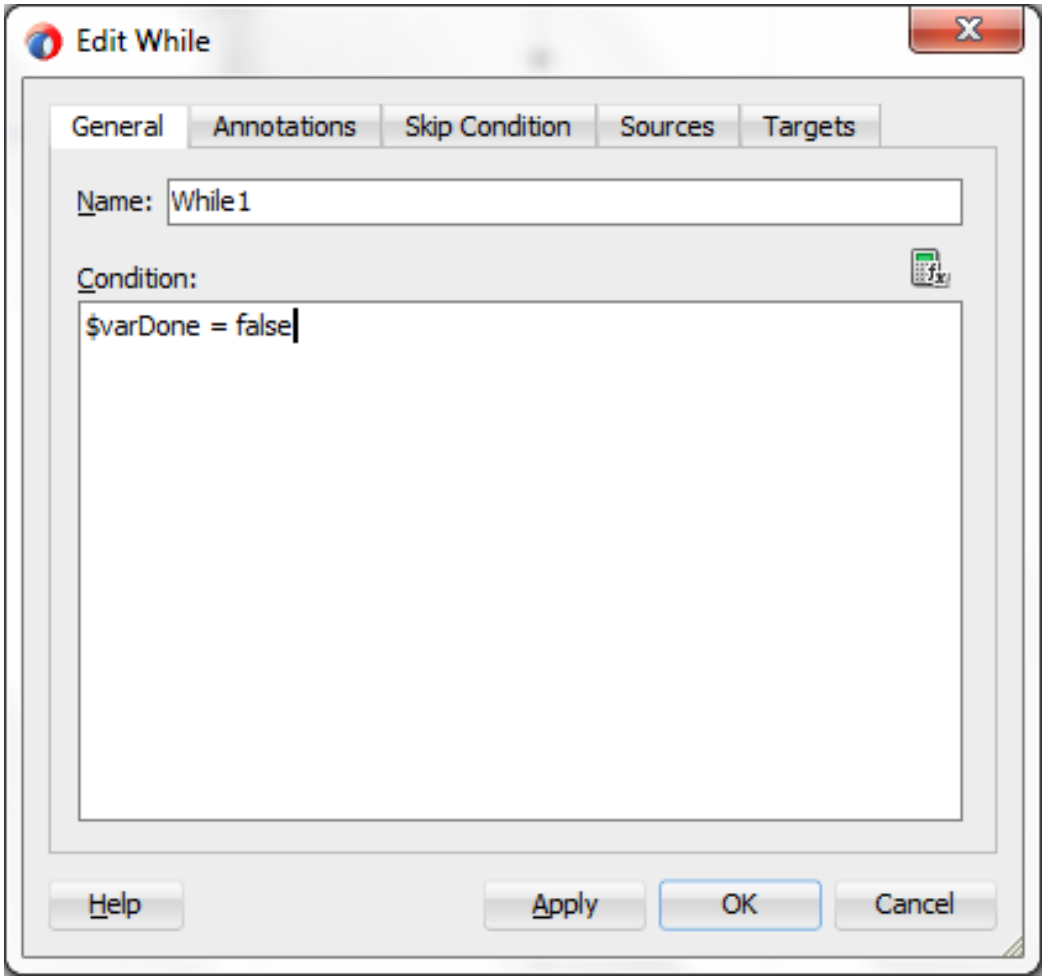
12. Add a While activity after the Assign activity, as shown in Figure 11-78. While activity is used to introduce iterations in the BPEL flow.

Figure 11 - 78 Add a While activity



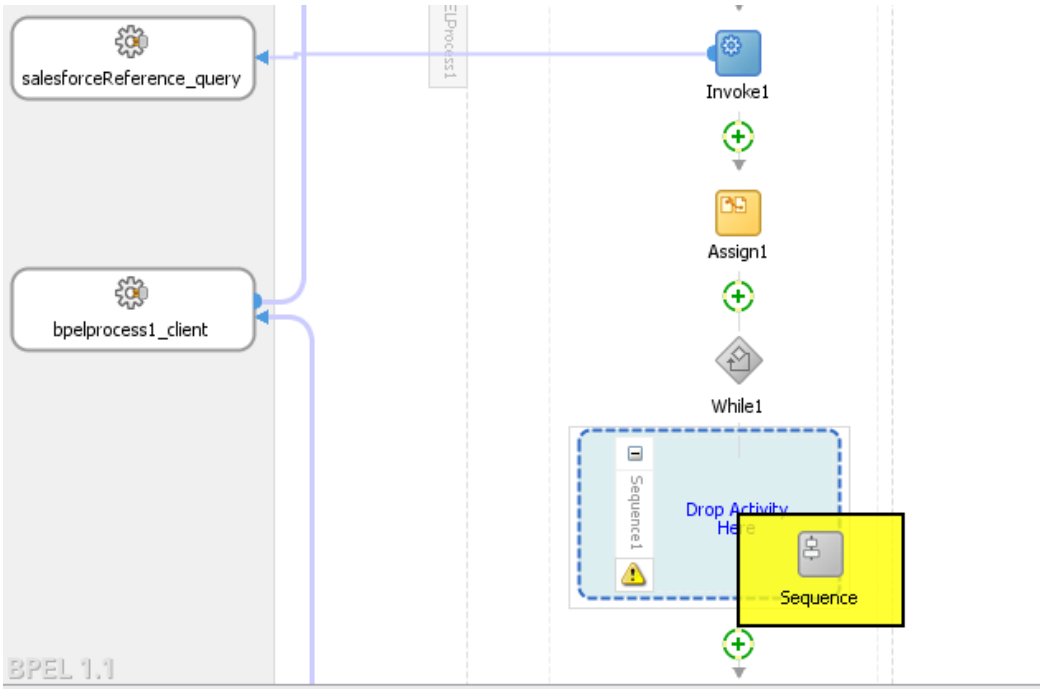
13. Double-click on the While Activity, a dialog box appears where you can specify condition for the while loop. Here, provide the condition, as shown in Figure 11-79.

Figure 11 - 79 Edit While activity



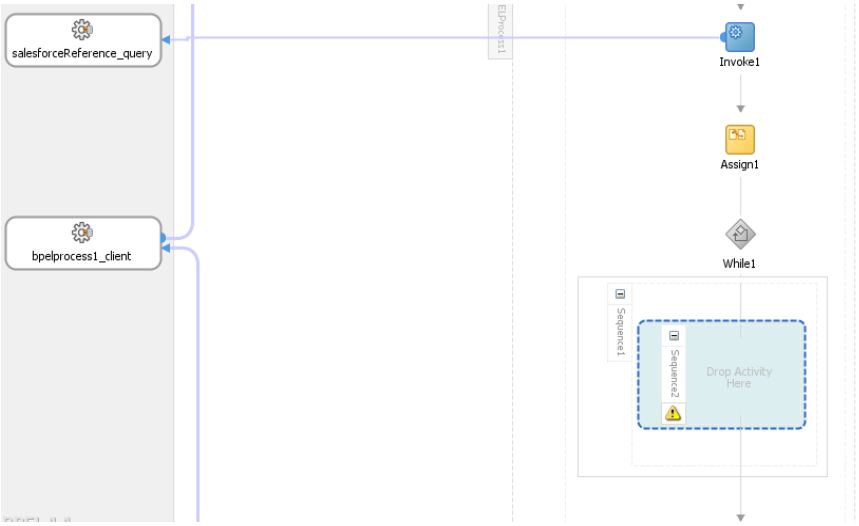
14. Click on **Apply** and then **OK**.
15. Drag and drop a **Sequence** Activity inside the **While**, as shown in Figure 11-80.

Figure 11 - 80 Add Sequence Activity



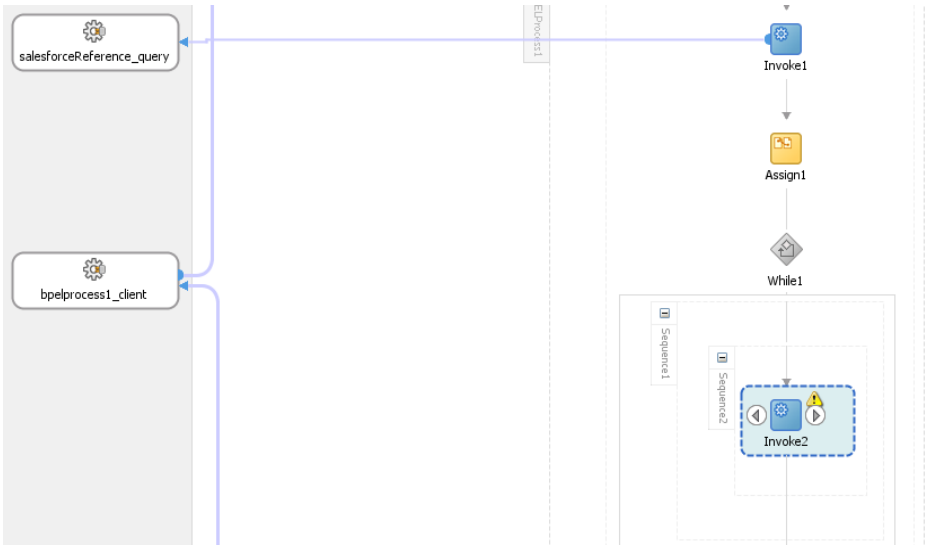
The While Activity will look like Figure 11-81.

Figure 11 - 81 While Activity



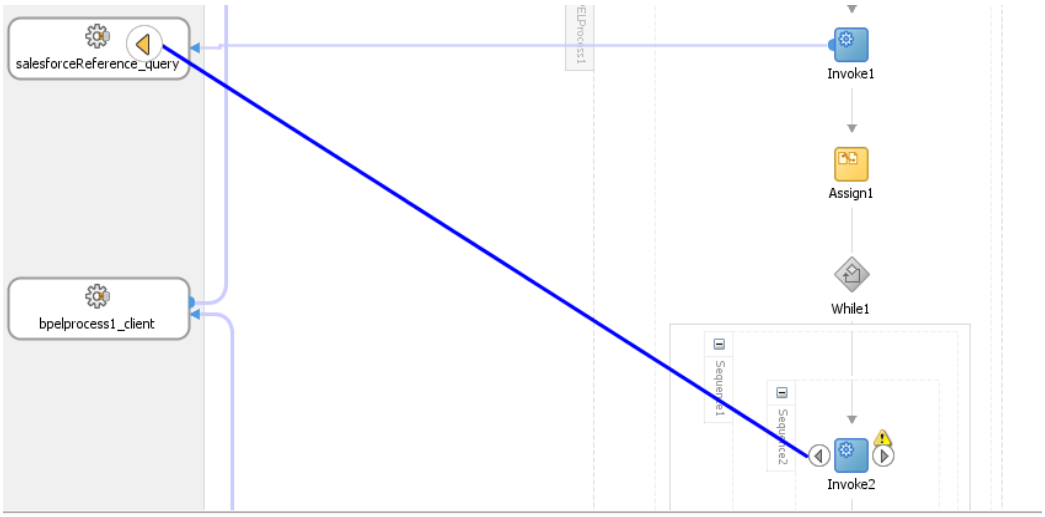
16. Drag and drop Invoke activity in the **Sequence**, as shown in Figure 11-82.

Figure 11 - 82 Add Invoke activity in the Sequence



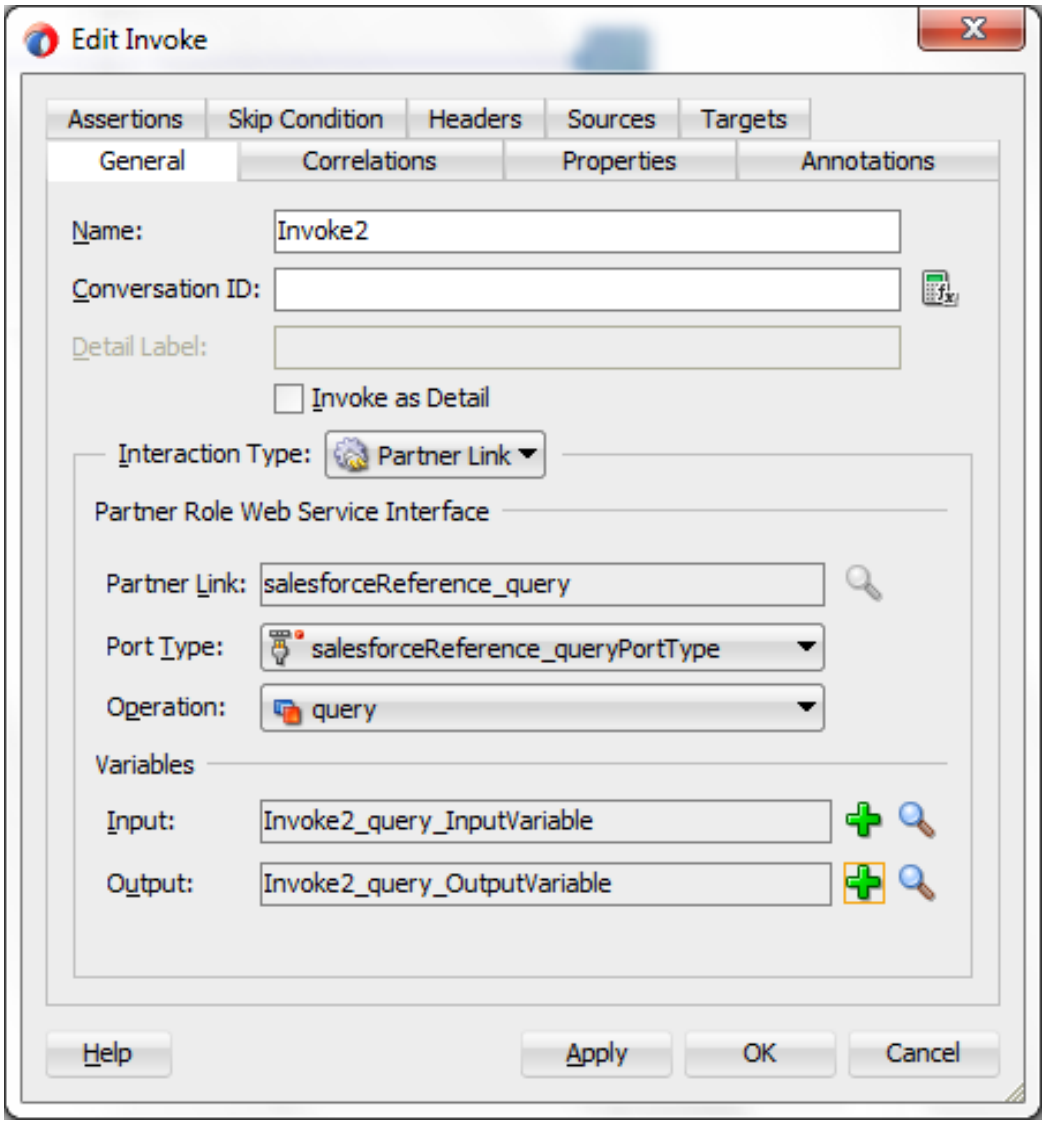
17. Wire this invoke activity to the partner link named “query”, as shown in Figure 11-83.

Figure 11 - 83 Wire the invoke activity



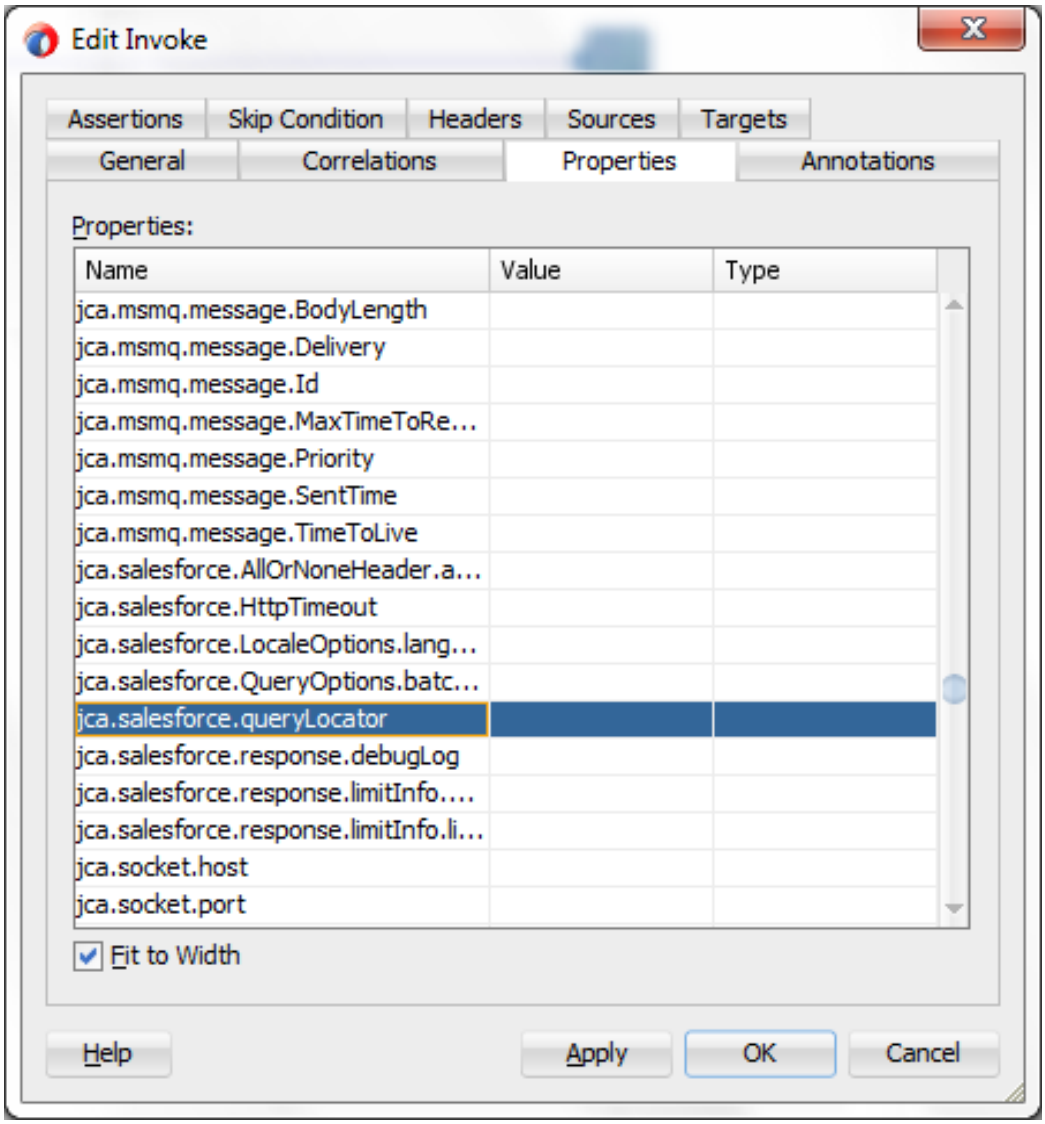
18. An “Edit Invoke” dialog will appear. Create input and output variables for this invoke activity by clicking on “+” sign, as shown in Figure 11-84.

Figure 11 - 84 Edit Invoke



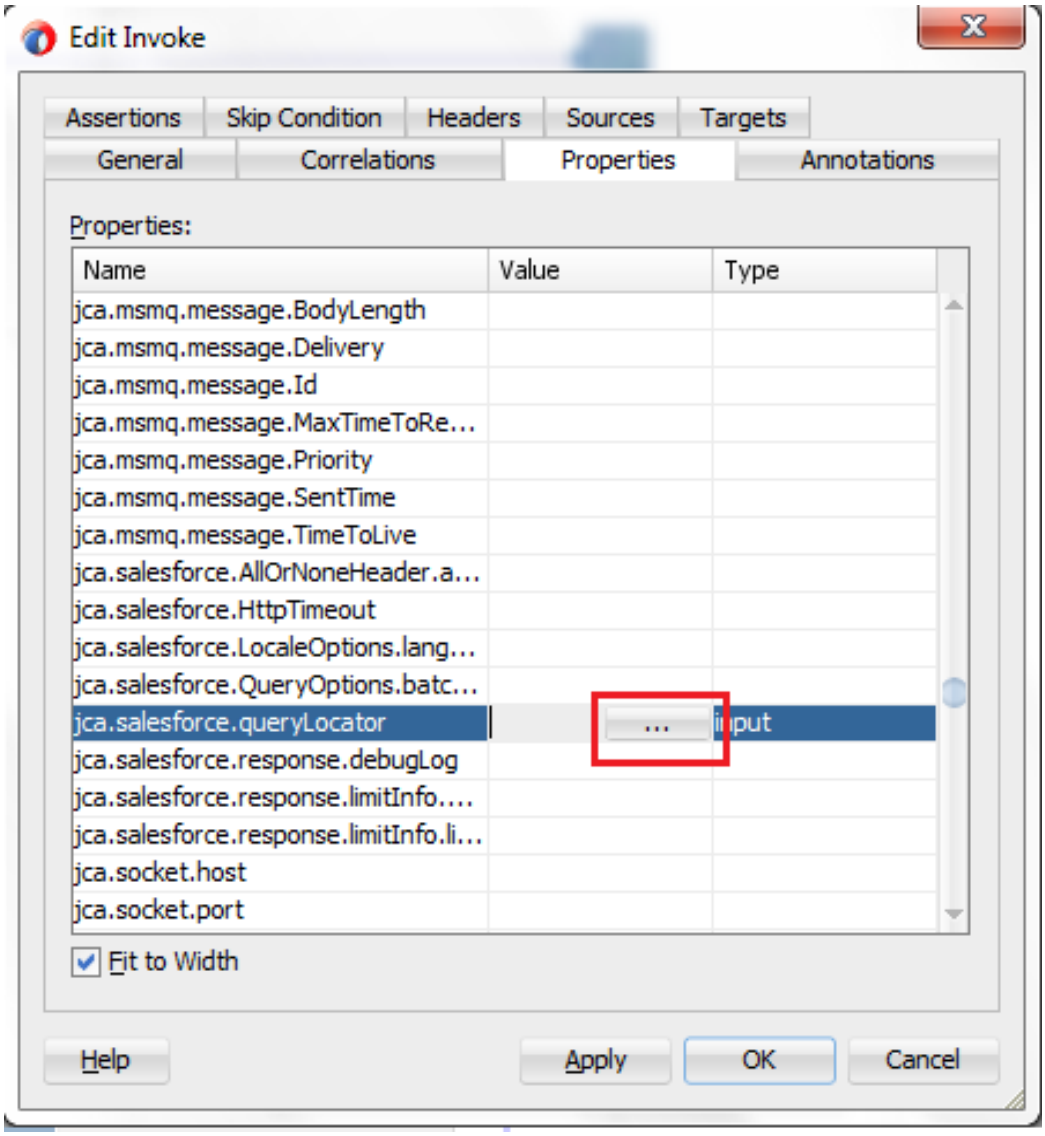
19. Click on **Properties** tab in the same dialog. Search for the property named "jca.salesforce.queryLocator", as shown in Figure 11-85.

Figure 11 - 85 Properties Tab



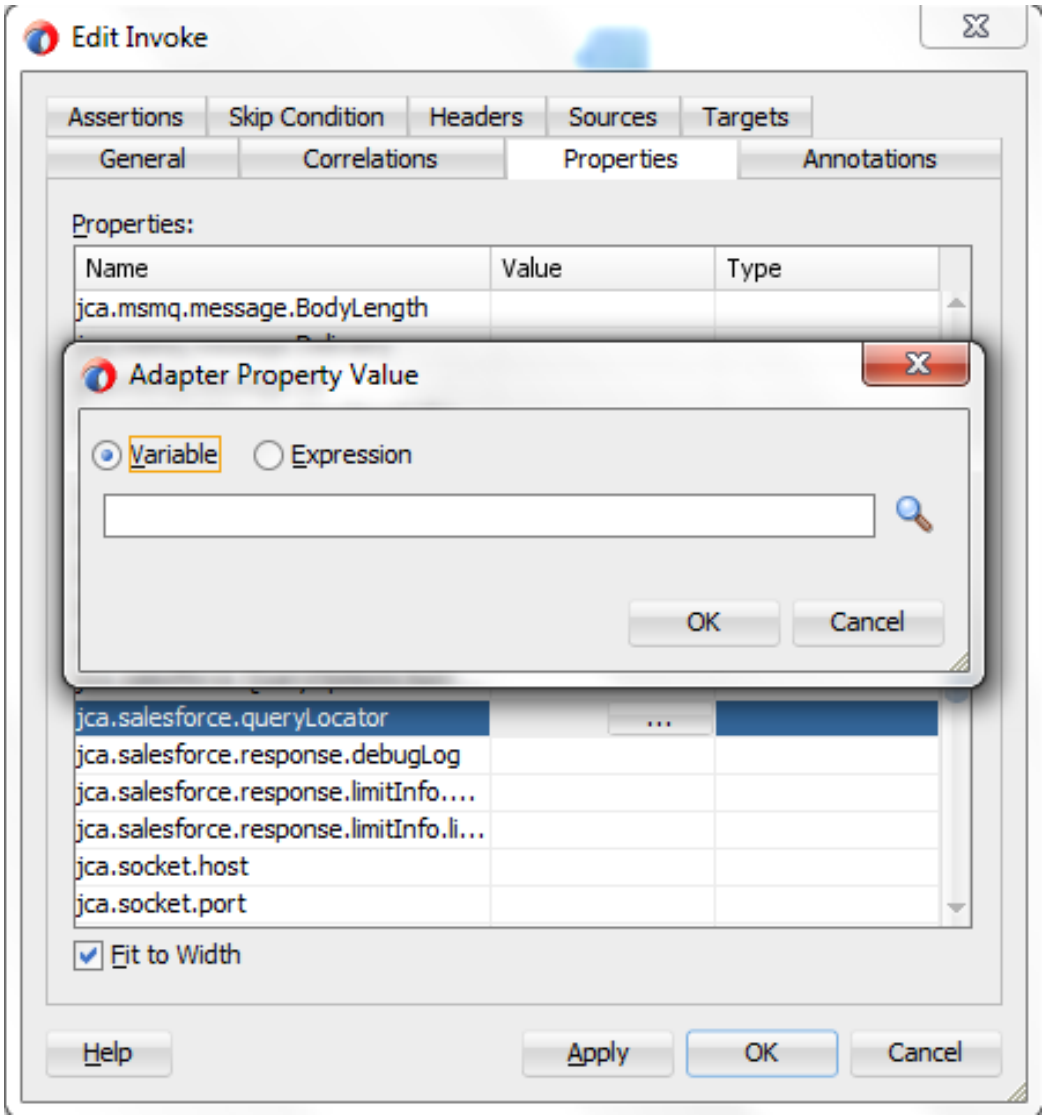
20. Browse for the variable by double-clicking on the button shown in Figure 11-86.

Figure 11 - 86 Browse for the variable



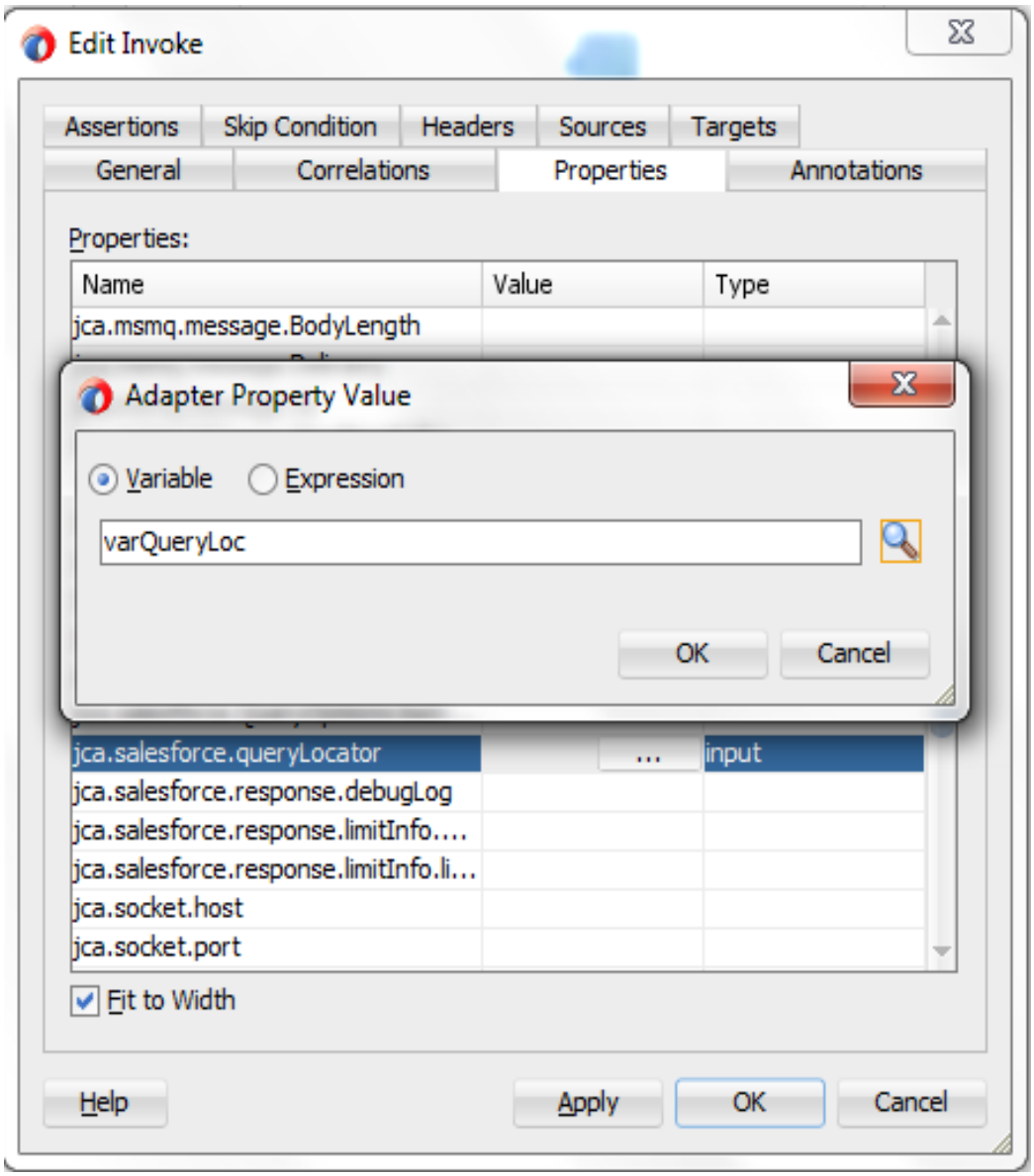
21. **Adapter Property Value** dialog box is displayed. Click on the **Search**, as shown in Figure 11-87.

Figure 11 - 87 Search Property Value



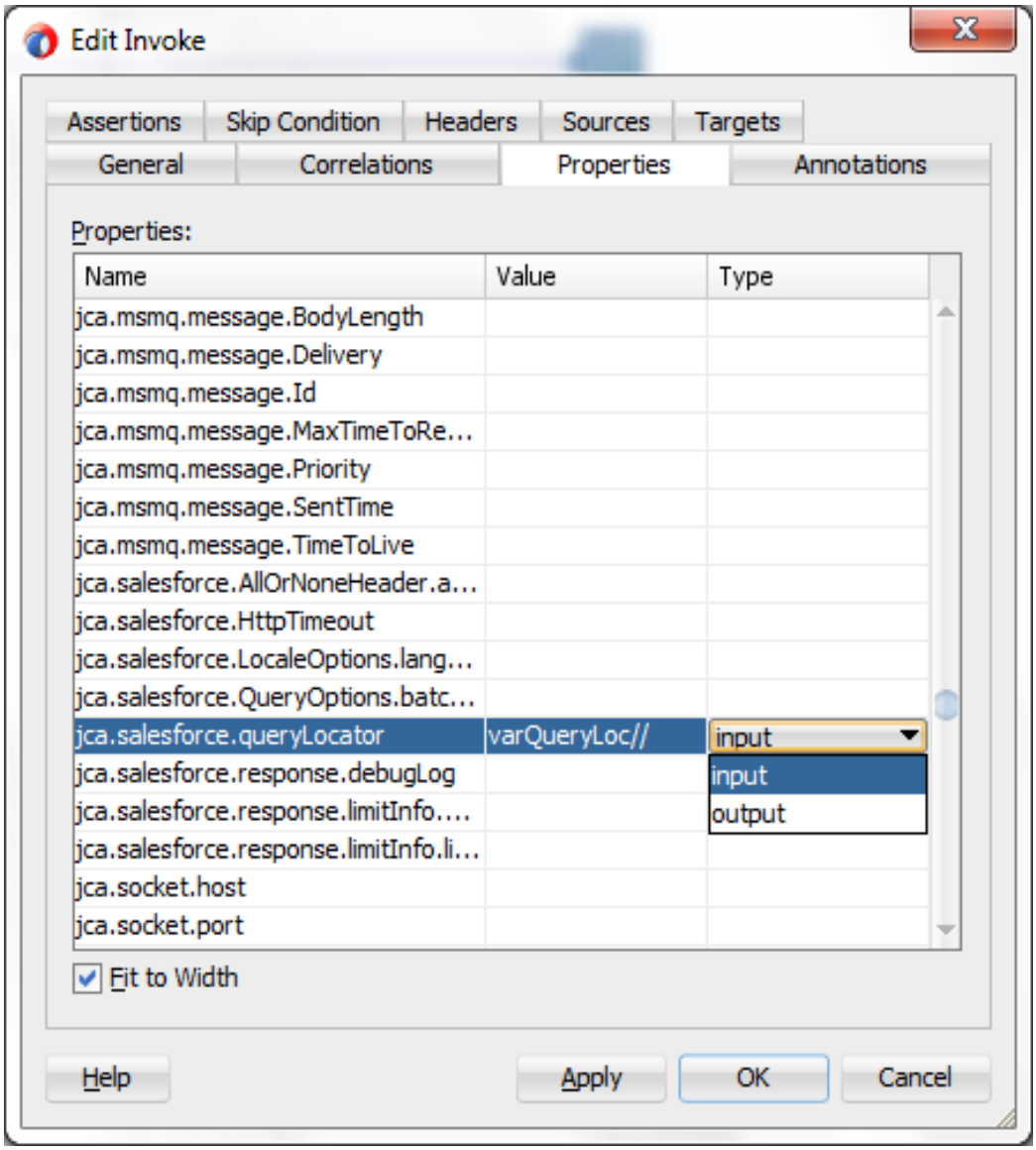
22. Select the variable **varQueryLoc** and click **OK**, as shown in Figure 11-88.

Figure 11 - 88 Select the variable varQueryLoc



23. Click on **OK** and select the **Type** as **input**, as shown in Figure 11-89.

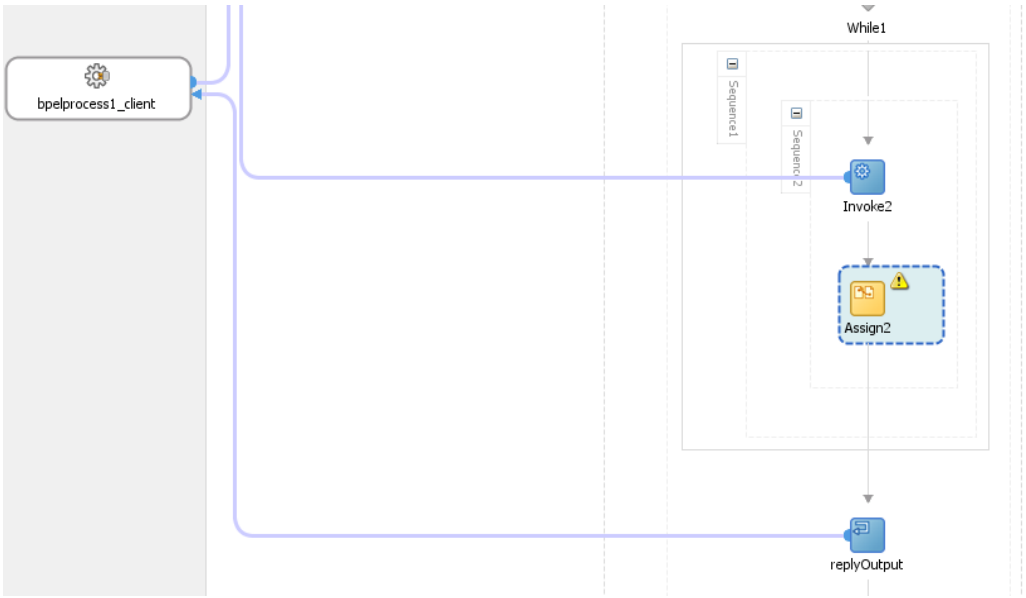
Figure 11 - 89 Select the Type as input



24. Click on **Apply** and then **OK**.

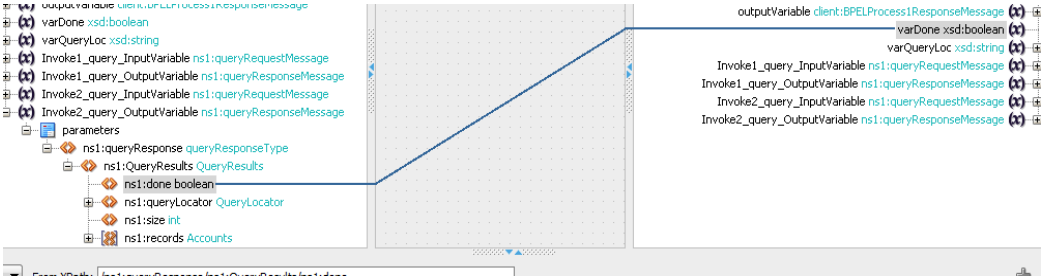
25. Drag and drop the **Assign** activity in the While Sequence, as shown in Figure 11-90.

Figure 11 - 90 Drag and drop Assign activity



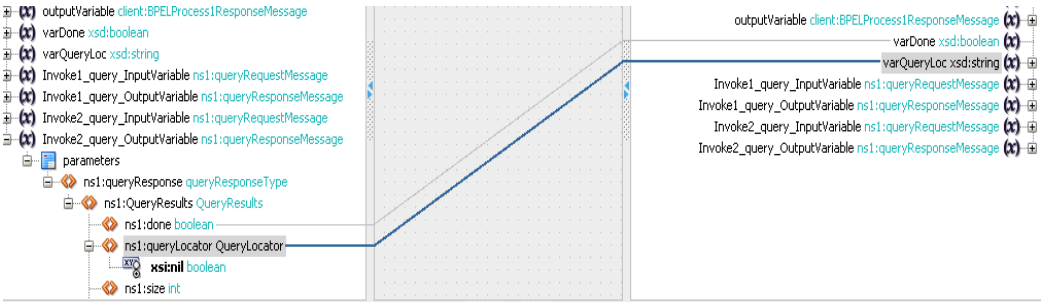
26. Double-click on **Assign activity**, and assign the value of "Invoke2_query_OutputVariable/done" variable to "varDone" variable, as shown in Figure 11-91.

Figure 11 - 91 Assign the Value



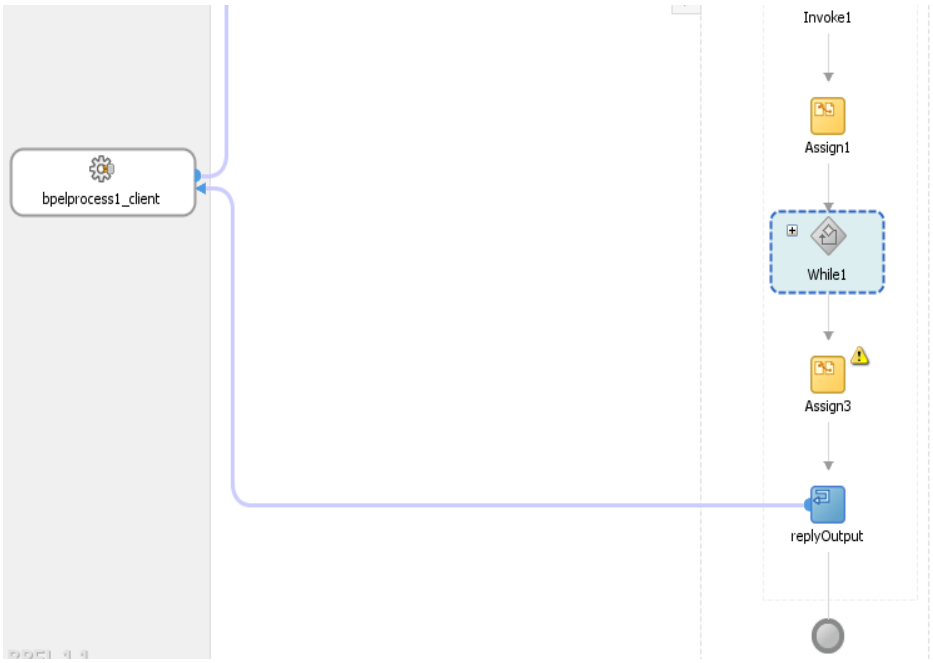
27. Assign the value of "Invoke2_query_OutputVariable/queryLocator" variable to "varQueryLoc" variable, as shown in Figure 11-92.

Figure 11 - 92 Assign the Value



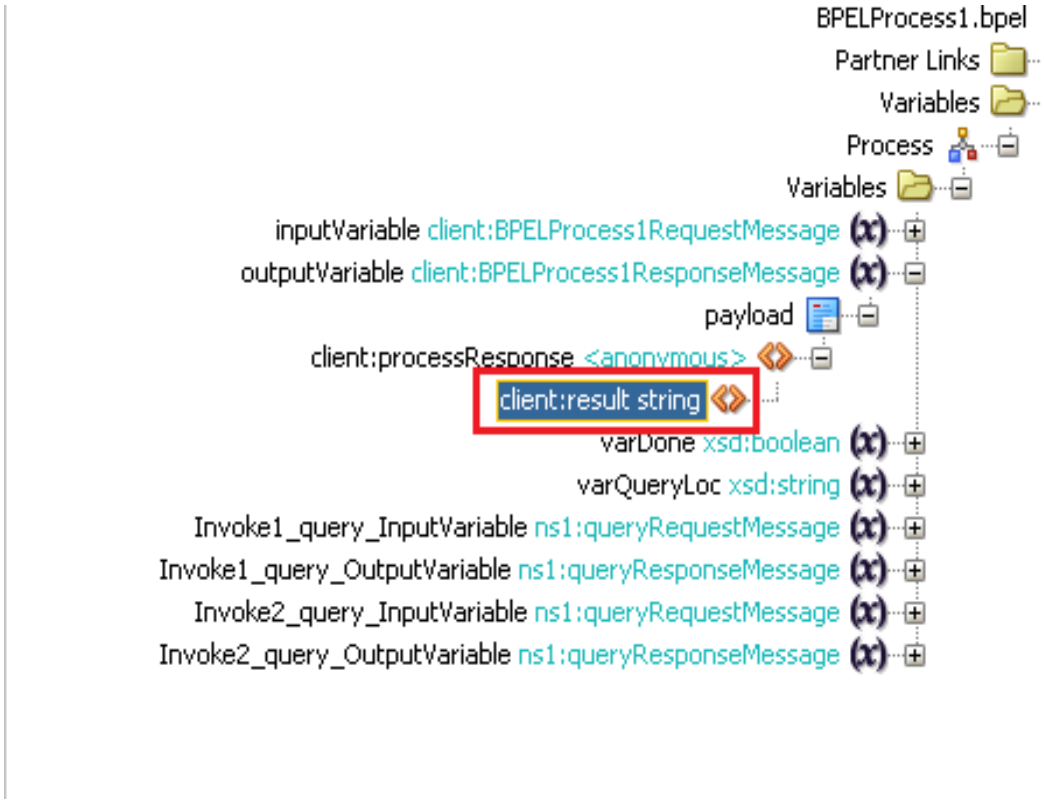
28. Click on **Apply** and then **OK**.
29. Drag and drop **Assign** activity right after the **While** activity, as shown in Figure 11-93.

Figure 11 - 93 Drag and drop Assign activity



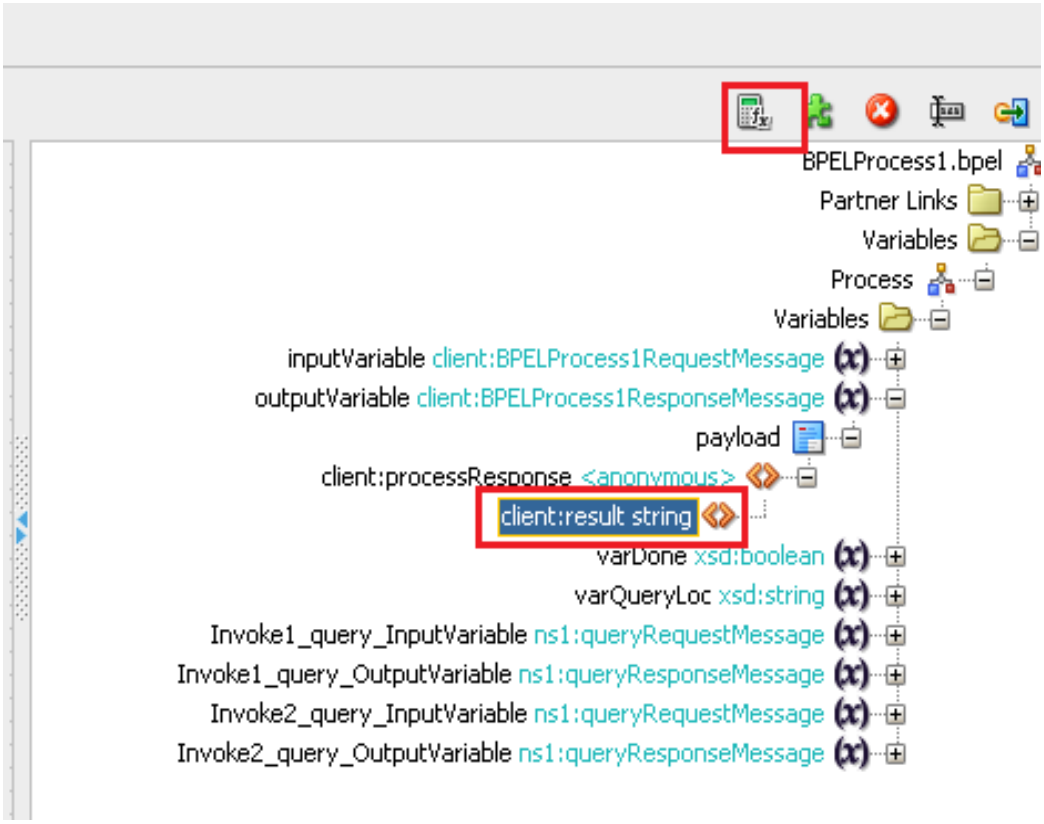
30. Double click on **Assign** activity and look for the variable **processresponse/result**, as shown in Figure 11-94.

Figure 11 - 94 Double click on Assign activity



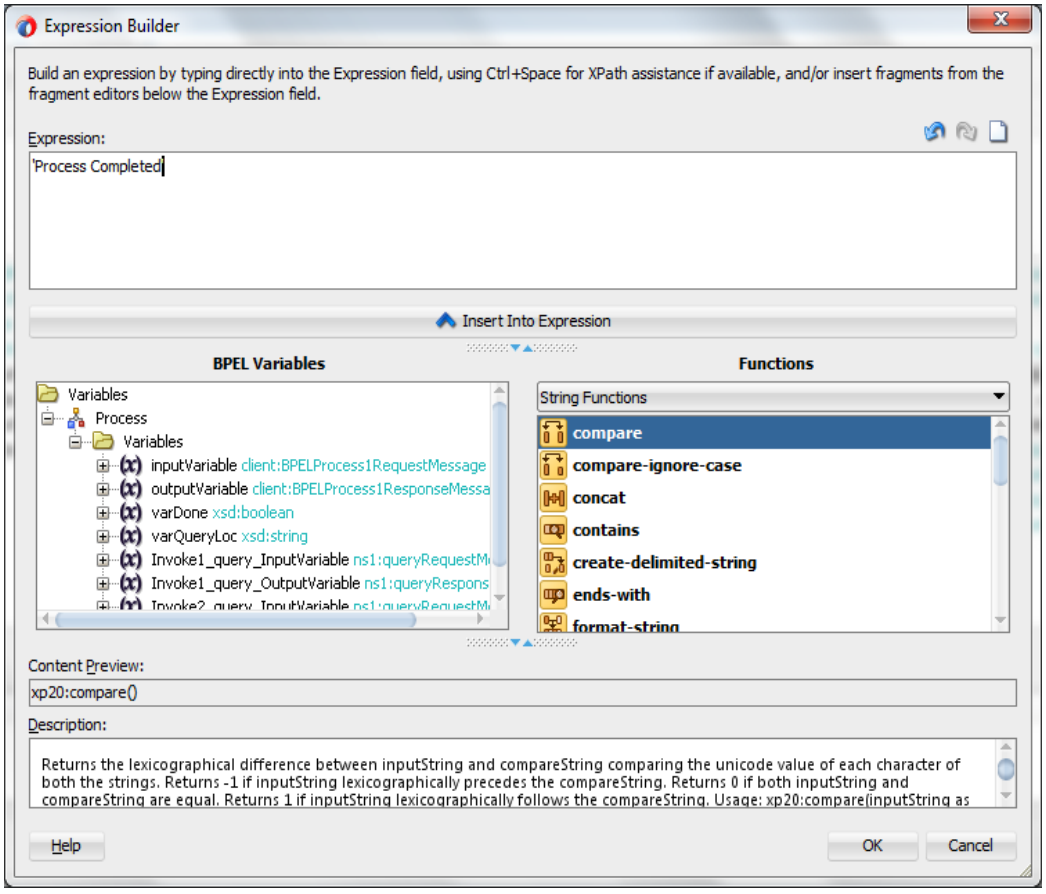
31. Drag and drop the Expression over to this variable, as shown in Figure 11-95.

Figure 11 - 95 Drag and drop the Expression over to this variable



32. **Expression Builder** dialog box is displayed. Provide an expression value, as shown in Figure 11-96.

Figure 11 - 96 Expression Builder Dialog



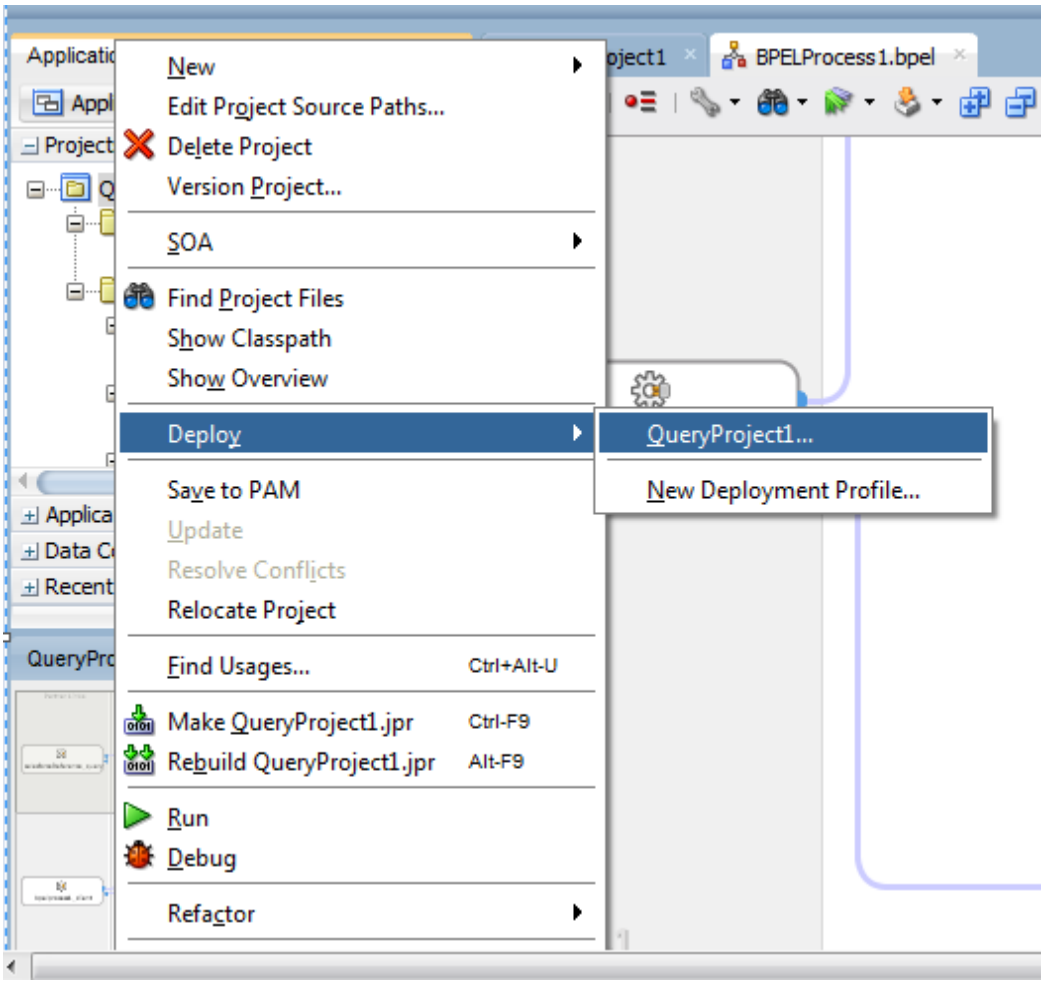
33. Click **OK**.
34. In the Assign activity, click on **Apply** and then **OK.**

Deploy the Composite

Perform the following steps to deploy the composite.

1. In the **Application Navigator** pane, right-click **Project1** and select **Deploy -> Project1**, as shown in Figure 11-97.

Figure 11 - 97 Navigator Screen



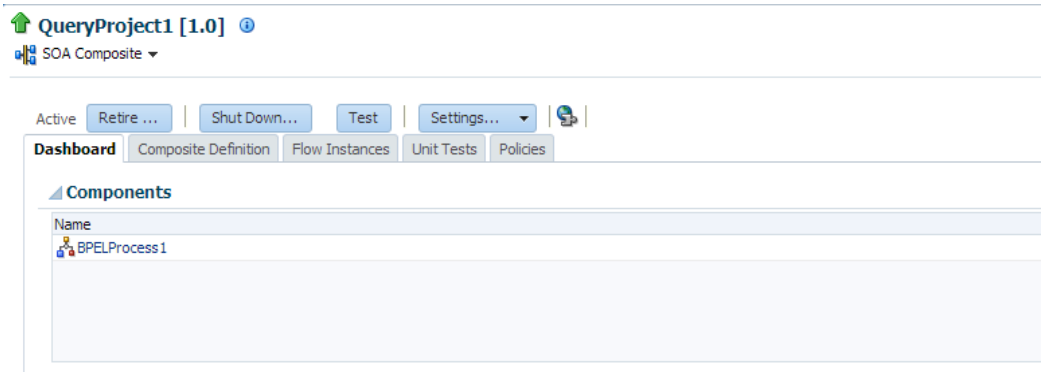
2. Select the **Deploy to Application Server** option and follow the instructions displayed on the screen. Using this option you can deploy the composite on the Application server after providing the details of the server.

Test the Composite

Perform the following process to test the composite.

1. Login to Enterprise Manager Console for the server you deployed your project on.
2. Open Q**ueryProject1**
3. Click the **Test** button to test the Web service, as shown in Figure 11-98.

Figure 11 - 98 Test Tab



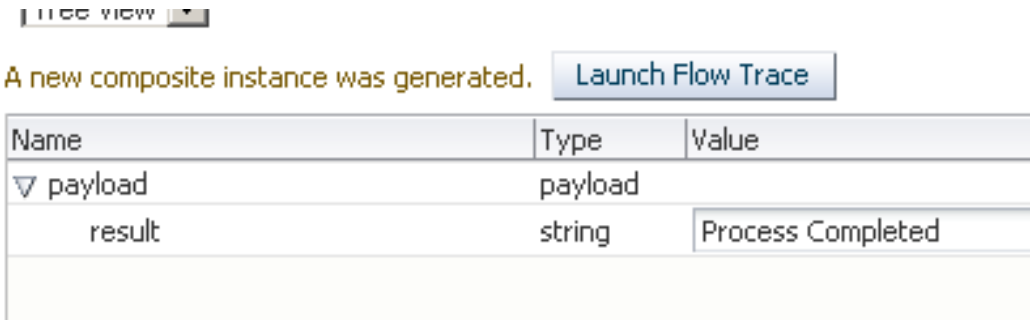
Since we are not mapping the input variable anywhere, you can provide any random input here.

4. Click on **Test Web Service** button, as shown in Figure 11-99.

Figure 11 - 99 Test Web Service

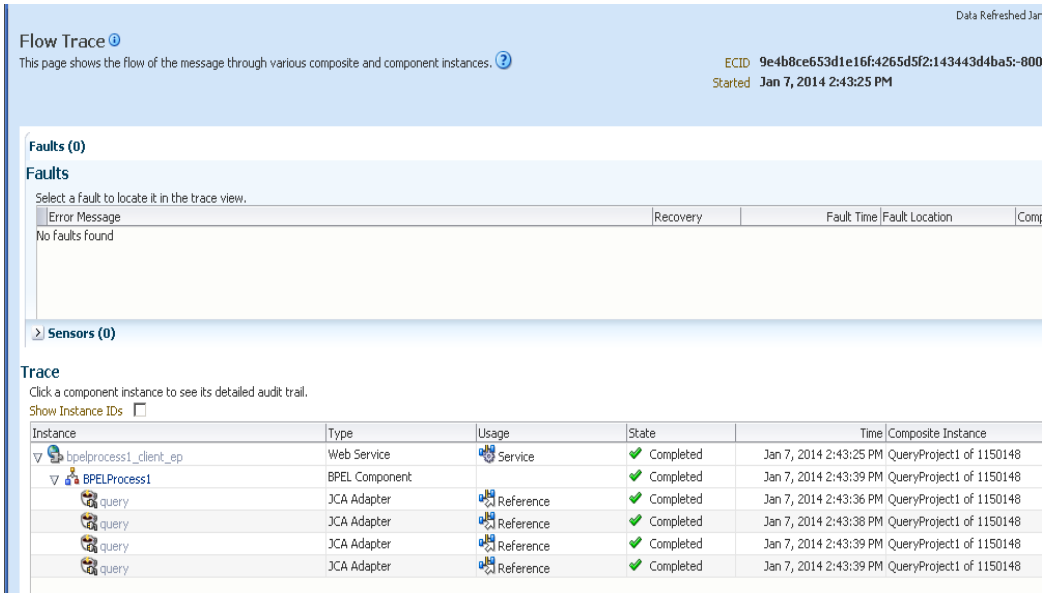
5. After successful execution the response contains the expression that you provided in the last assign activity, as shown in Figure 11-100.

Figure 11 - 100 Execution Window



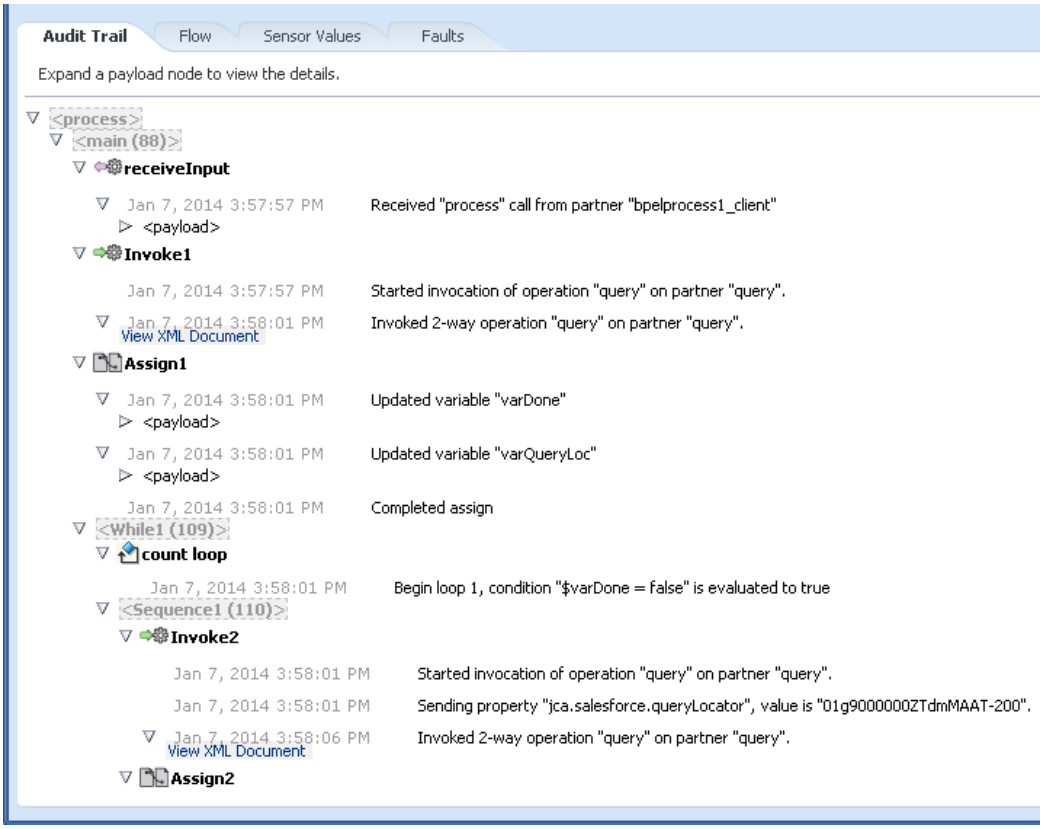
6. Click the **Launch Flow Trace** button to view the **Audit Trail**, as shown in Figure 11-101.

Figure 11 - 101 Launch Flow Trace



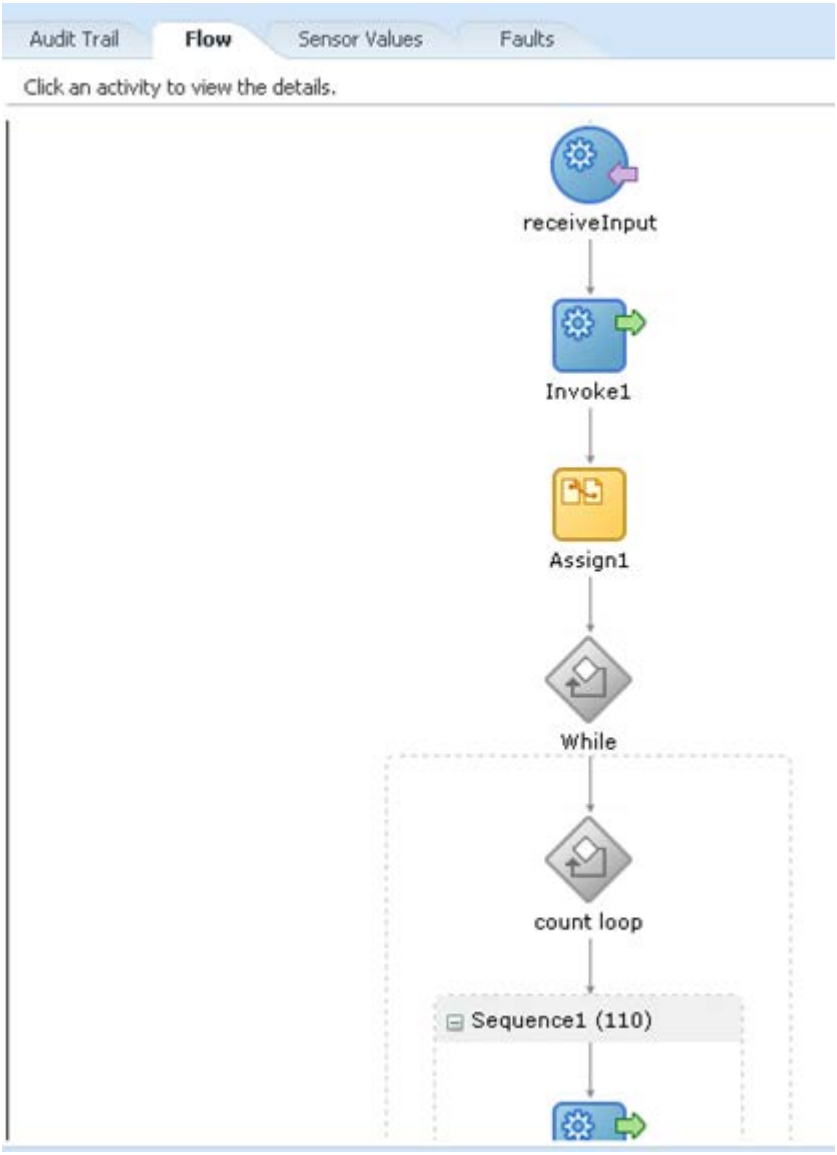
7. The **Audit Trail** will look like Figure 11-102.

Figure 11 - 102 Audit Trail



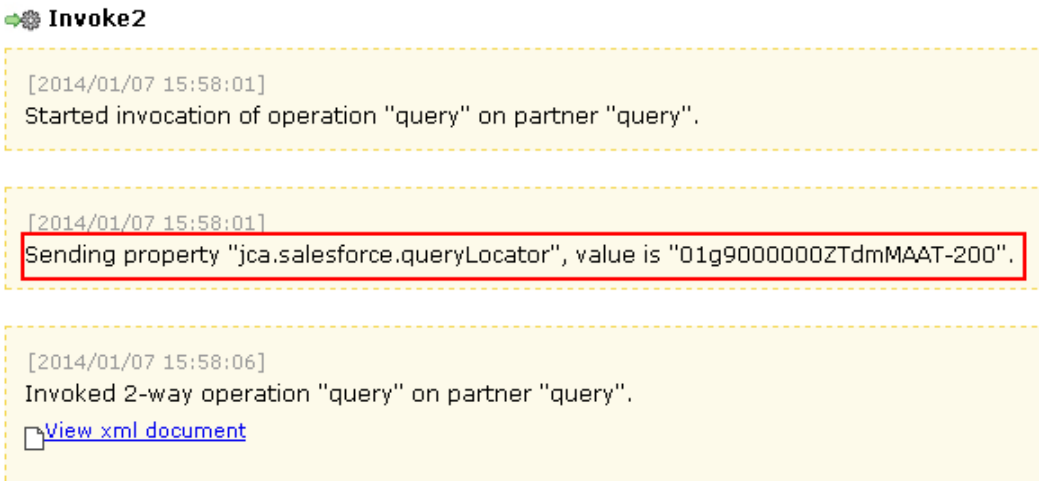
8. Click on **Flow** tab, The Flow tab will look like Figure 11-103.

Figure 11 - 103 Flow Tab



9. Click on any invoke activity inside the While loop, here you can see how the value of queryLocator is being passed, as shown in Figure 11-104.

Figure 11 - 104 Invoke 2



11.3. Creating Integration using Oracle Cloud Adapter for Salesforce.com without internet connection or when Salesforce.com is down.

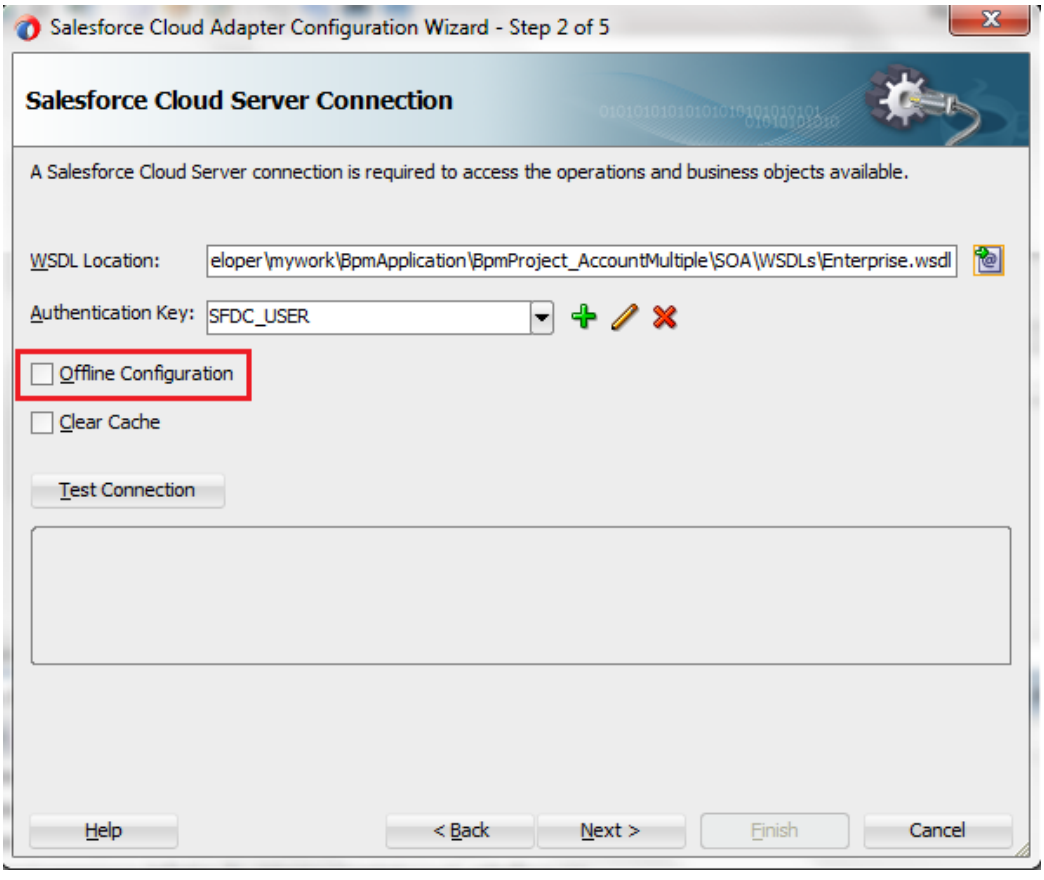
Overview:

The following scenario walks you through a simple integration wherein user creates a new Account on Salesforce.com using Oracle Cloud Adapter for Salesforce.com using the offline configuration capability of the Oracle Cloud Adapter for Salesforce.com. This method of adapter configuration can be used in case you do not have access to the internet while creating SOA composites or if Salesforce.com is down due to some reason. It can also be used to save unnecessary login calls to Salesforce.com while creating the composites.

Configuring the Oracle Cloud Adapter for Salesforce.com in Offline Mode:

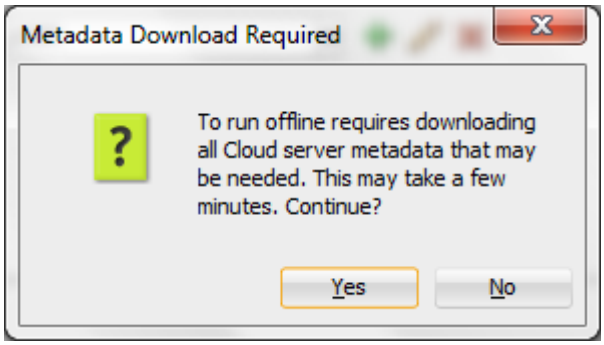
1. Follow steps 10 to 20 of the section "Creating the BPEL Process" of chapter 5-Integration with Different Service Components of this guide".
2. Click on the **Offline Configuration** checkbox, as shown in Figure 11-105.

Figure 11 - 105 Offline Configuration checkbox



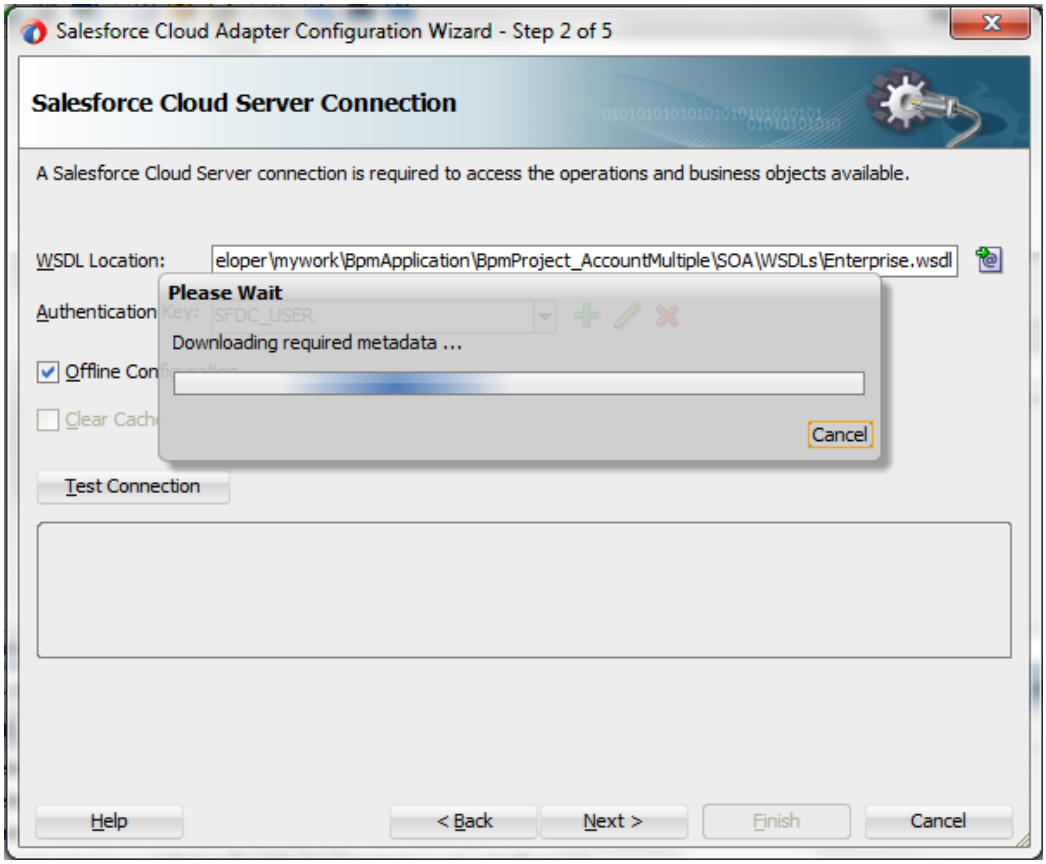
3. Click **Yes** to download the metadata to cache, as shown in Figure 11-106.

Figure 11 - 106 Metadata Dialog Box



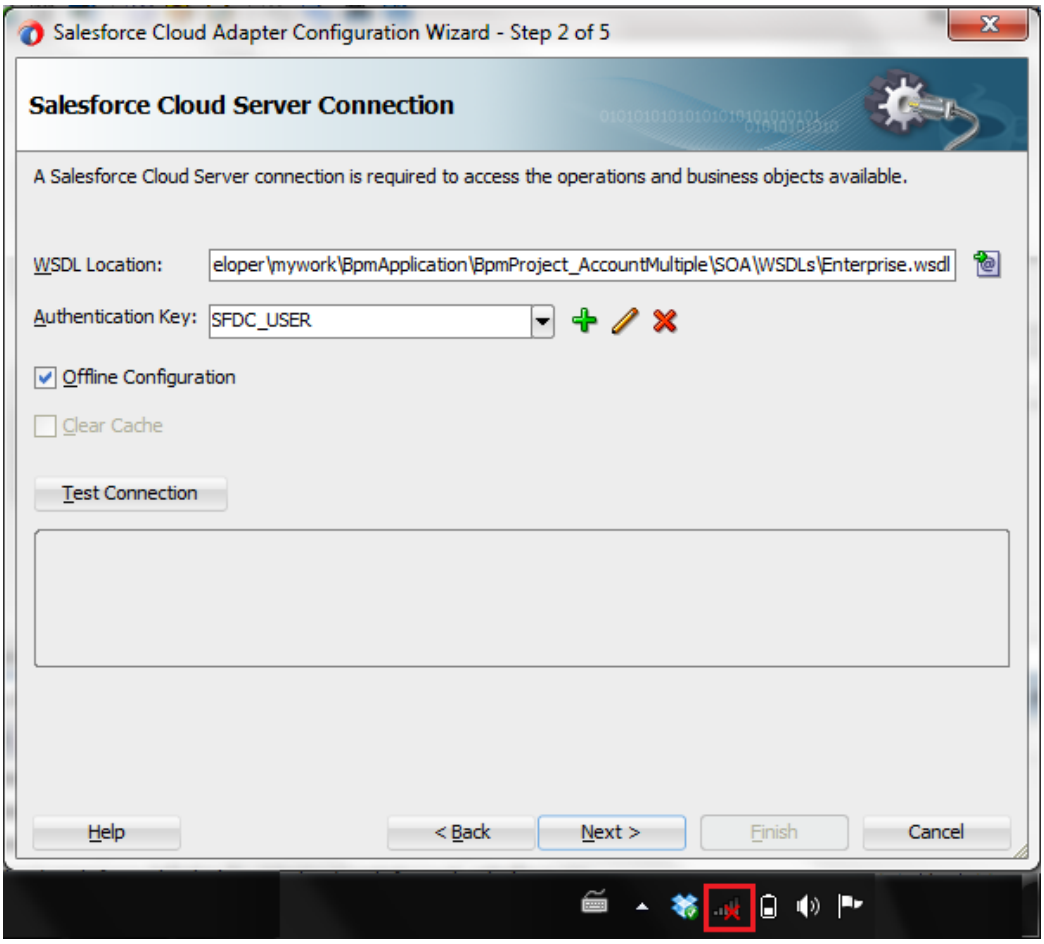
4. Wait while the metadata is downloaded and stored in cache, as shown in Figure 11-107.

Figure 11 - 107 metadata is downloaded



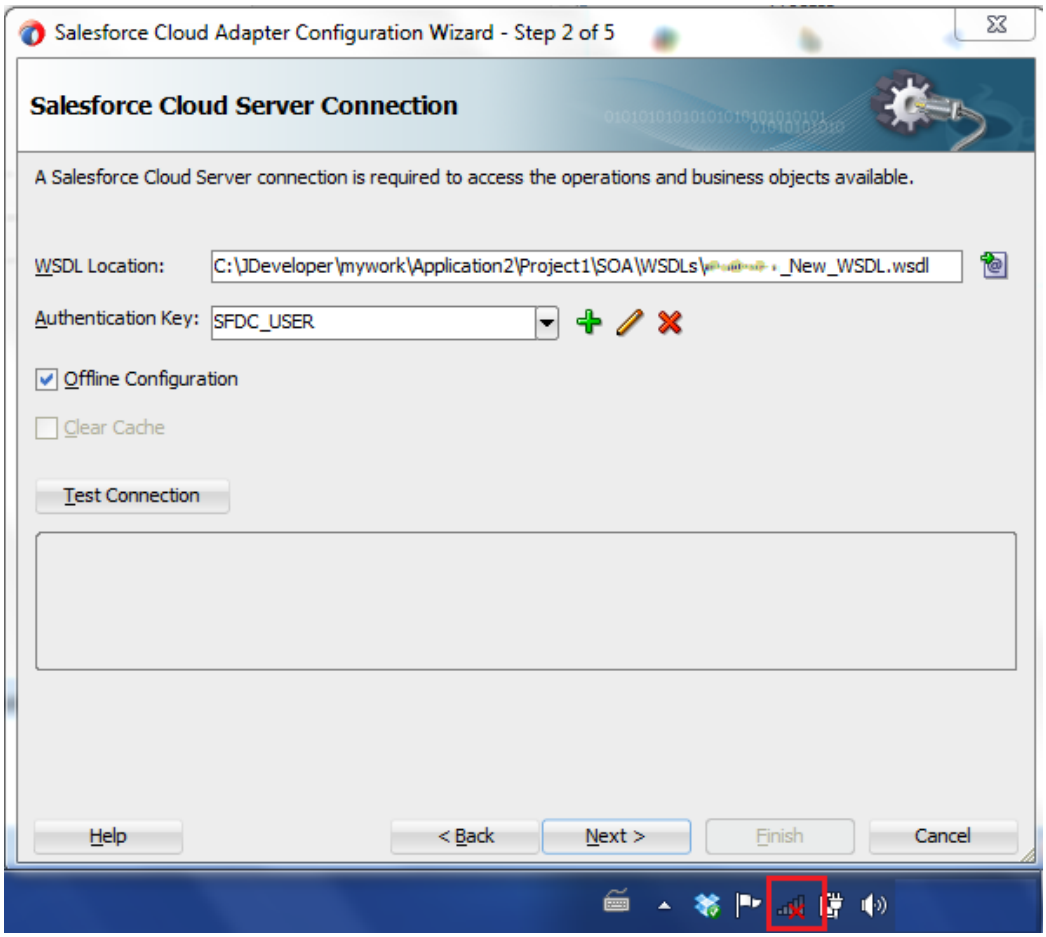
5. Click **Next**. [Notice that the user is disconnected from the internet (marked by red box)], as shown in Figure 11-108.

Figure 11 - 108 User is disconnected from the internet



6. The adapter configuration wizard moves to Cloud Operation Configuration page, as shown in Figure 11-109.

Figure 11 - 109 Cloud Operation Configuration page



7. Follow step 24 to 46 of the section "Creating the BPEL Process".

11.4. How to use MDS for importing WSDL into JDeveloper?

Oracle Metadata Services (MDS) provides a unified store for Metadata and ensures reliable access to metadata for fusion middleware artifacts like XSD, WSDL, etc. This Use Case intends to make you familiar with how Enterprise WSDL placed in MDS can be used in Oracle cloud adapter for Salesforce.com. It first provides a brief introduction about placing the Enterprise WSDL in MDS and then explains you how this file can be used in the Oracle Cloud Adapter for Salesforce.com.

Steps to Place Enterprise WSDL in MDS:

1. Create SOA Application.
2. Create an MDS connection.
3. Transfer artifacts to MDS.

Create SOA Application

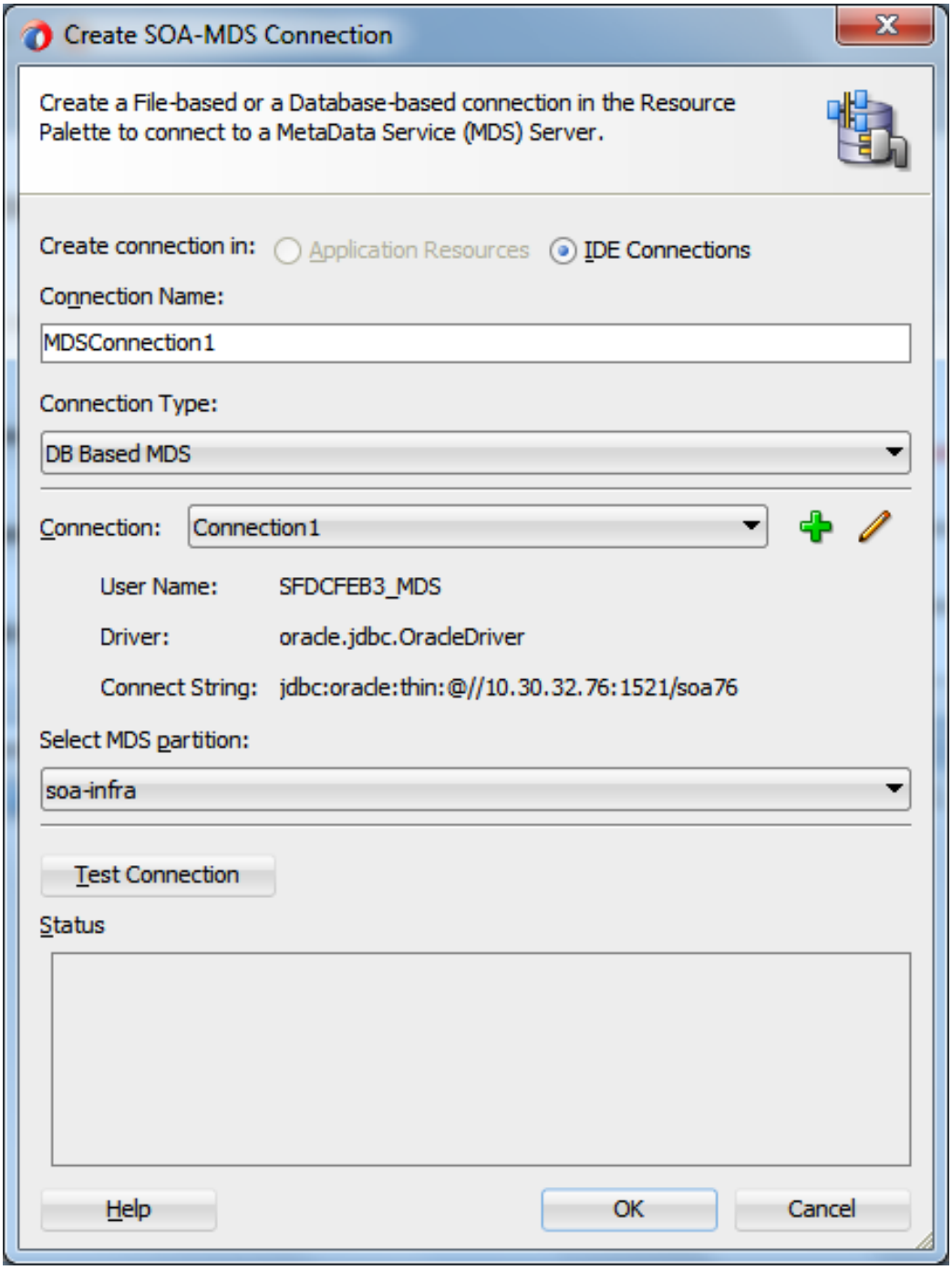
Follow the steps from 1-32 of section 11.1.1 Creating the BPEL Process for the use case 11.1 How can I build integration in which the SOA client is not interested in the response?

Create an MDS connection.

Now you need to create a new MDS connection for you application.

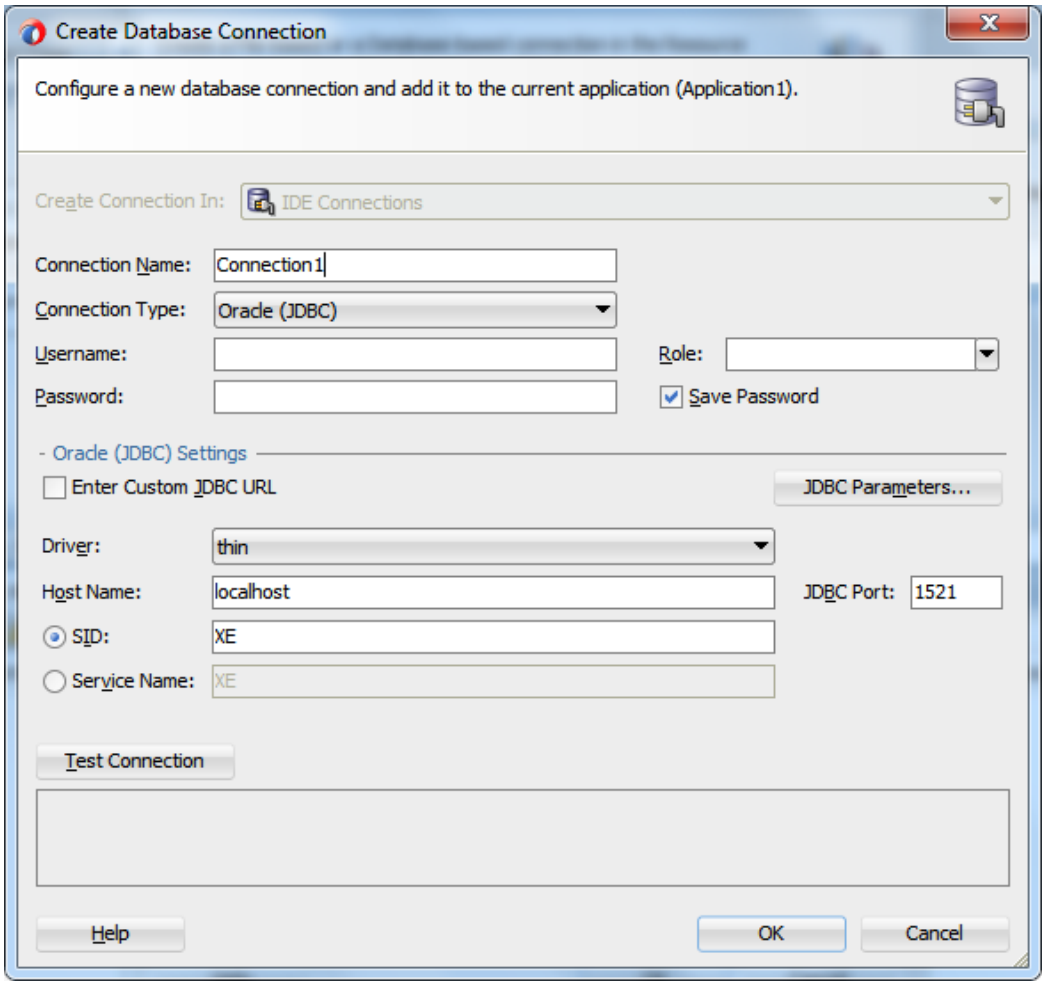
1. For this, press Ctrl+N and search for SOA-MDS Connection option and select it, a new window will pop up asking for MDS connection details Popup window has been shown in Figure 11-110.

Figure 11 - 110 Create SOA-MDS Connection



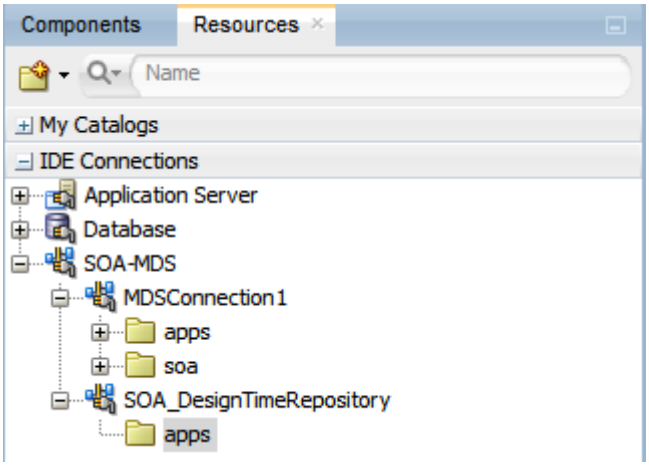
2. For this Use Case we will be using the connection type as DB Based MDS. To enter DB connection details, please click on "+". You will see the Create database connection pop-up as shown in Figure 11-111. Please enter the connection details as per your installation and DB configuration of the SOA SUITE.

Figure 11-111 Create Database Connection



3. Once the connection has been established with MDS, under the Resources tab, in IDE Connections, you will see the SOA-MDS connection as shown in the Figure 11-112.

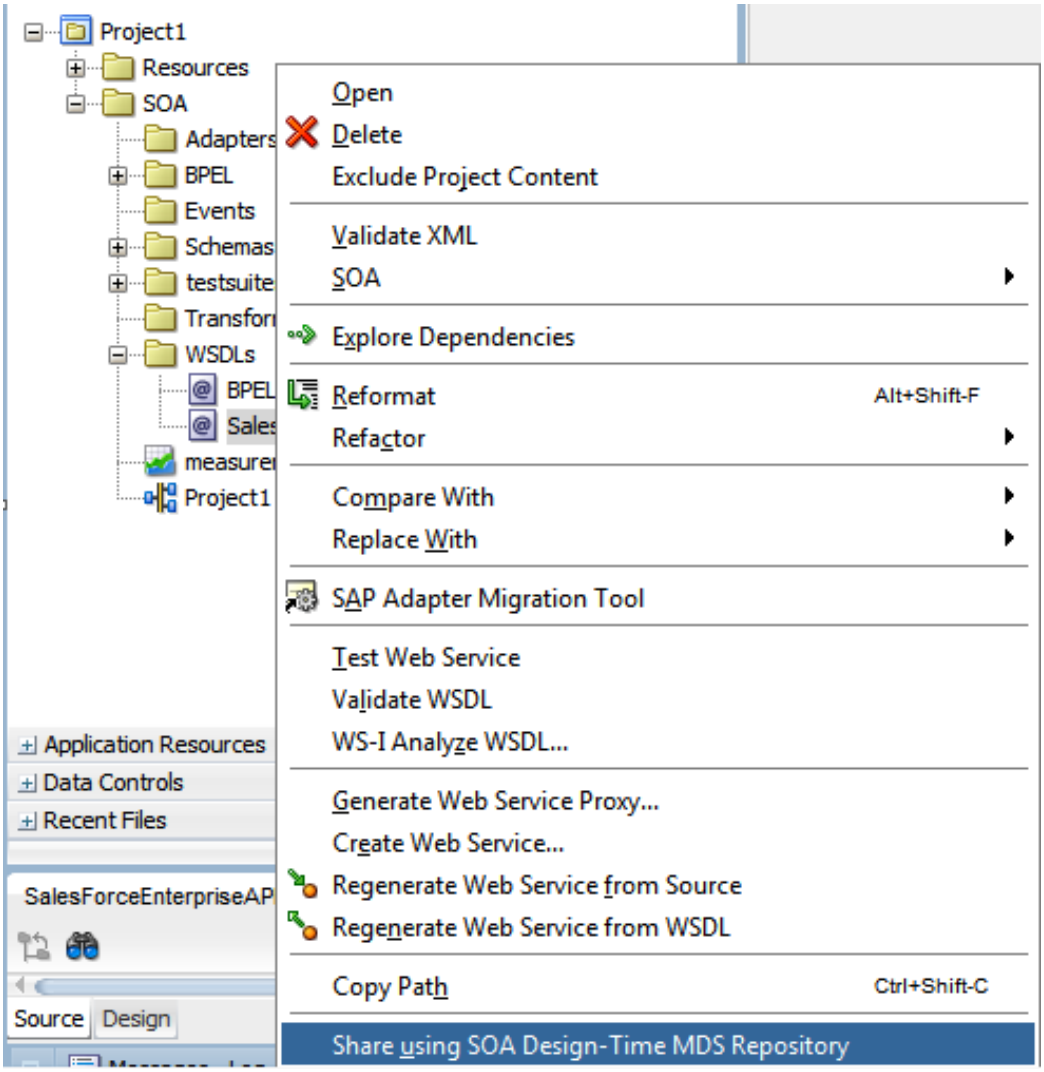
Figure 11-112 Create Database Connection



Transfer Enterprise WSDL to MDS

1. In your project, Under WSDL's directory, right-click on the **Enterprise WSDL** and select the Share using SOA Design-Time MDS Repository as shown in the Figure 11-113.

Figure 11-113 SOA Design-Time MDS Repository

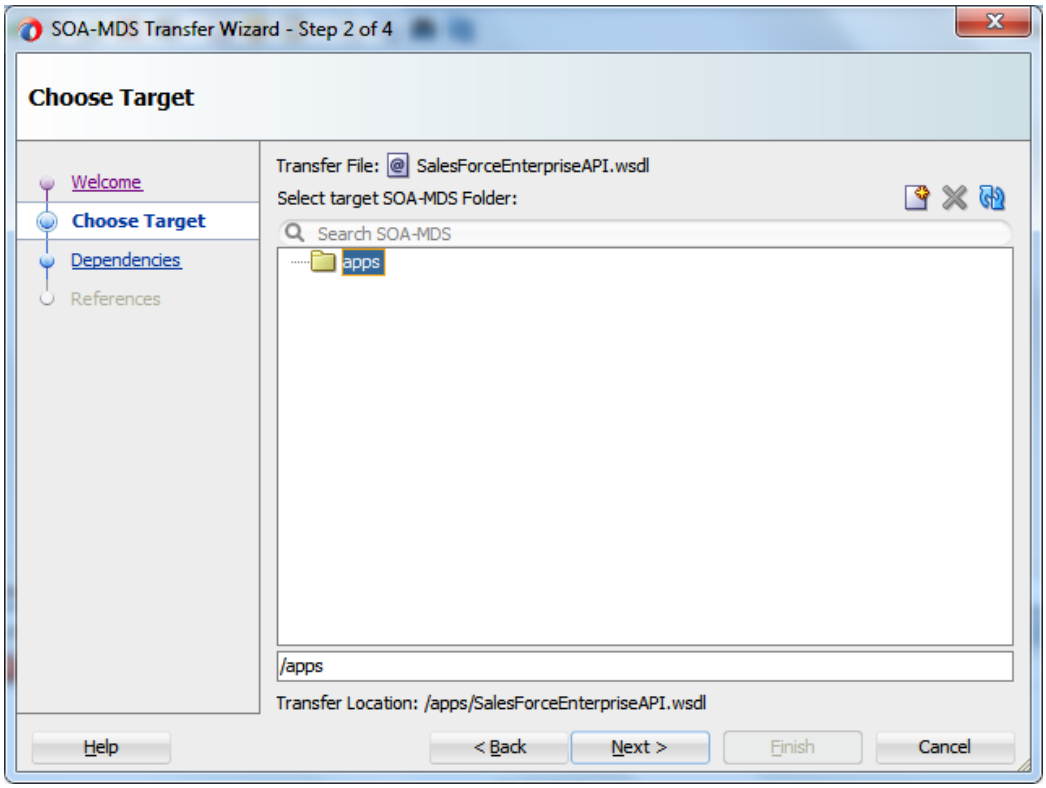


2. Now you will see a SOA-MDS Transfer Wizard as shown in the Figure 11-114.

Figure 11-114 SOA-MDS Transfer Wizard

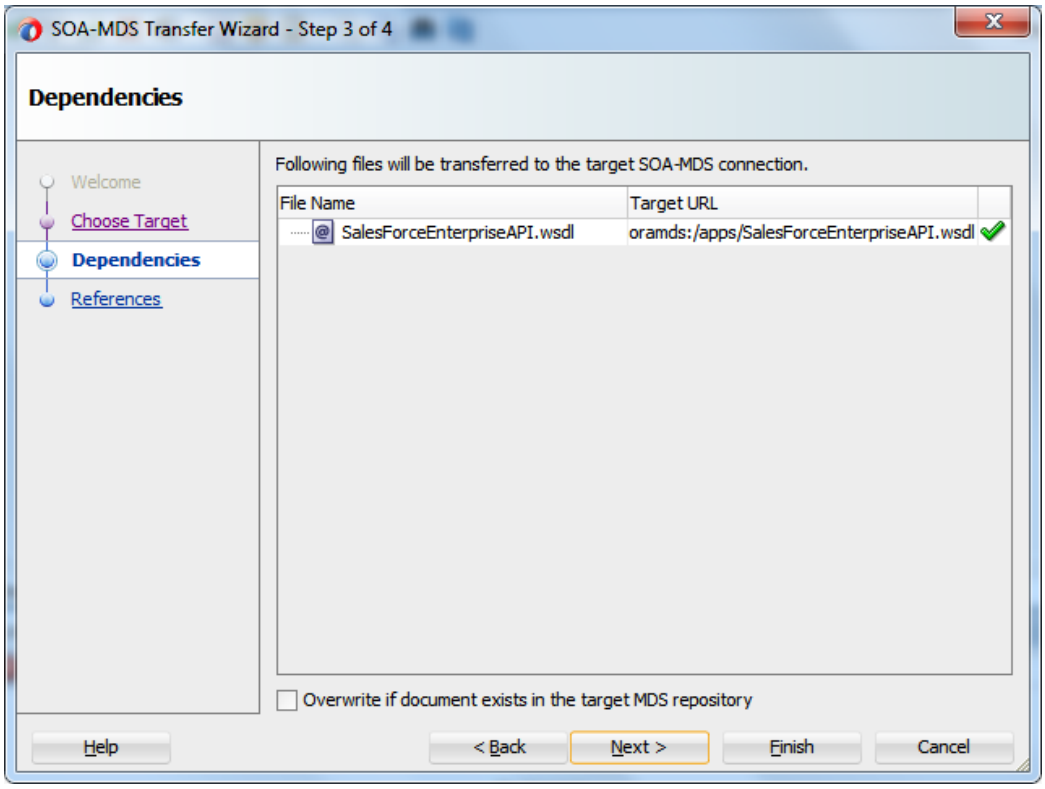
3. Click on the **Next** button and you will see ‘apps’ folder as shown in the Figure 11-115. You can also notice the Transfer Location.

Figure 11 - 115 SOA-MDS Transfer Wizard



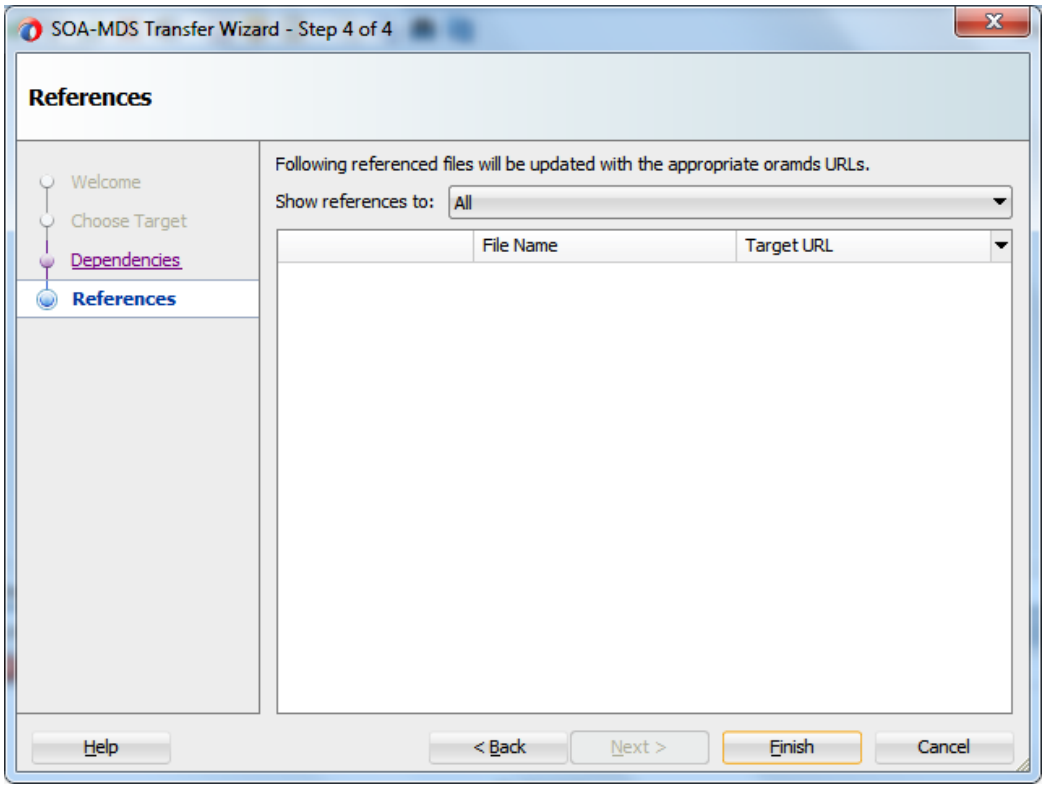
4. Click **Next**, You will see what all files are being transferred and you can also note MDS URL of the WSDL File as shown in Figure 11-116.

Figure 11 - 116 SOA-MDS Transfer Wizard



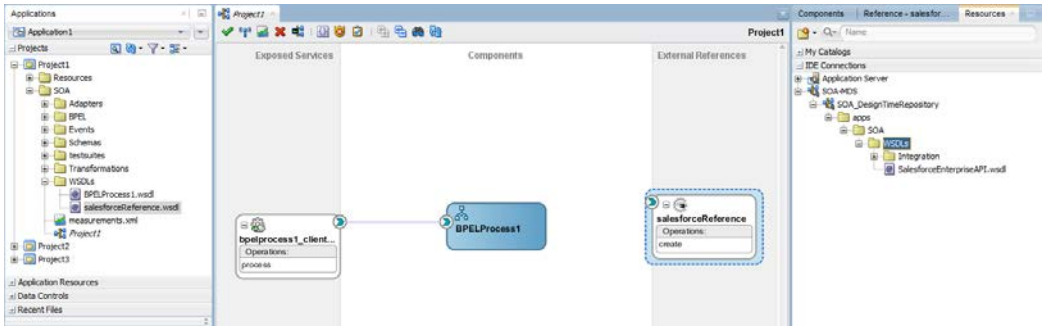
5. Click **Next**, You can see what all files would be updated with the oramds URLs Figure 11-117.

Figure 11 - 117 SOA-MDS Transfer Wizard



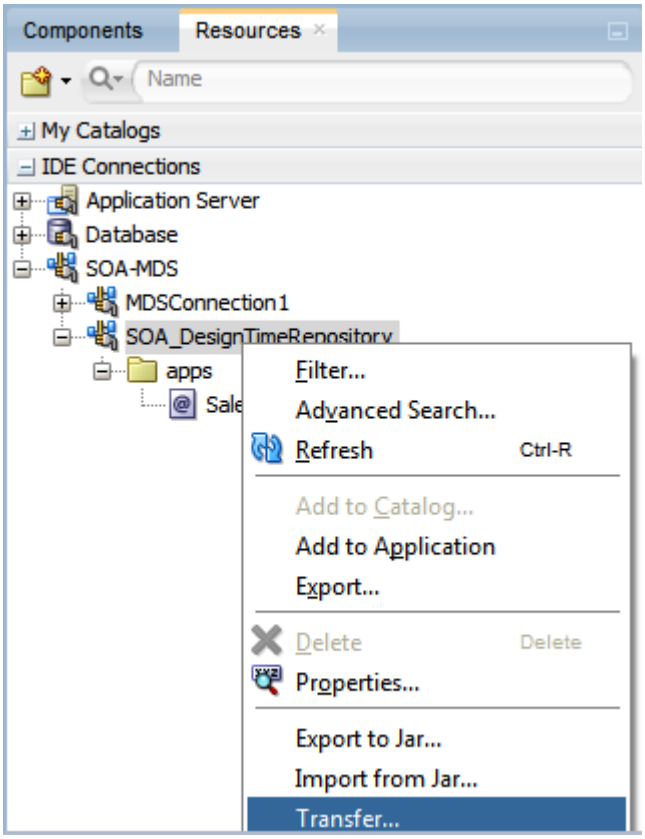
6. Click **Finish** to transfer the WSDL to SOA_DesignTimeRepository. Please note that once the file is transferred then it is no more part of your SOA project. Check your resources palette now. Expand SOA-MDS connection to see the transferred file as shown in the Figure 11-118.

Figure 11 - 1118 After Transferring WSDL to SOA_DesignTimeRepository



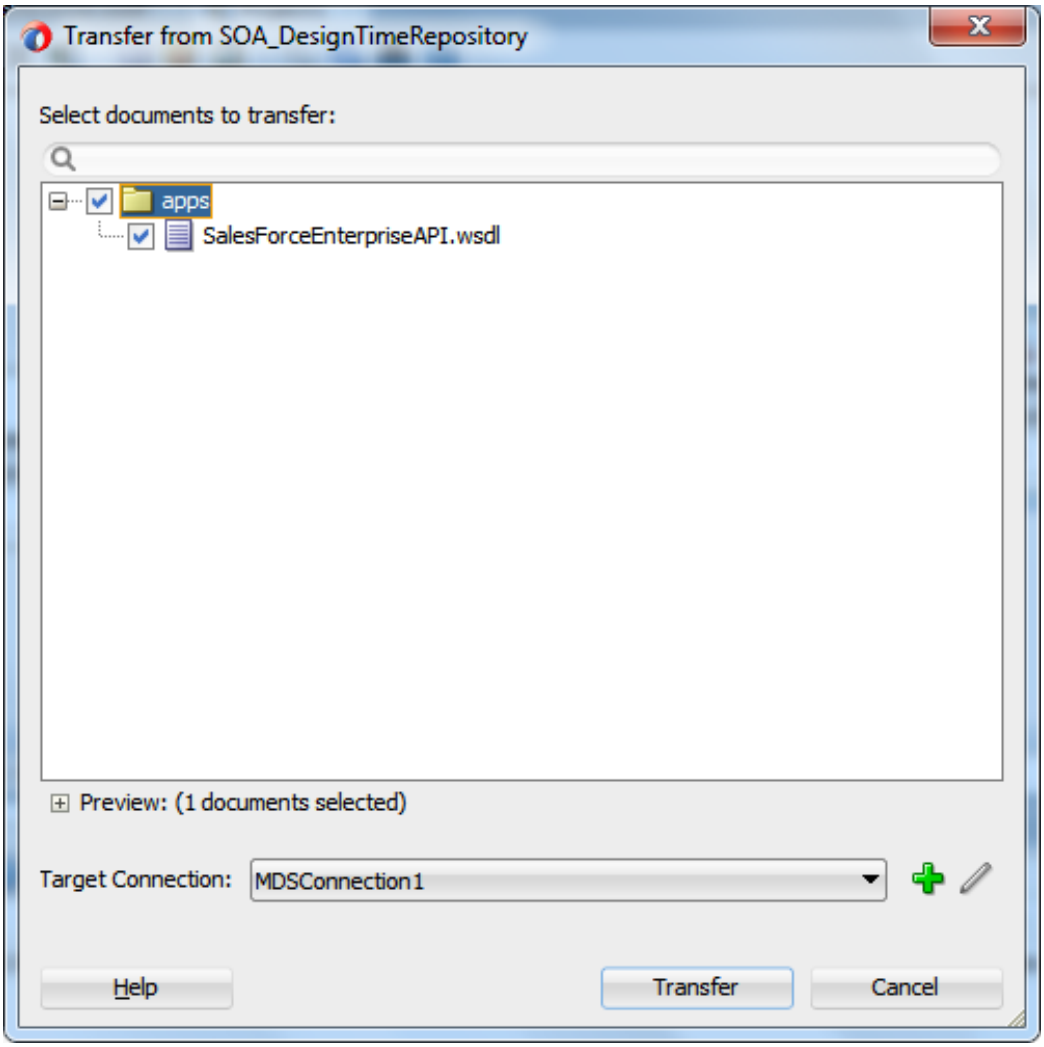
7. Now to transfer WSDL File from SOA_DesignTimeRepository to Target MDS Connection, Right-Click on the SOA_DesignTimeRepository and select the **Transfer** option as shown in the Figure 11-119.

Figure 11 - 1129 Transfer from SOA_DesignTimeRespository



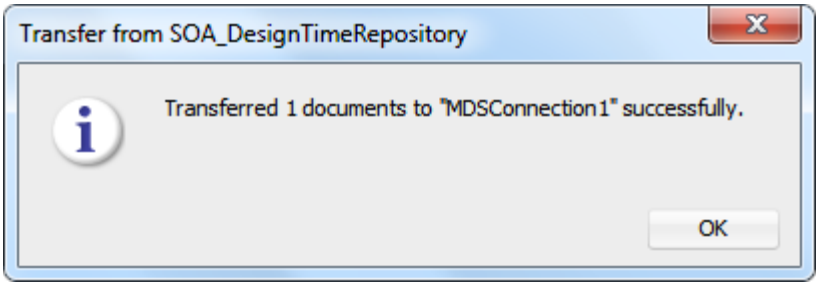
8. In the pop-window, select the Enterprise WSDL and click on **Transfer** as shown in the Figure 11-120.

Figure 11 - 120 Transfer from SOA_DesignTimeRespository



10. You will get a confirmation message after file has been transferred to target MDS connection as shown in the Figure 11-121

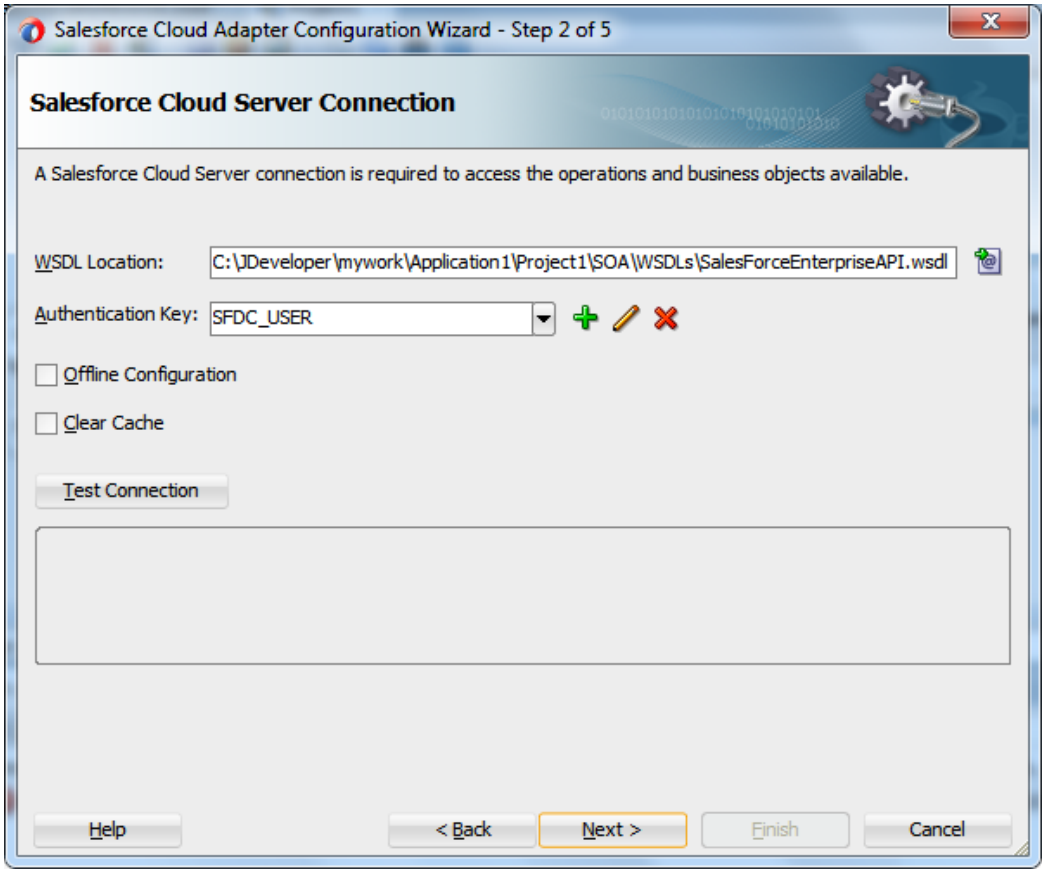
Figure 11 - 121 Confirmation for Transfer of WSDL File to MDS Connection



Using Enterprise WSDL placed at MDS in Oracle Cloud Adapter for Salesforce.com.

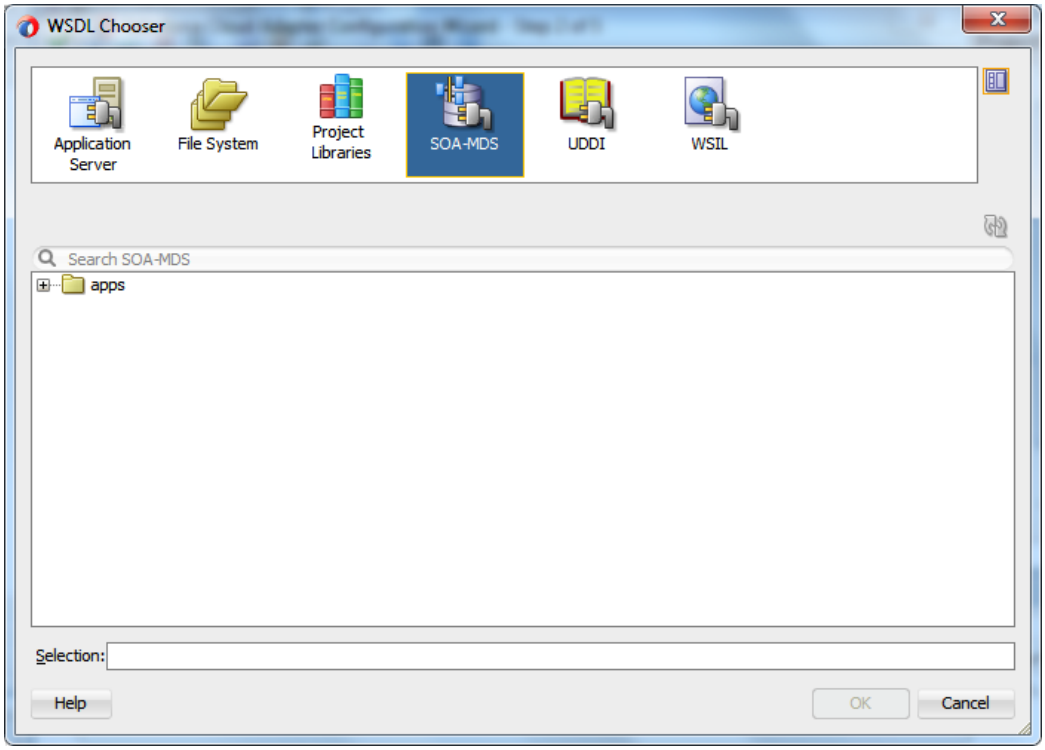
1. To make use of the WSDL file that is placed in MDS, Open the Adapter in Edit mode and navigate to the connection page as shown in Figure 11-122.

Figure 11 - 122 Connection Page in Edit Mode



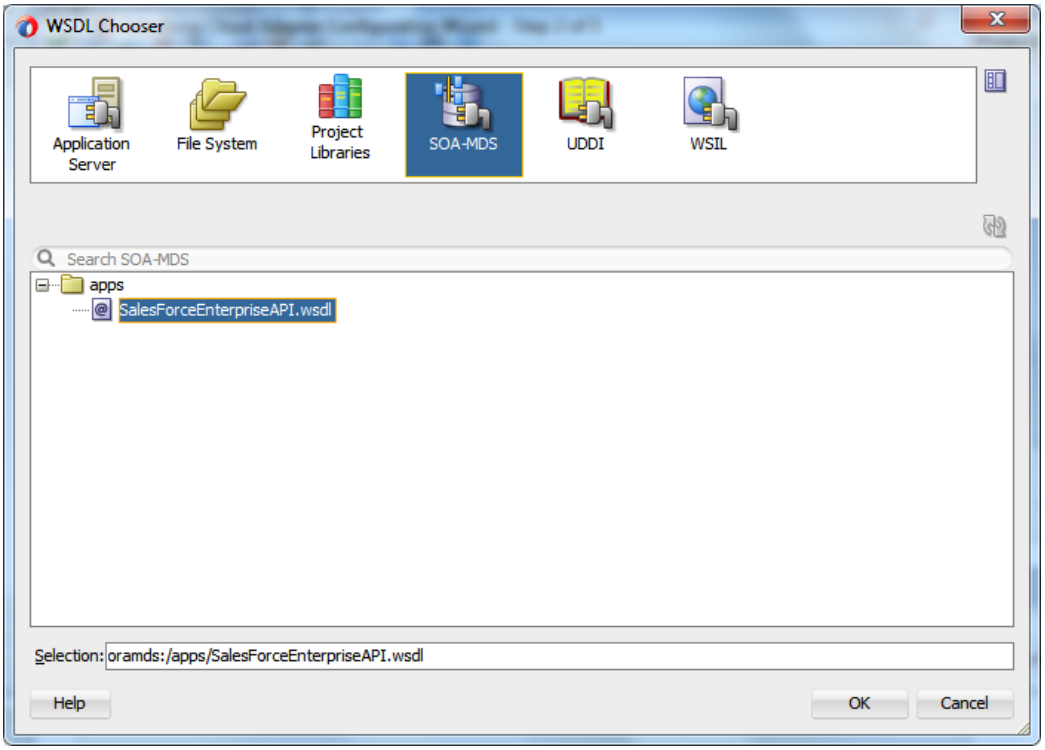
2. Click on the **WSDL Chooser** button and select SOA-MDS tab as shown in Figure 11-123.

Figure 11 - 123 Connection page in edit mode



3. Expand apps folder and select the **Enterprise WSDL** as shown in Figure 11-124. This is the WSDL that was transferred to MDS in the previous steps.

Figure 11 - 124 WSDL Chooser Page



4. Subsequent steps for configuration would remain the same as in section 11.1.1 Creating the BPEL Process for the Use Case 11.1 How can I build integration in which the SOA client is not interested in the response?

5. In your JCA file, you can notice that the targetWSDLURL property has been updated with MDS file location see as shown in Figure 11-125.

Figure 11 - 125 Connection page in edit mode

```
<adapter-config name="salesforceReferencePortType" adapter="salesforce" wsdlLocation="..
  <connection-factory location="cloud/CloudAdapter">
    <non-managed-connection managedConnectionFactoryClassName="oracle.cloud.connector.sa
      <property name="targetWSDLURL" value="oramds:/apps/SalesForceEnterpriseAPI.wsdl"/>
      <property name="csfkey" value="SFDC_USER"/>
      <property name="csfMap" value="SOA"/>
      <property name="applicationVersion" value="29.0"/>
    </non-managed-connection>
  </connection-factory>
```

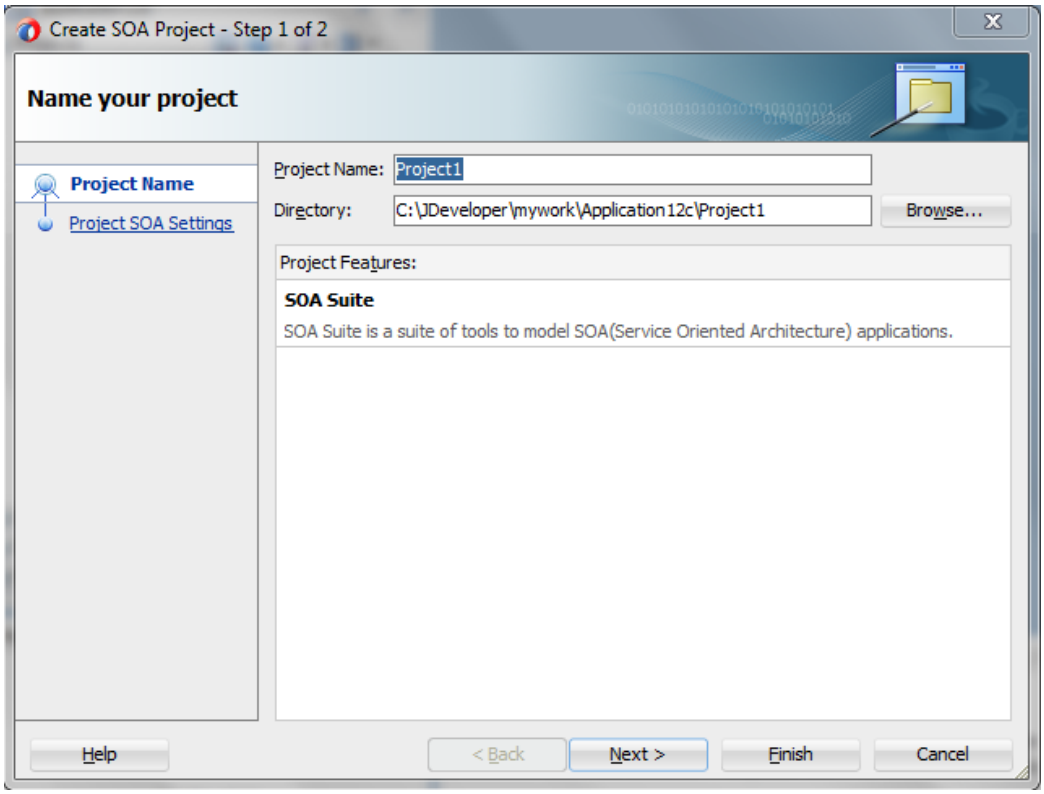
6. Subsequent steps for deployment and testing would remain the same as mentioned in the use case 11.1 How can I build integration in which the SOA client is not interested in the response?.

11.5. How to use Debug Header (Response header)?

Perform the following steps to use debug header and response headers.

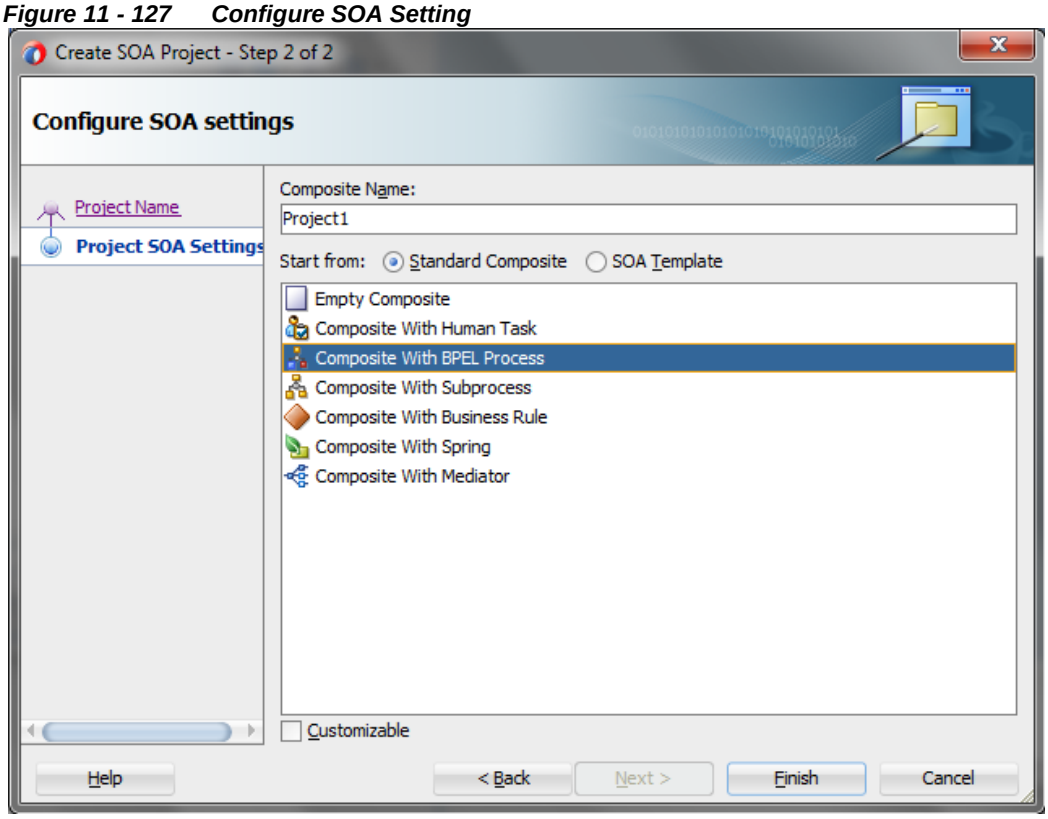
1. Create a New project in an existing application or in a new application.
2. Provide a suitable name to your project, as shown in Figure 11- 126.

Figure 11 - 126 Name your project



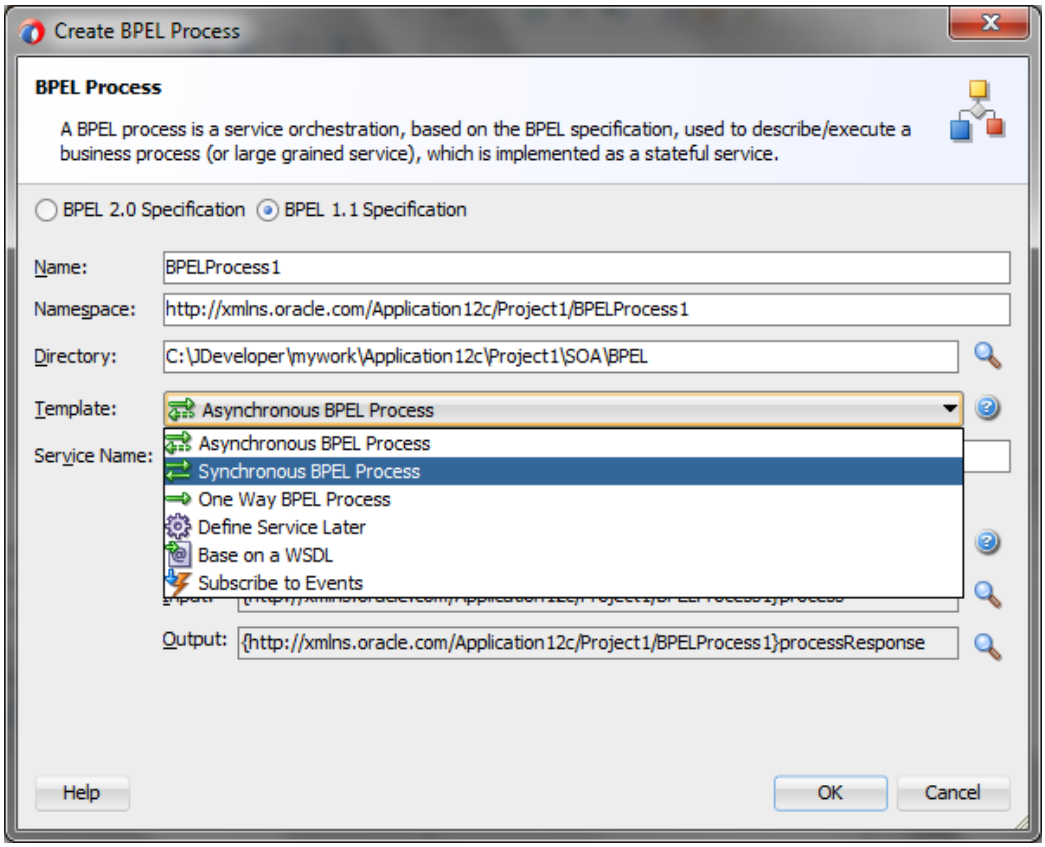
3. Click **Next** and select **Composite with BPEL Process** from Standard Composite list, as shown in Figure 11- 127.

Figure 11 - 127 Configure SOA Setting



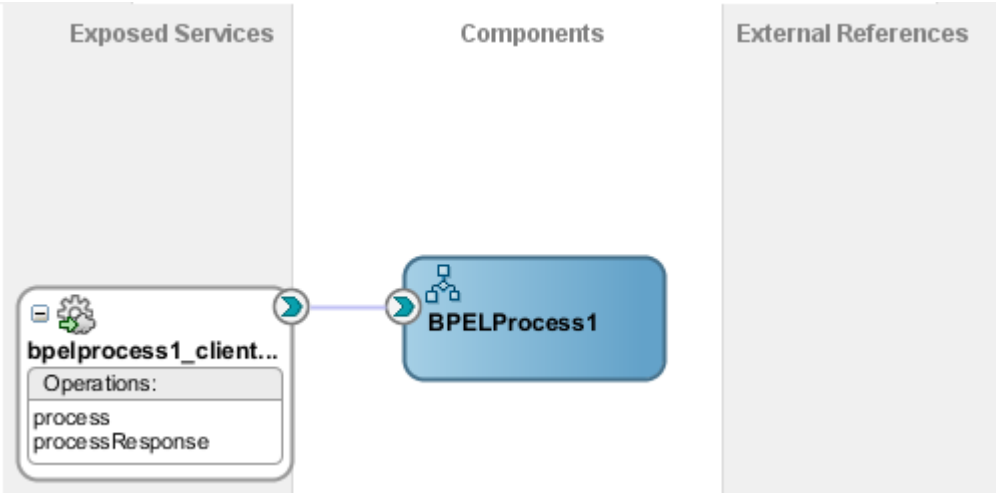
4. Click **Finish**.
5. Select the **Synchronous BPEL Process** from **Template** drop-down and click **OK**, as shown in Figure 11- 128.

Figure 11 - 128 Create BPEL Process



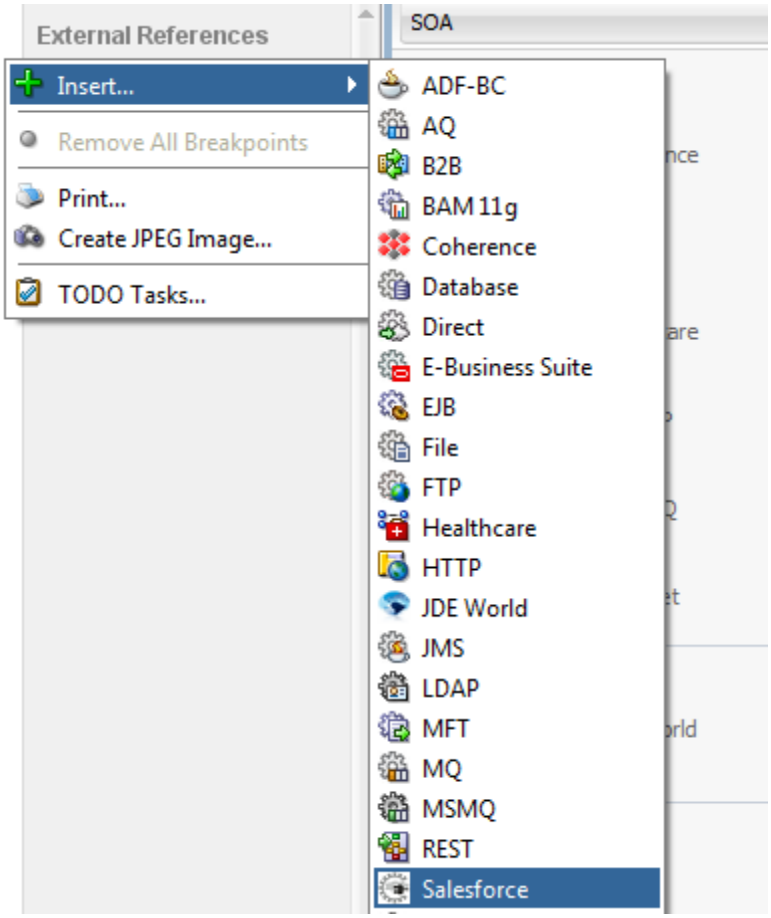
The composite.xml shown in Figure 11- 129.

Figure 11 - 129 Composite.xml



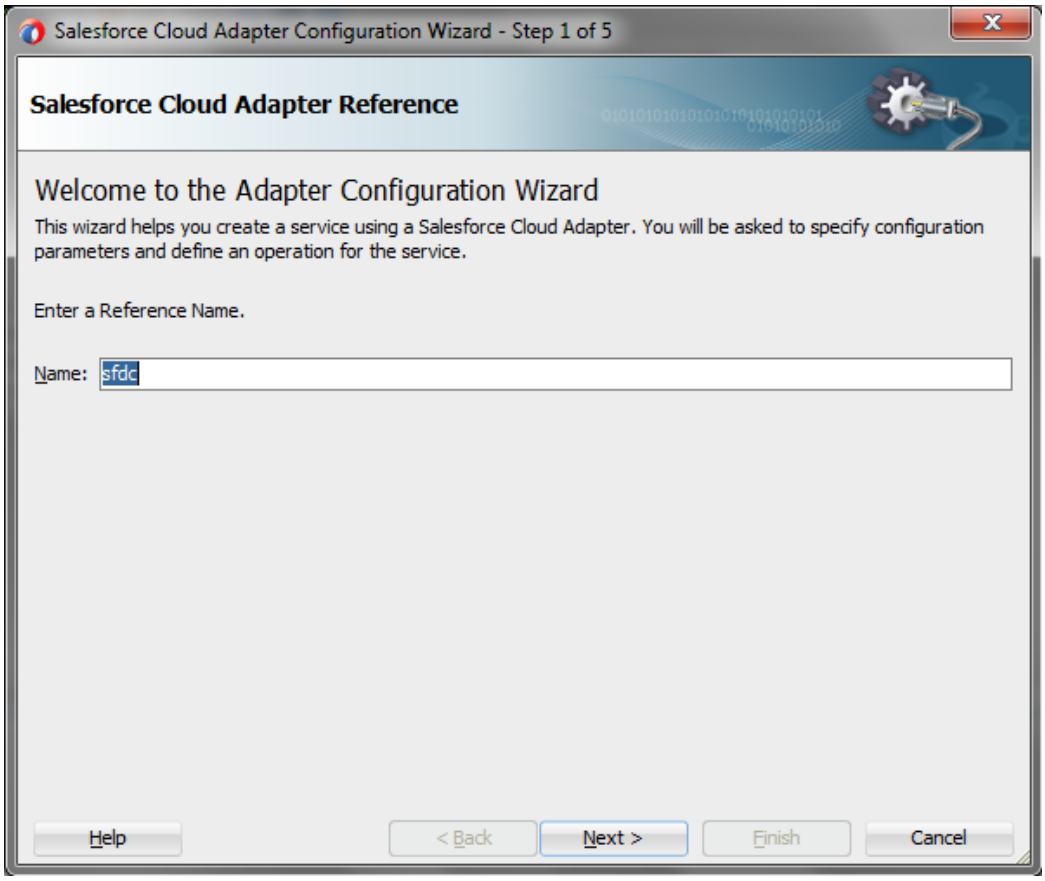
6. In **External References** swim-lane of the composite.xml file, right-click and select **Salesforce** adapter, as shown in Figure 11- 130.

Figure 11 - 130 Salesforce adapter



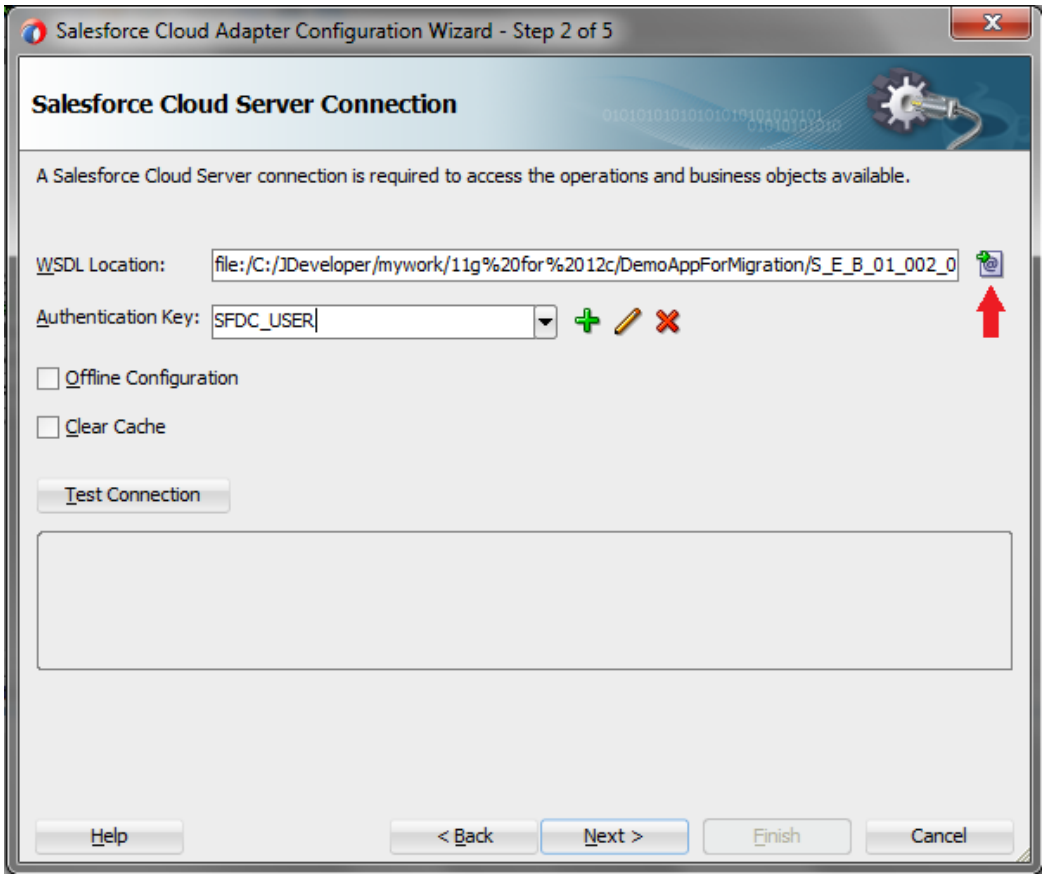
7. The Salesforce Cloud Adapter Configuration Wizard - Welcome page is displayed, as shown in Figure 11- 131.

Figure 11 - 131 Welcome Page



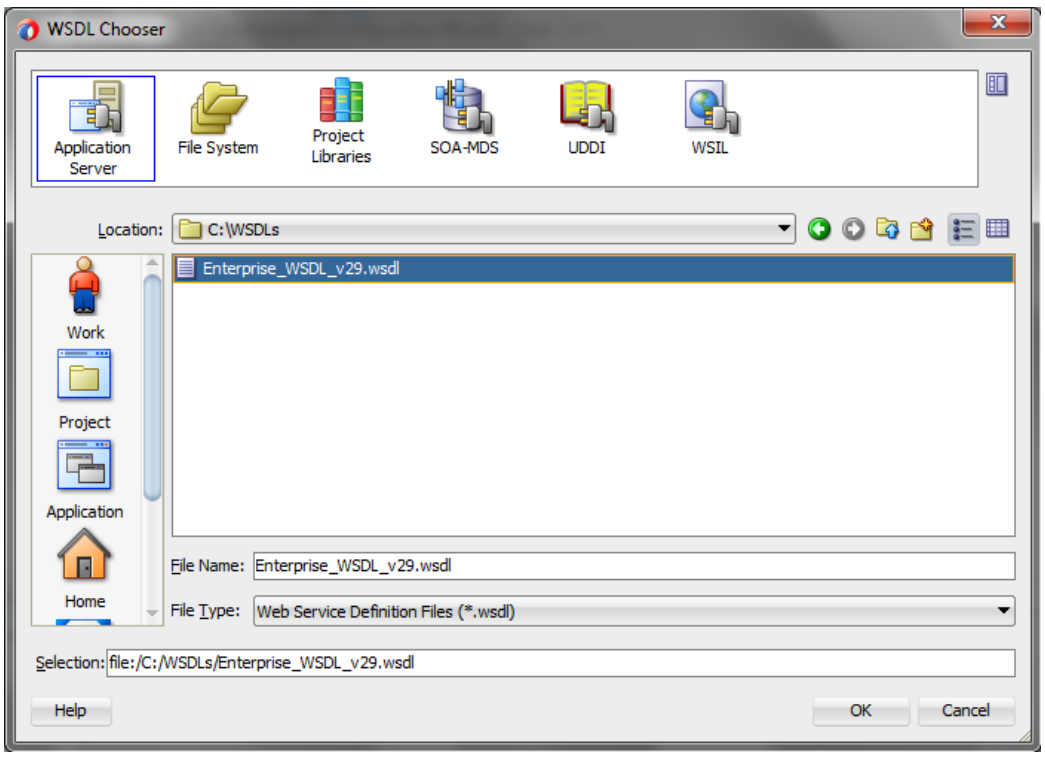
8. Click **Next**.
9. The **Salesforce Cloud Server Connection** page is displayed. The **WSDL Location** and **Authentication Key** textboxes are already populated. It picks up these values from the cache. You can re-enter these values. If you want to use a different value, click the **Find existing WSDLs** icon, which is located to the right of the **WSDL Location** field, as shown in Figure 11- 132.

Figure 11 - 132 Salesforce Cloud Server Connection Page



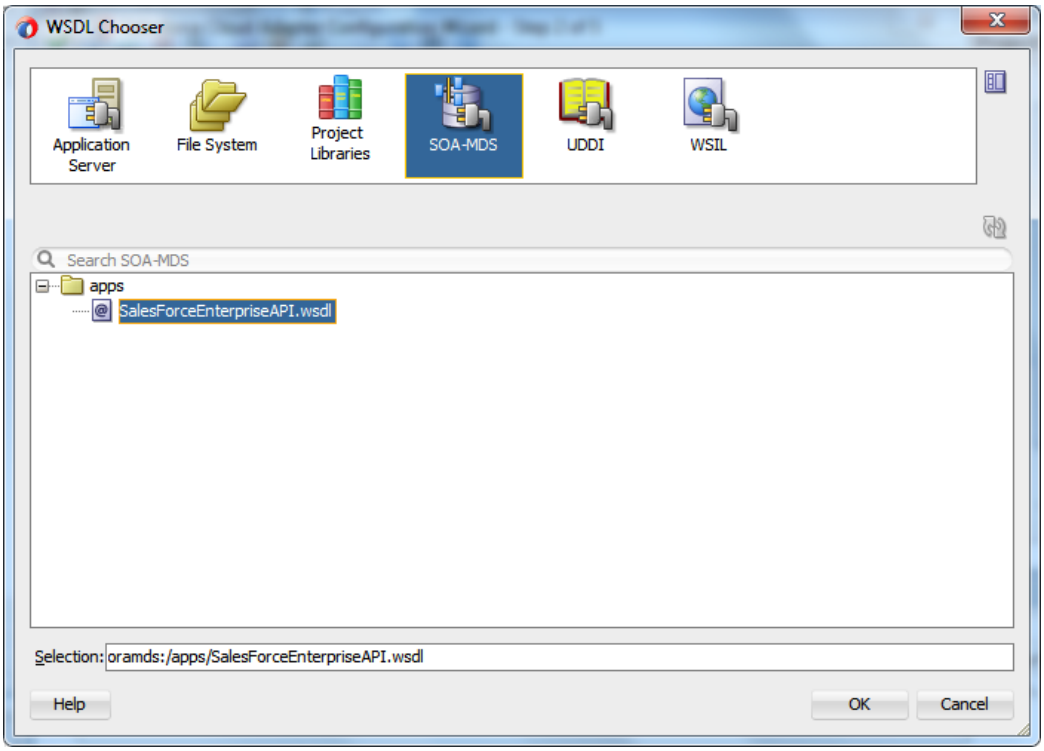
10. The **WSDL Chooser** dialog is displayed, browse and select the downloaded Enterprise WSDL and click **OK**, as shown in Figure 11- 133.

Figure 11 - 133 SOA Resource Browser



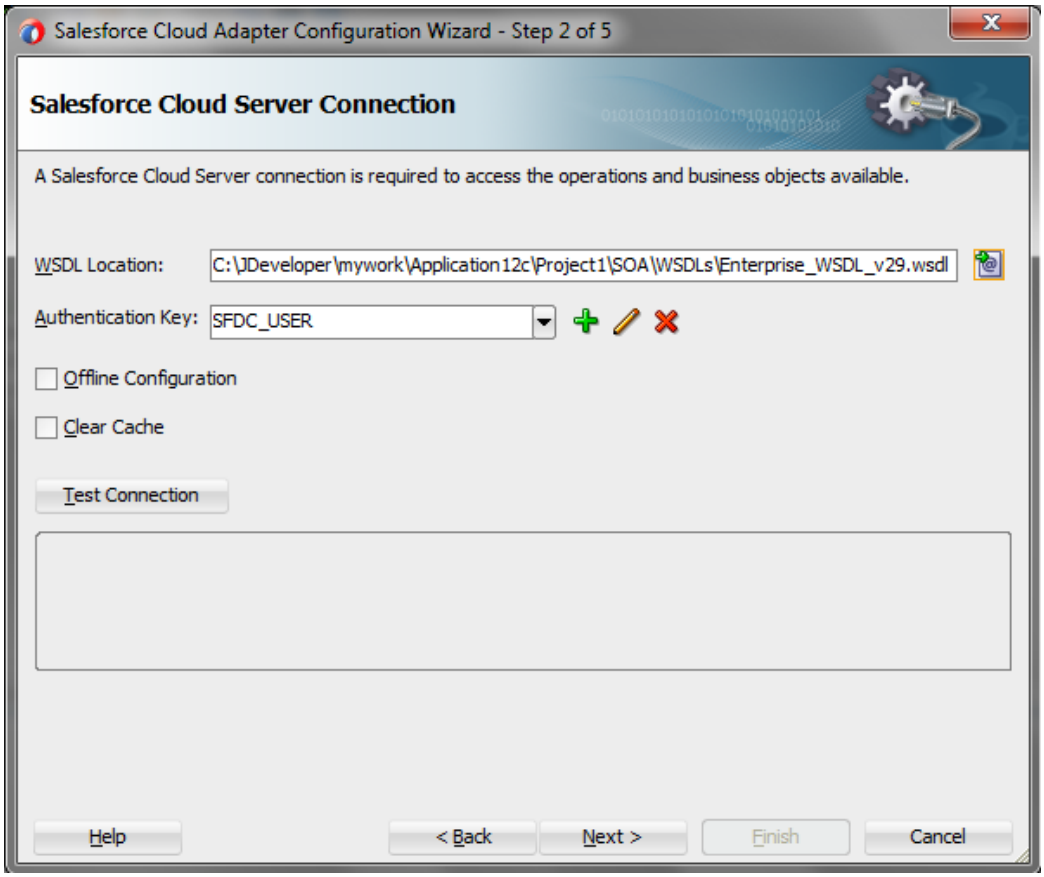
Note that as an alternative, you can store WSDL at an MDS location and access it, as shown in Figure 11- 134.

Figure 11 - 134 SOA Resource Browser



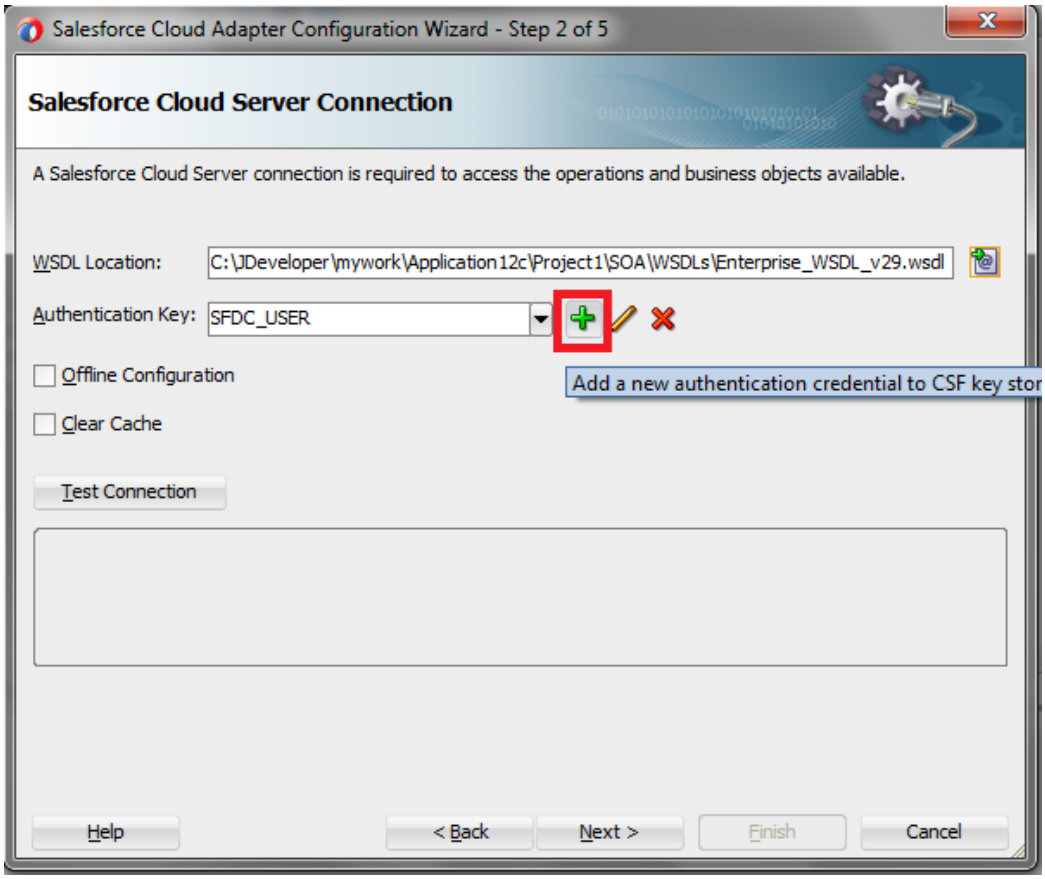
11. Copy WSDL to Your Project Folder.
12. Traverse to IDE Connections → SOA-MDS. Select the appropriate SOA-MDS connection where you placed the Enterprise WSDL. Select the WSDL file to be used in the adapter configuration and click **OK**.
13. The WSDL location should be of the form ‘oramds:/apps/SalesforceEnterpriseAPI.wsdl’, as shown in Figure 11- 135.

Figure 11 - 135 WSDL location



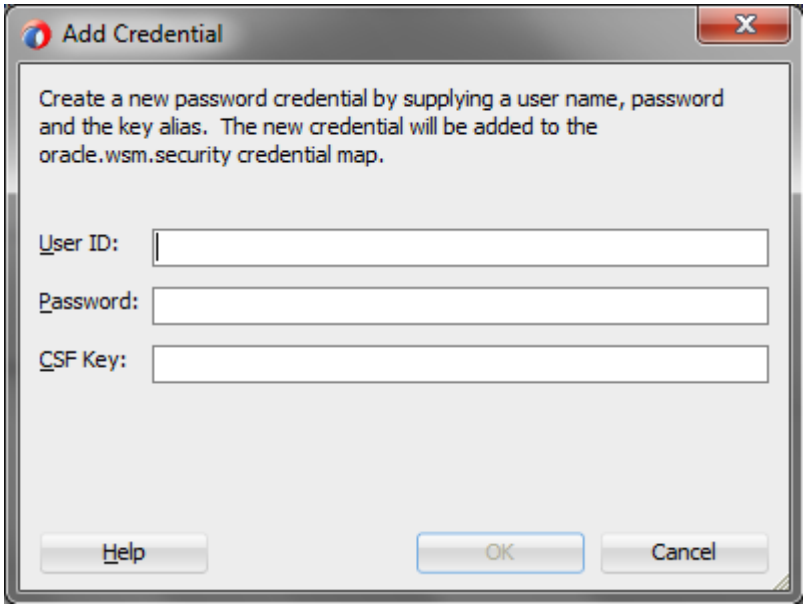
14. Click **OK**.
15. Click **"+"** button to create a new **Authentication Key**, as shown in Figure 11- 136.

Figure 11 - 136 Create a New Authentication Key



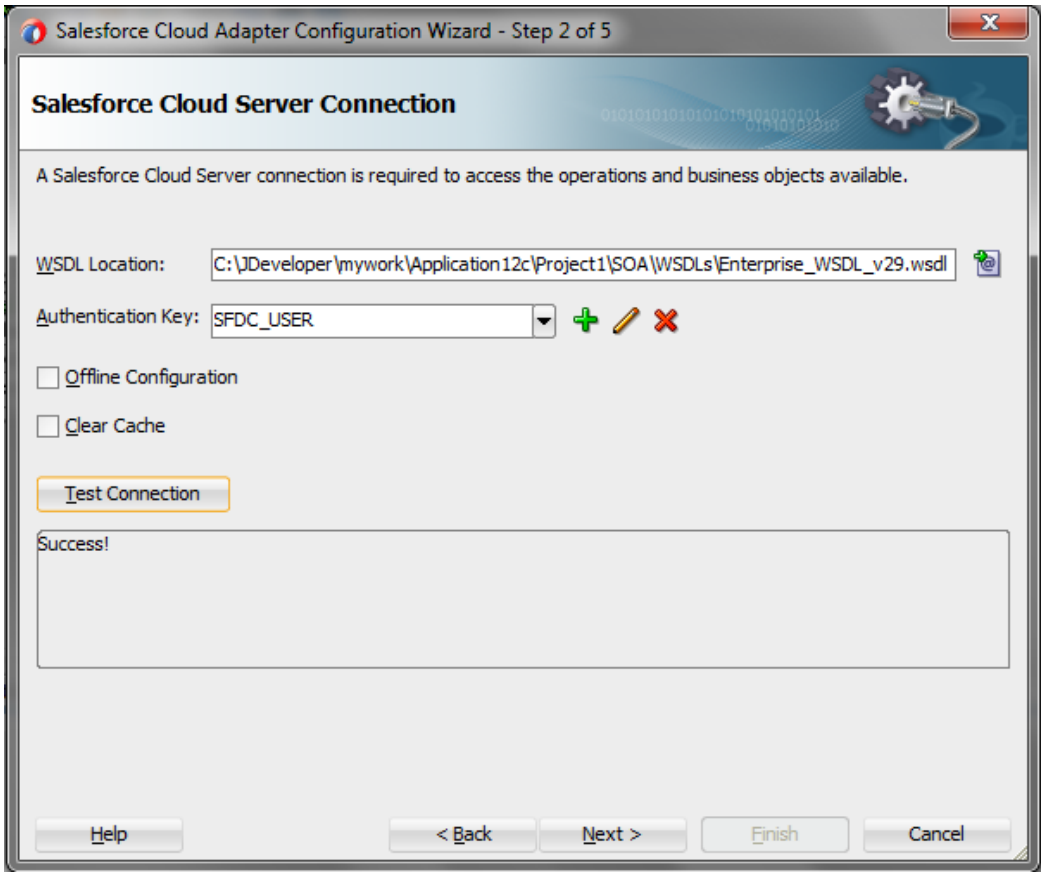
16. The **Add Credential** page is displayed, as shown in Figure 11- 137. Provide a suitable name and the Salesforce.com credentials. The password should be a combination of Salesforce.com password and Salesforce.com Security Token.

Figure 11 - 137 Add Credential



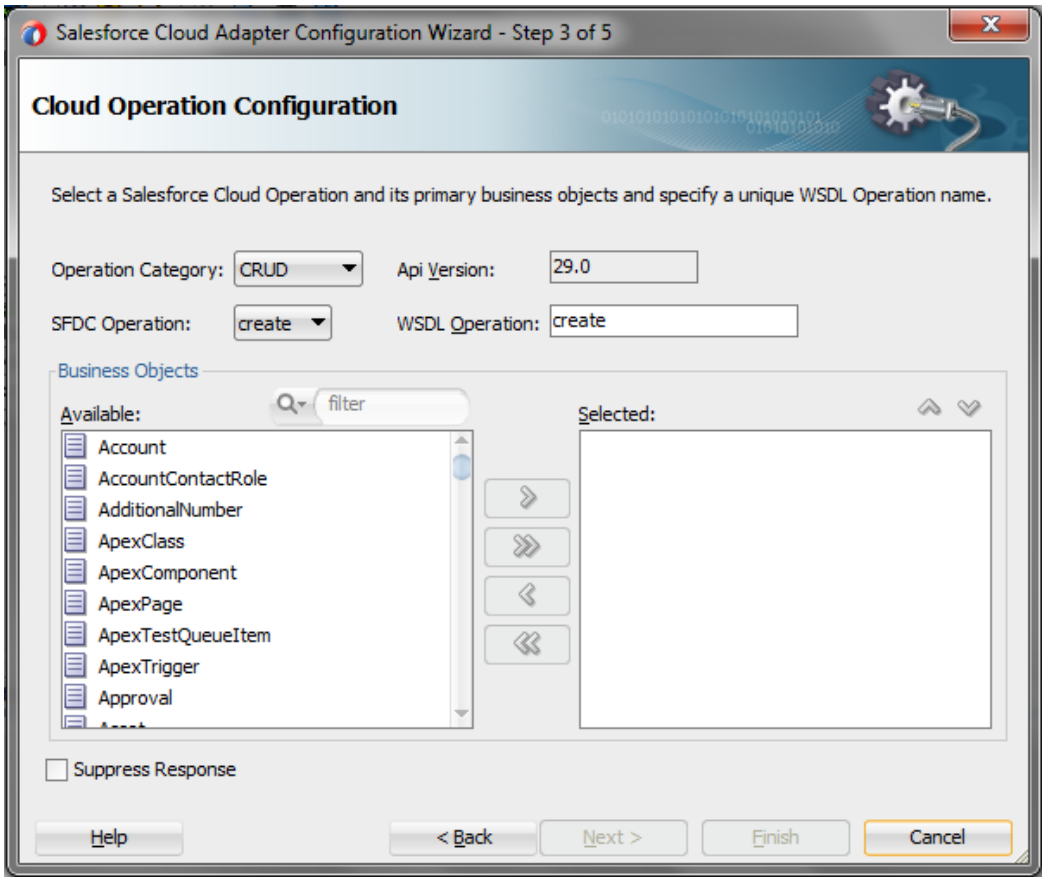
17. Click **Test Connection** button to validate the Authentication Key, as shown in Figure 11- 138.

Figure 11 - 138 Test Connection



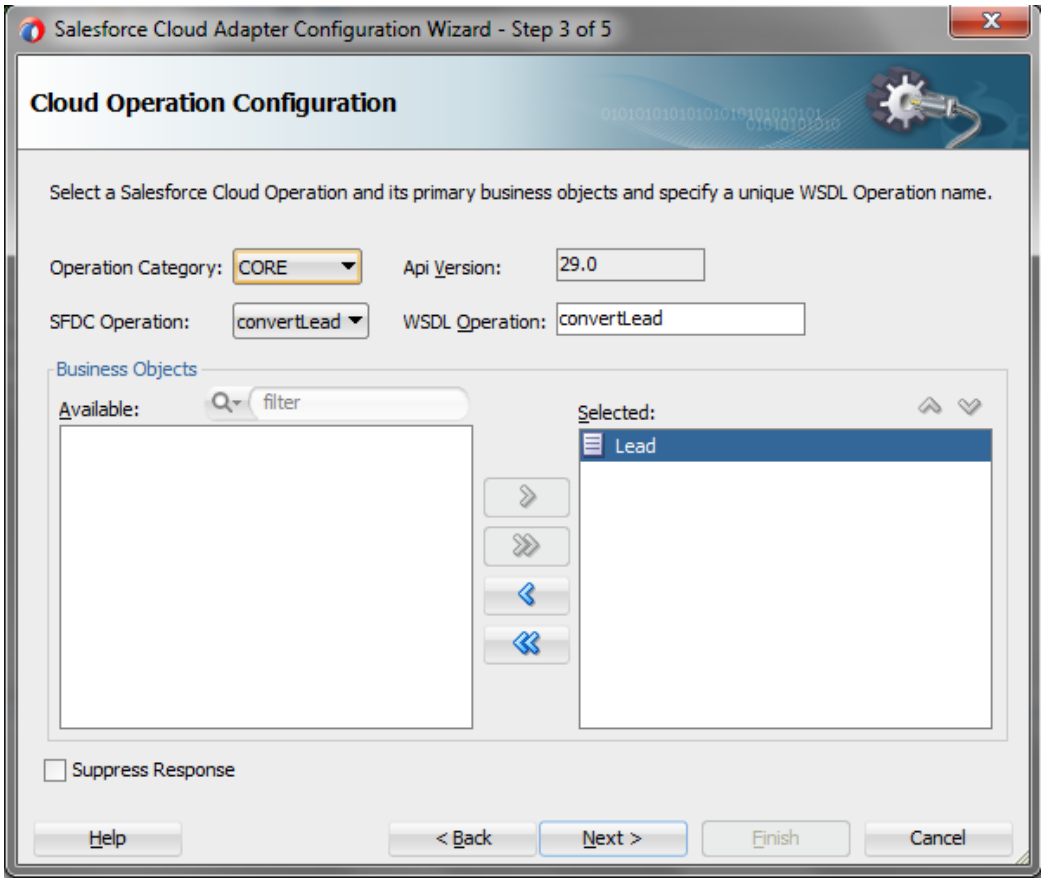
18. Click **Next**.
19. Click **OK** in the next screen.
20. The **Cloud Operation Configuration** page is displayed, as shown in Figure 11- 139.

Figure 11 - 139 Cloud Operation Configuration Page



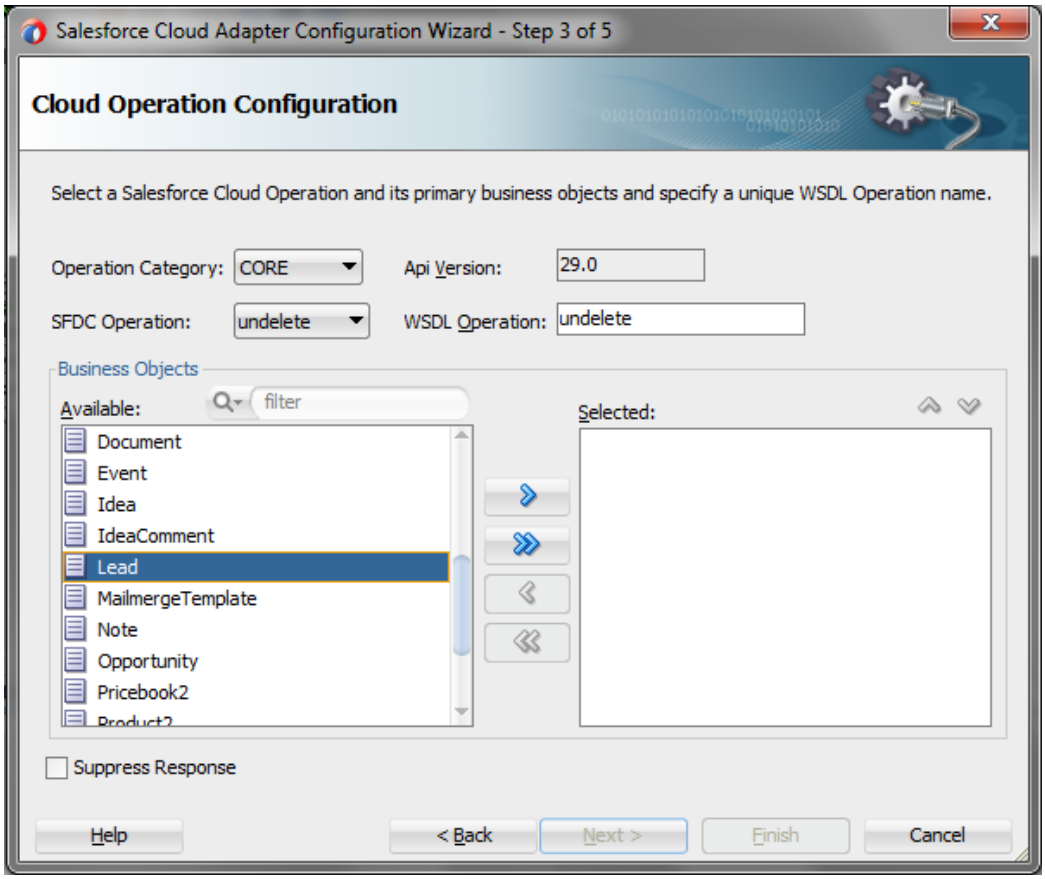
21. From the list of **Operation Category**, select **CORE**, as shown in the Figure 11- 140.

Figure 11 - 140 Select CORE



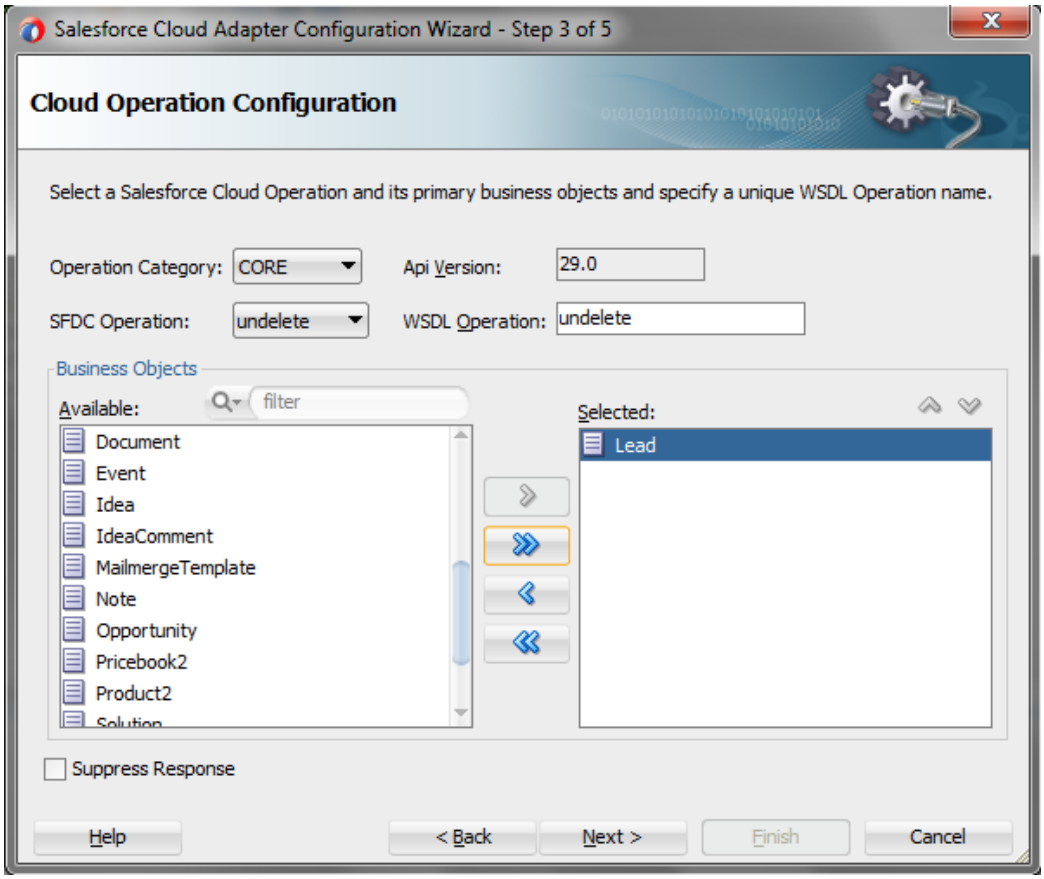
22. For **SFDC Operation**, the **convertlead** operation will automatically be selected. Select **SFDC Operation** as **undelete**.
23. Select **Lead** from **Available** object, as shown in the Figure 11- 141.

Figure 11 - 141 Select Lead object



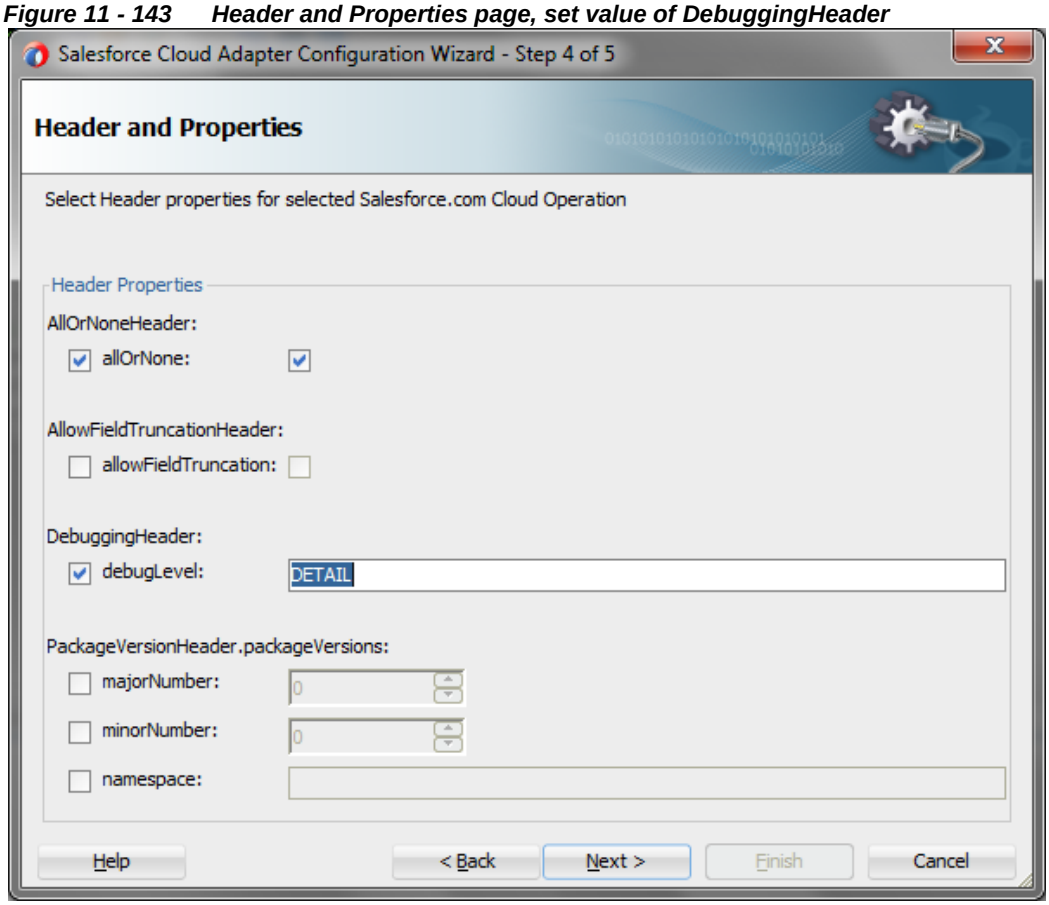
24. After selecting **Lead** object, move it to the **Selected** object area, as shown in Figure 11- 142.

Figure 11 - 142 Select Lead object



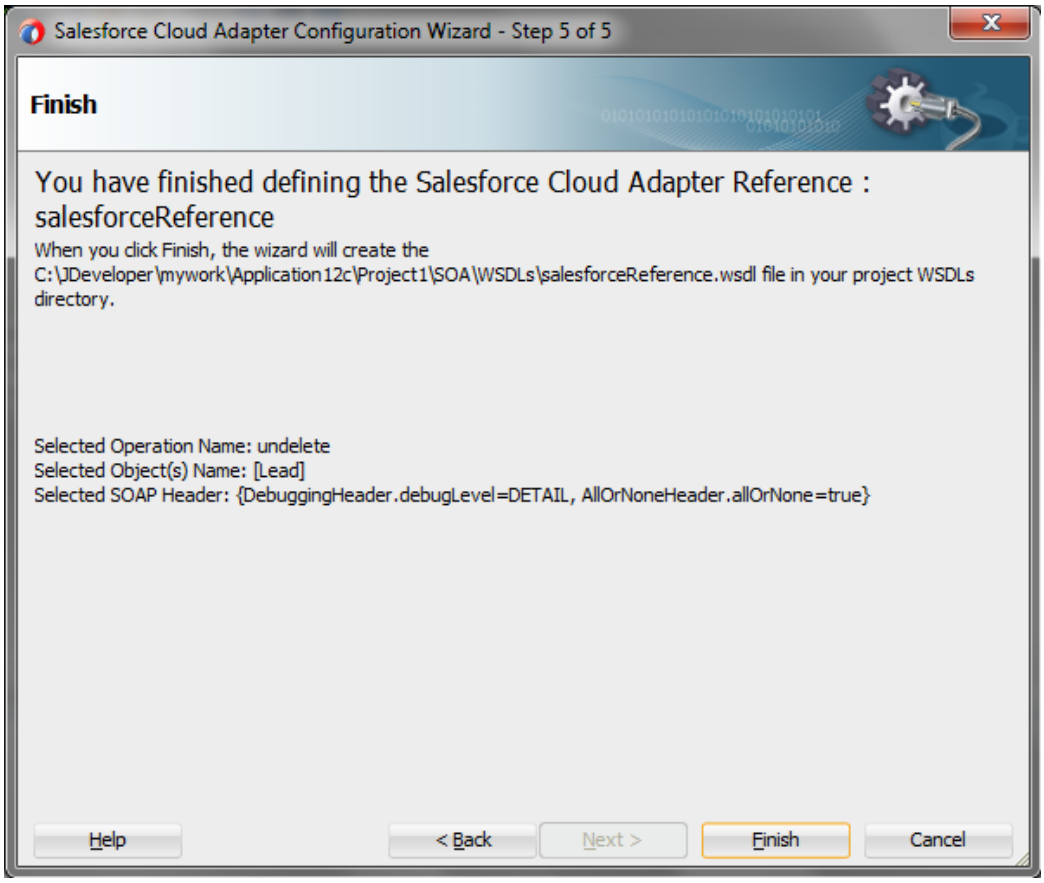
25. Click **Next**. The Headers and Properties page is displayed.
26. On the Headers and Properties, enter the value of DebuggingHeader – debugLevel as **DETAIL**. As shown in Figure 11- 143.

Figure 11 - 143 Header and Properties page, set value of DebuggingHeader



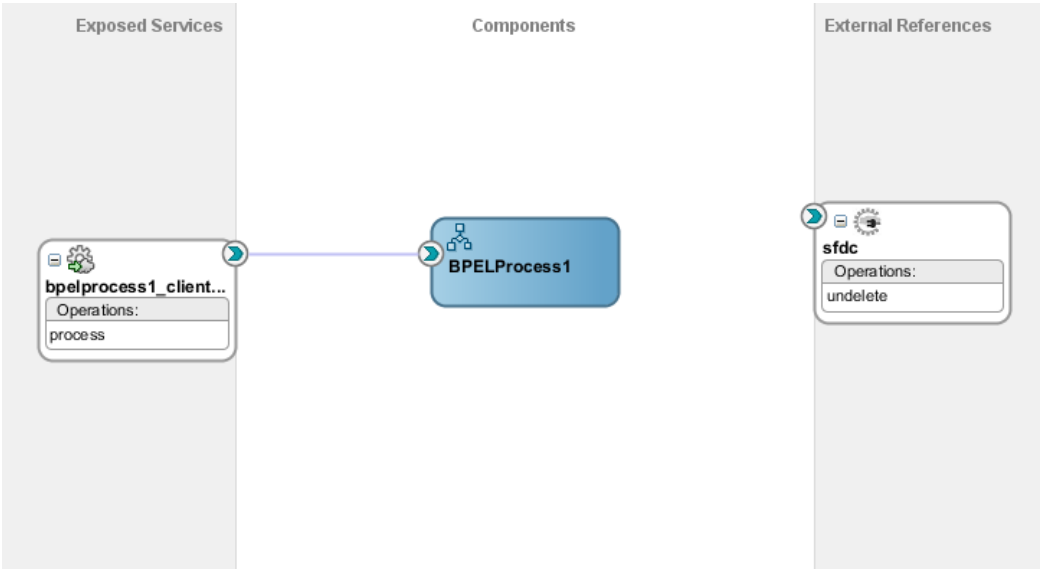
27. Click **Next**.
28. The finish page is displayed. It provides a complete summary of the operation selected, object on which the operation would operate and the headers selected for that operation, as shown in Figure 11- 144.

Figure 11 - 144 Finish Page



29. Click the **Finish** button to complete adapter configuration.
30. After clicking on **Finish** button, the following screen appears, as shown in Figure 11-145.

Figure 11 - 145 Composite Screen

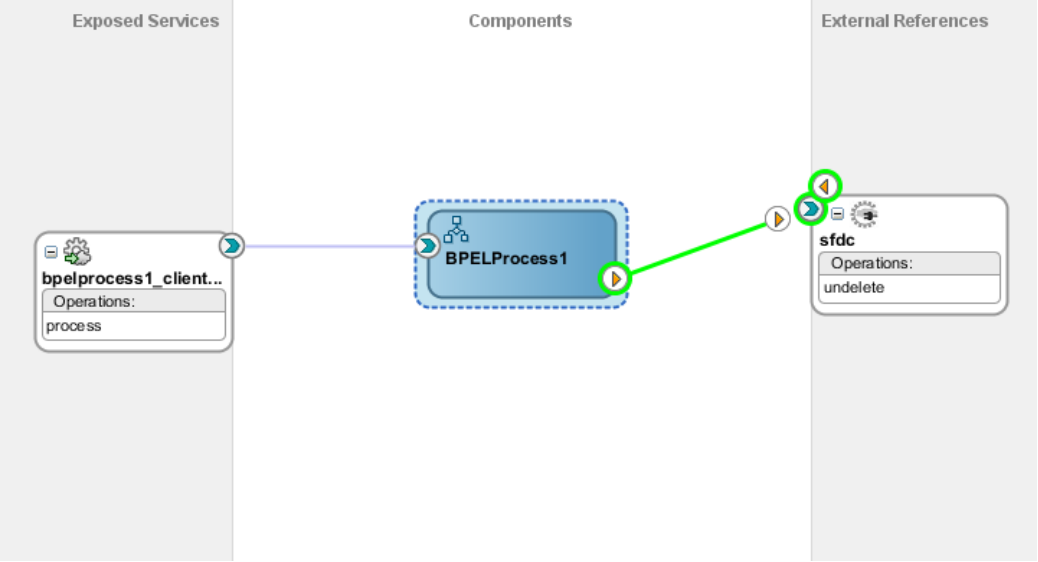


Integration with BPEL

Perform the following steps for integration with BPEL:

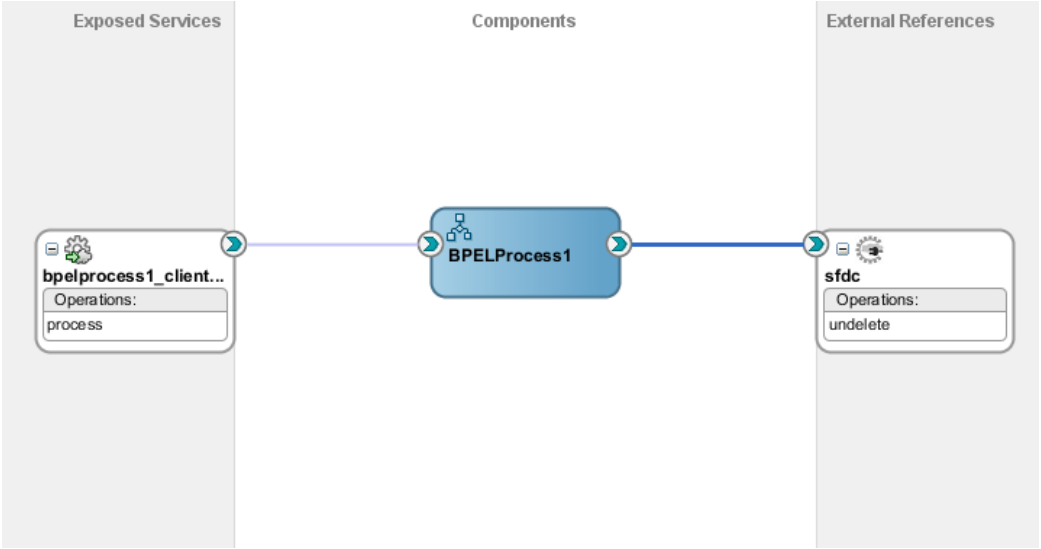
1. Connect **BPELProcess1** and **SFDC** via a wire, as shown in Figure 11- 146.

Figure 11 - 146 Wiring BPELProcess1 and adapter



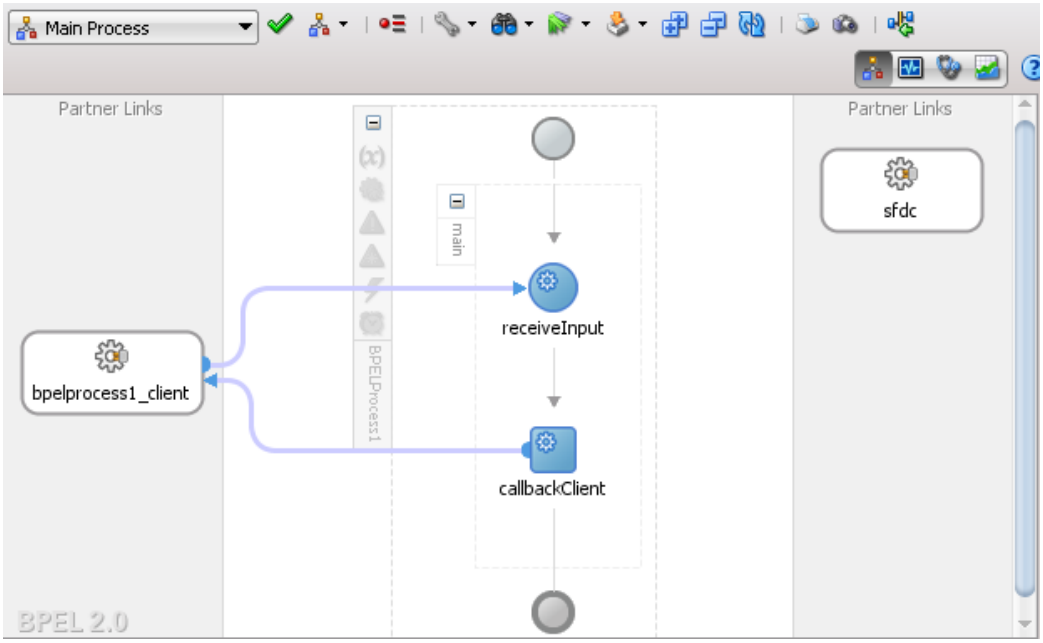
2. After wiring, your composite will look like Figure 11- 147.

Figure 11 - 147 Wiring BPELProcess1 and undelete



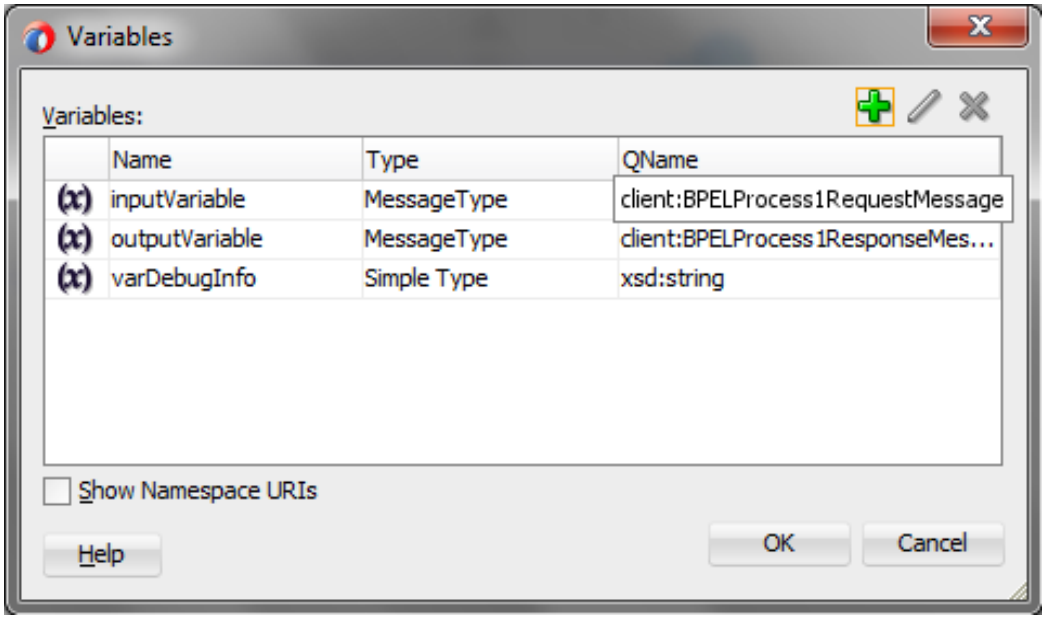
3. Double-click and open **BPELProcess1**. The **sfdc** adapter should be present as part of the Partner Link, as shown in Figure 11- 148.

Figure 11 - 148 Open BPELProcess1



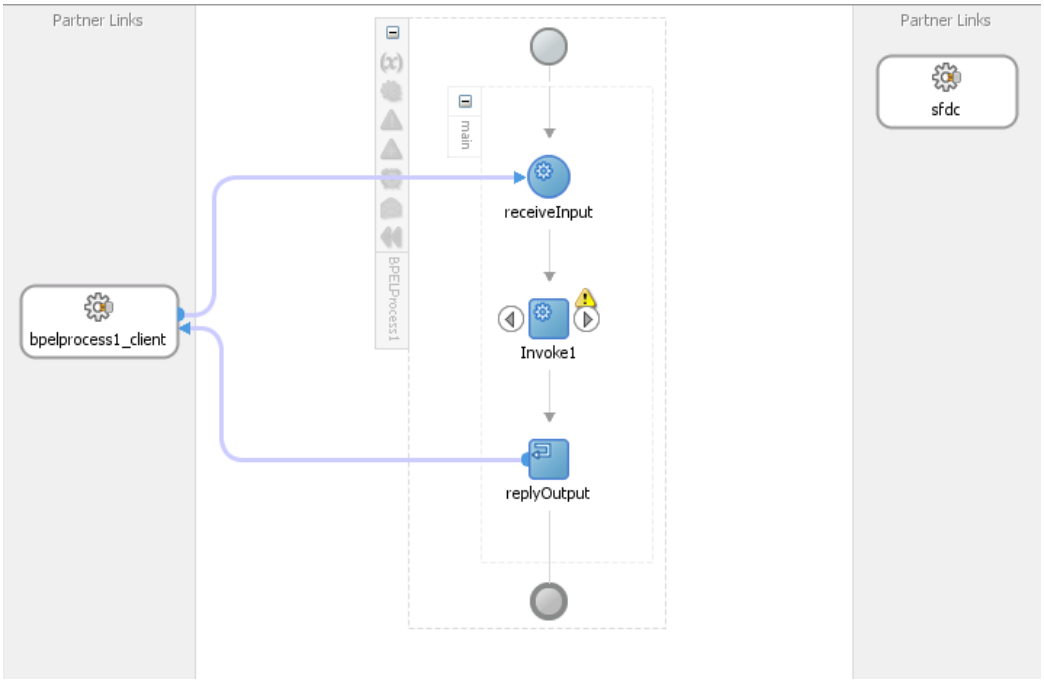
4. Create a variable **varDebugInfo** to track the debugLog returned in debuggingHeader from Salesforce.com, as shown in Figure 11- 149.

Figure 11 - 149 Variables Screen



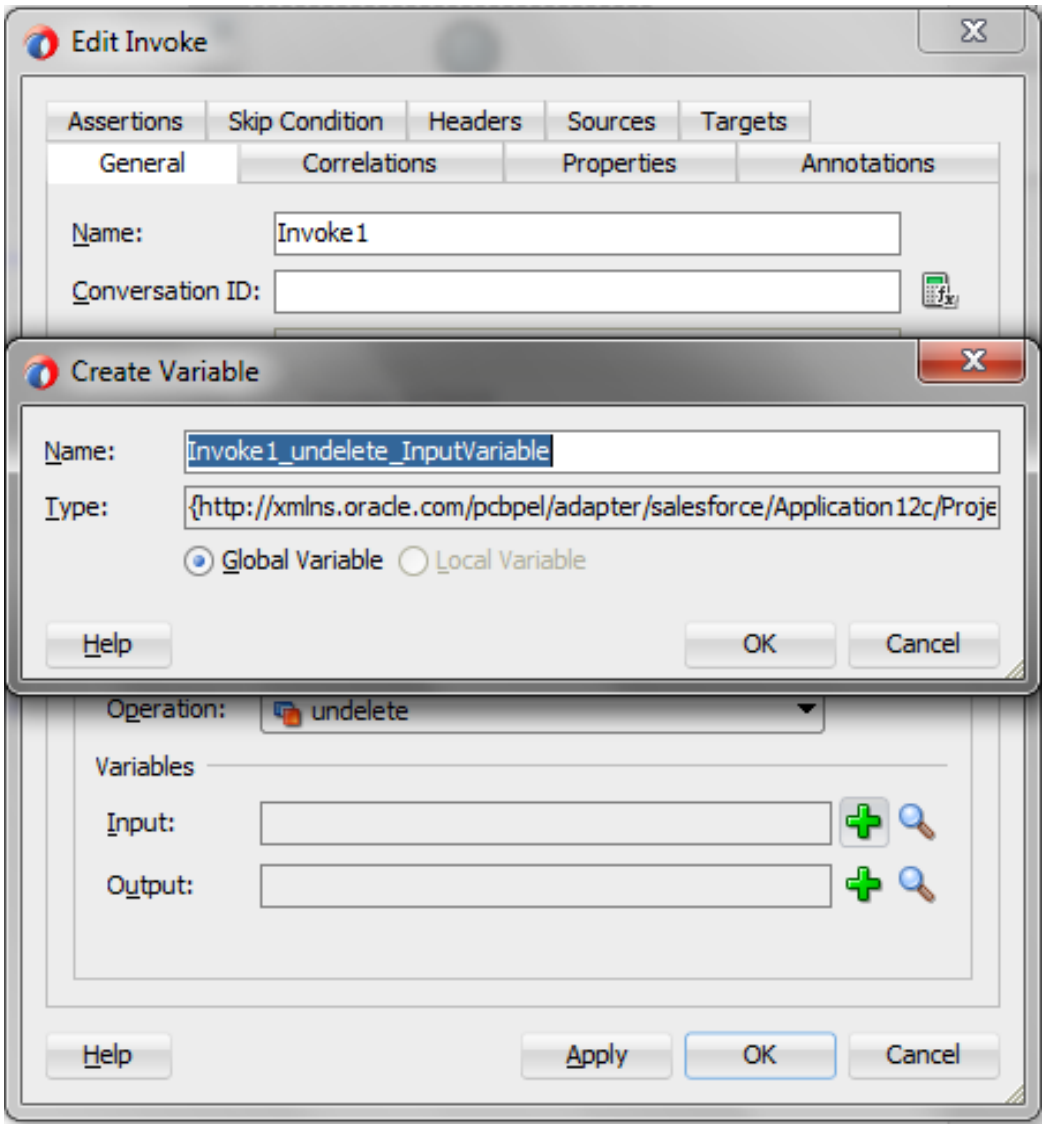
5. Add an **invoke activity** to invoke the query Partner Link, as shown in Figure 11-150.

Figure 11 - 150 Invoke Activity



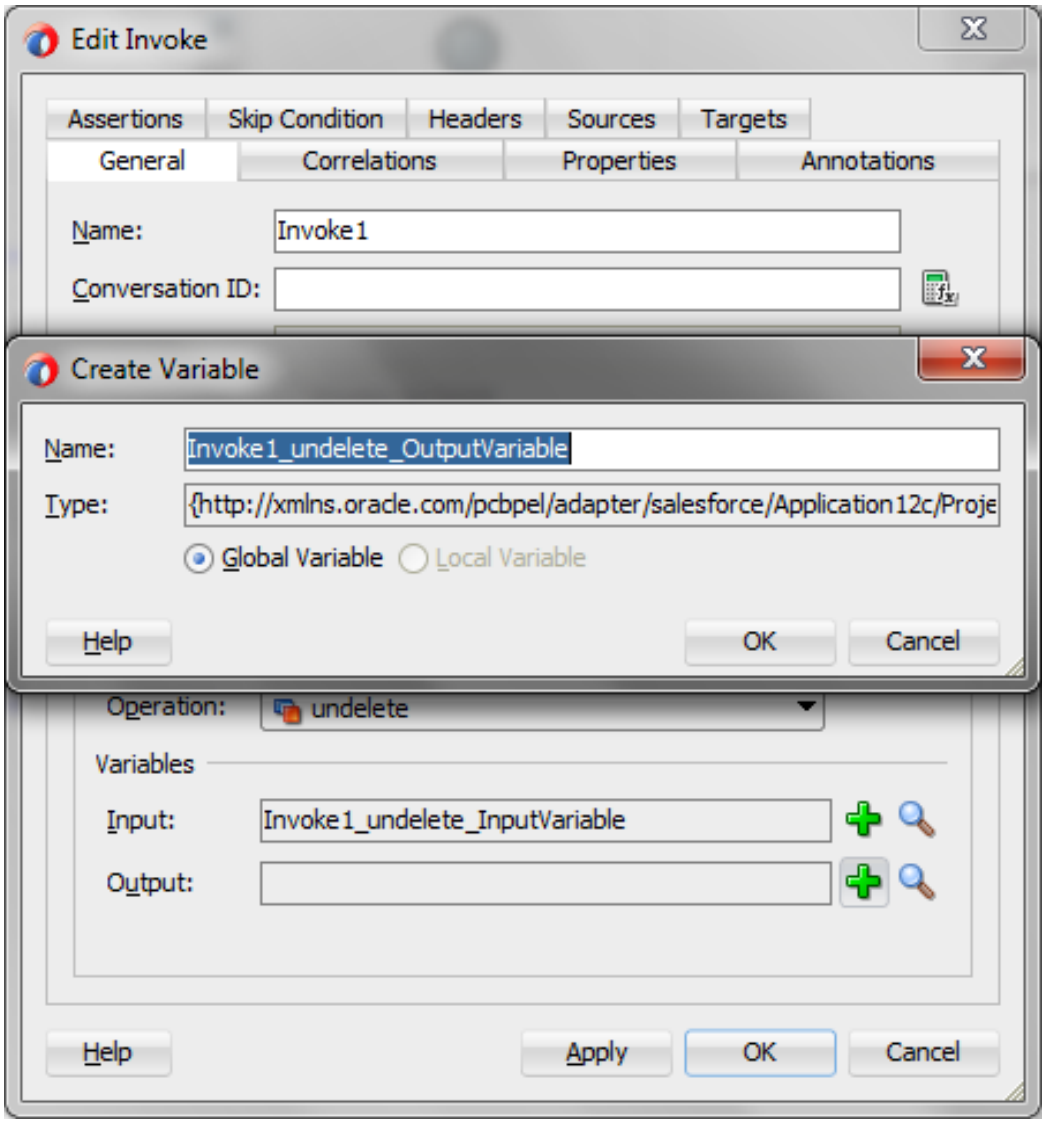
6. Edit Invoke dialog is displayed. Create an input variable to the partner link by clicking the '+' button adjacent to **Input** textbox in the **Variables** section. The **Create Variable** dialog is displayed, as shown in Figure 11- 151.

Figure 11 - 151 Create Variable



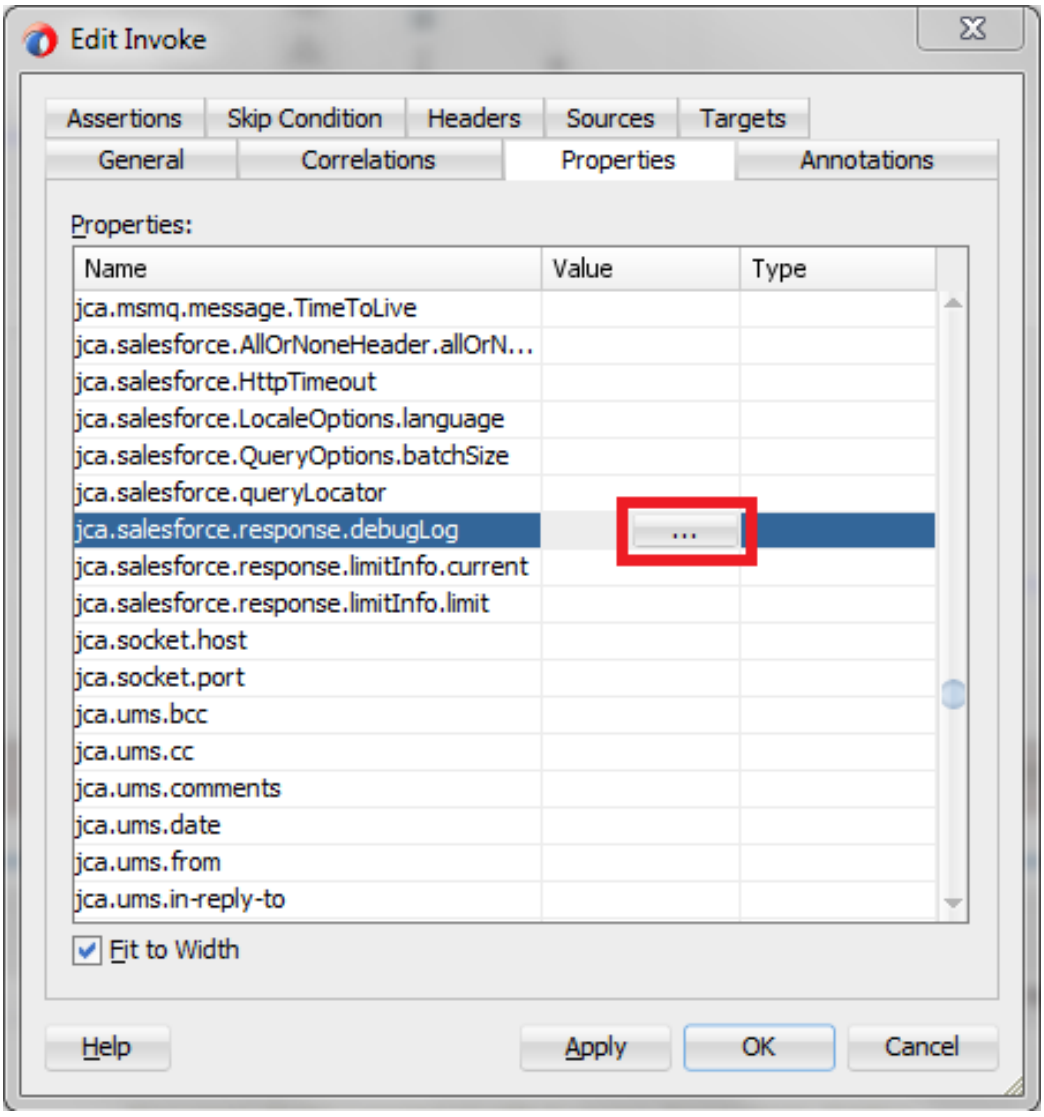
7. Create an output variable from the partner link by clicking the '+' button adjacent to **Output** textbox in the **Variables** section. The **Create Variable** dialog is displayed, as shown in Figure 11- 152 .

Figure 11 - 152 Create Variable



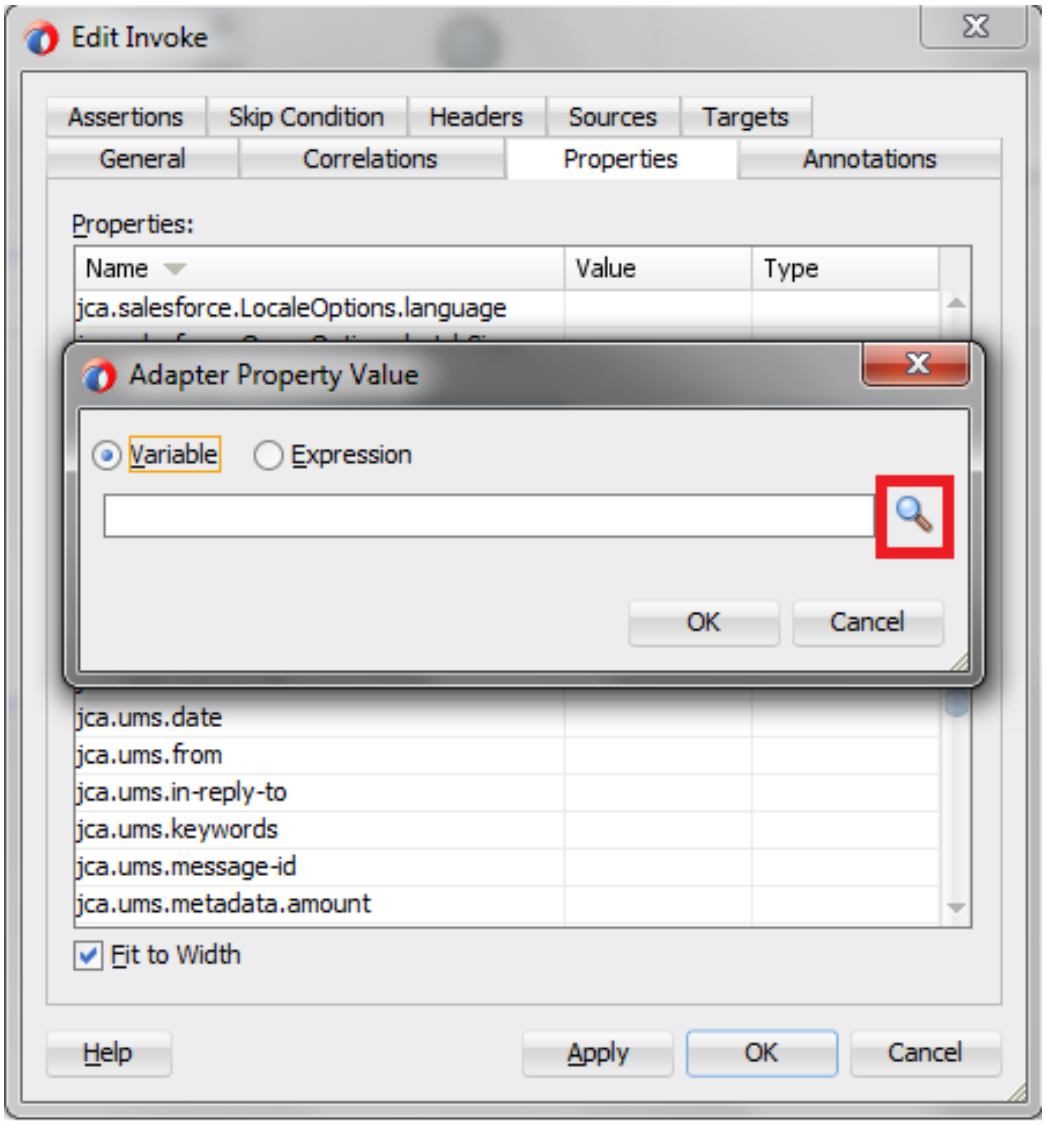
8. Go the properties tab and locate jca.salesforce.response.debugLog and double click on the "**…**" button under values as shown in Figure 11- 153.

Figure 11 - 153 Properties Tab



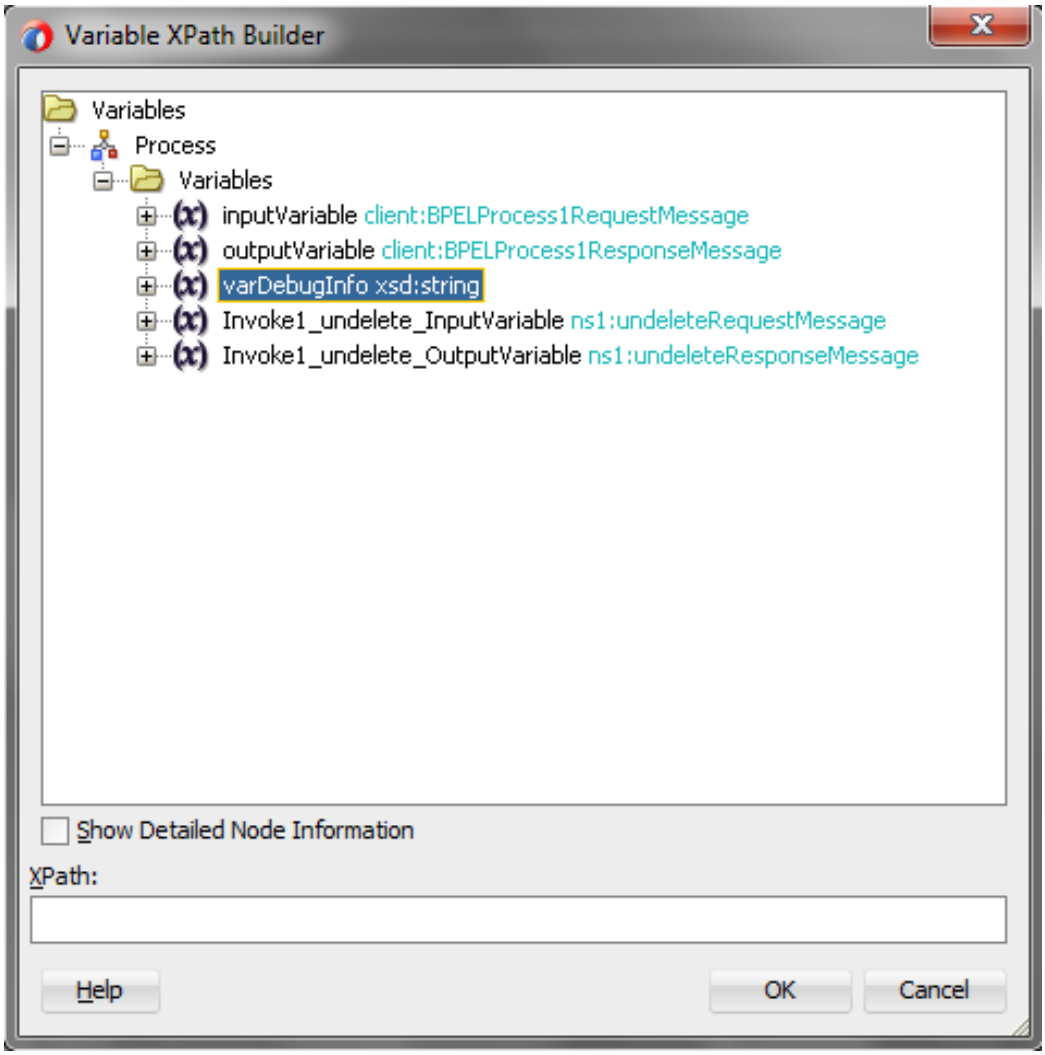
9. In the AdapterPropertyValue dialog, click on search for the variable as shown in Figure 11- 154.

Figure 11 - 154 Search Property Value



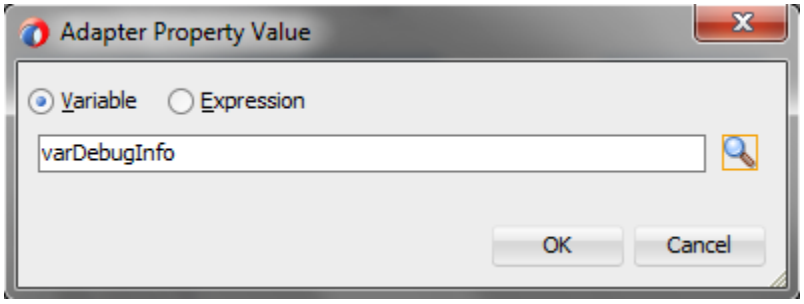
10. In the next dialog, click **varDebugInfo** variable and click **OK** as shown in Figure 11- 155.

Figure 11 - 155 Variable XPath Builder



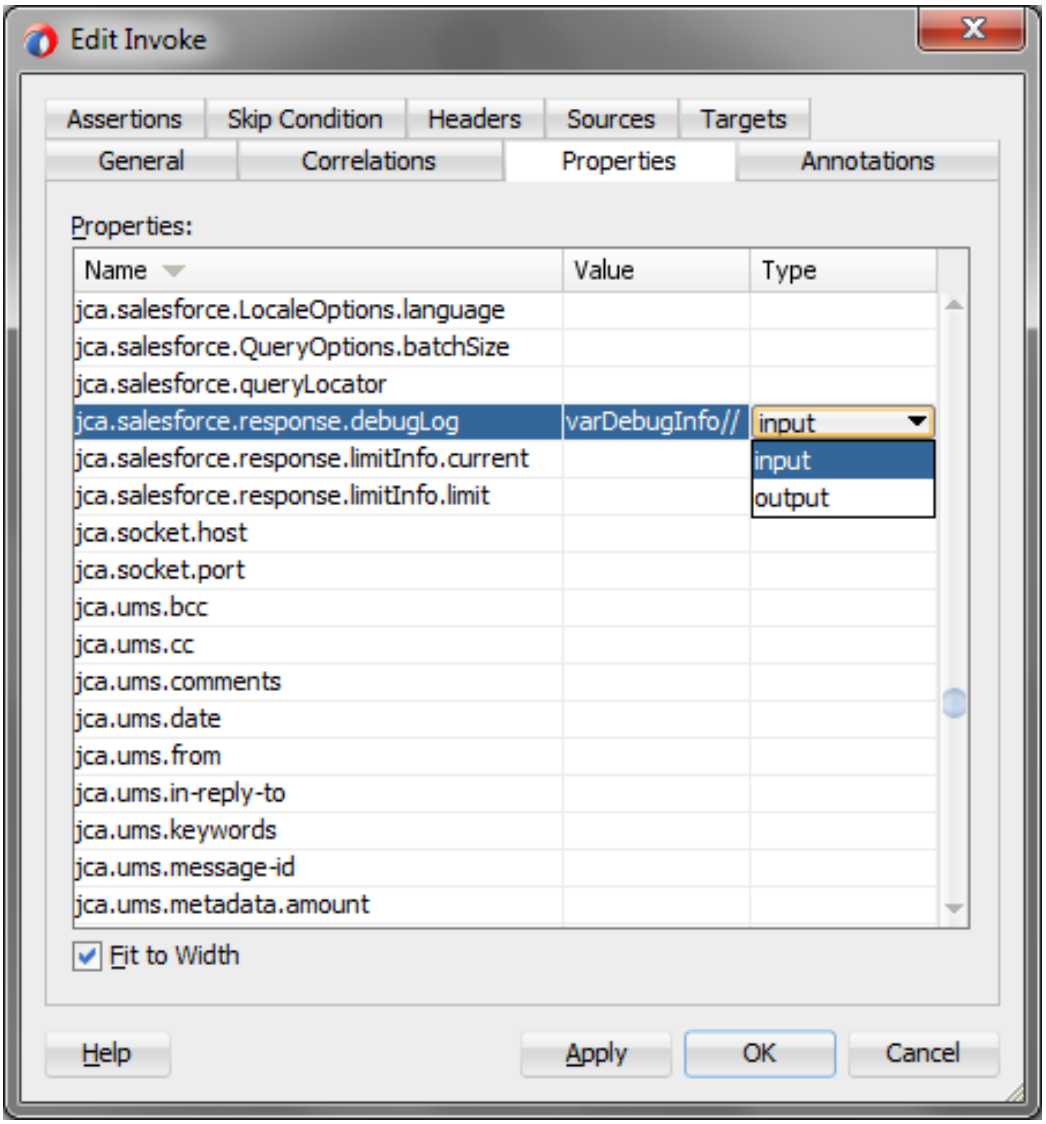
11. After the variable is selected, click **OK** as shown in Figure 11- 156.

Figure 11 - 156 Select variable



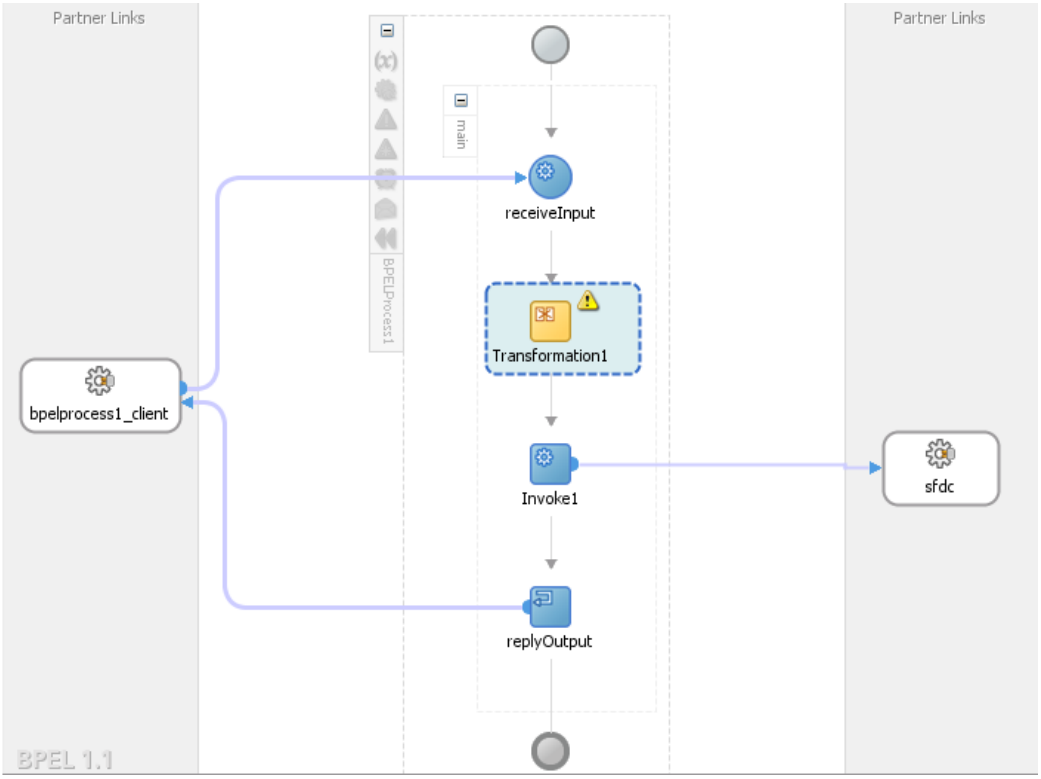
12. Change the type to **Output** and click **OK**, as shown in Figure 11- 157.

Figure 11 - 157 Select Output



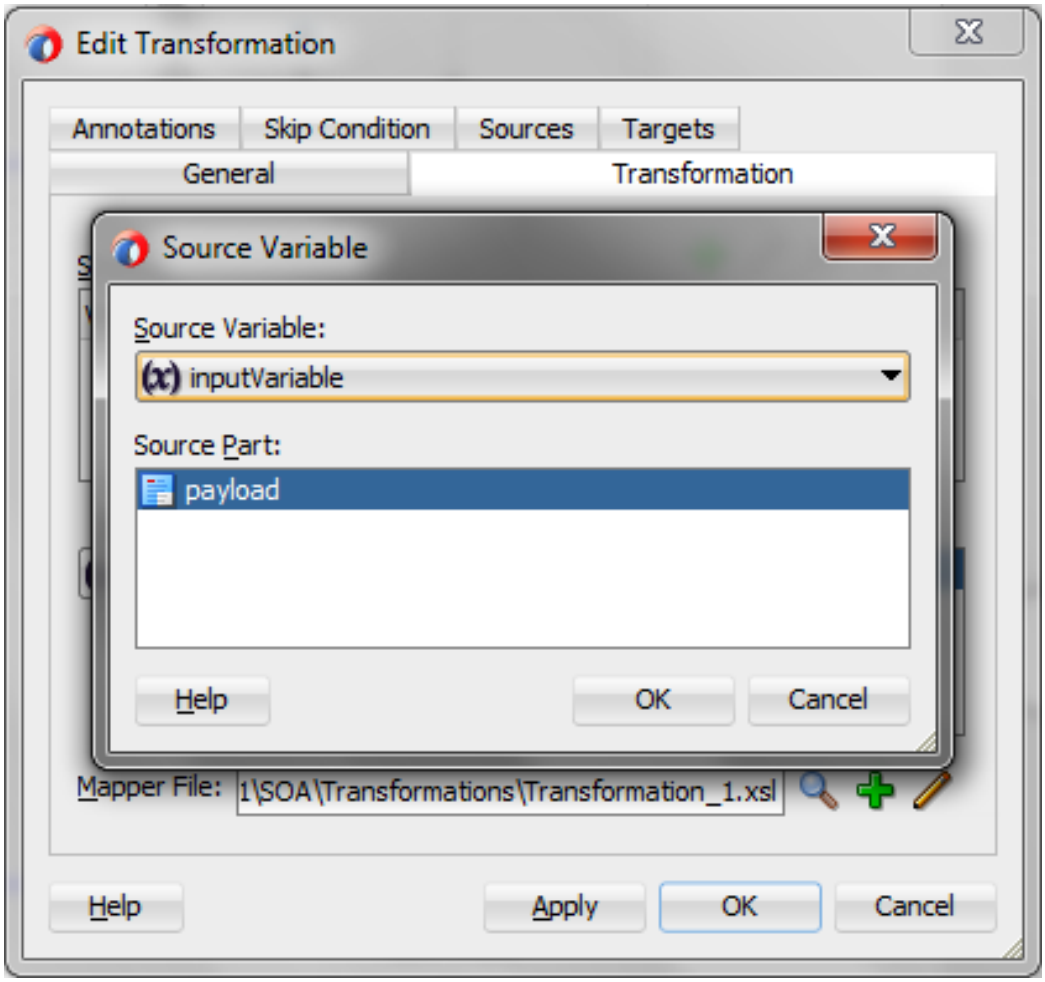
13. Introduce Transform activity right before invoke activity, as show in Figure 11- 158

Figure 11 - 158 Introduce transform activity



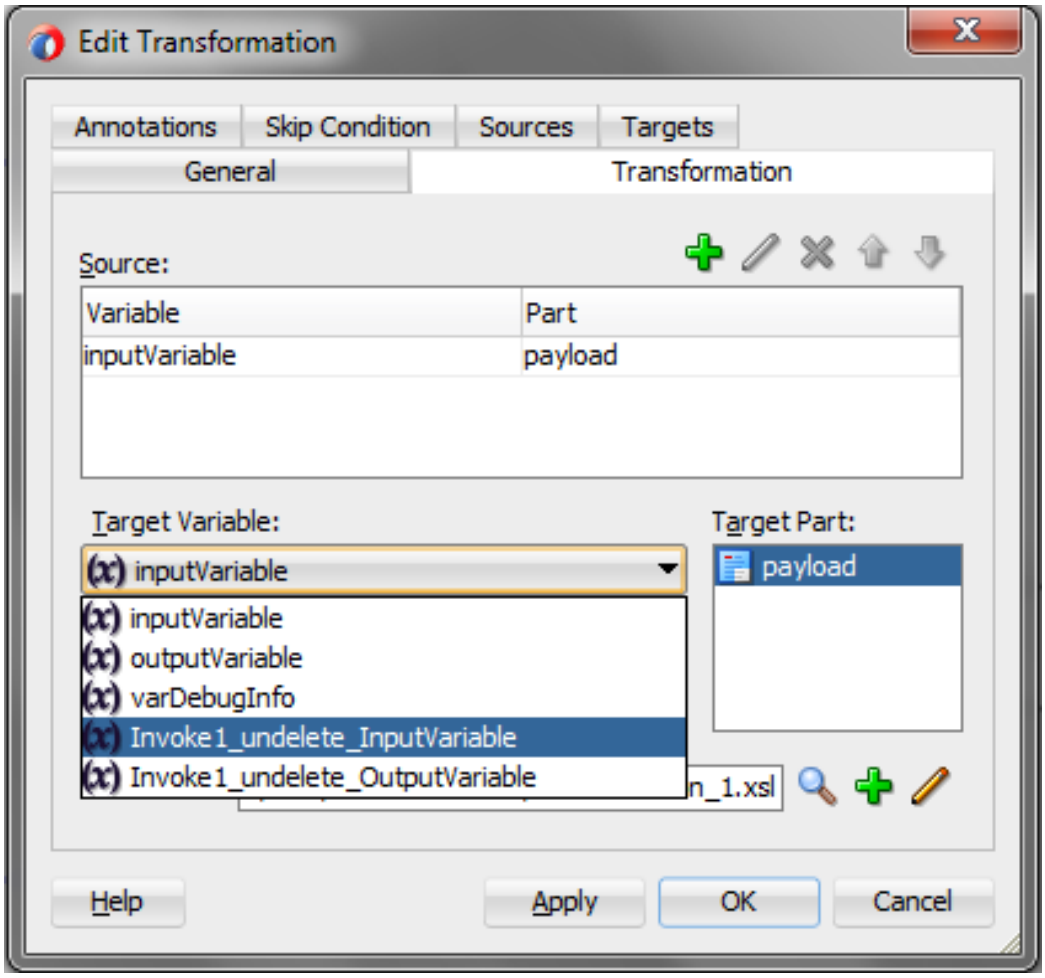
14. Double-click on **Transform1** and add the source variable as inputVariable, as shown in Figure 11- 159.

Figure 11 - 159 Transformation Tab



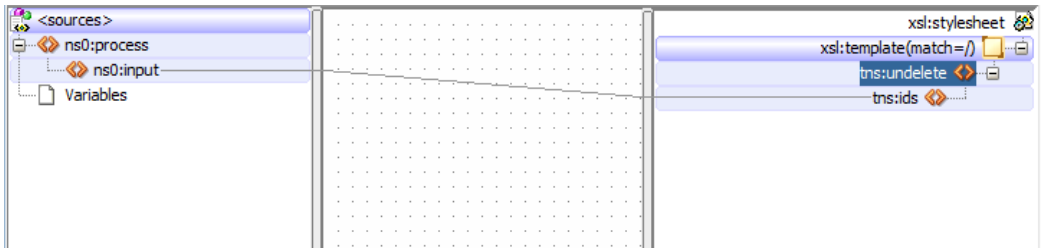
15. Add the target variable “Invoke1_undelete_InputVariable” as shown in Figure 11-160, and click **OK**.

Figure 11 - 160 Add target variable



16. Map the input to the source variable in the mapper file, as shown in Figure 11- 161.

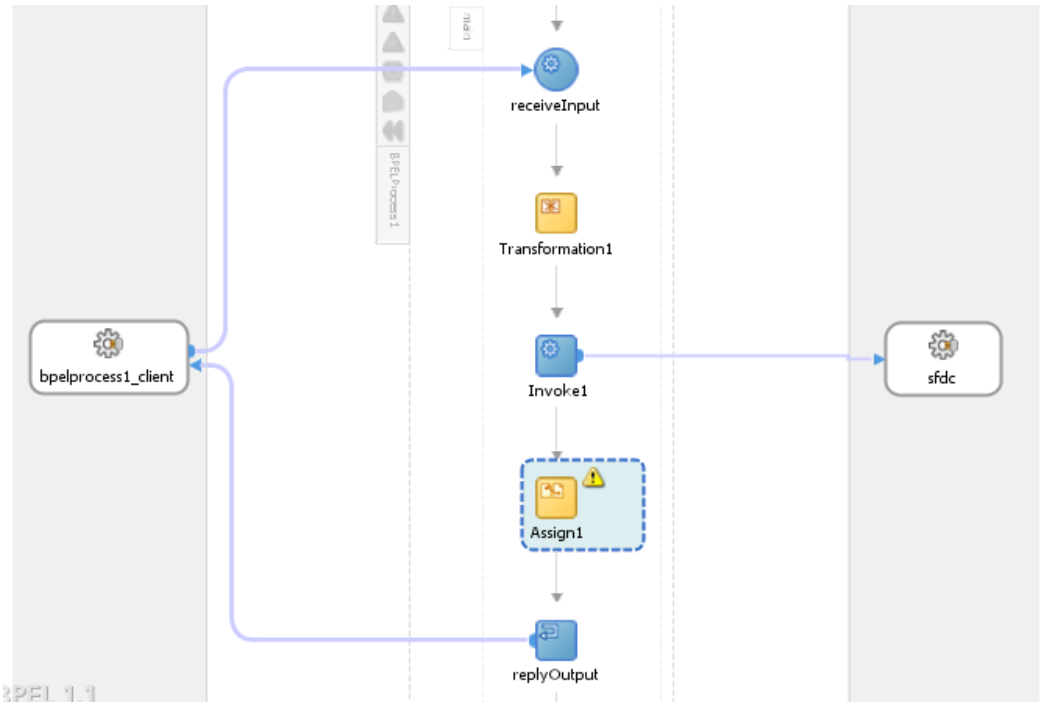
Figure 11 - 161 Map the input



17. Save all and move back to BPELProcess1.

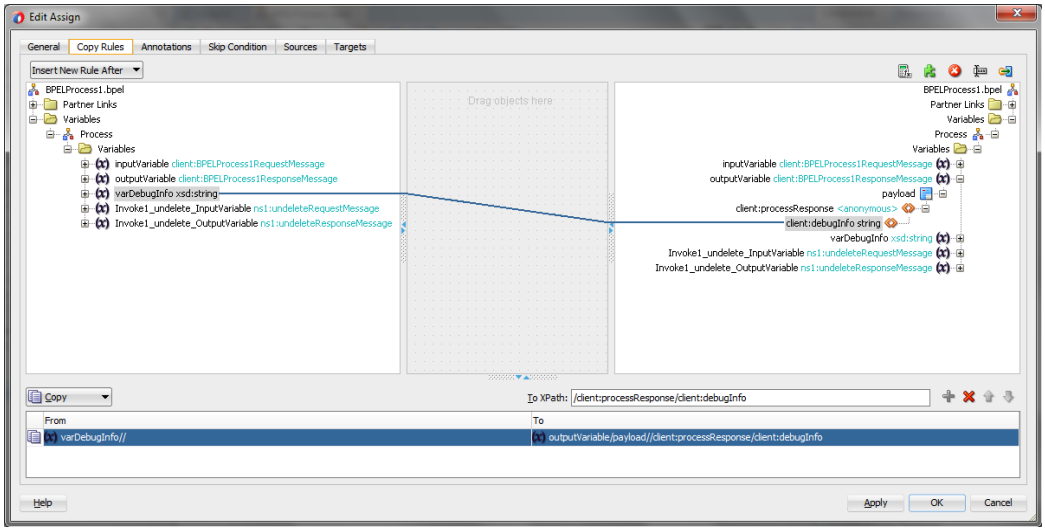
18. Introduce **Assign** activity right after the invoke activity, as shown in Figure 11- 162.

Figure 11 - 162 Introduce Assign activity



19. Inside the Assign activity, assign the value of "OutputVariable/payload//client:processResponse/client:debugInfo" variable to "varDebugInfo" variable, as shown in Figure 11- 163.

Figure 11 - 163 Wire the Assign activity



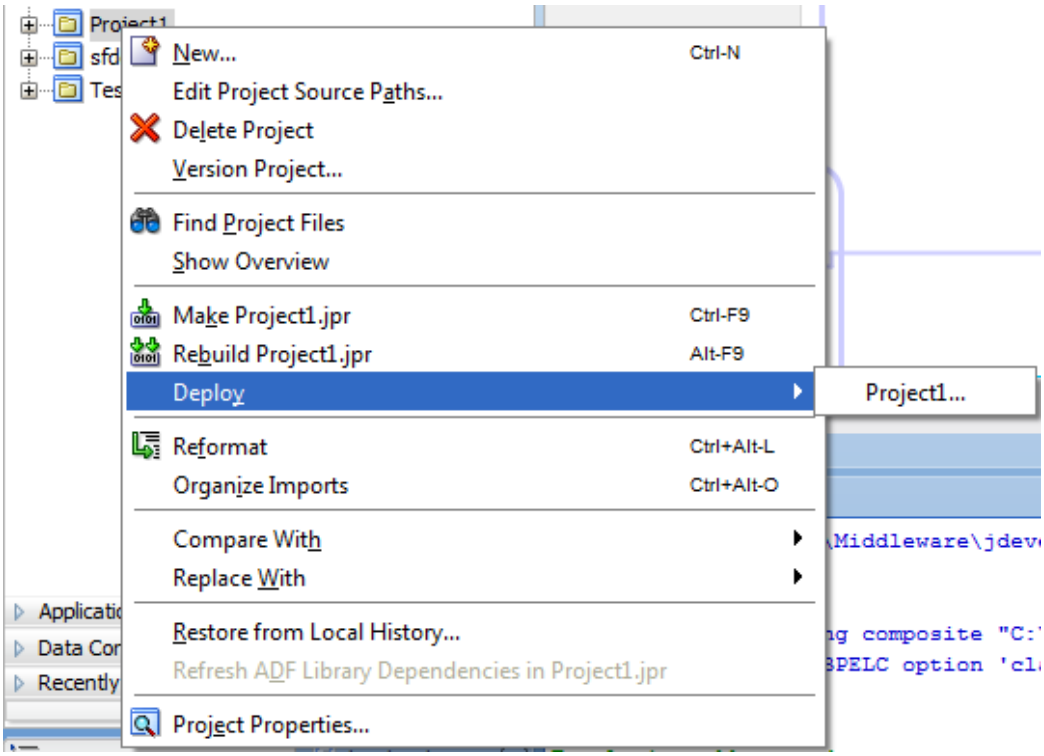
20. In the Assign activity, click on **Apply** and then **OK.**

Deploy the Composite

Perform the following steps to deploy the composite.

1. In the **Application Navigator** pane, right-click **Project1** and select **Deploy -> Project1**, as shown in Figure 11- 164.

Figure 11 - 164 Navigator Screen



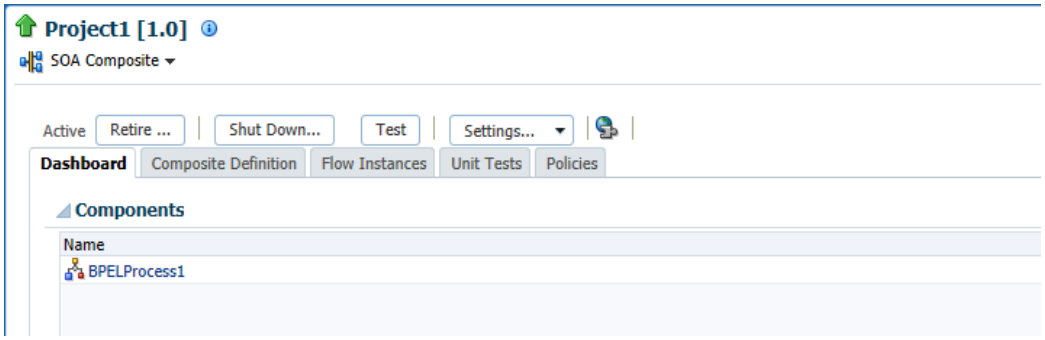
2. Select the **Deploy to Application Server** option and follow the instructions on the screen. Using this option you can deploy the composite on the Application server after providing the details of the server.

Test the Composite

Perform the following steps to test the composite.

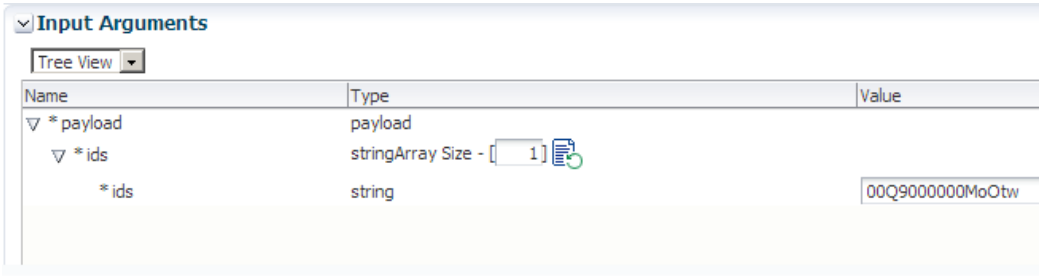
1. Login to Enterprise Manager Console for the server on which you deployed your project.
2. Open **Project1.**
3. Click the **Test** button to test the Web service, as shown in Figure 11- 165.

Figure 11 - 165 Test Tab



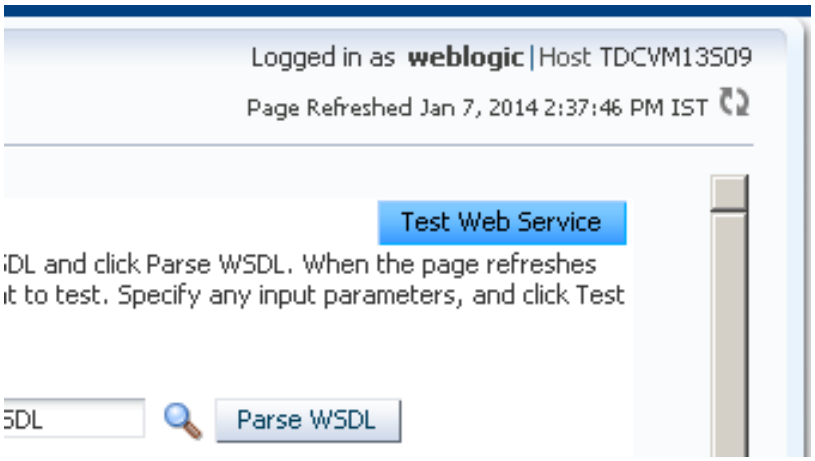
4. Enter the ID of the objects you wish to undelete, as shown in Figure 11- 166.

Figure 11 - 166 Enter ID



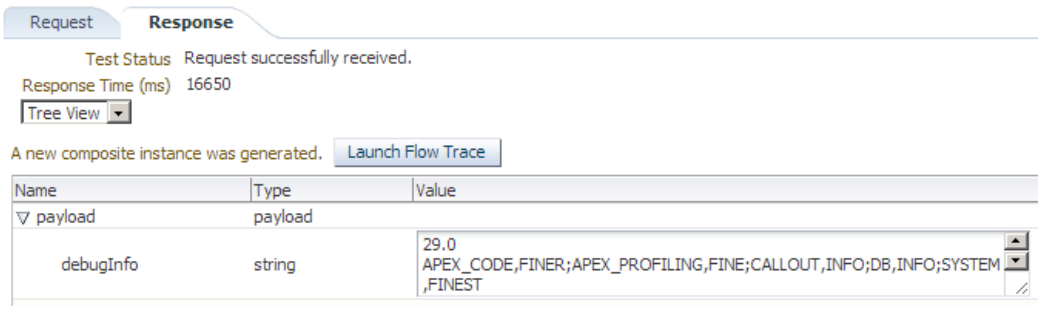
5. Click on **Test Web Service** button, as shown in Figure 11-167.

Figure 11 - 167 Test Web Service



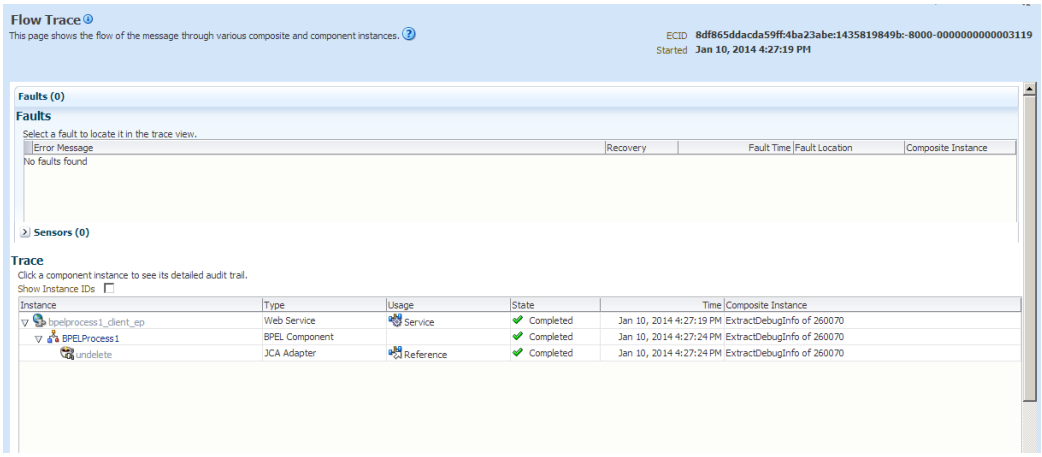
6. After successful execution, the response contains the expression that you provided in the last assign activity, as shown in Figure 11-168.

Figure 11 - 168 Execution Window



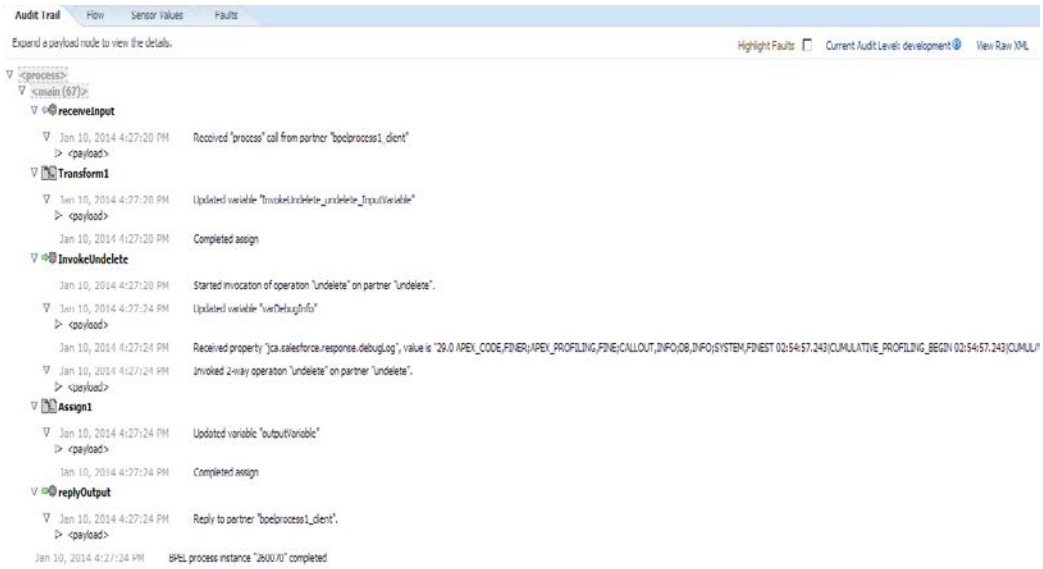
7. Click the **Launch Flow Trace** button to view the **Audit Trail**, as shown in Figure 11-169.

Figure 11 - 169 Launch Flow Trace



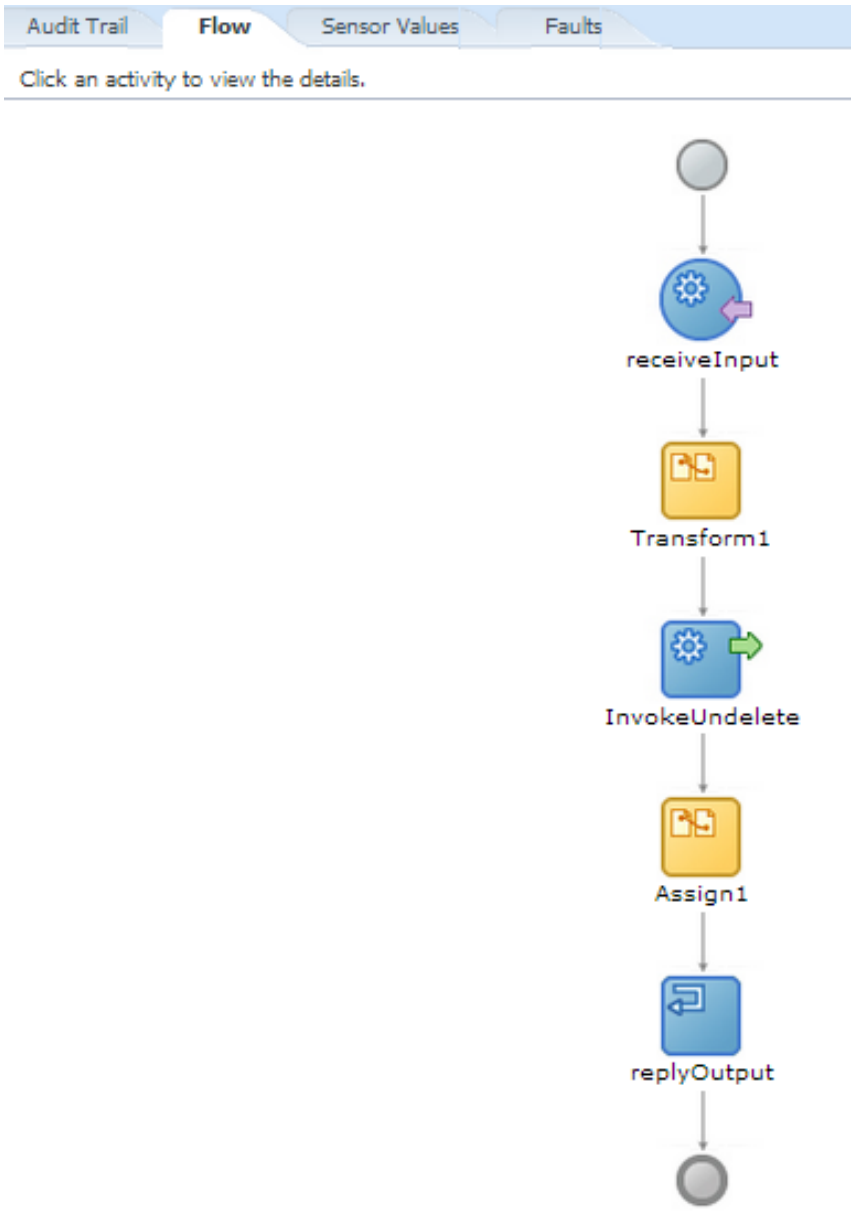
8. The **Audit Trail** will look like Figure 11-170.

Figure 11 - 170 Audit Trail



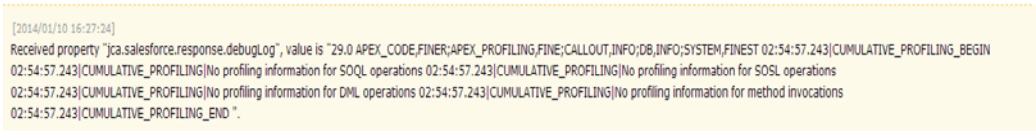
9. Click on **Flow** tab, the Flow tab will look like Figure 11-171.

Figure 11 - 171 Flow Tab



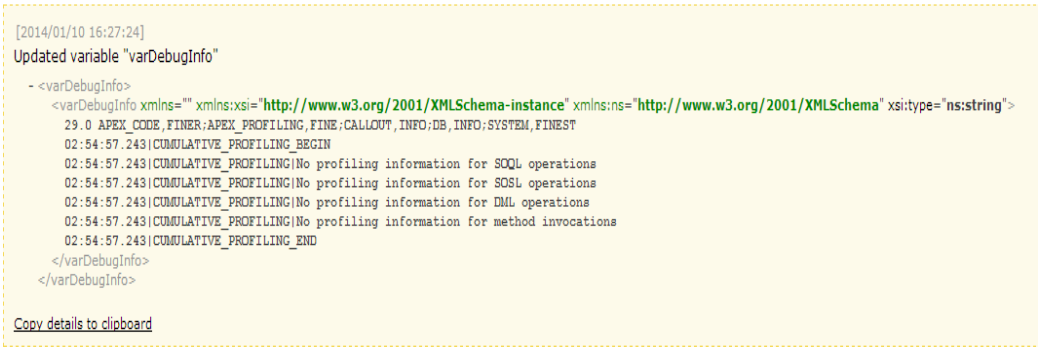
10. Click on invoke activity, here you can see how the value of “debugLog” property being returned, as shown in Figure 11-172.

Figure 11 - 172 Invoke Received Property


[2014/01/10 16:27:24]
Received property "jca.salesforce.response.debugLog", value is "29.0 APEX_CODE,FINER;APEX_PROFILING,FINE;CALLOUT,INFO;DB,INFO;SYSTEM,FINEST 02:54:57.243|CUMULATIVE_PROFILING_BEGIN
02:54:57.243|CUMULATIVE_PROFILING|No profiling information for SOQL operations 02:54:57.243|CUMULATIVE_PROFILING|No profiling information for SOSL operations
02:54:57.243|CUMULATIVE_PROFILING|No profiling information for DML operations 02:54:57.243|CUMULATIVE_PROFILING|No profiling information for method invocations
02:54:57.243|CUMULATIVE_PROFILING_END ".

11. You can also see how the value of “debugLog” property is written to varDebugInfo value, as shown in Figure 11-173.

Figure 11 - 173 Invoke updated variable



Part II

BPM Use Cases

This section is intended to walk you through use cases specific to integrating to Salesforce via Oracle BPM. The examples provided in this chapter would be beneficial to a developer creating BPM processes to integrate to salesforce. This chapter is subdivided into the sections:

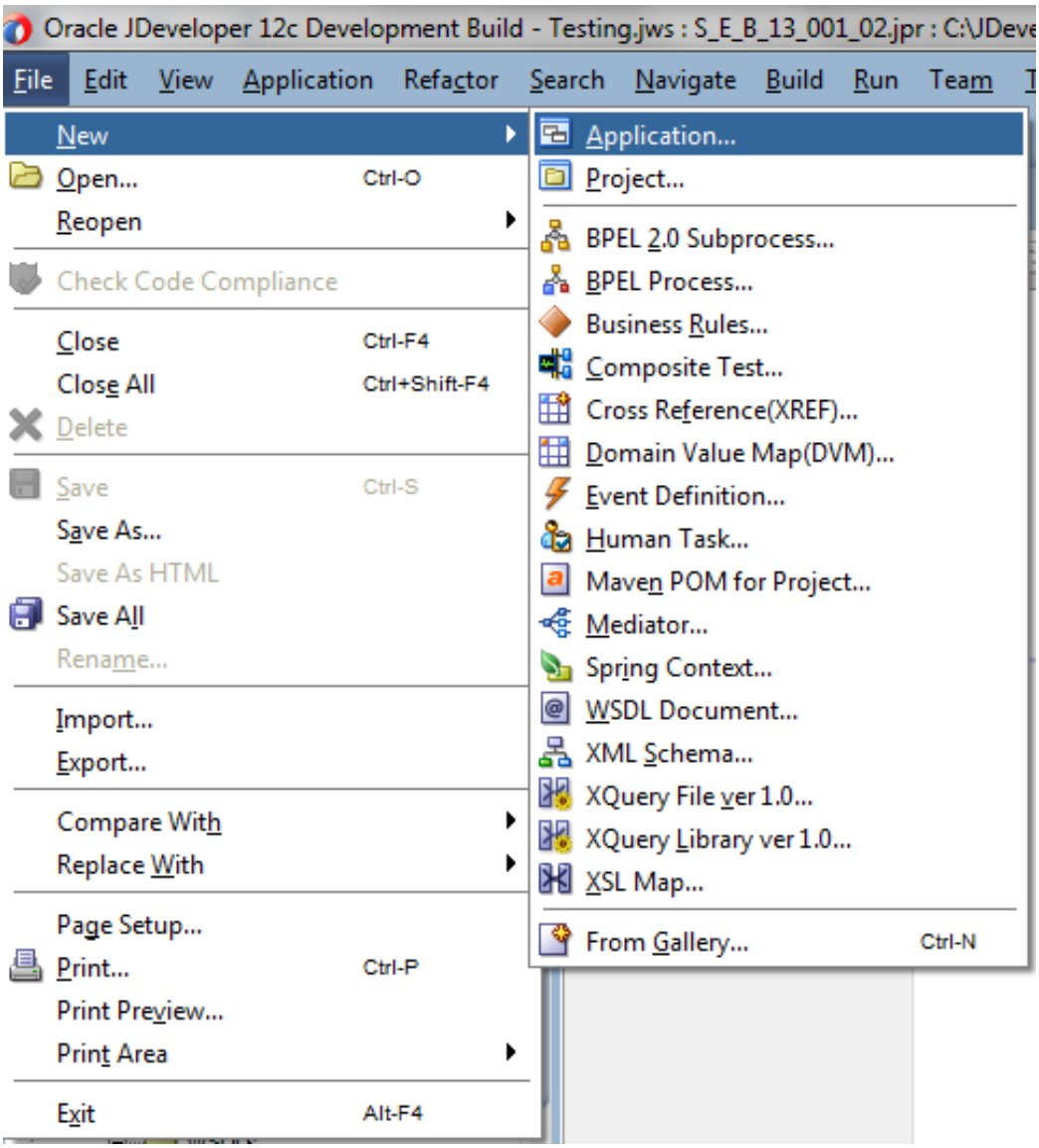
- Section 11.6, "Define Composite for BPM"
- Section 11.7, “Configure Oracle Cloud Adapter for Salesforce.com”
- Section 11.8, “Integration with BPM”
- Section 11.9, “Deploy the Composite”
- Section 11.10, “Test the Composite”

11.6. Define Composite for BPM

Perform the following steps to define a composite for BPM.

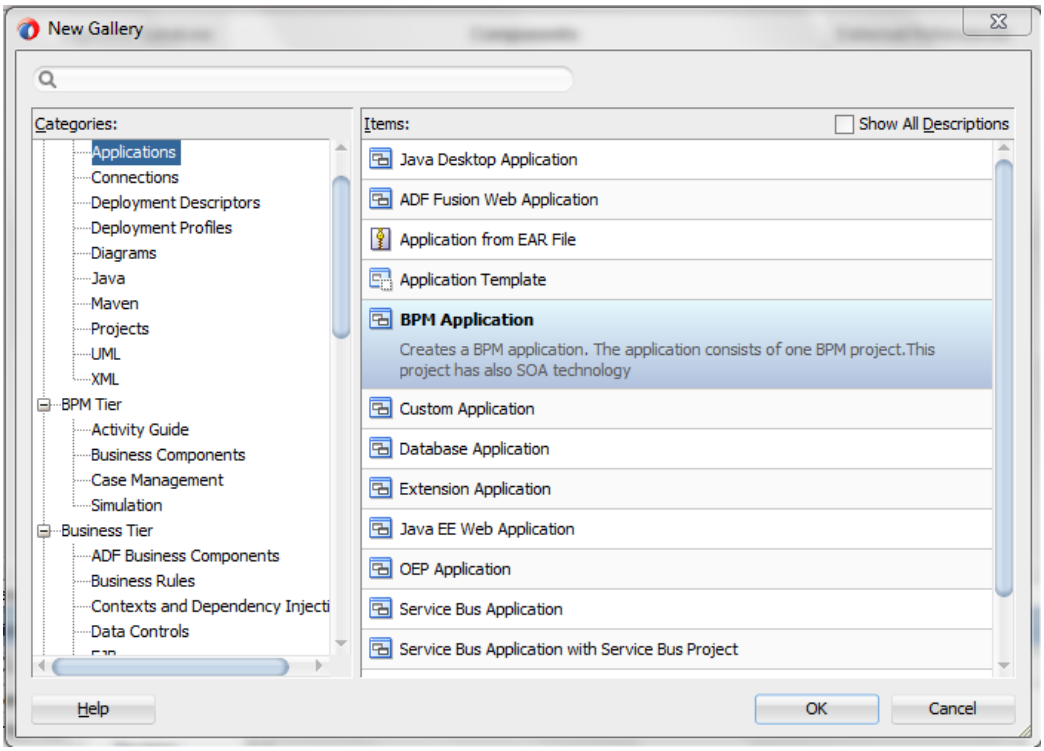
1. In the **File** menu of JDeveloper, click **New** and select **Application**.

Figure 11 - 1 Navigation Window



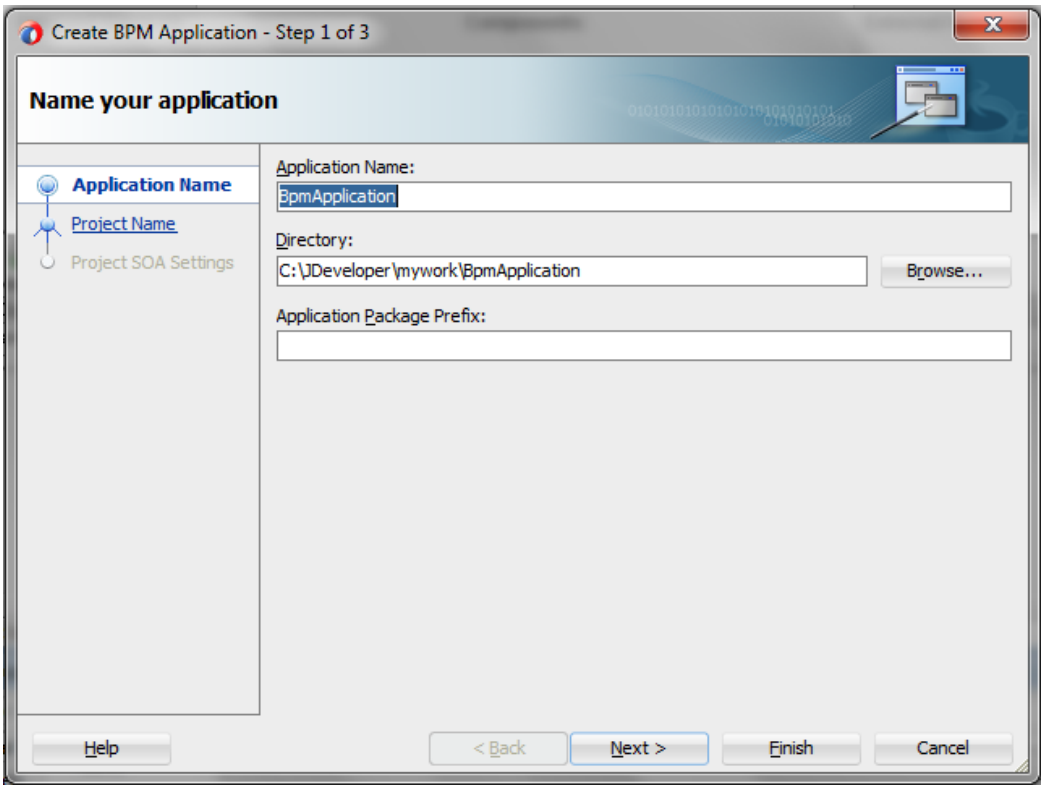
2. The **New Gallery** page is displayed. Select **BPM Application** from the **Items** list, as shown in the screenshot below.

Figure 11 - 2 Create BPM Application



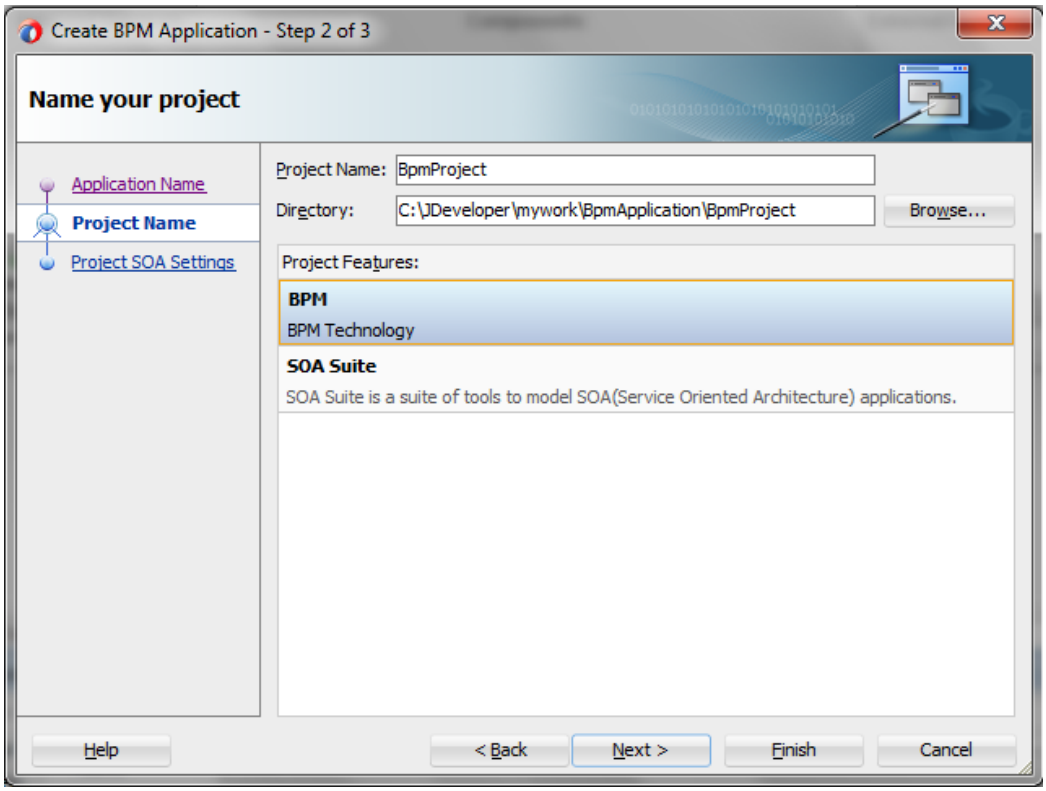
3. Provide a suitable name to your application, as shown in screenshot below.

Figure 11 - 3 Name your application



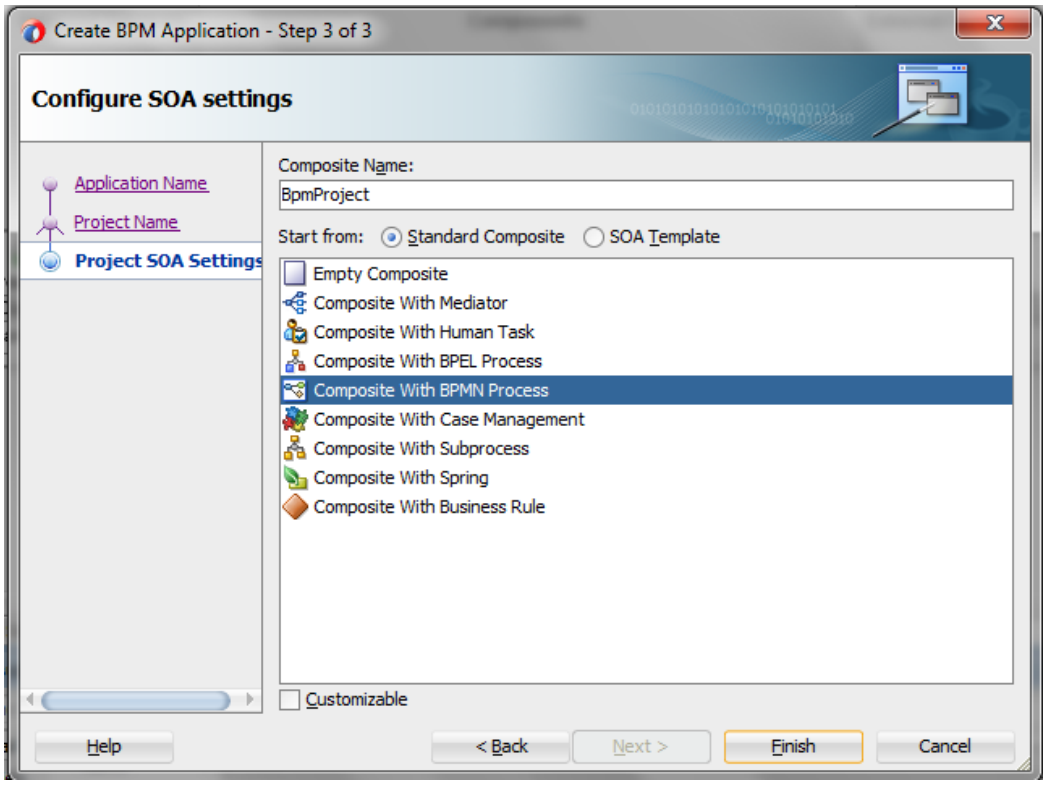
4. Click **Next** and provide a suitable name to your project, as shown in screenshot below.

Figure 11 - 4 Name your project



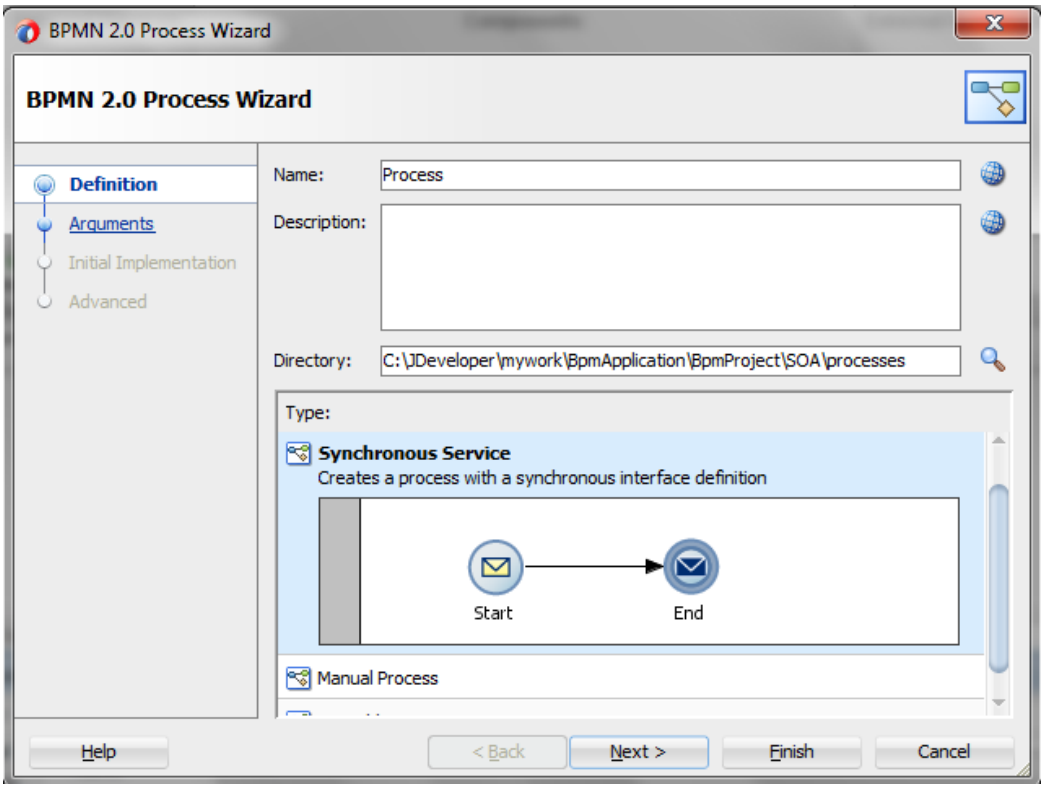
5. Click **Next.**
6. Select **Composite with BPMN Process** from **Standard Composite** list, as shown in screenshot below.

Figure 11 - 5 Configure BPM Setting



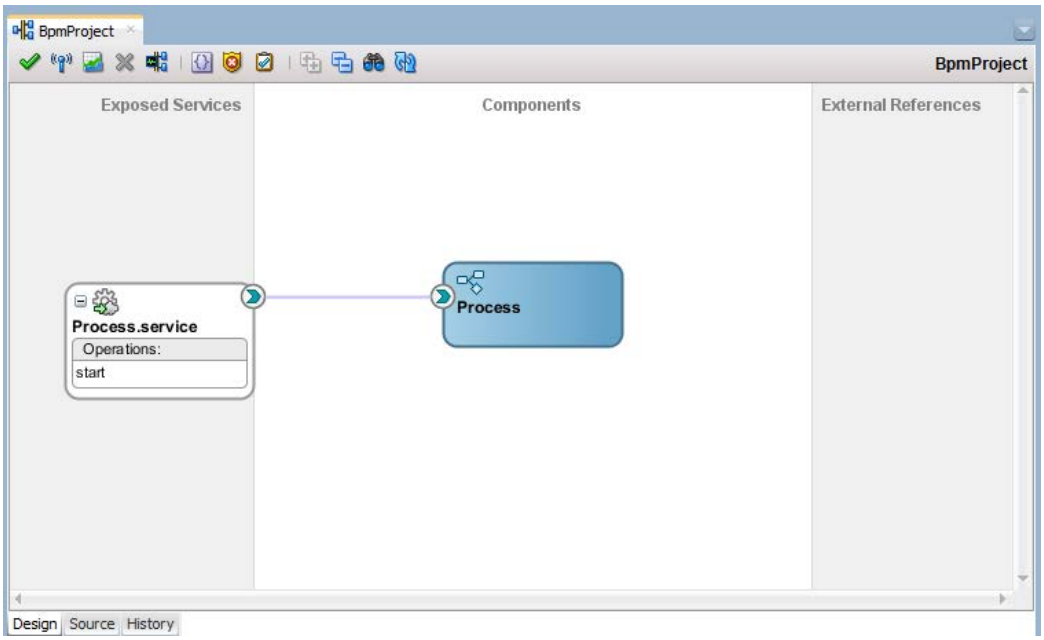
7. Click **Finish**.
8. Select the **Synchronous Service** from **Type** section and click **Finish**, as shown in screenshot below.

Figure 11 - 6 Configure BPM Setting



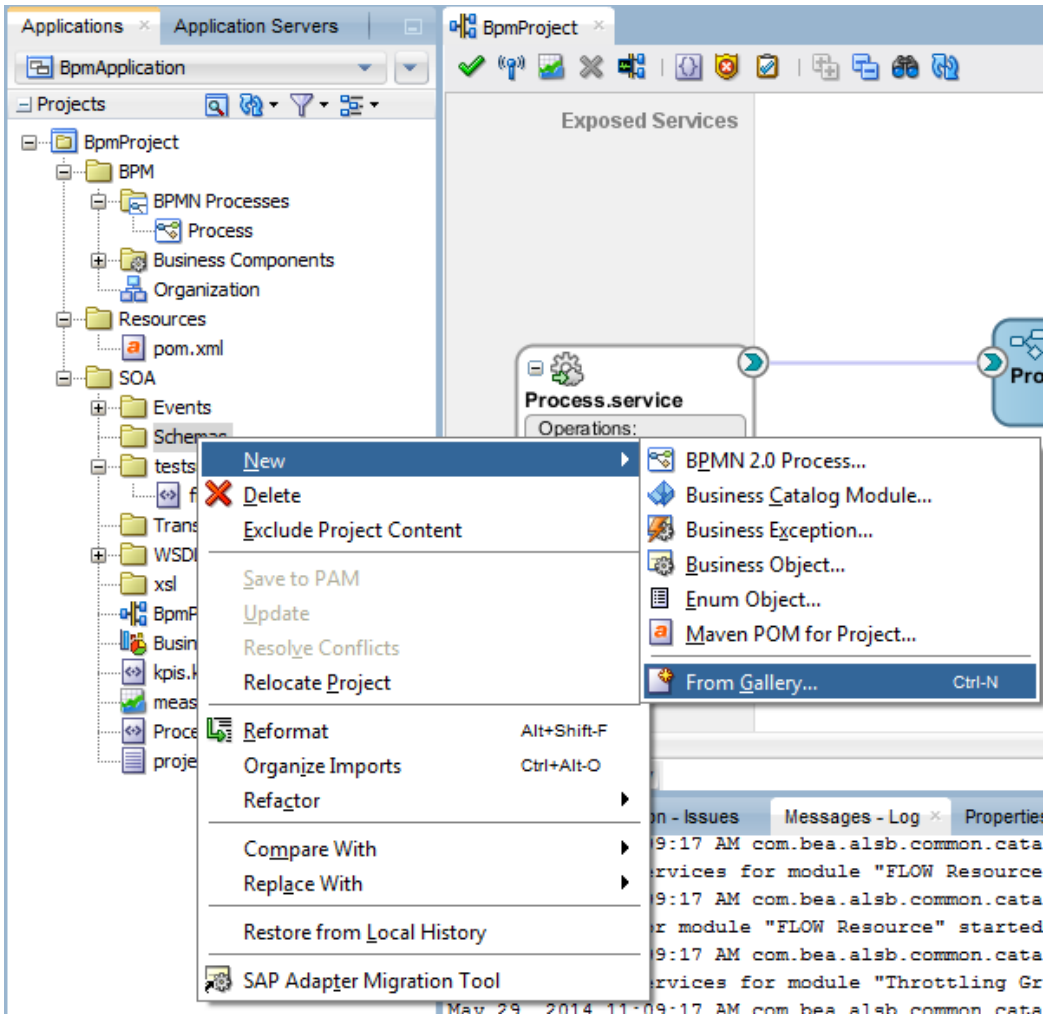
9. The Composite.xml looks like the one displayed in the screenshot below

Figure 11 - 7 Composite.xml



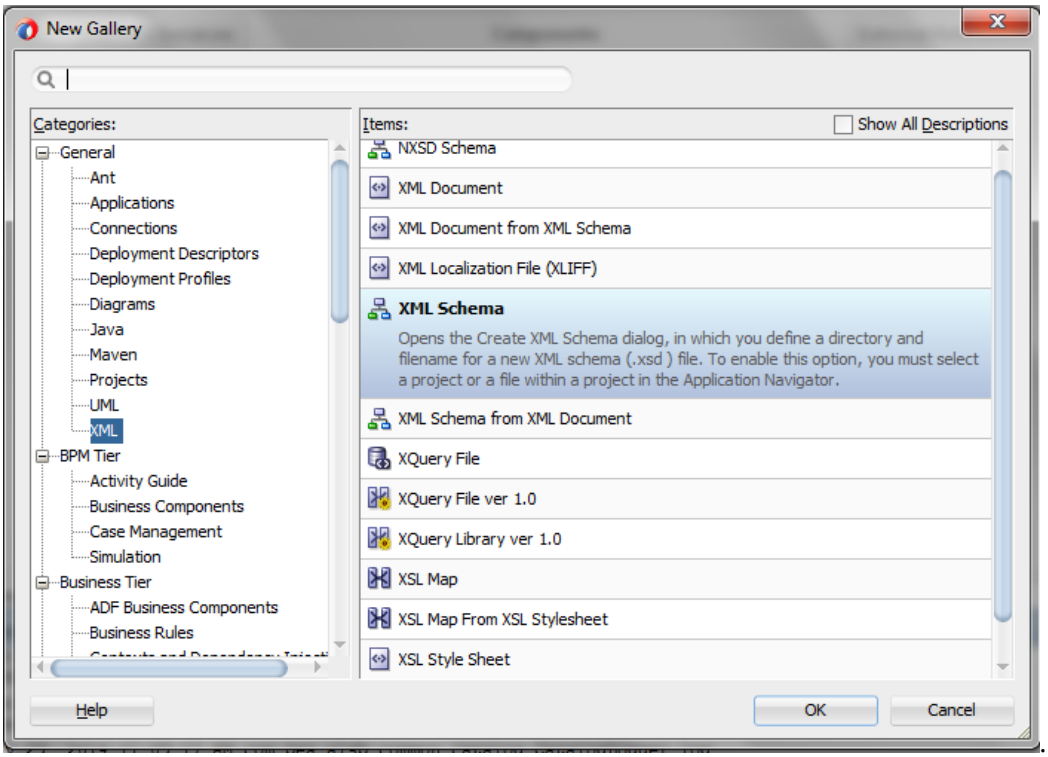
10. Next we would create the schema for our process. In the **Application Navigator**, under your project folder, right-click on the schema folder and select **New →From Gallery** as shown in screenshot below.

Figure 11 - 8 Application Navigator



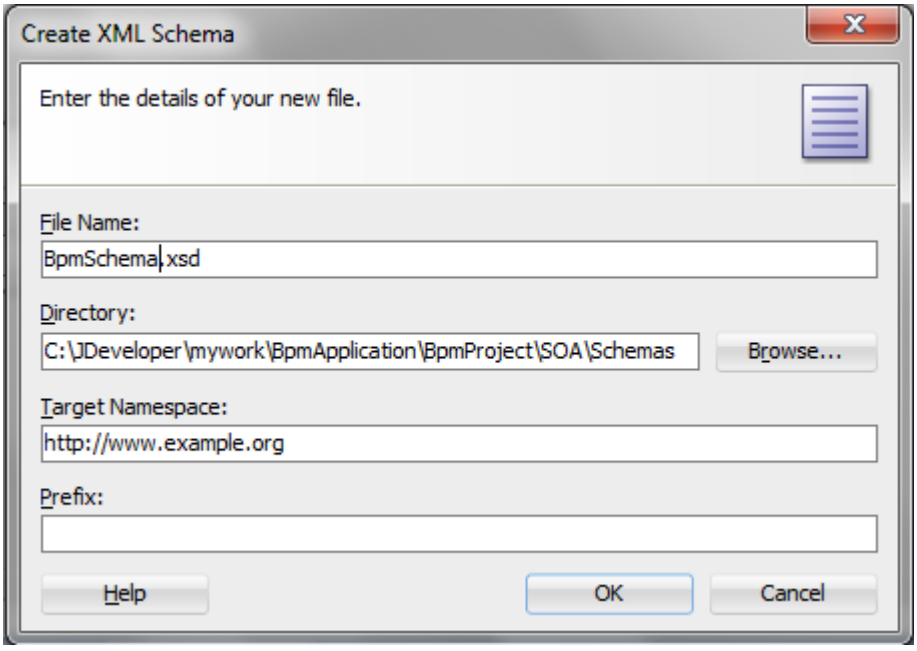
11. The **New Gallery** page is displayed. Select **XML Schema** from the **Items** list, and click **OK** as shown in screenshot below.

Figure 11 - 9 Create XML Schema



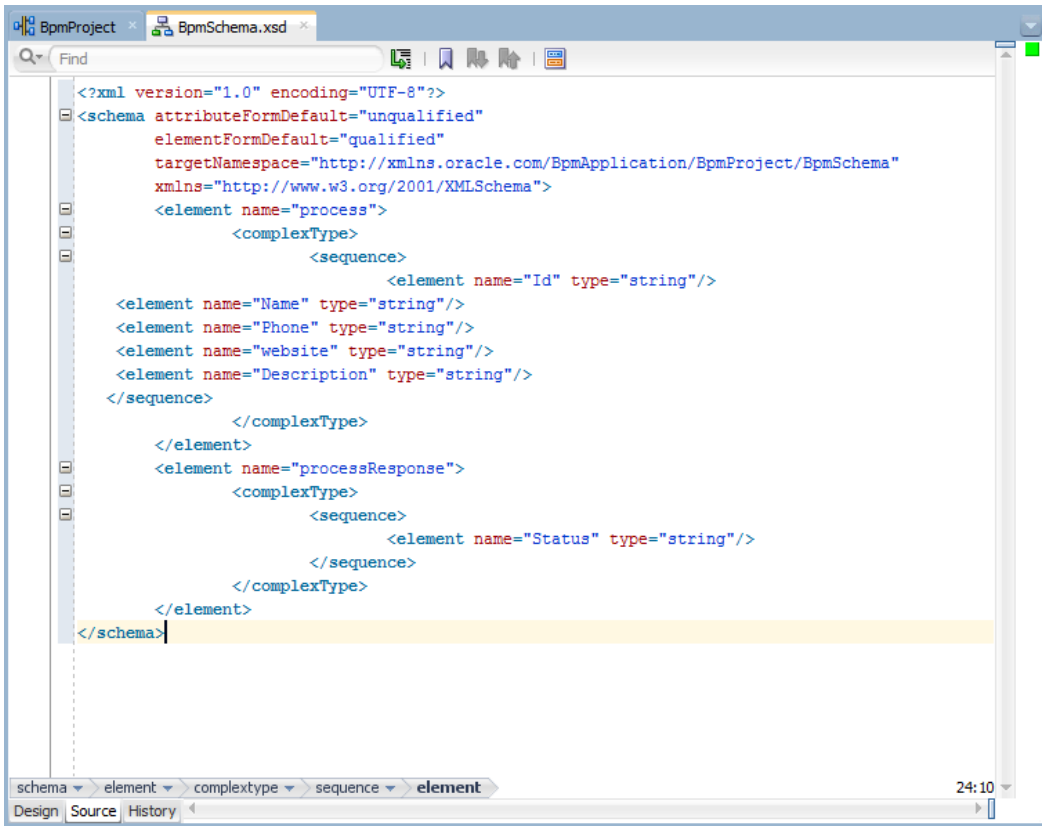
12. The **Create XML Schema** page is displayed, provide a suitable name to the schema and click **OK,** as shown in screenshot below.

Figure 11 - 10 Name Your Schema



13. Edit the schema file as per the business requirement. It is the responsibility of the front-end application to impose the data validations and to ensure that the input sent to SFDC via SOA is correct. The structure of the schema used in this use case is shown in the screenshot below.

Figure 11 - 11 Schema.xsd

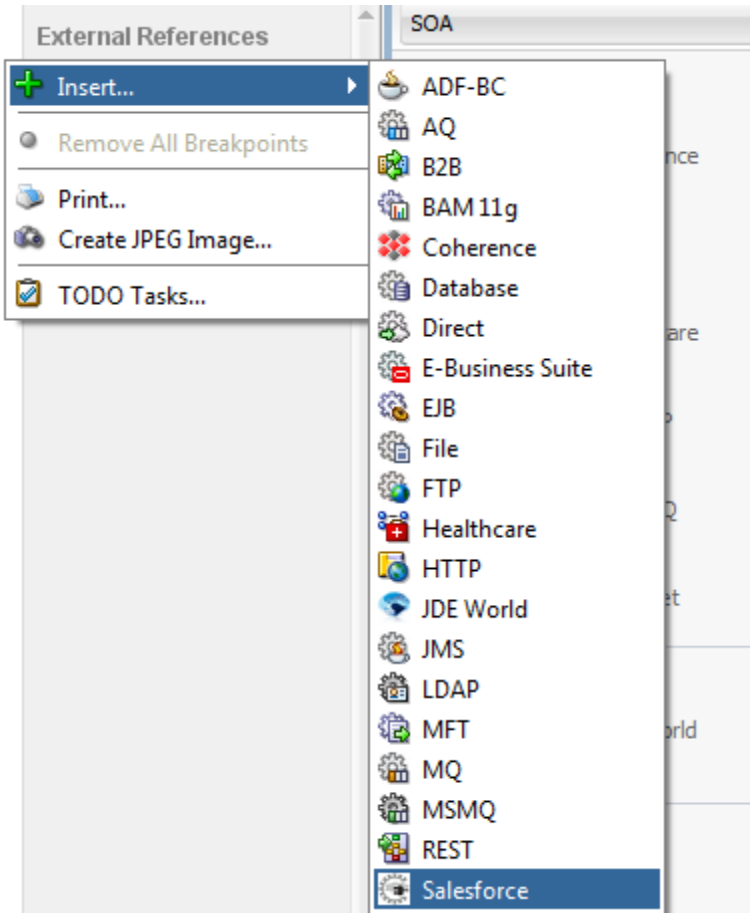


11.7. Configure Oracle Cloud Adapter for Salesforce.com

Perform the following steps to configure a new Oracle Cloud Adapter for Salesforce.com:

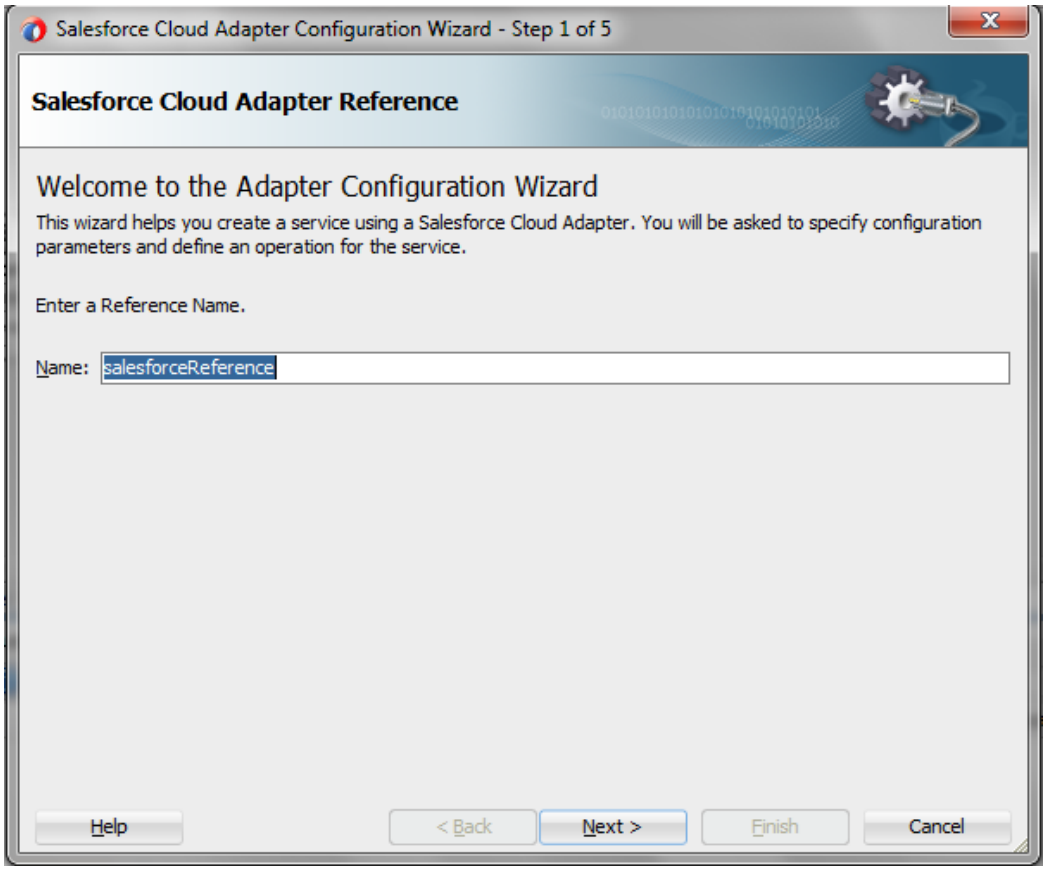
23. In **External References** swim-lane of the composite.xml file, right-click and select **Salesforce** adapter, as shown in the screenshot below.

Figure 11 - 12 Salesforce Adapter



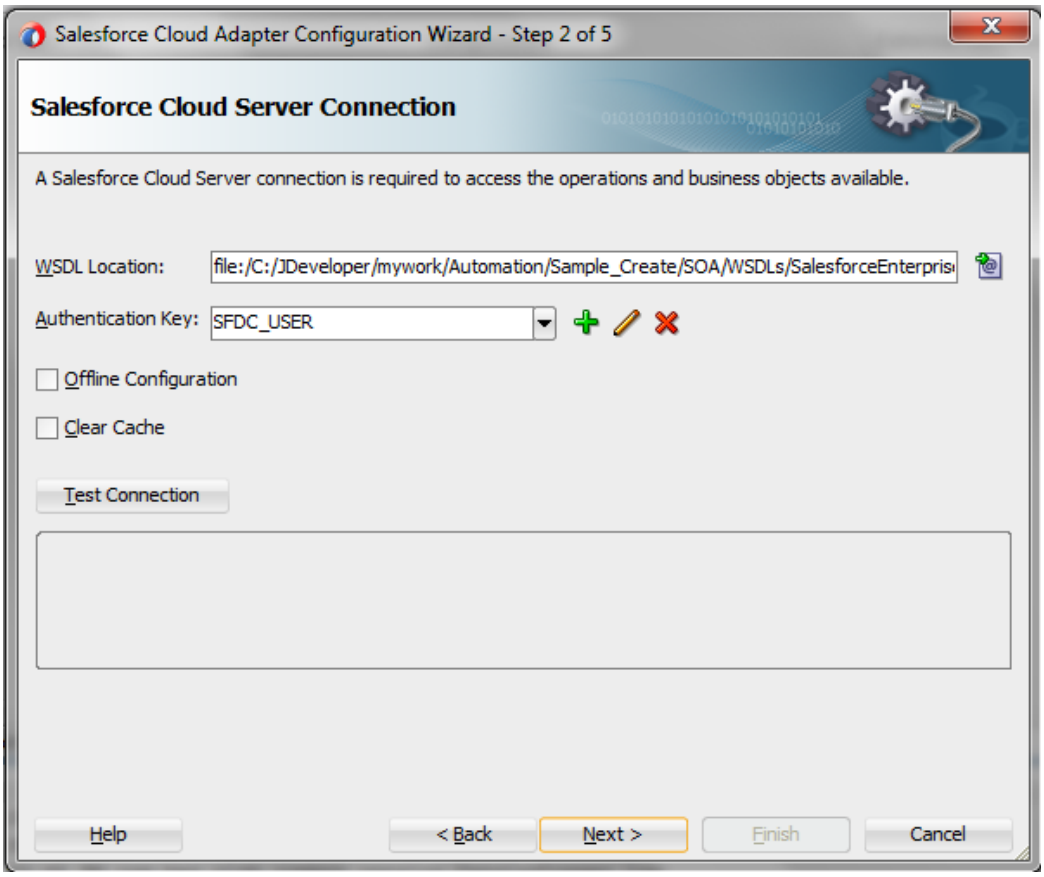
24. The Salesforce Cloud Adapter Configuration Wizard - Welcome page is displayed, as shown in the screenshot below.

Figure 11 - 13 Welcome Page



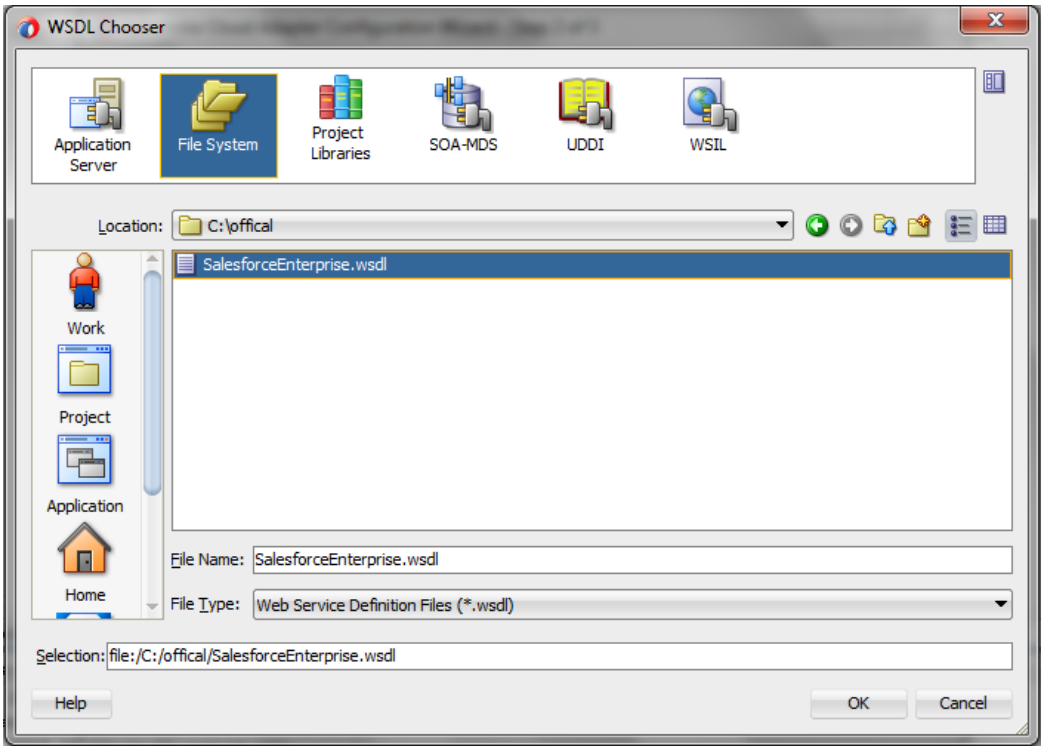
25. Click **Next**.
26. The **Salesforce Cloud Server Connection** page is displayed. The **WSDL Location** and **Authentication Key** text boxes are auto-filled. It picks up these values from the cache. You can re-enter these values. If you want to use a different value, click the **Find existing WSDLs** icon, which is located to the right of the **WSDL Location** field, as shown in the screenshot below.

Figure 11 - 14 Salesforce Cloud Server Connection Page



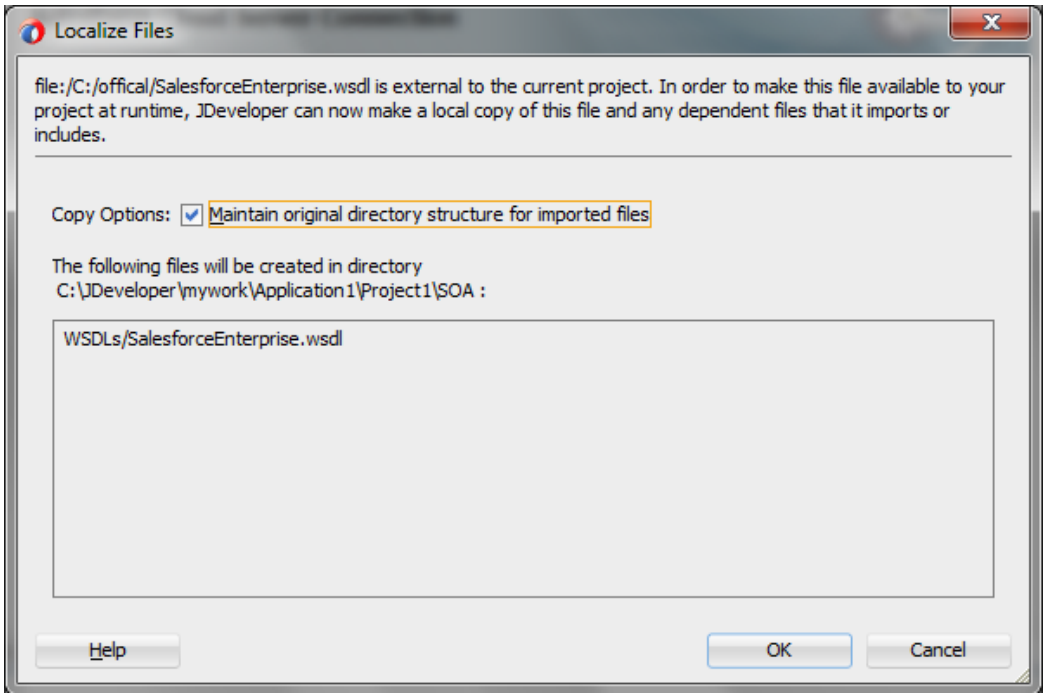
27. The **WSDL Chooser** dialog is displayed, browse and select the downloaded Enterprise WSDL and click **OK**, as shown in the screenshot below.

Figure 11 - 15 SOA Resource Browser



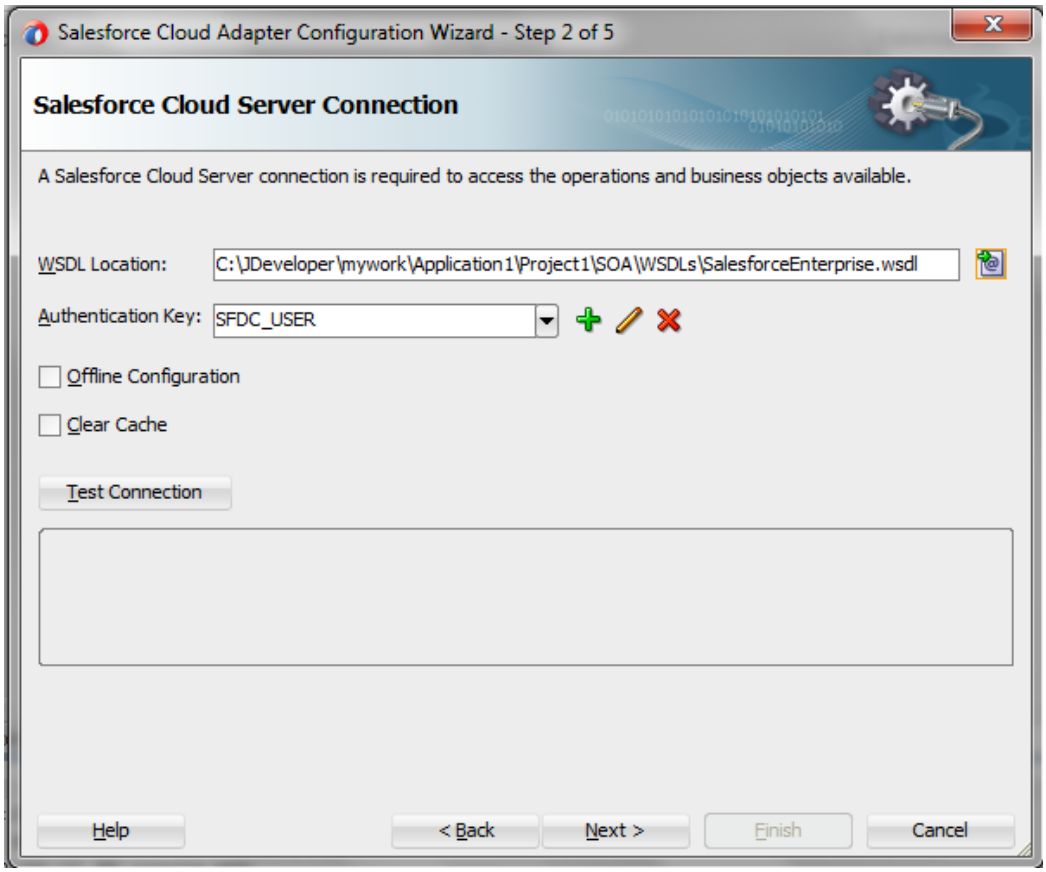
28. Click **OK**. The following screen appears as shown in the screenshot below.

Figure 11 - 16 Localize Files Dialog



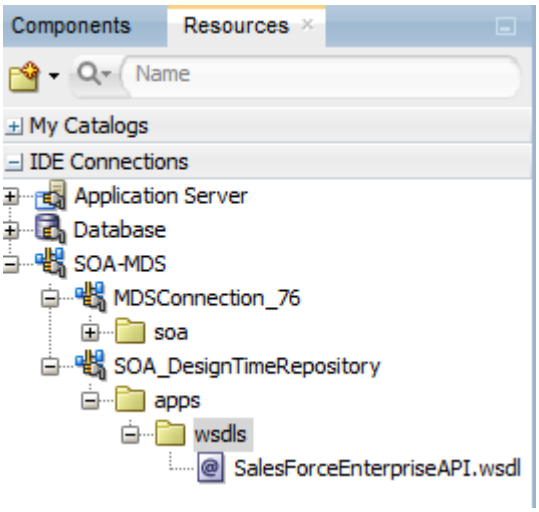
29. Click **OK**. It will be returned to the Salesforce Cloud Server Connection page.

Figure 11 - 17 Salesforce Cloud Server Connection Page



Note that as an alternative, you can store WSDL at an MDS location and access it, as shown in the screenshot below.

Figure 11 - 18 SOA Resource Browser

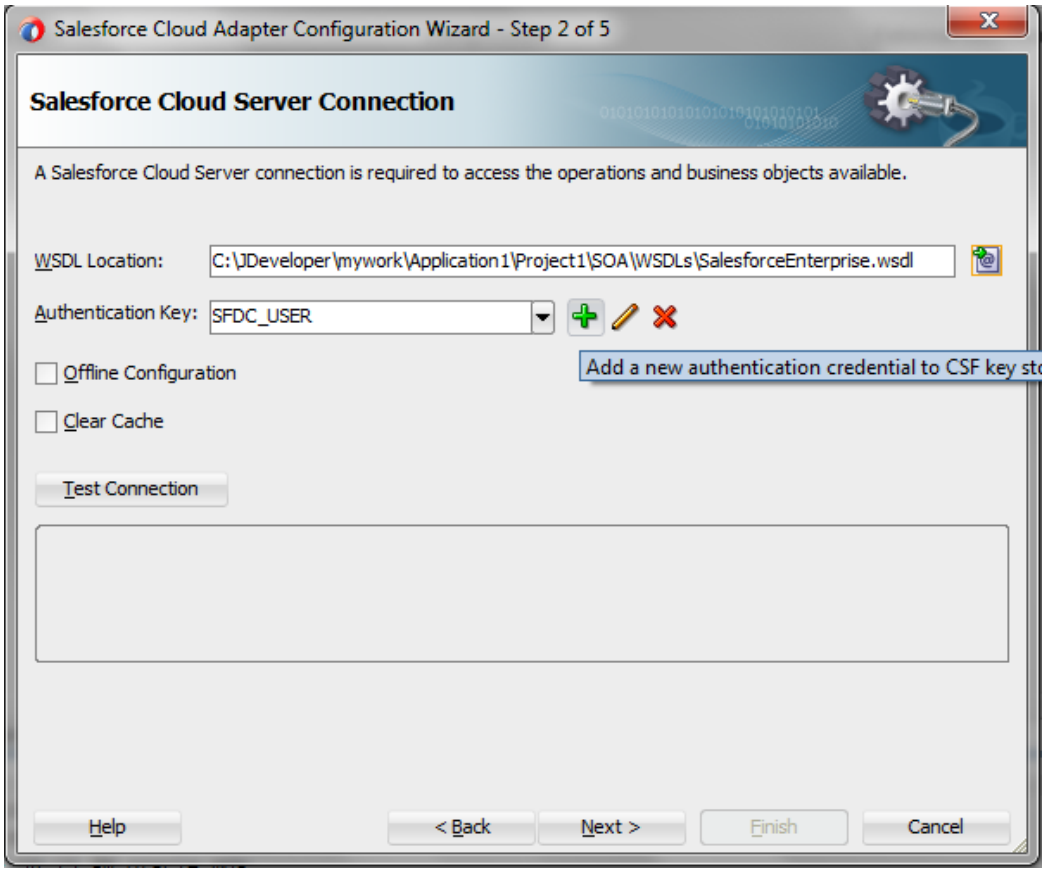


30. Proceed to **IDE Connections → SOA-MDS**. Select the appropriate SOA-MDS connection where you placed the Enterprise WSDL. Select the WSDL file to be used in the adapter configuration and click **OK**.
31. The WSDL location should be of the form: 'oramds:/apps/SOA/WSDLs/Integration/SalesforceReference.wsdl', as shown in the screenshot below.

Figure 11 - 19 WSDL location

32. Click **OK**.
33. Click **"+"** button to create a new Authentication Key, as shown in the screenshot below.

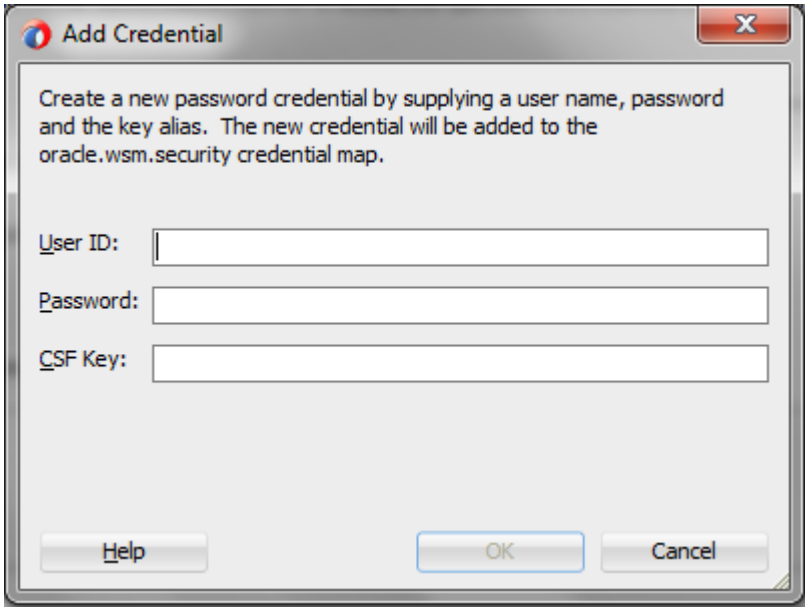
Figure 11 - 20 Create a New Authentication Key



34. The **Add Credential** dialog box is displayed, as shown in the screenshot below. Provide a suitable name and the Salesforce.com credentials and click **OK**.

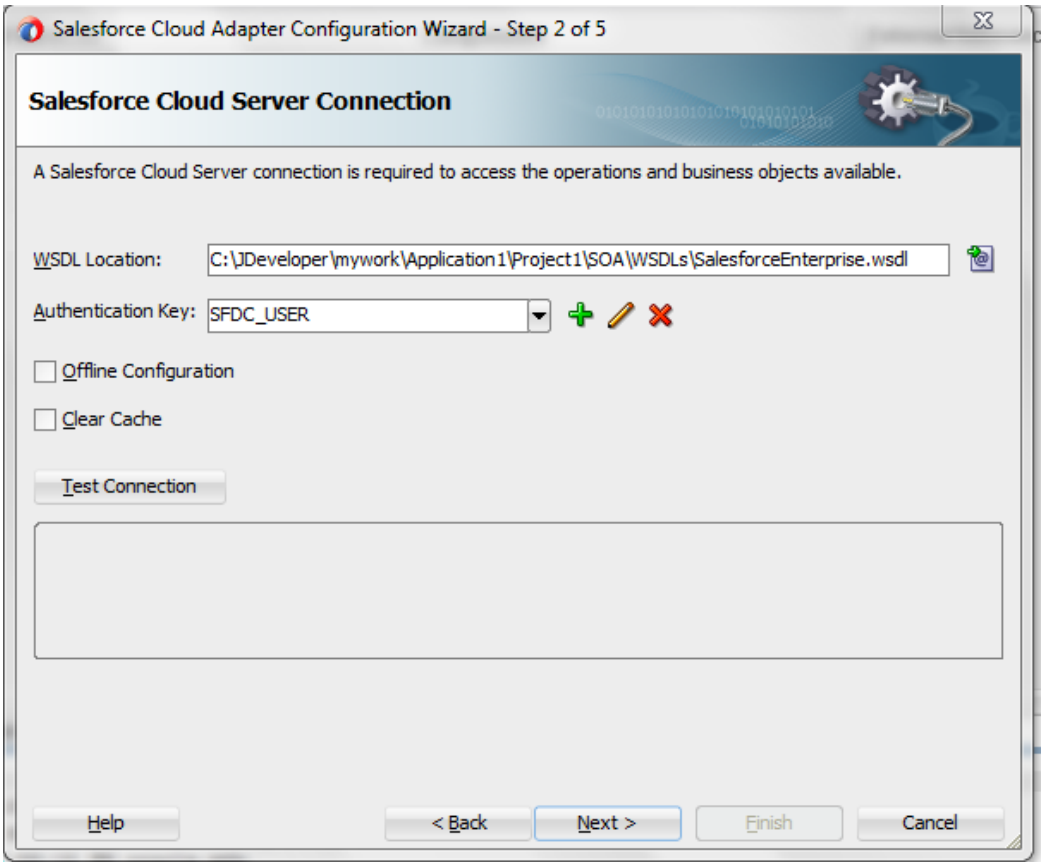
Note: The password should be a combination of Salesforce.com password and Salesforce.com Security Token.

Figure 11 - 21 Add Credential



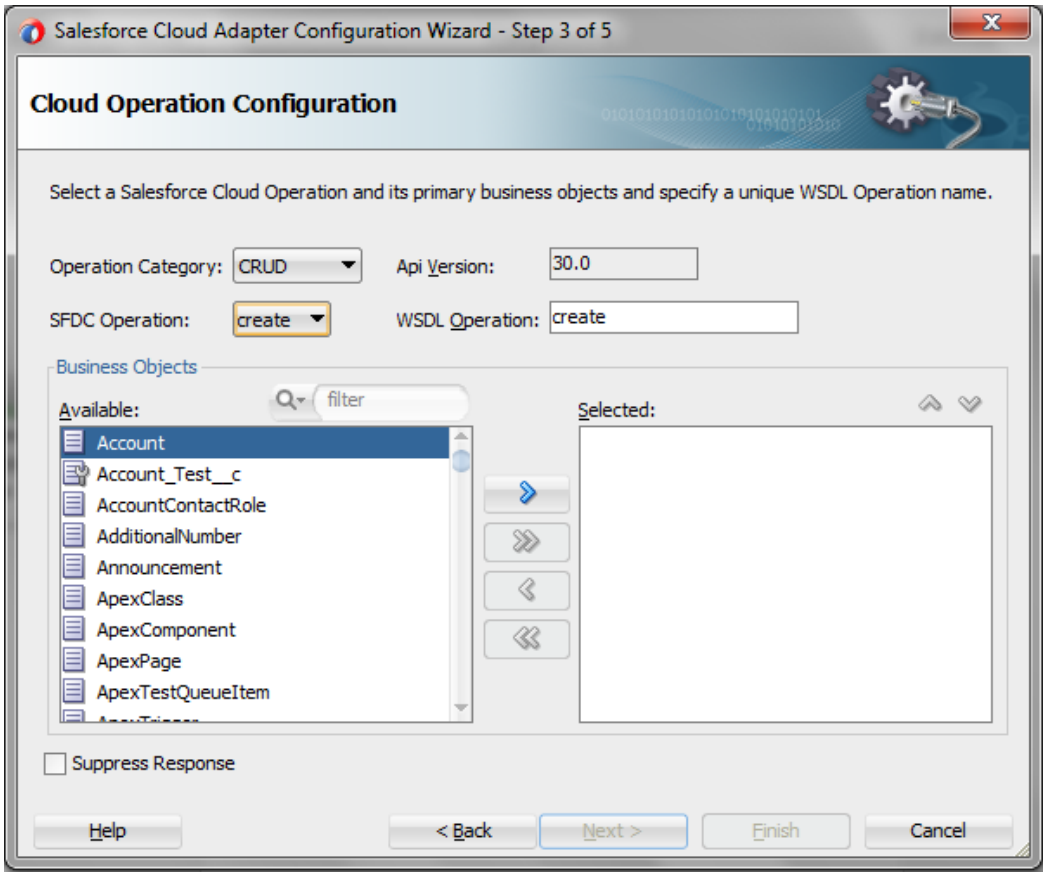
35. Click **Test Connection** button to validate the Authentication Key, as shown in the screenshot below.

Figure 11 - 22 Test Connection



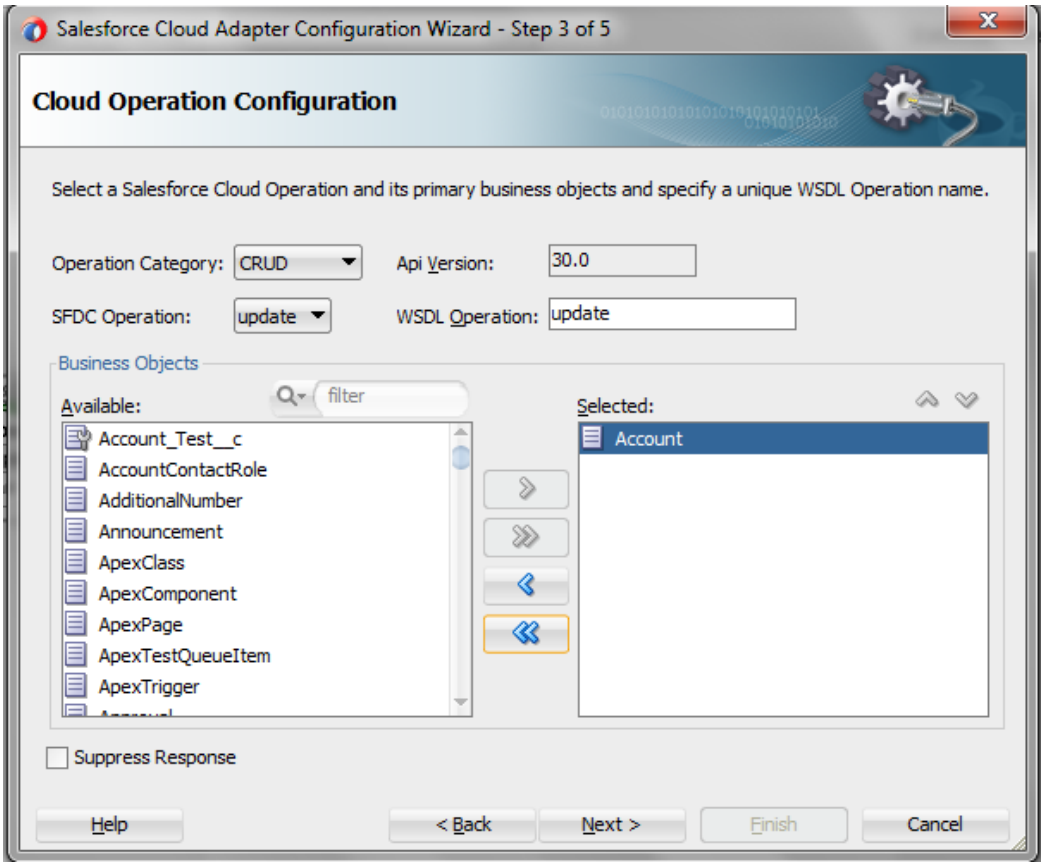
36. Click **Next**.
37. The **Cloud Operation Configuration** page is displayed, as shown in the screenshot below.

Figure 11 - 23 Cloud Operation Page



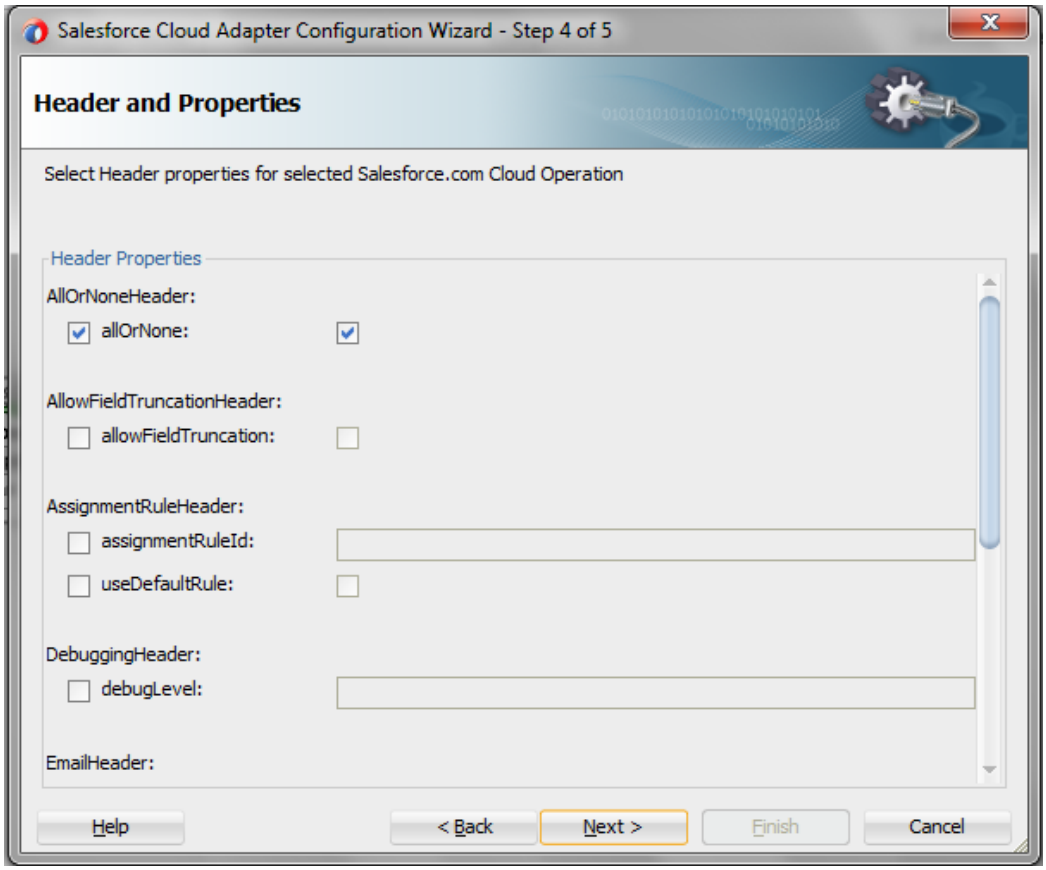
38. Since the scenario is to update an Account on Salesforce.com, select **Operation Category** as **CRUD** and **SFDC Operation** as **Update**. Move **Account** from the list of **Available** objects to the list of the **Selected** objects. The **WSDL Operation** by default is **create** (same as SFDC Operation). You can edit the same by providing an operation name suitable to your business requirement, as shown in the screenshot below.

Figure 11 - 24 Cloud Operation Page



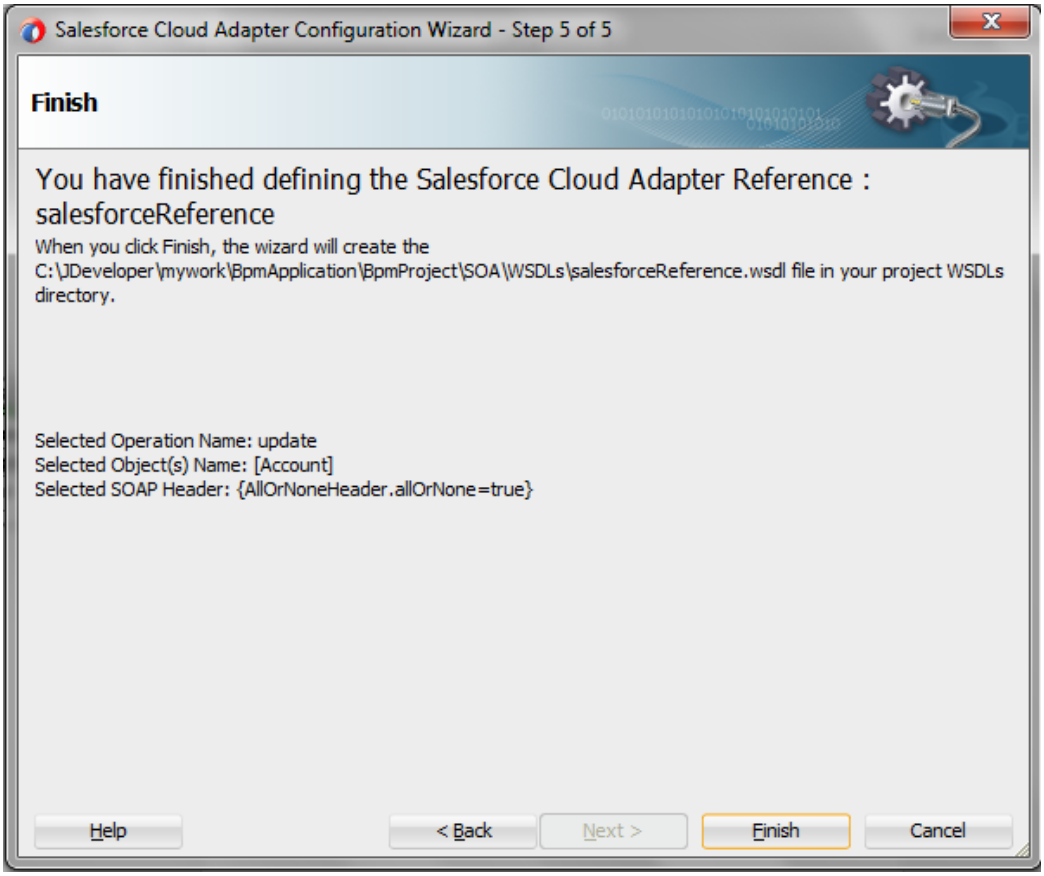
39. Click **Next**. The **Header and Properties** page is displayed, as shown in the screenshot below.

Figure 11 - 25 Header and Properties



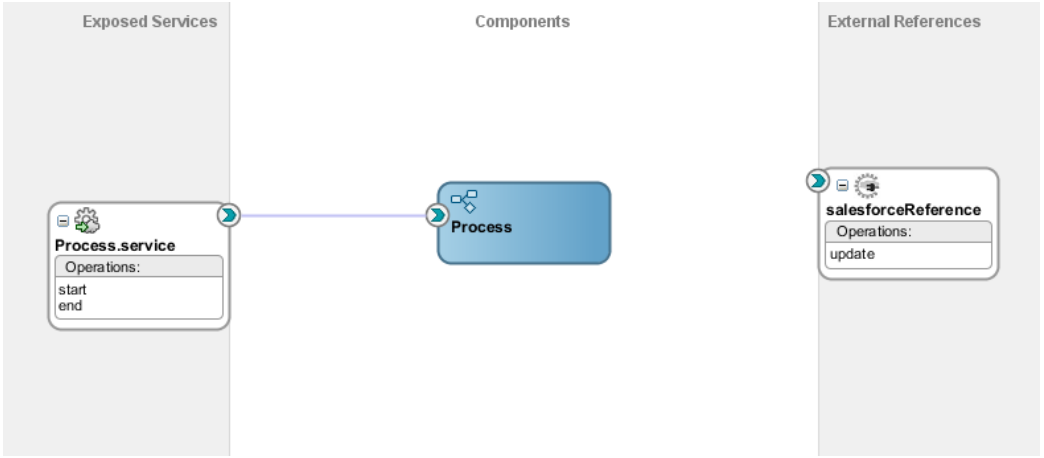
40. Select the appropriate header according to your requirement. Headers displayed in this page depend on the operation selected in the previous page.
41. Click **Next**.
42. The finish page is displayed. It provides a complete summary of the operation selected, object on which the operation would operate and the headers selected for that operation, as shown in the screenshot below.

Figure 11 - 26 Finish Page



43. Click the **Finish** button to complete the Adapter Configuration Wizard.
44. After clicking on **Finish** button, the following screen appears, as shown in the screenshot below.

Figure 11 - 27 Composite.xml

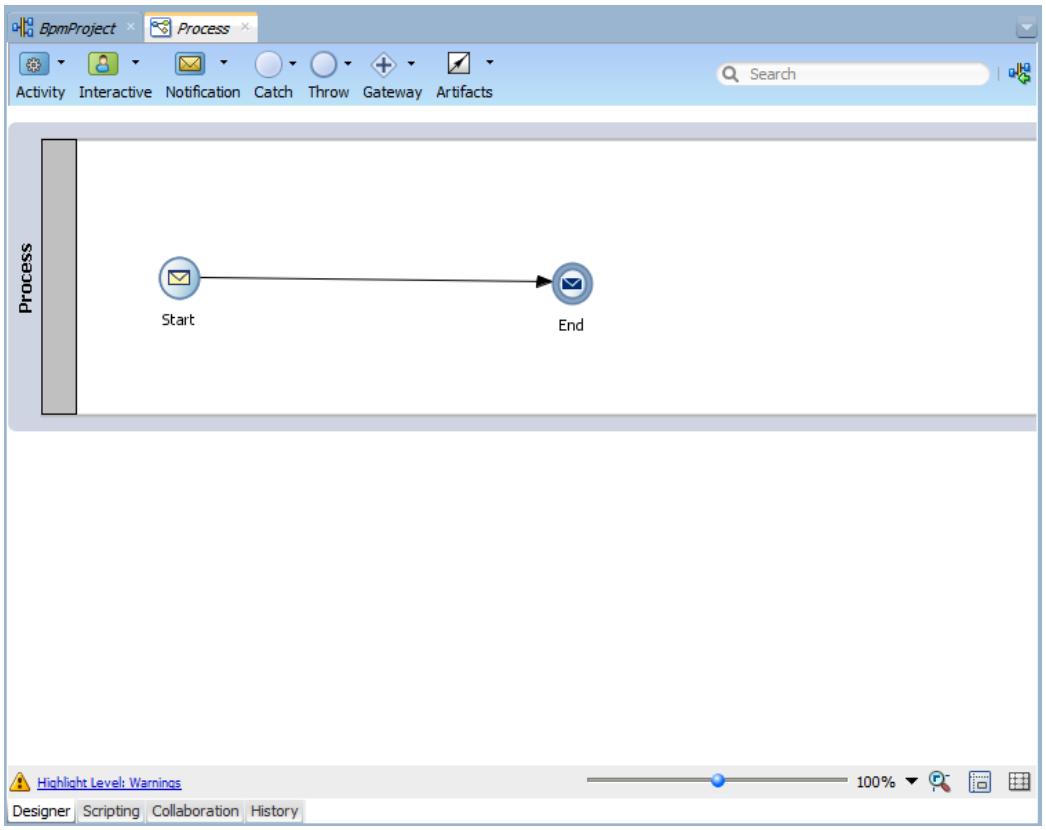


11.8. Integration with BPM

Perform the following steps for integration with BPM:

1. In the composite.xml file of your project, double-click the object named **Process.** Process.bpm page opens, as shown in the screenshot below.

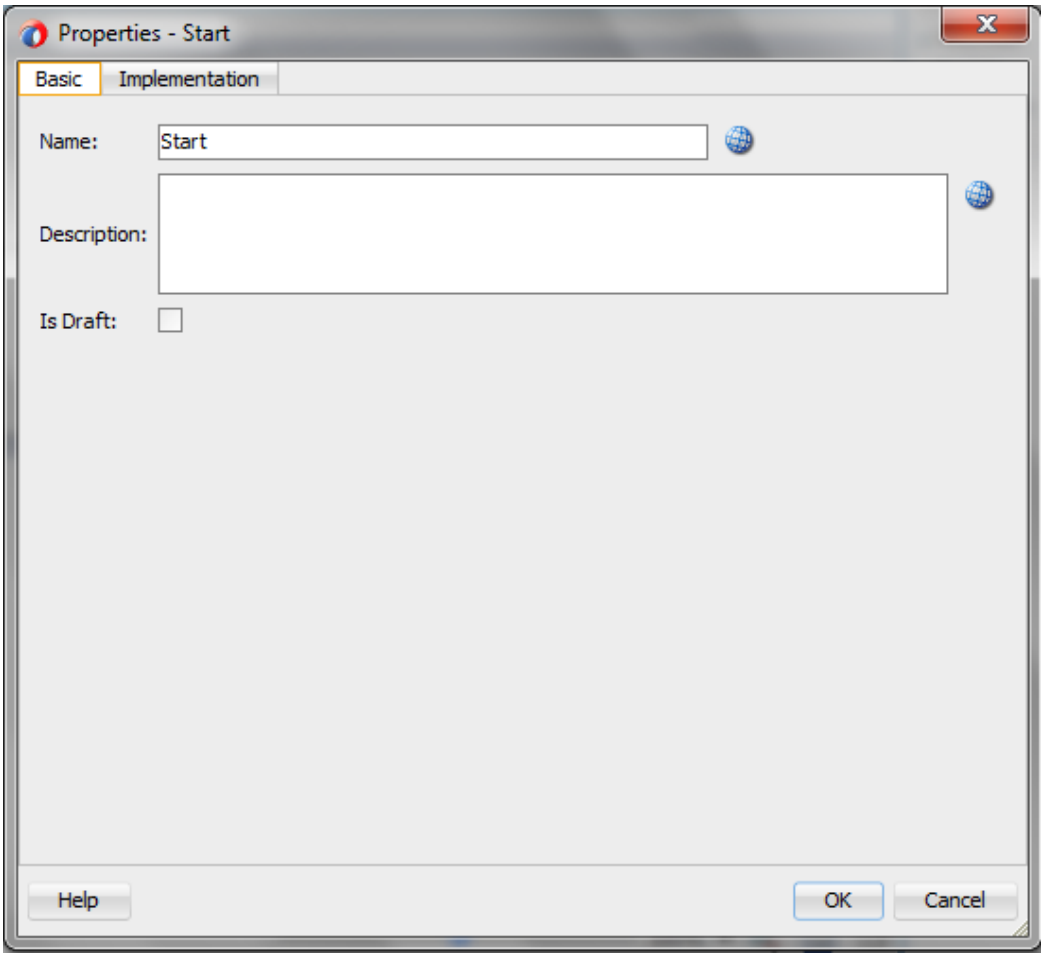
Figure 11 - 28 Open Process



2. Double-click on the **Start** activity. **Properties-Start** page opens as shown in the screenshot below.

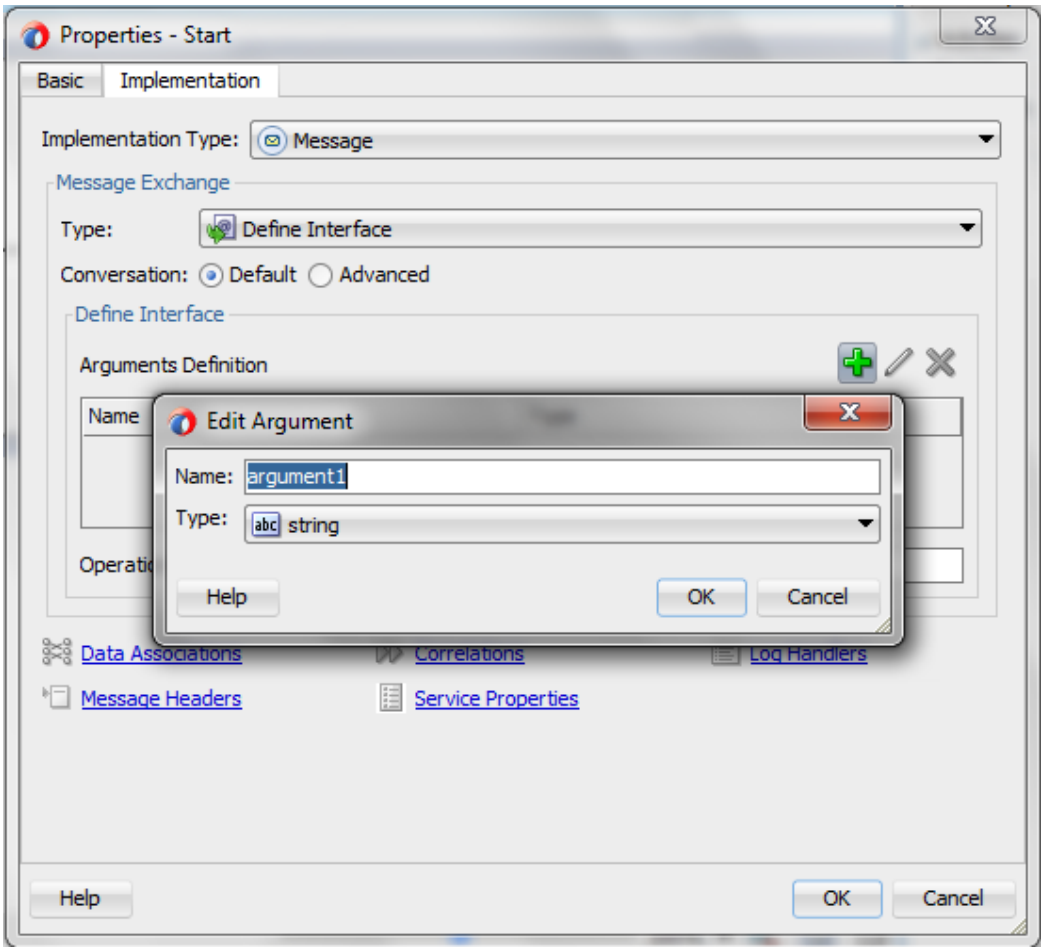
Note: The Model starts and ends with two circular icons. The Left Circle is a **Start Activity** and the circle which is on the right side is the **End Activity.** These two activities are joined with the help of a line that represents the flow of activities through the process known as Sequence Flow.

Figure 11 - 29 Properties – Start



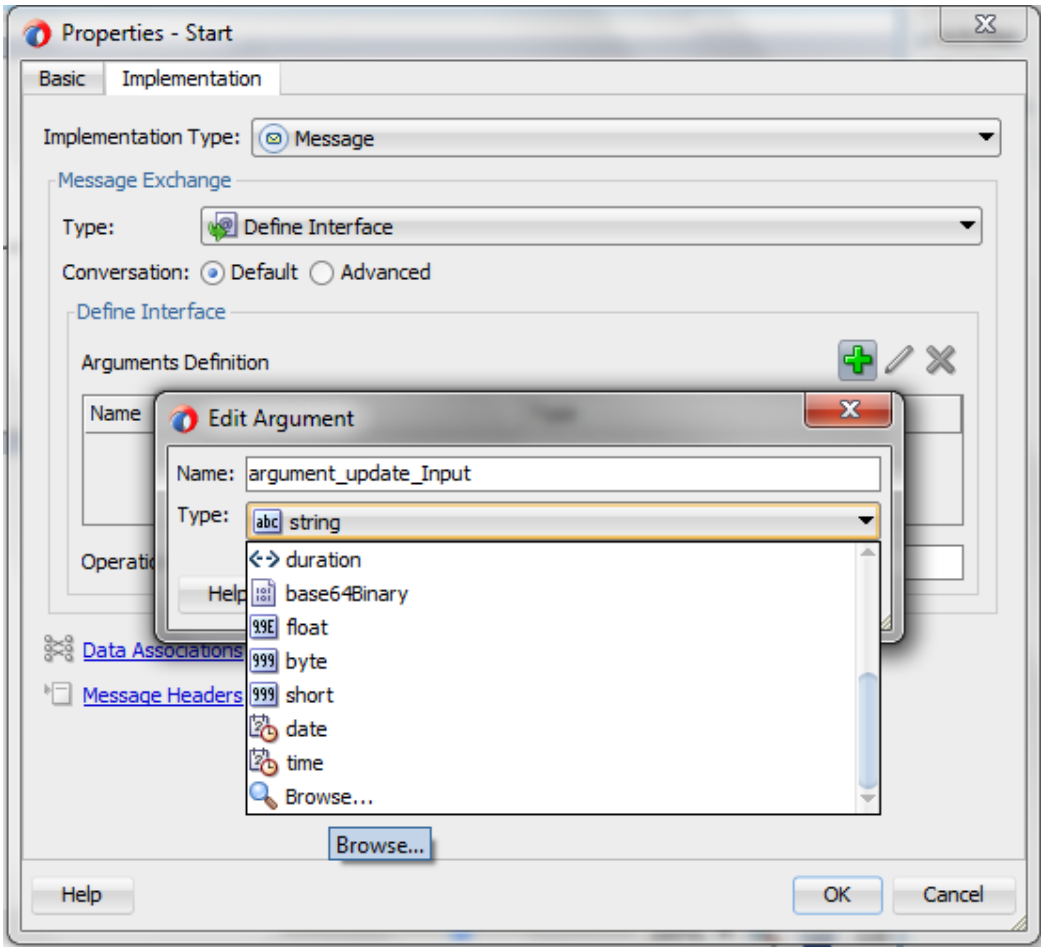
3. Left-click on the **Implementation** tab. Now, in the **Argument Definition,** click on the '+' sign to create the input argument/s.

Figure 11 - 30 Edit Argument for Start



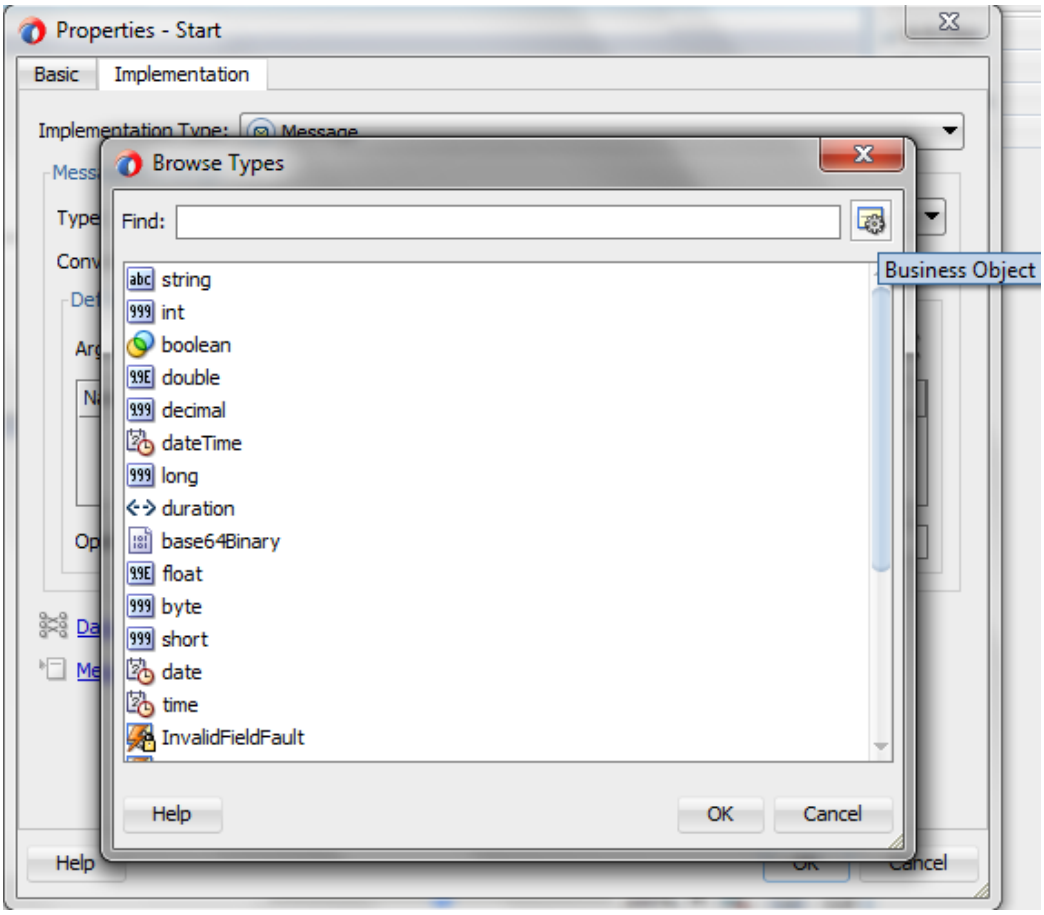
4. In the **Edit Argument** page, provide a name to the argument in the **Name** section and in the **Type** section, scroll down and select the **Browse** option.

Figure 11 - 31 Edit-Argument Page



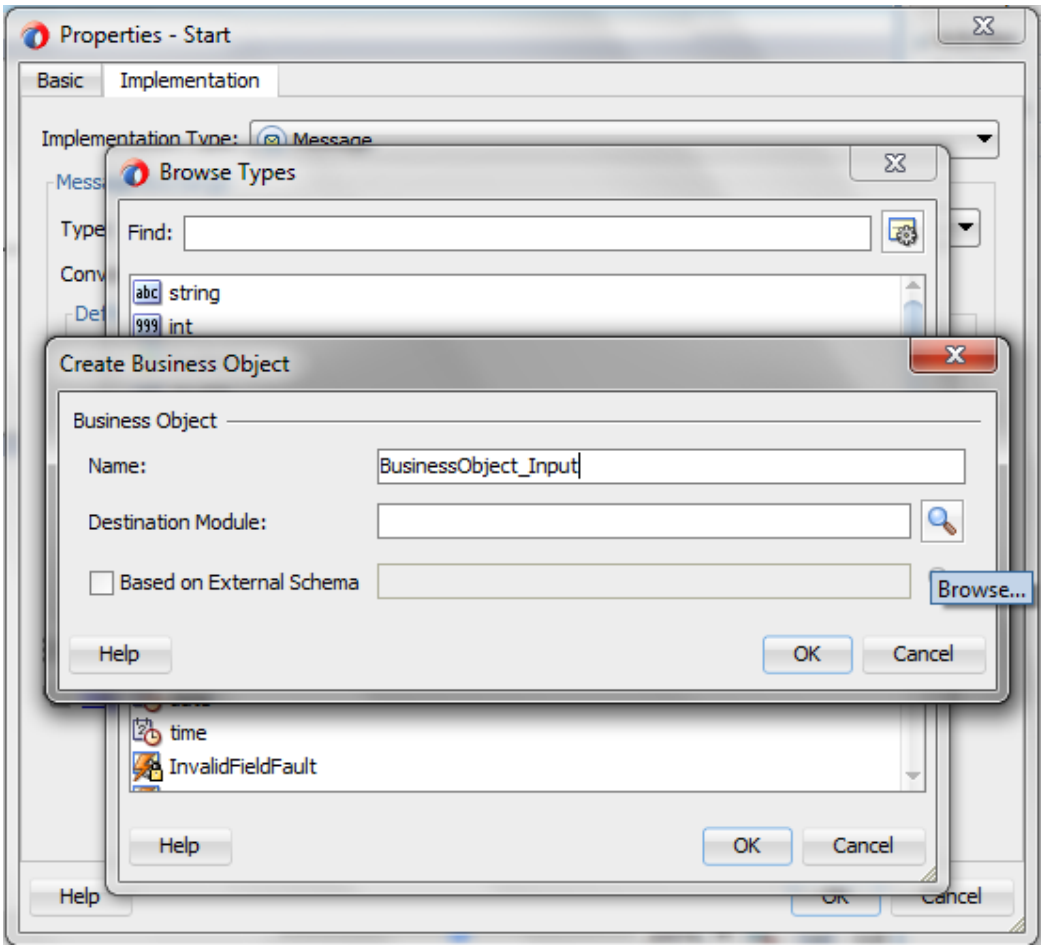
5. Next, a new **Browse Type** page opens, click on the **Business Object** button to create one.

Figure 11 - 32 Browse Type for Argument



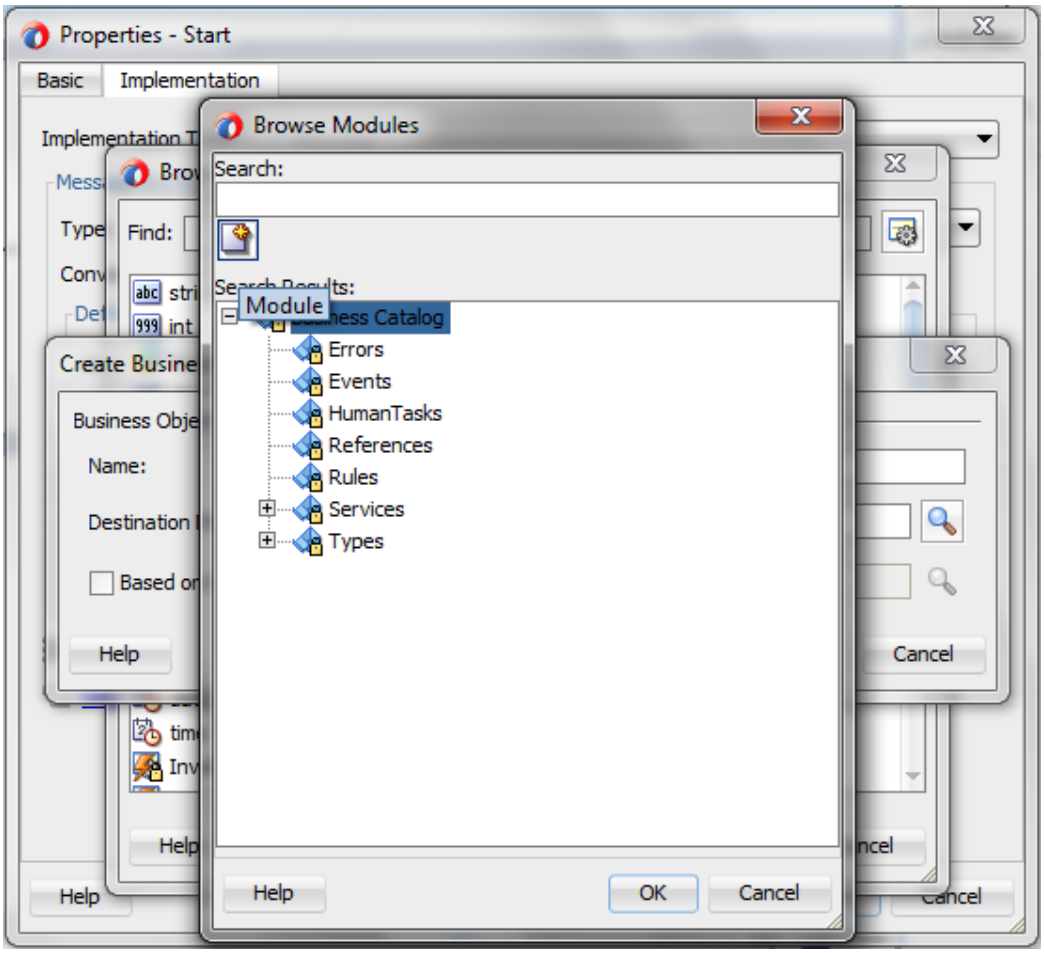
6. In the **Create Business Object** page, provide a **Name** and click on the **Magnifying** icon in front of the **Destination Module** to browse for a module in which the business object exists.

Figure 11 - 33 Business Object



7. In the **Browse Module** page, create a new module by clicking on the Module button as shown in the screenshot below.

Figure 11 - 34 Create Module



8. In the create module page give a suitable **Name** to the module and click on **OK** button. Select the module you just created in the Browse module page and click **OK** as shown in the subsequent figures.

Figure 11 - 35 Name the Module

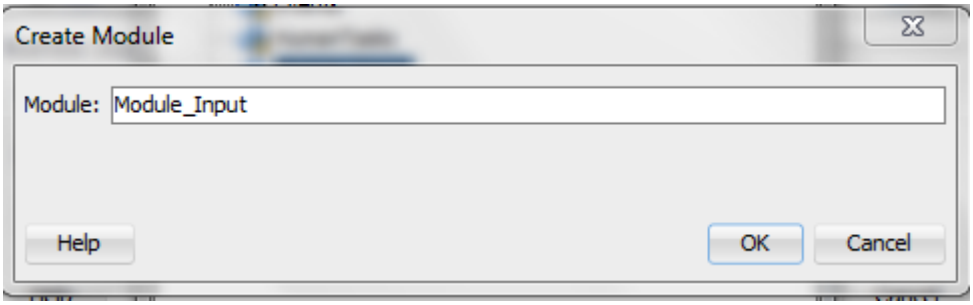
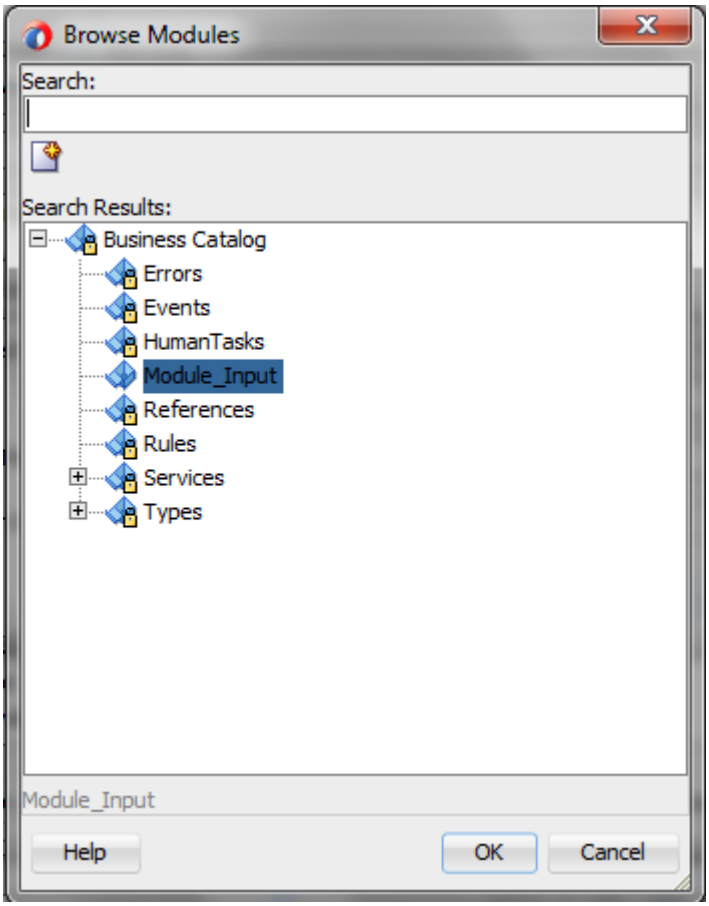
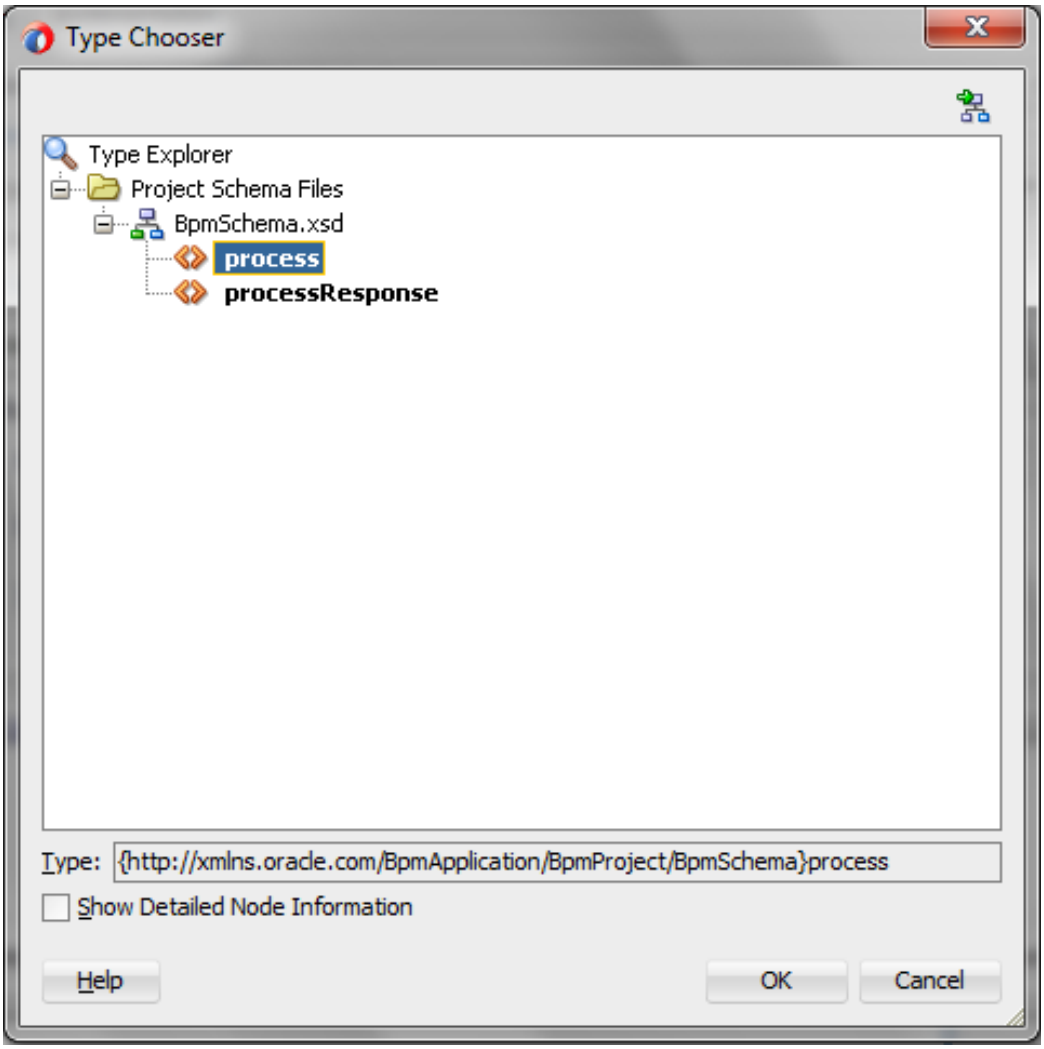


Figure 11 - 36 Selecting Module



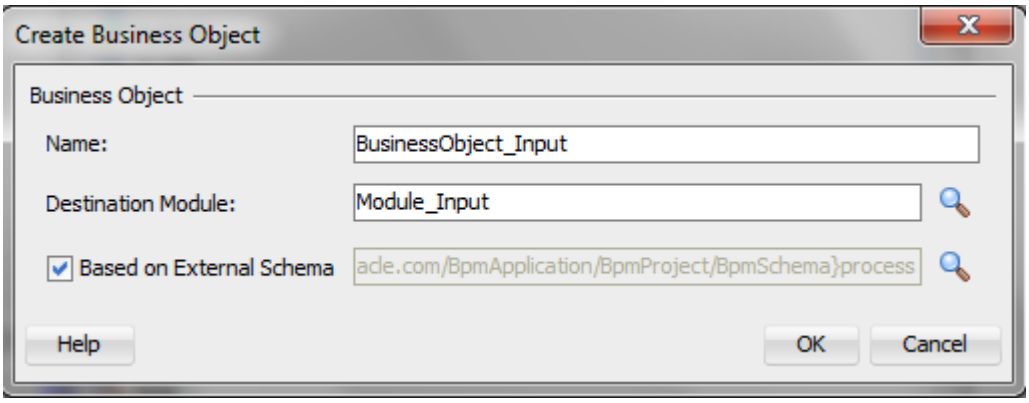
9. In the **Create Business Object** page, click on the checkbox **Based on External Schema** and browse to the respective schema file and select the input element (**Process** in this case) from the Schema and click on **OK**.

Figure 11 - 37 Select Schema for Business Object



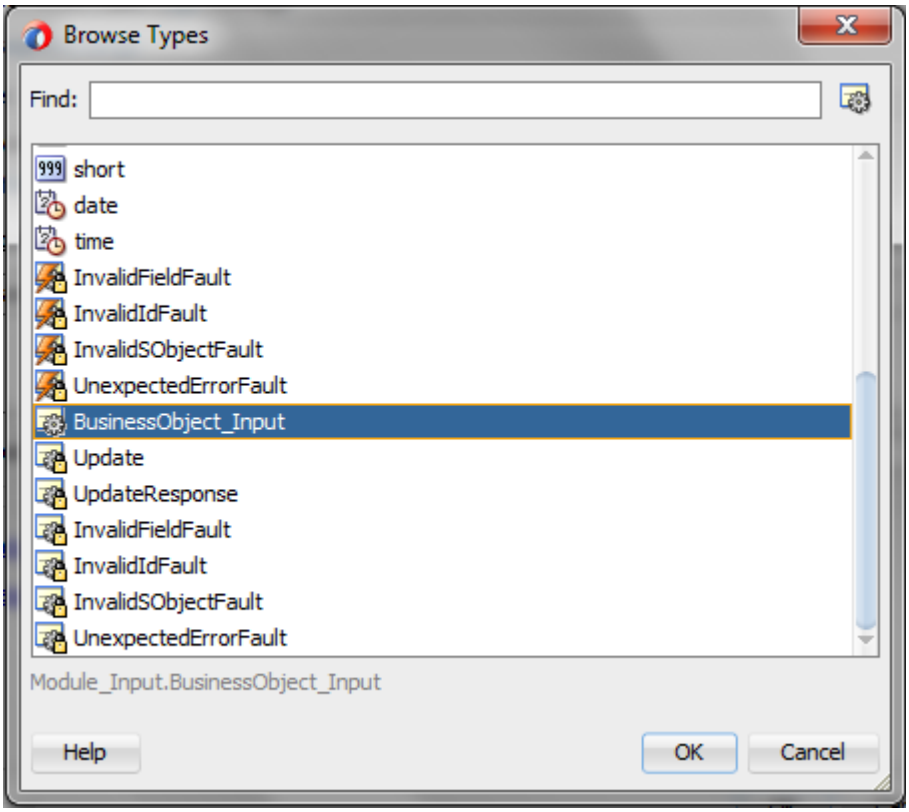
10. The **Create Business Object** page is displayed, as shown in the screenshot below. Click **OK**.

Figure 11 - 38 Create Business Object



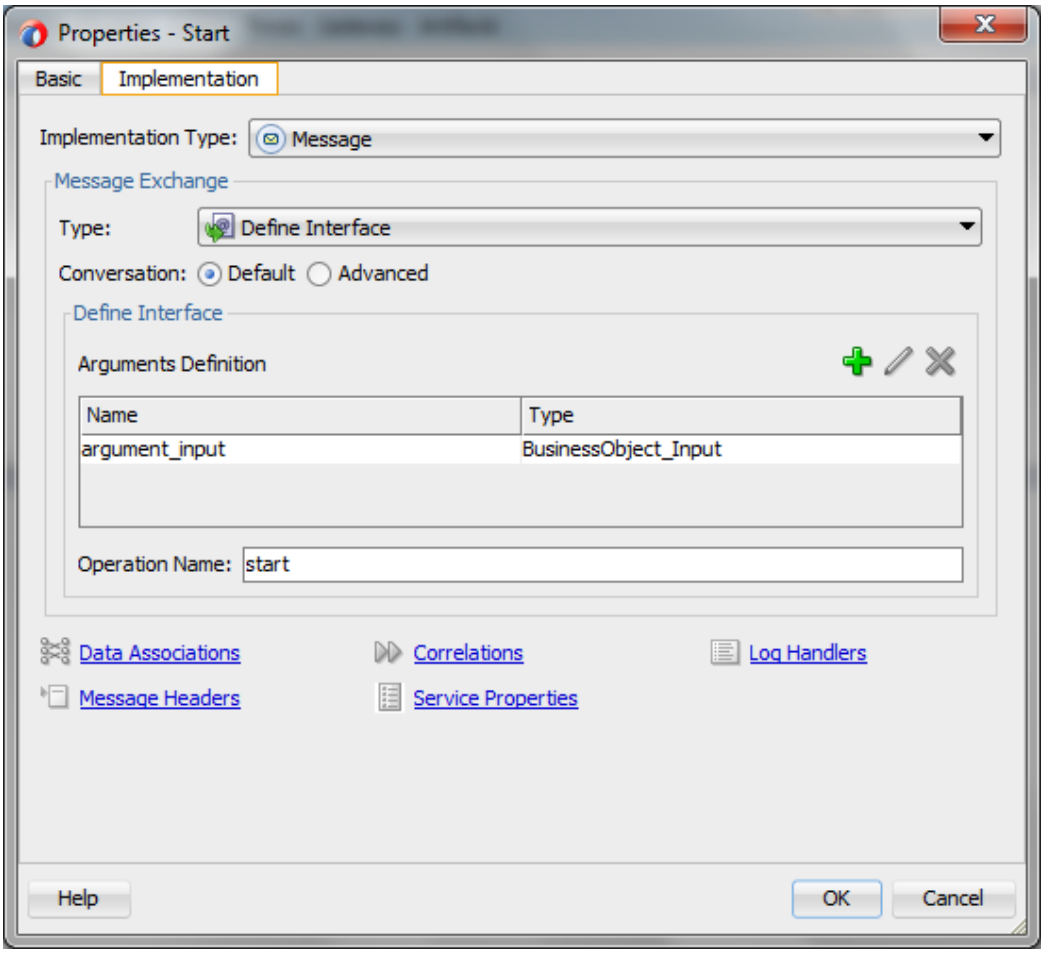
11. Now a Business Object is created. Select the Business Object that you have created for the input and click **OK** as shown in the screenshot below.

Figure 11 - 39 Selecting Business Object For Argument



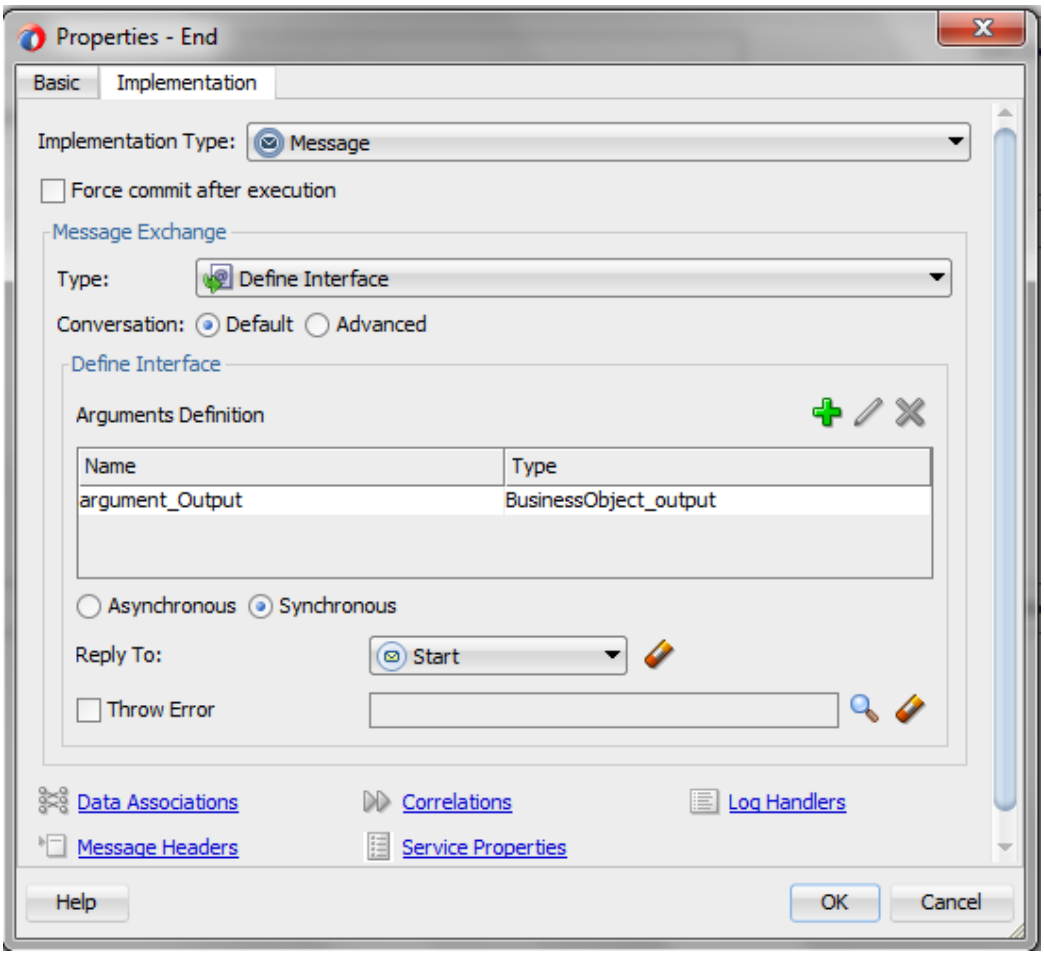
12. **Properties-Start** page is displayed. It contain an argument for the input.

Figure 11 - 40 Properties – Start



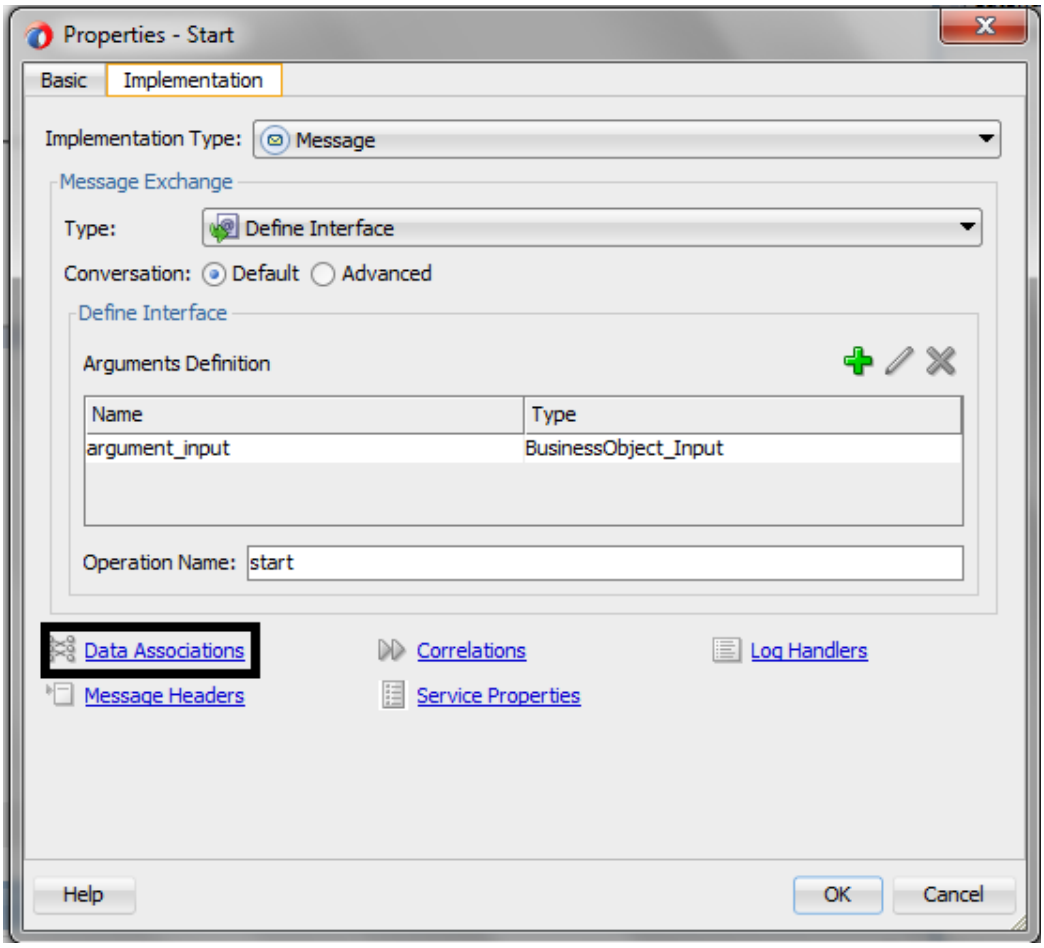
13. Follow the same steps to create **Business object** for output in the **End** activity of your process. Choose the respective schema file and select the Output element (**processResponse** in this case) from the Schema in **Create Business Object** page.

Figure 11 - 41 Properties – End



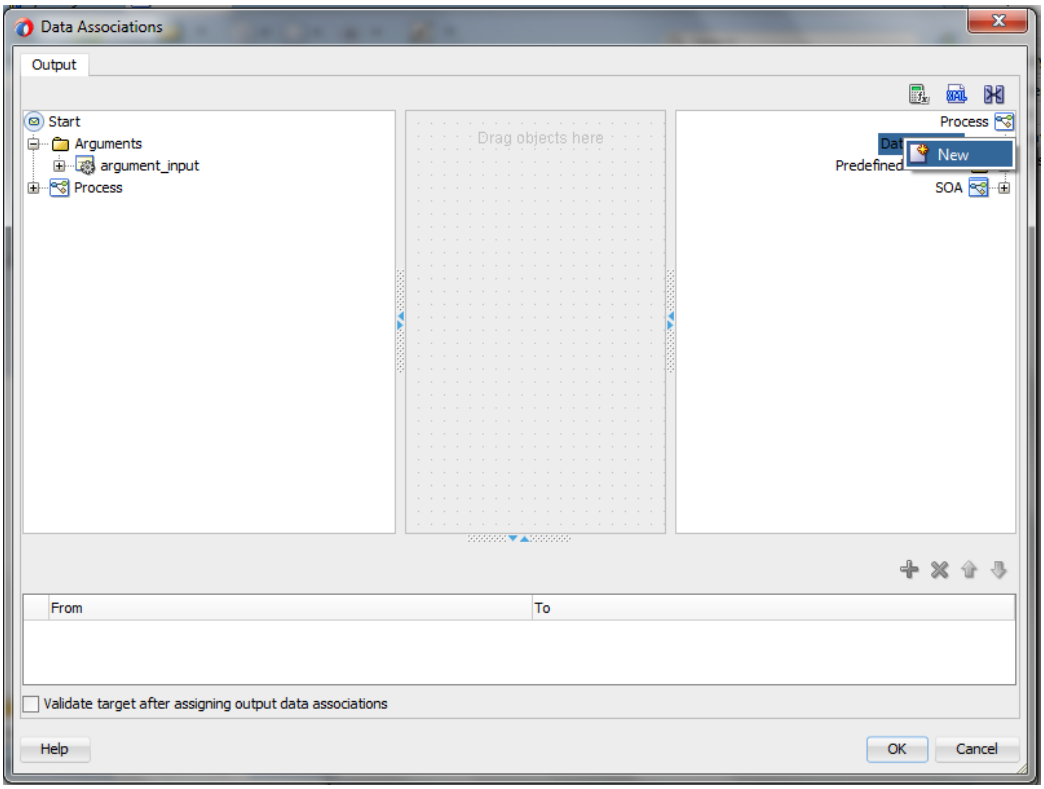
14. Again proceed to the **Start** activity and click on the **Implementation** tab. Click on **Data Association** as shown in the screenshot below.

Figure 11 - 42 Data Associations for Start Activity



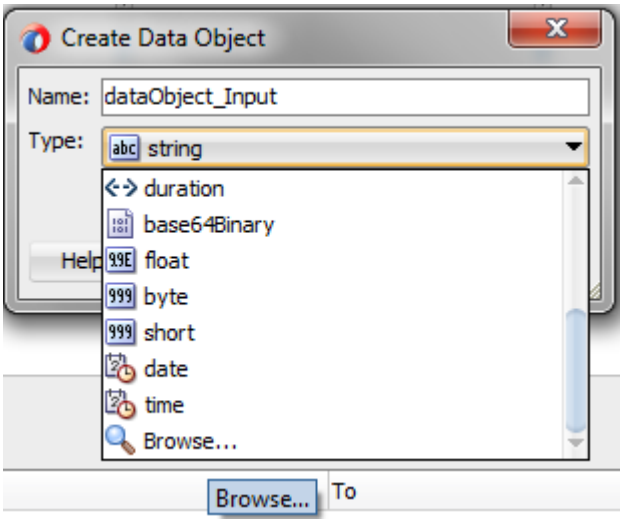
15. **New Data Association** page opens. In the data objects under **Process,** create **Data Objects** for Input, Output and Response.
16. Right-Click on the data object and click **New** as shown in the screenshot below.

Figure 11 - 43 Data Association



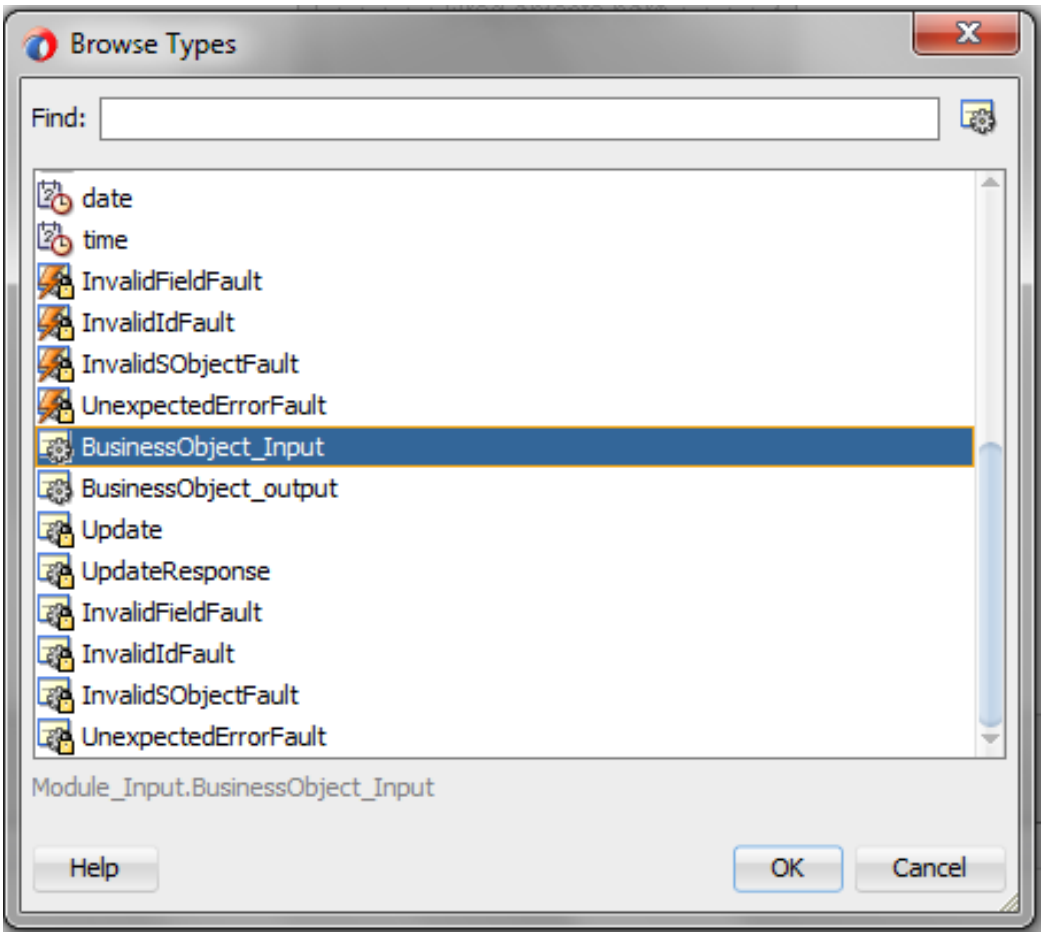
17. Provide a suitable **Name** for the data object (Input in this case). In the **Type** section, scroll down and select the **Browse** option.

Figure 11 - 44 Creating dataObject



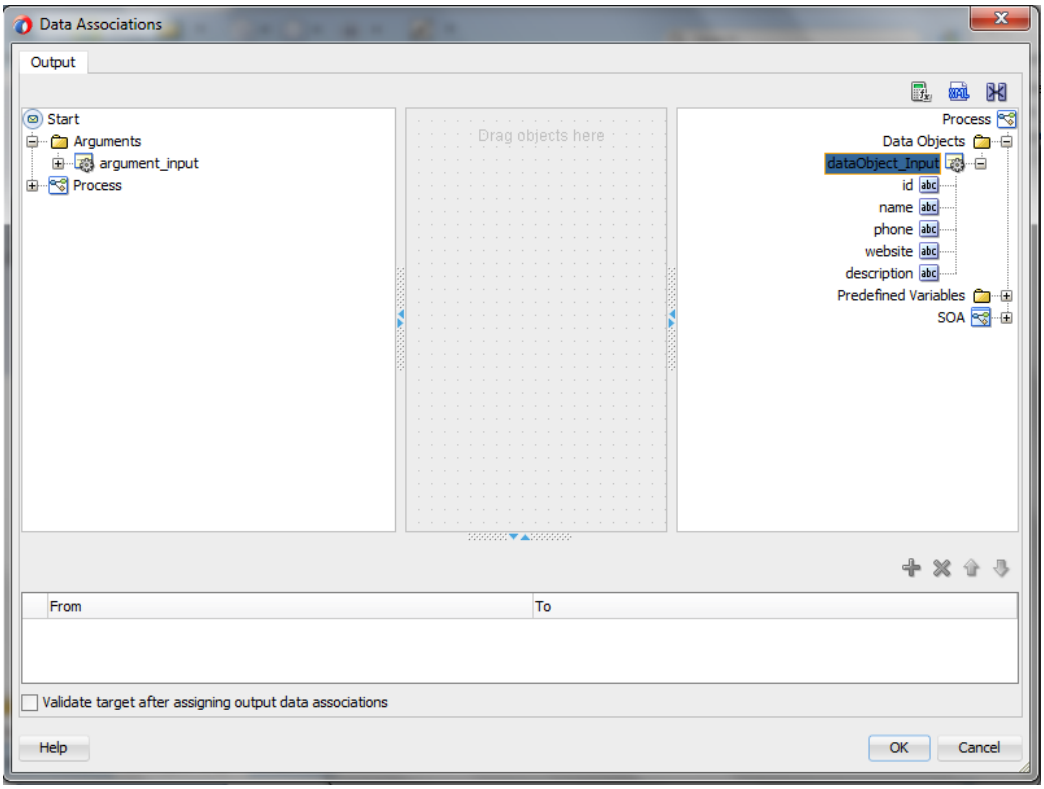
18. **Browse Type** page opens. Select the business object that you have created and click **OK** button as shown in the screenshot below.

Figure 11 - 45 Selecting BusinessObject for dataObject



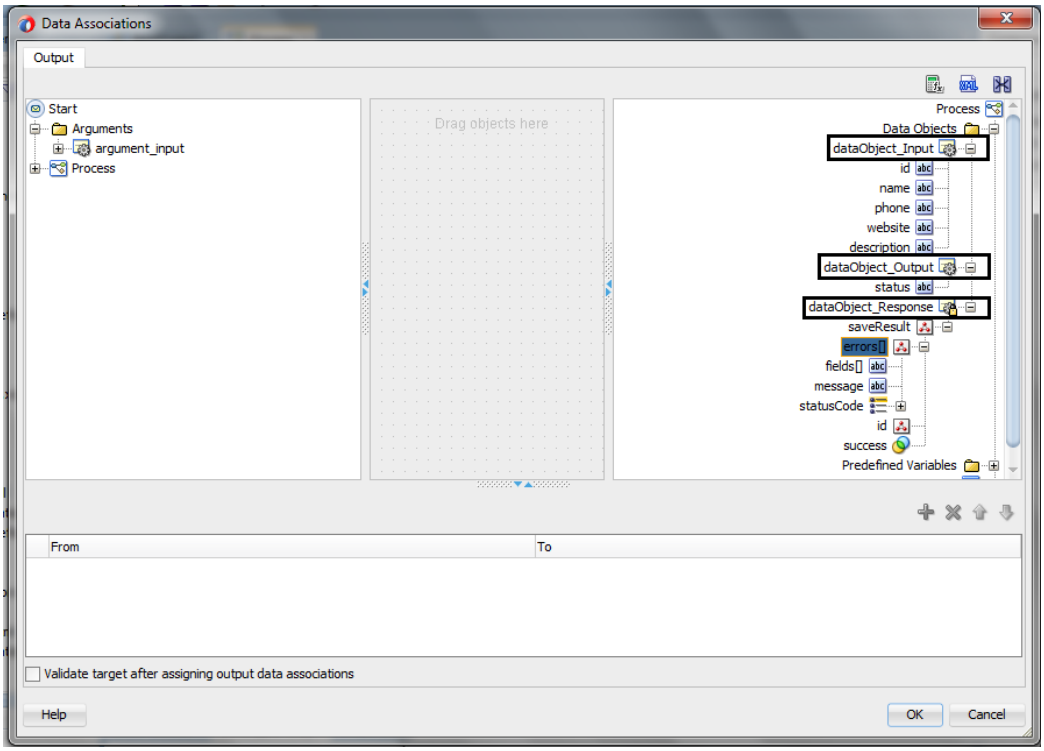
19. Click **OK** on **Create Data Object** page.
20. Data object is created for Input which can be seen under **Data Objects** of **Process** as shown in the screenshot below.

Figure 11 - 46 Data Associations for Start Activity



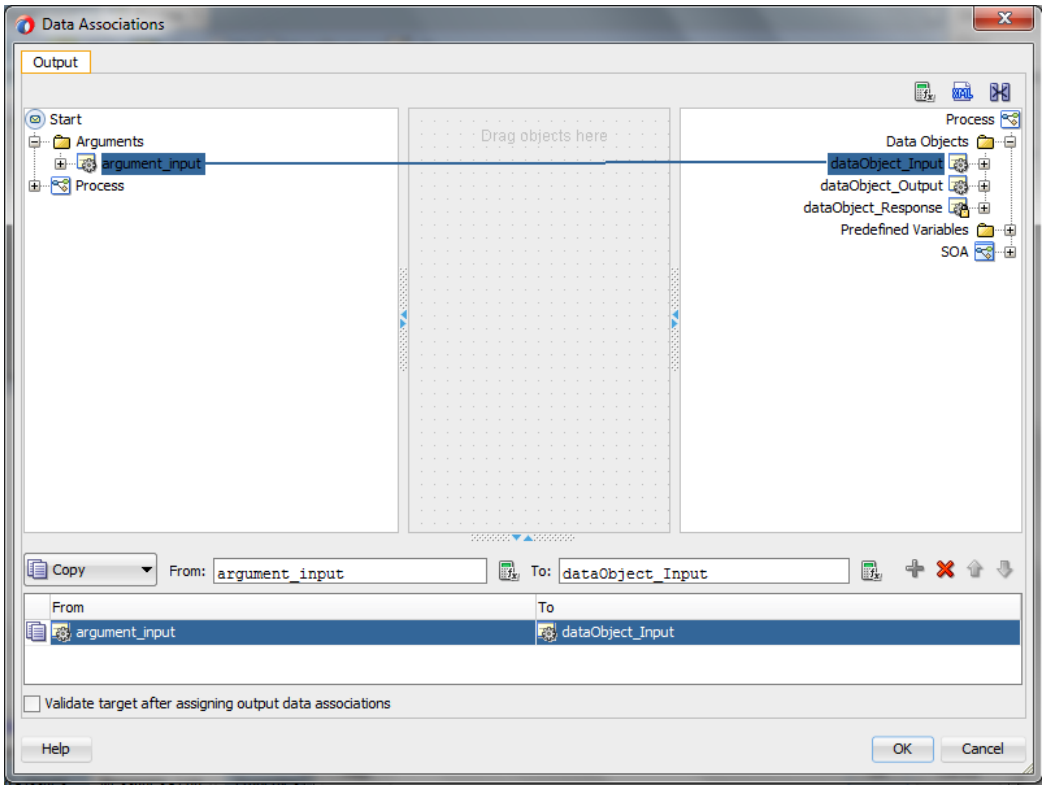
21. Follow the same steps for creating the **Data Objects** for Output and Response (**updateResponse** in this case).

Figure 11 - 47 Data Associations For Start Activity



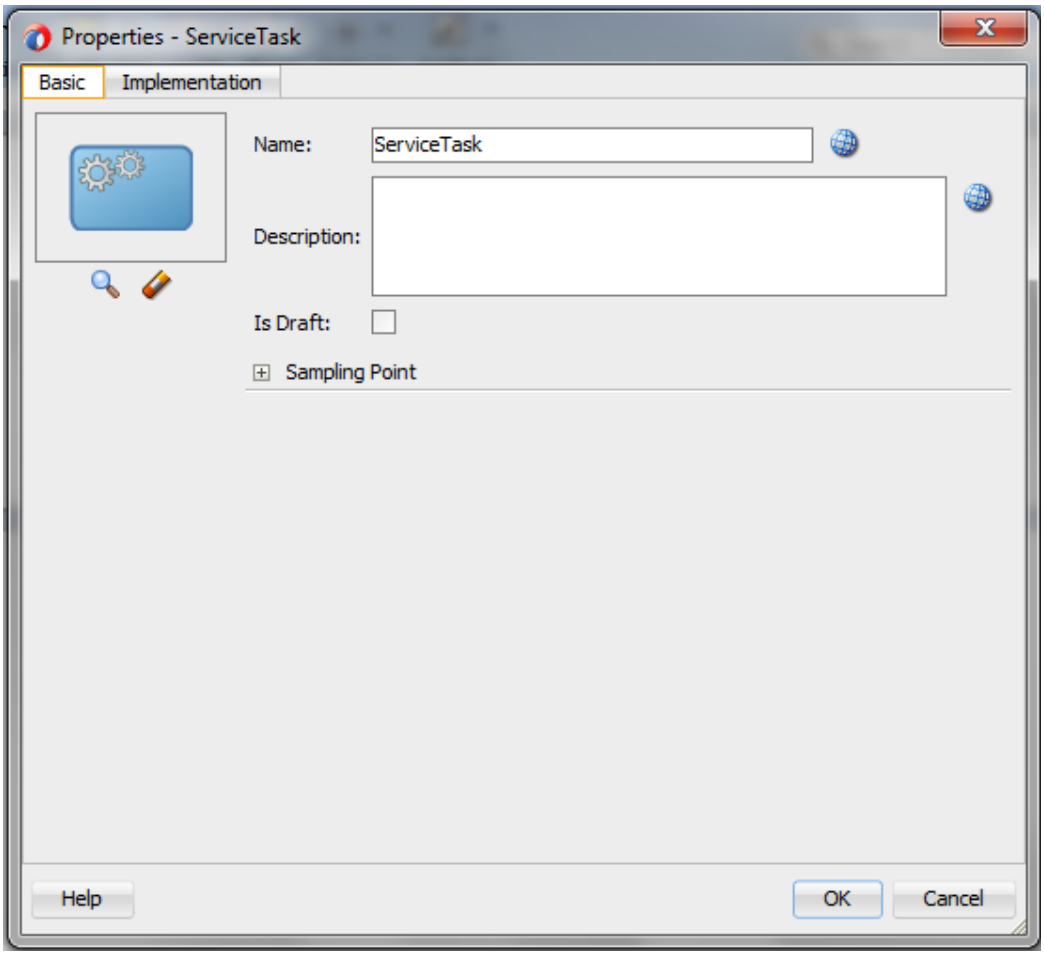
22. Map the **argument_input** under **Start** to **dataObject_Input** and click on **OK** button as shown in the screenshot below.

Figure 11 - 48 Mapping Source To Target



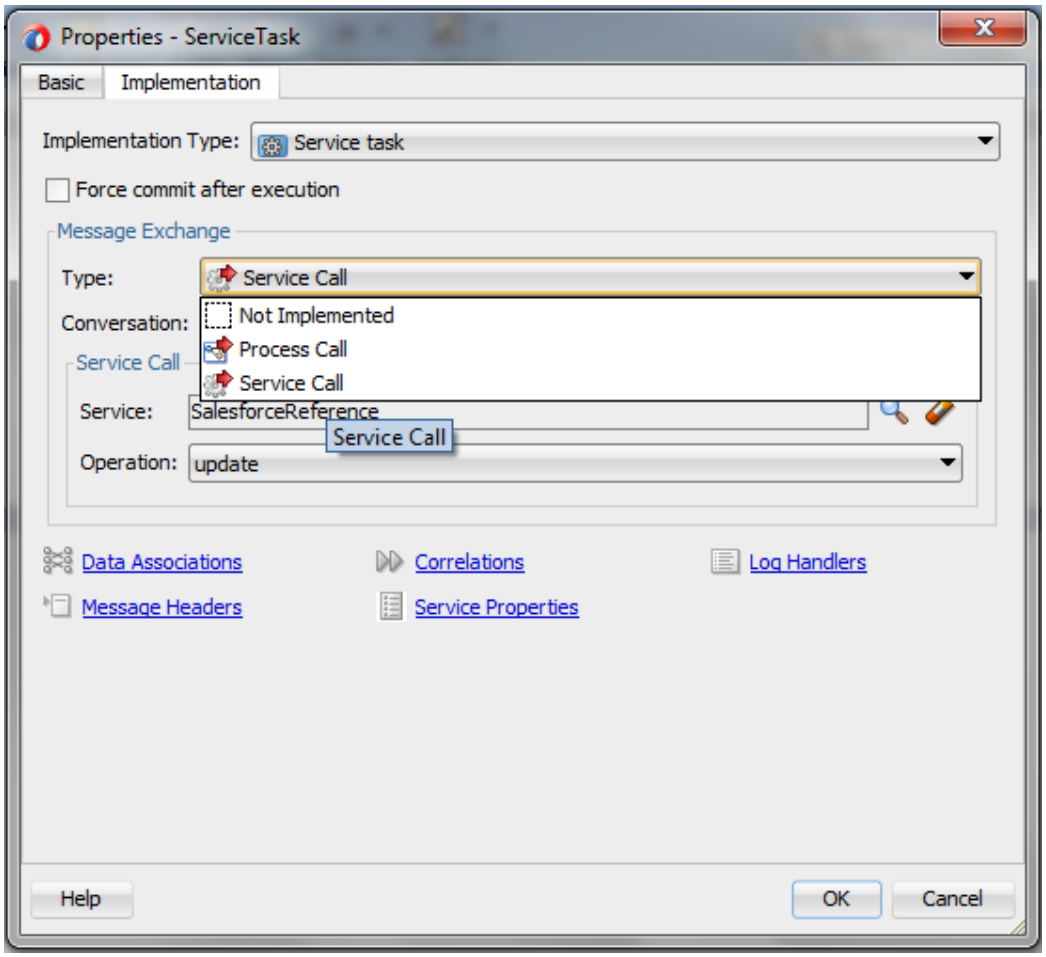
14. Drag the **Service** activity from the **Activity** tab and place it in between **Start** and **End** Activity.
15. A new page **Properties – ServiceTask** opens, as shown in the screenshot below.

Figure 11 - 49 Service Task



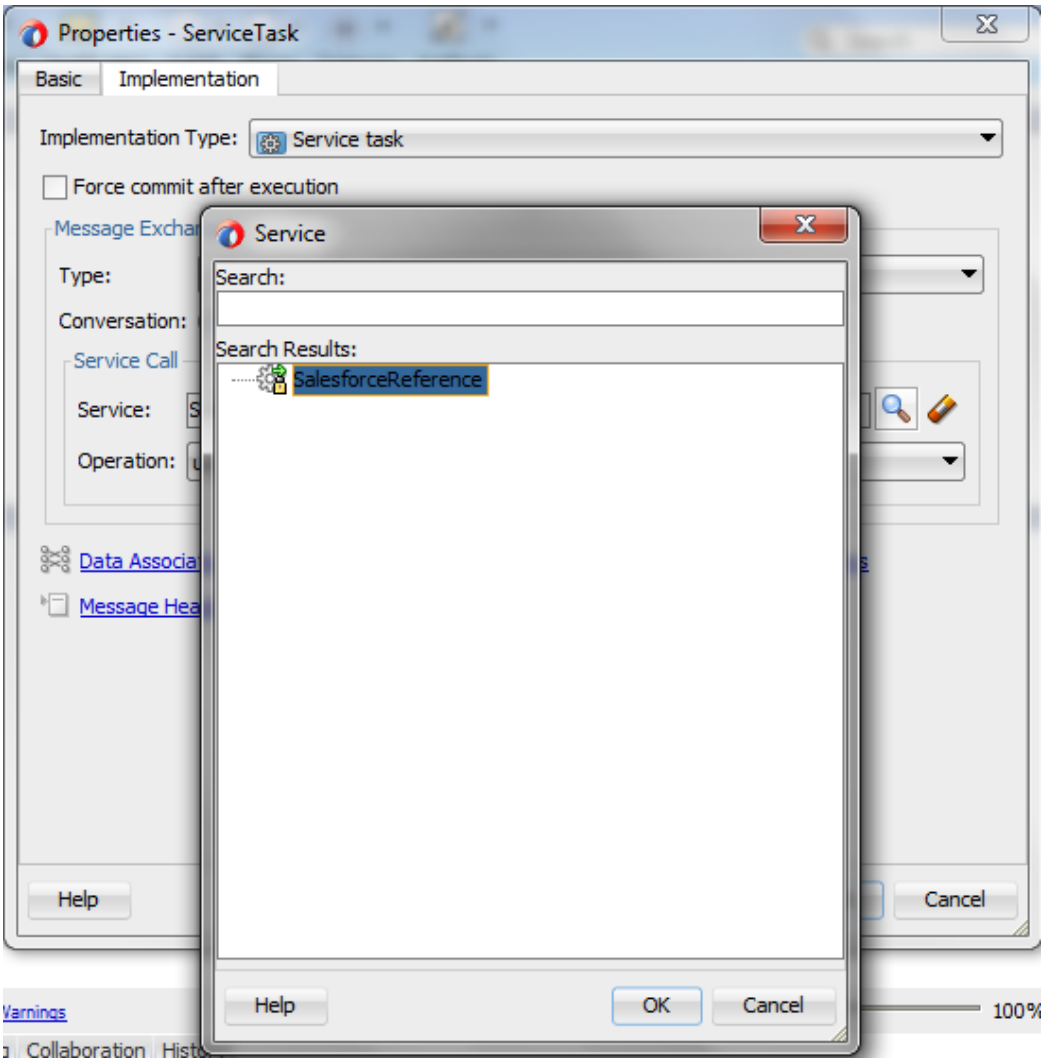
16. Click on **Implementation** tab and select **Service call** in **Type** Section.

Figure 11 - 50 Selecting Type for Service – Task



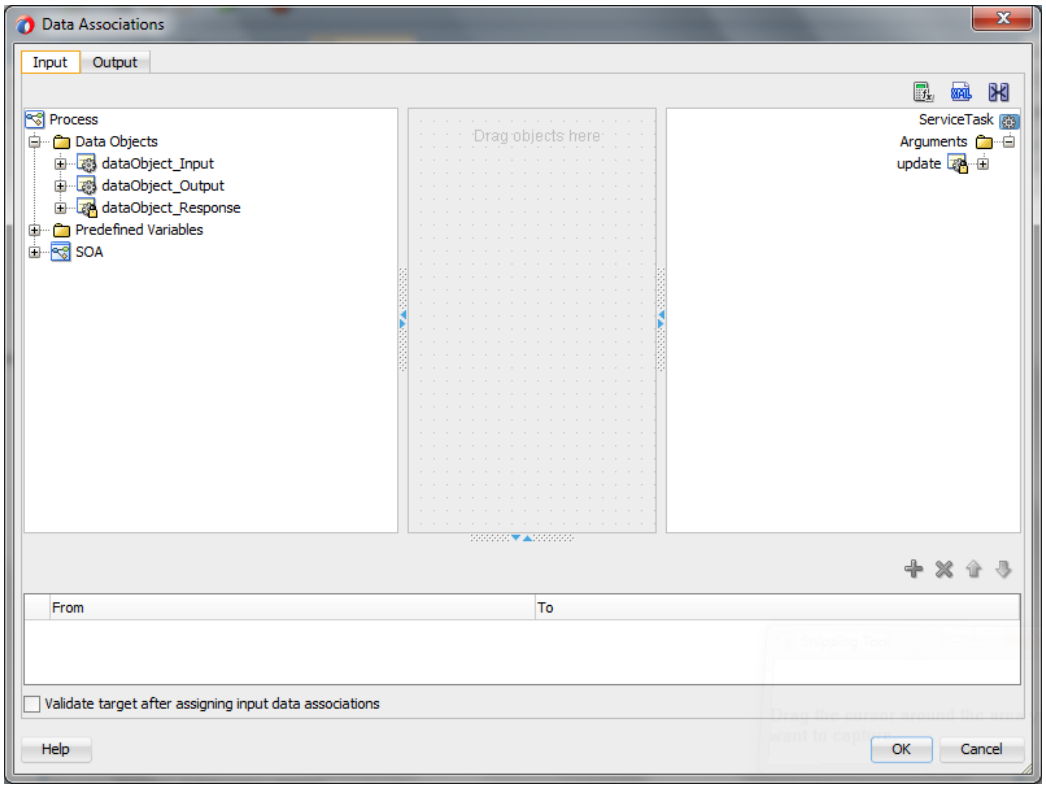
17. In the **Service** section, click on **Magnifier** button and select your service for Salesforce as shown in the screenshot below.

Figure 11 -51 Selecting Service for Service - Task



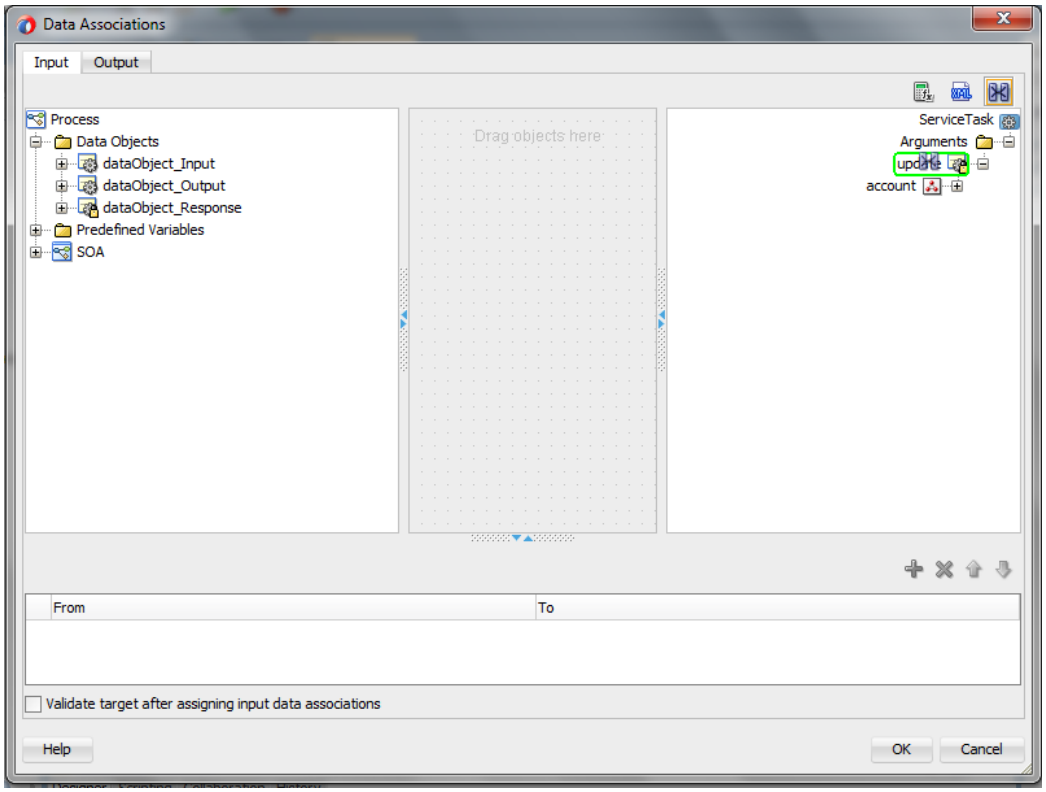
18. Click **Data Association** and a new **Data Association** page opens as shown in the screenshot below.

Figure 11 - 52 Data Association



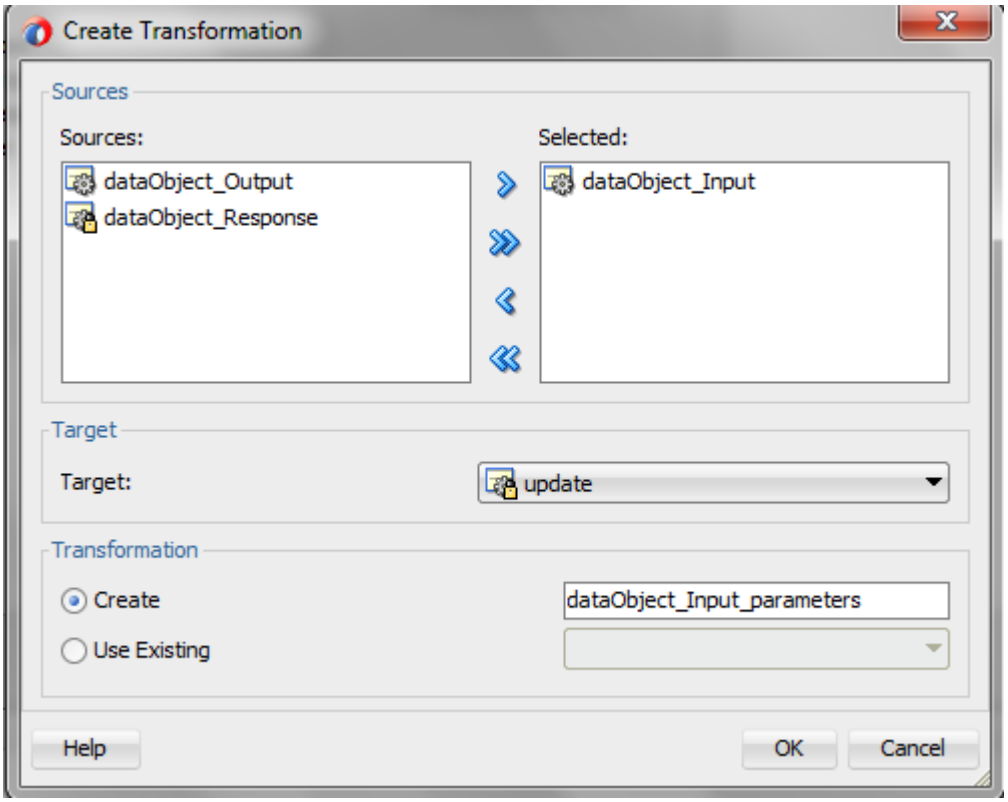
19. Click on the **Transformation** button and drag it to **update** under Arguments in **ServiceTask**.

Figure 11 - 53 Adding Transformation



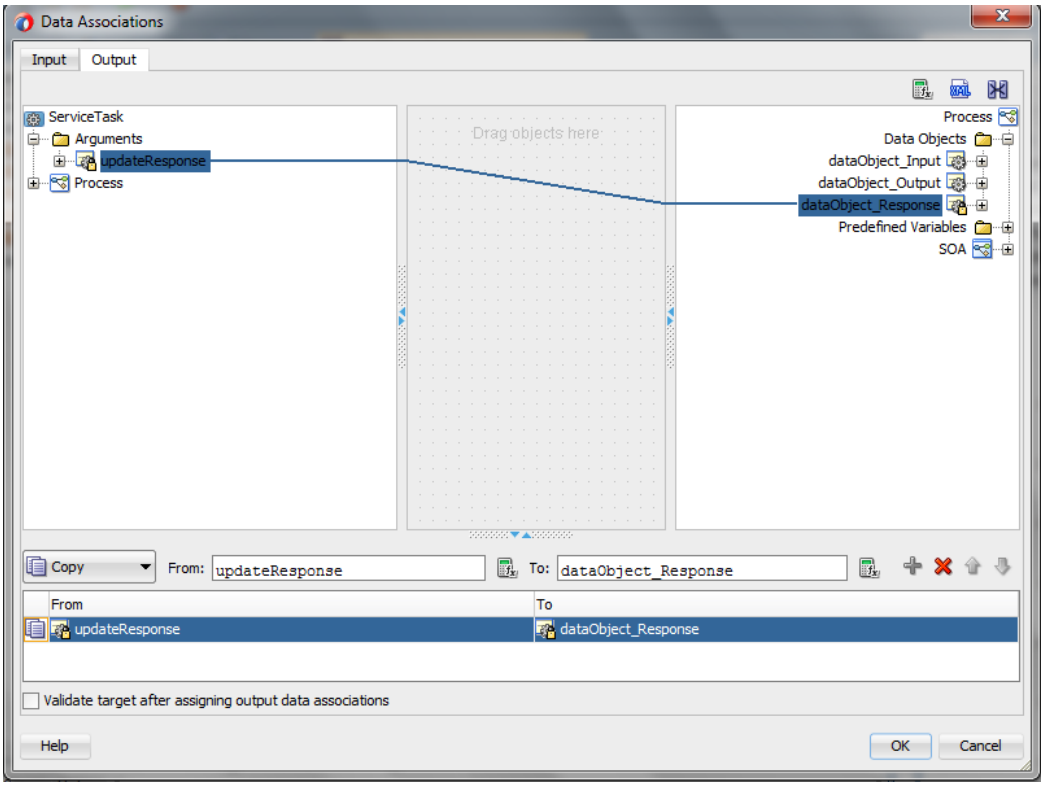
20. **Create Transformation** page opens. Now move **dataObject_Input** from the list of **Source** side to the list of **Selected** and click **OK** as shown in the screenshot below.

Figure 11 - 54 Create Transformation



21. Click on the **Output** tab and map the **updateResponse** from the **serviceTask** to the **dataObject_Response** in **process** and click **OK** button as shown in the screenshot below.

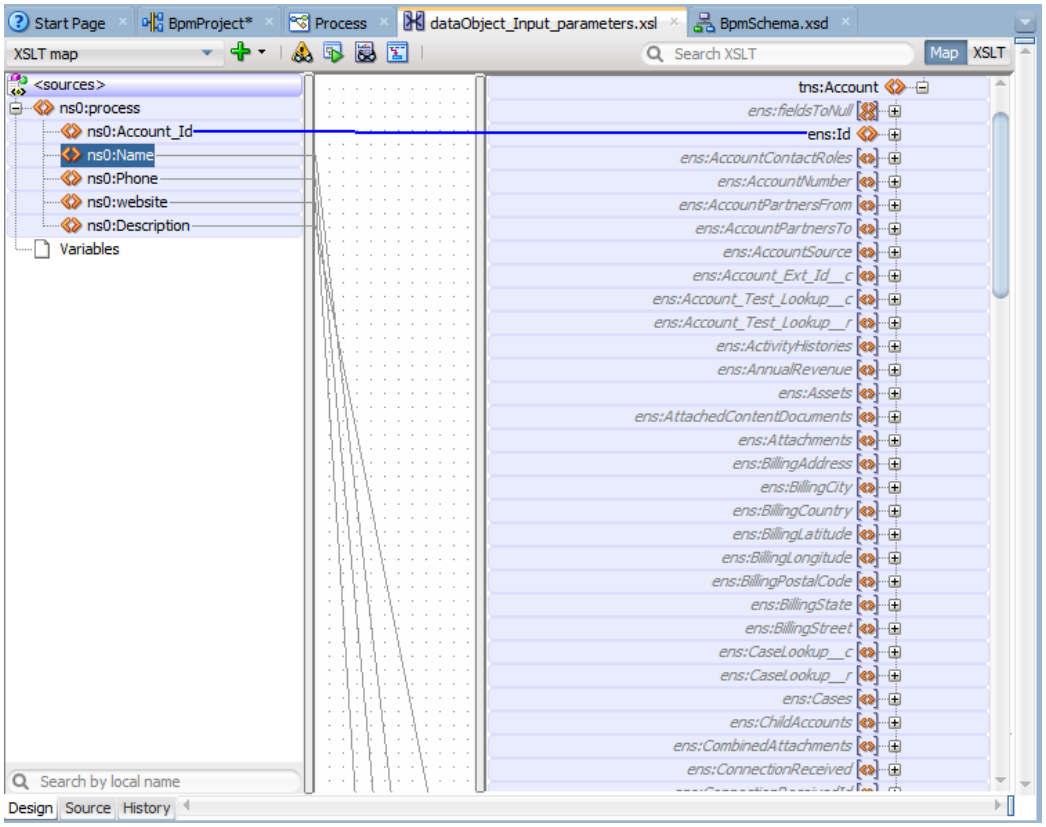
Figure 11 - 55 Data - Association



22. Now Perform the mapping from Source side to the Target side:
    - Map Account_Id with Id.

- Map Name with Name.
- Map Phone with Phone.
- Map website with Website.
- Map Description with Description as shown in figure

Figure 11 - 56 Mapping From Source to Target



23. From the **Activity** Tab, click and drag the **Script** Activity in between **ServiceTask** and **End** Activity, as shown in the following screenshots.

Figure 11 - 57 Using Script Activity

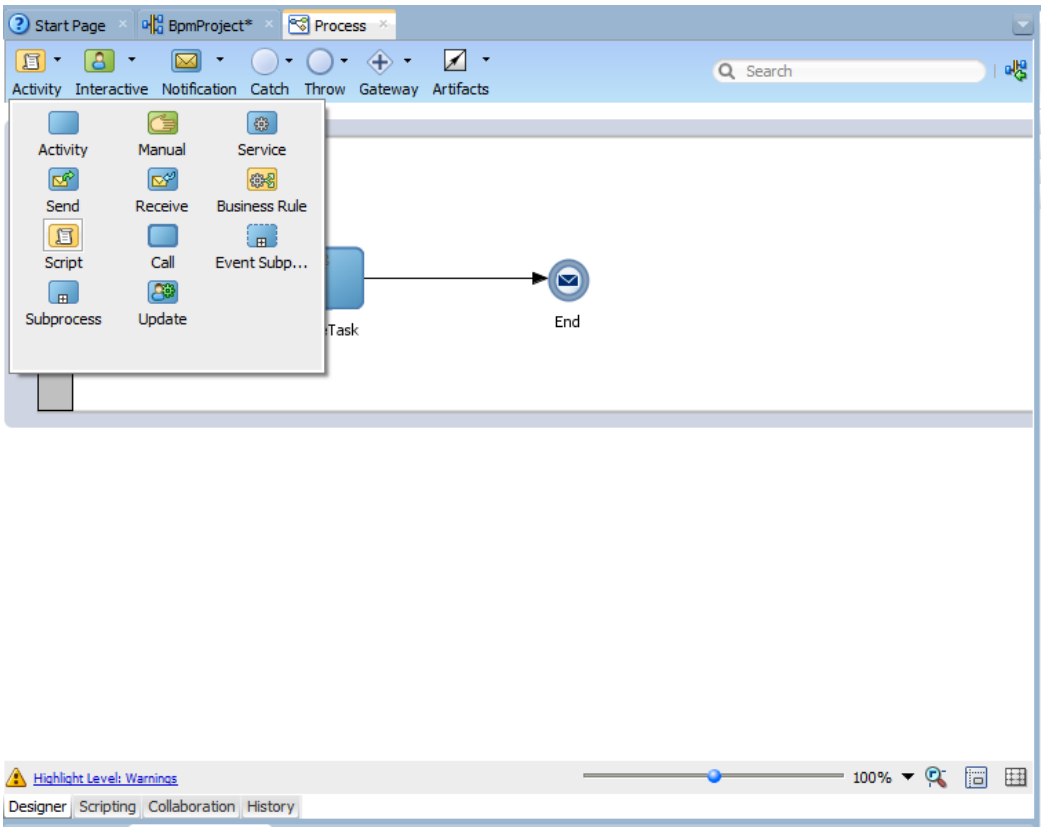
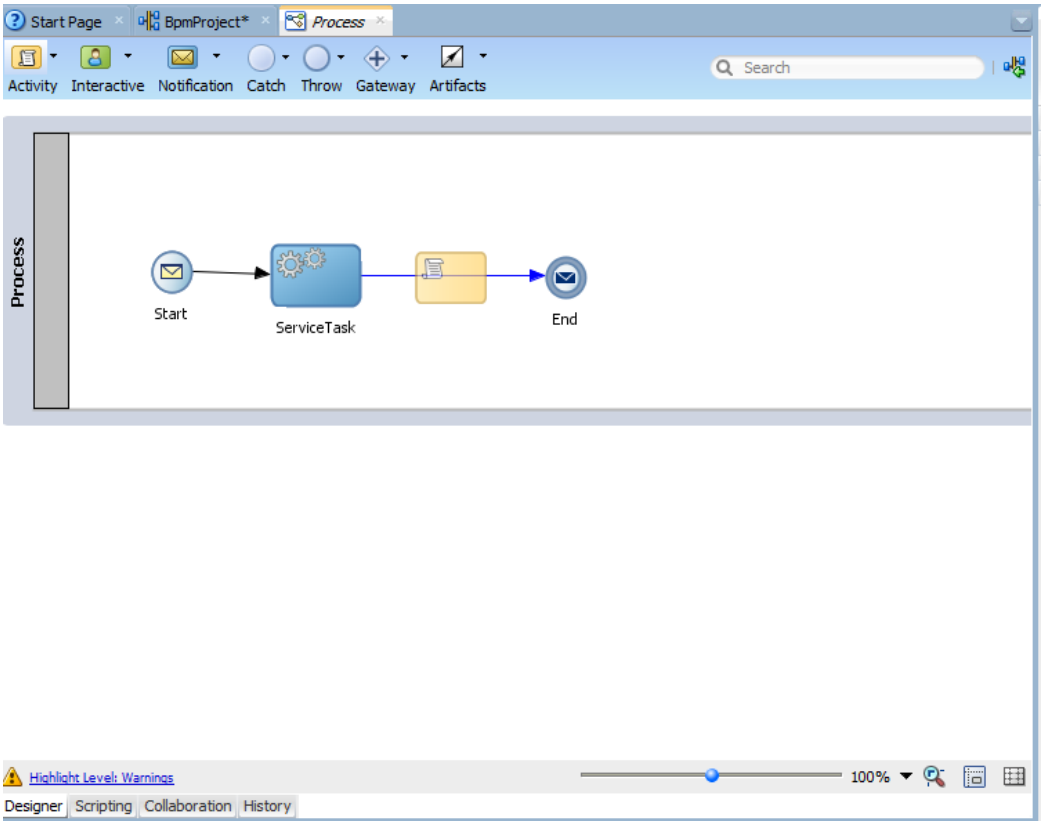
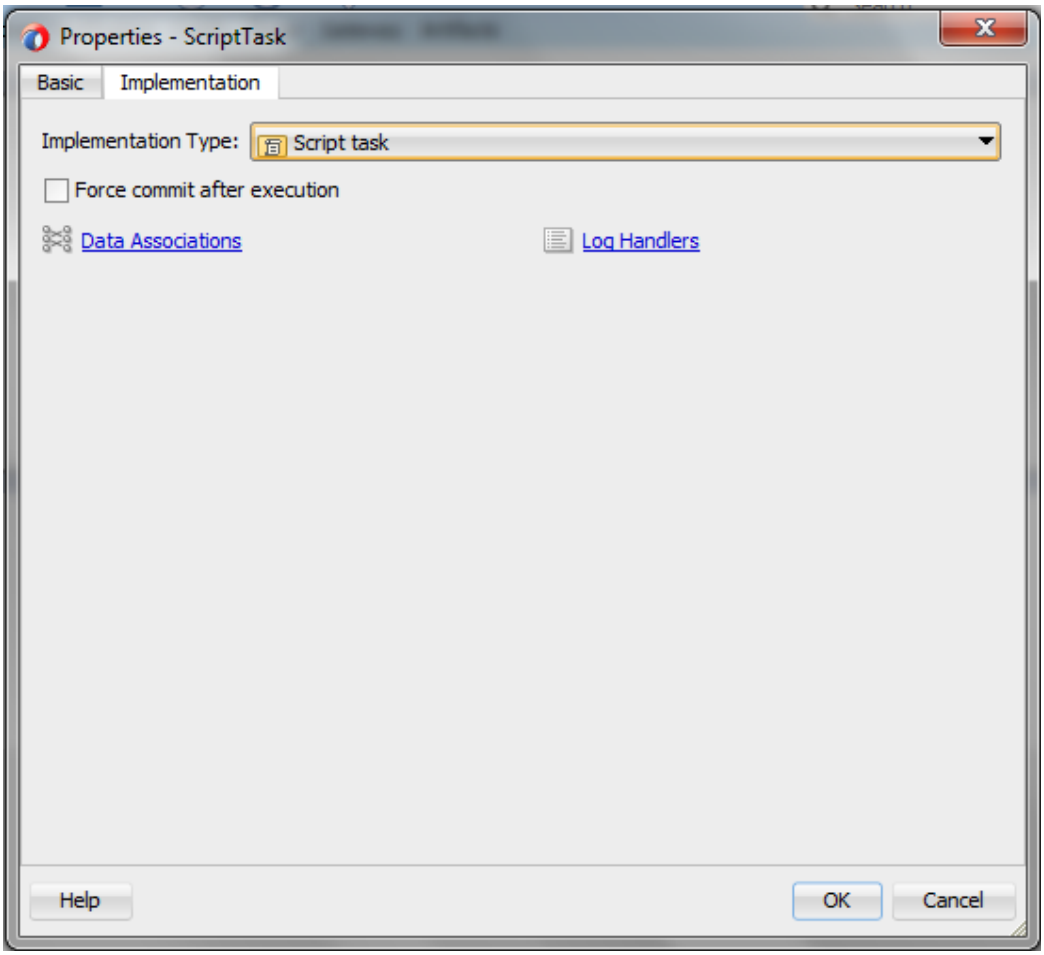


Figure 11 - 58 Process



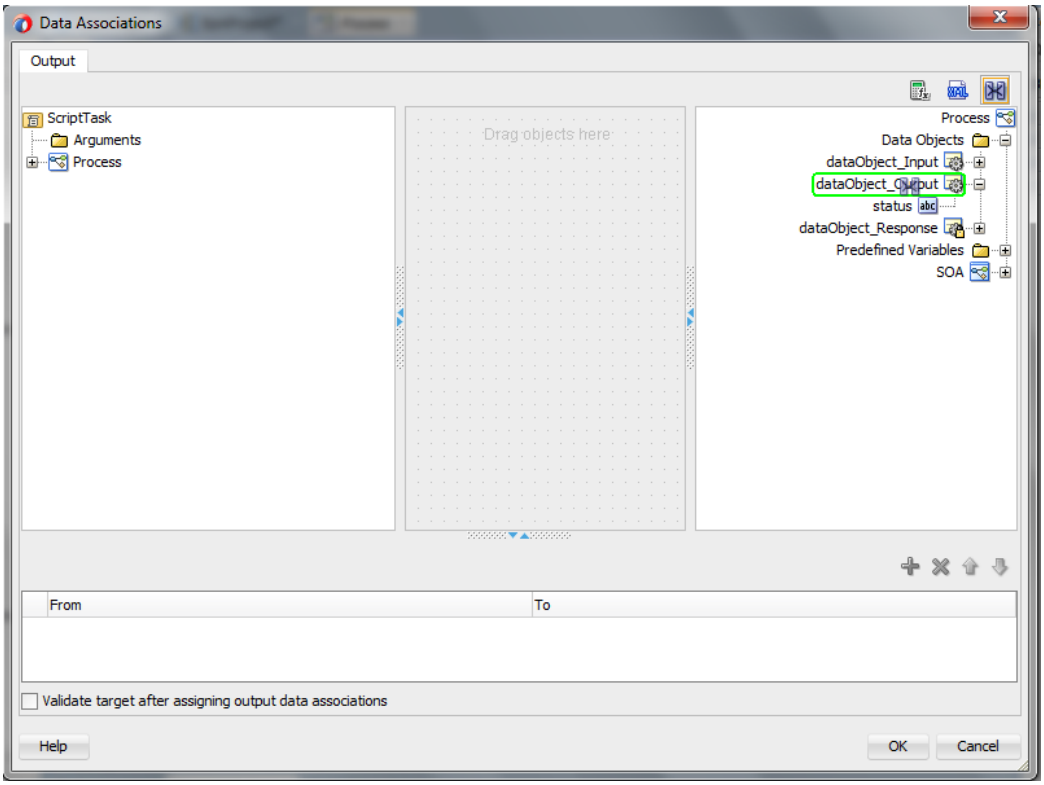
24. The **Properties-ScriptTask** page opens. Click on **Implementation** tab and then click on **Data Association** as shown in the screenshot below**.**

Figure 11 - 59 Properties - ScriptTask



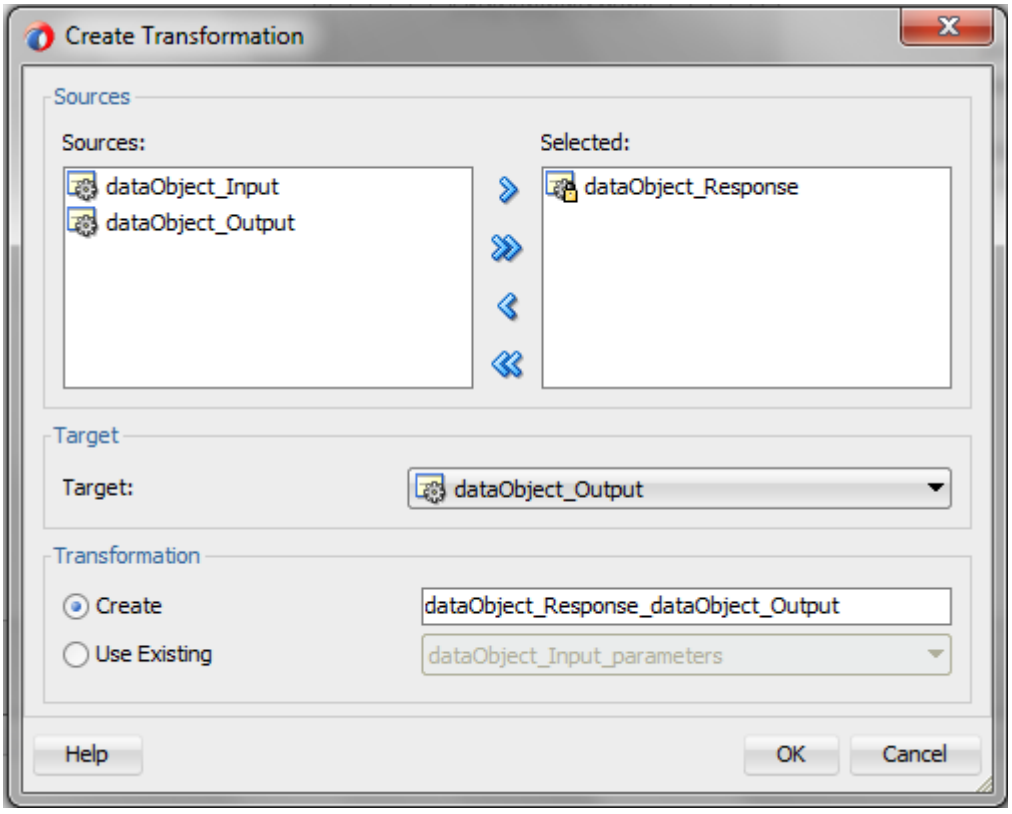
25. A new **Data Association** page opens, click and drag the transformation icon to **dataObject_Output** under **DataObjects** of **Process** as shown in the screenshot below.

Figure 11 - 60 Adding Transformation



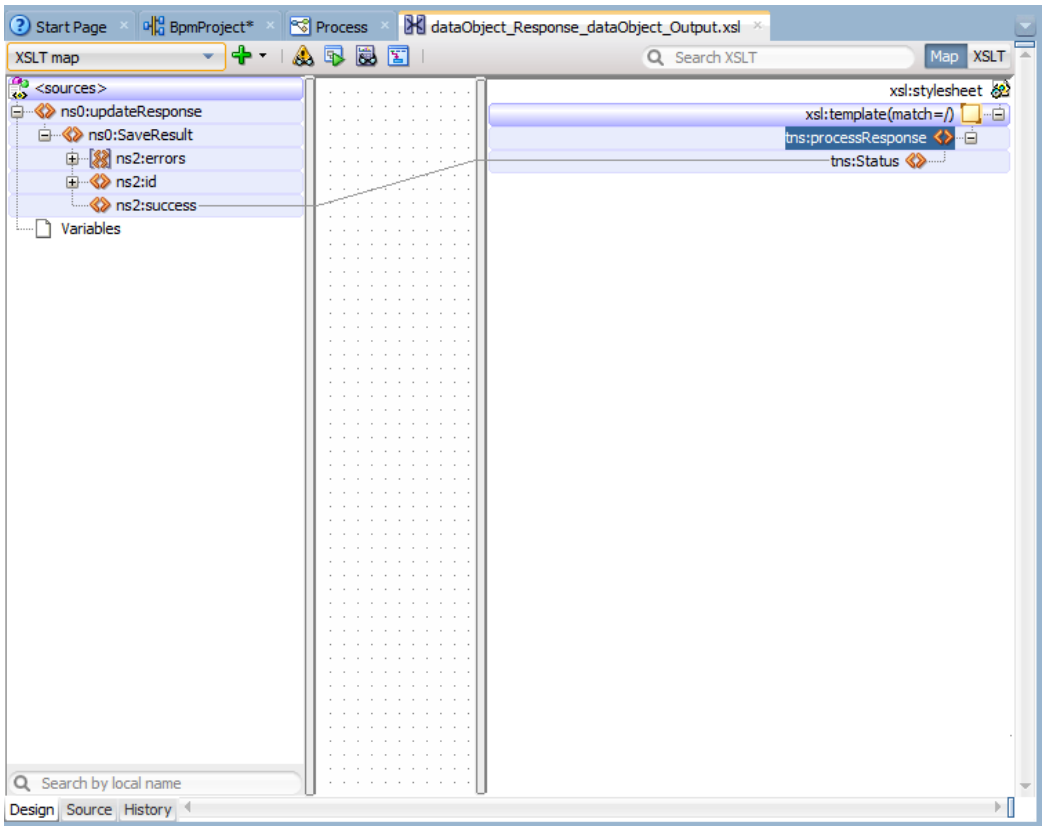
26. The **Create Transformation** page opens. Move **dataObject_Response** from the list of **Source** side to the list of **Selected** and click **OK** button as shown in the screenshot below.

Figure 11 - 61 Create Transformation



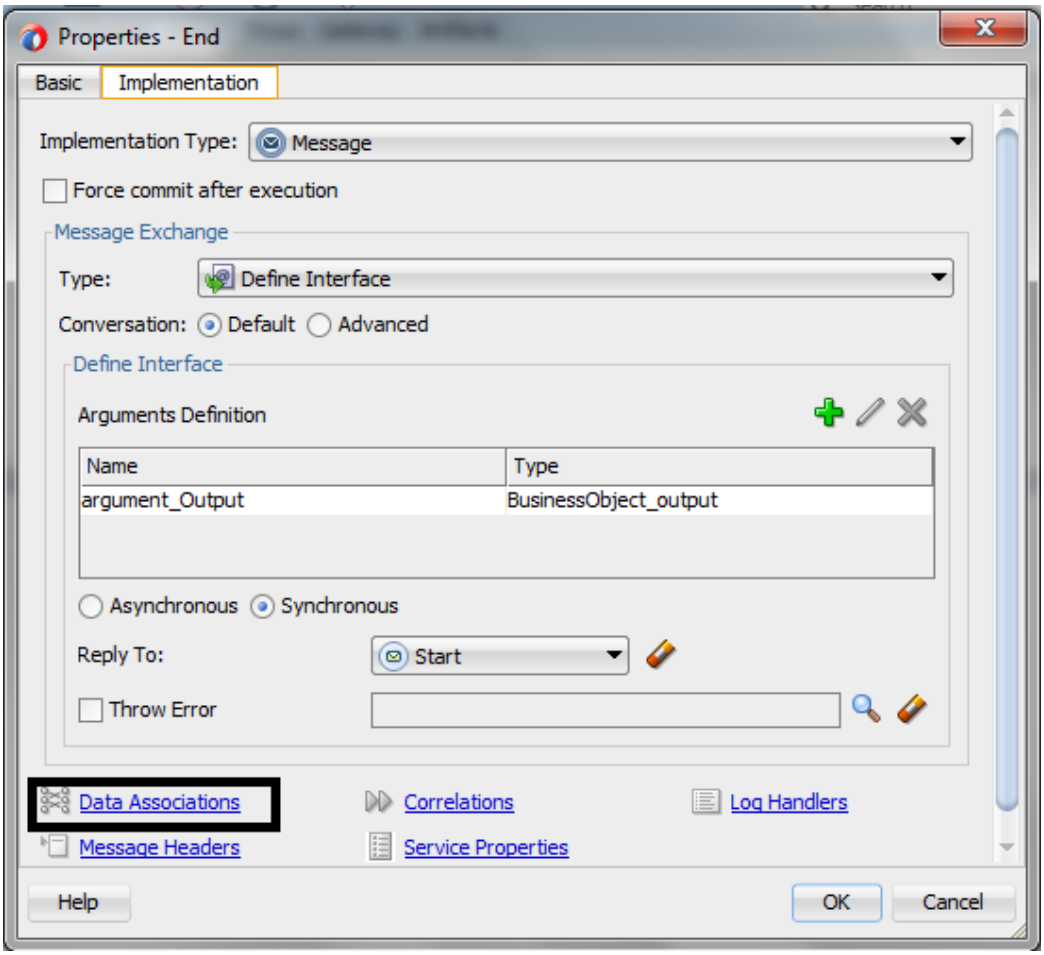
27. Perform the mapping from Source side to the Target side.

Figure 11 - 62 Mapping from Source to Target



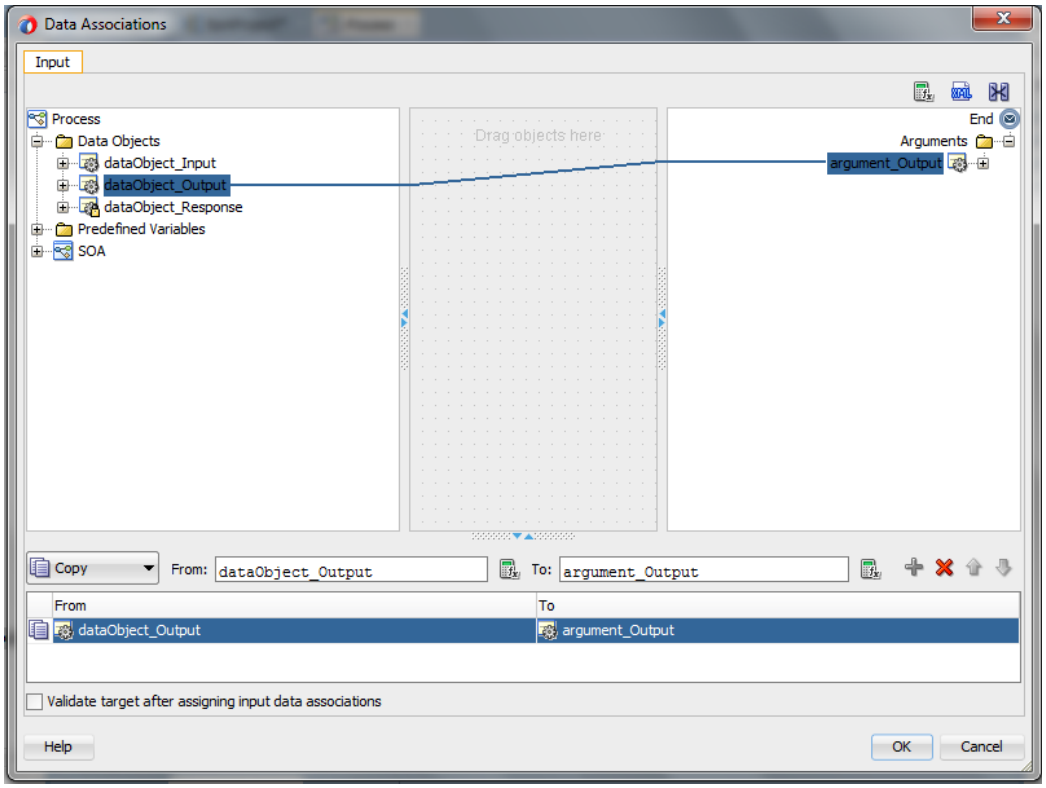
28. Double-click the **End** activity and click on **Implementation → Data Association,** as shown in the screenshot below.

Figure 11 - 63 Properties - End



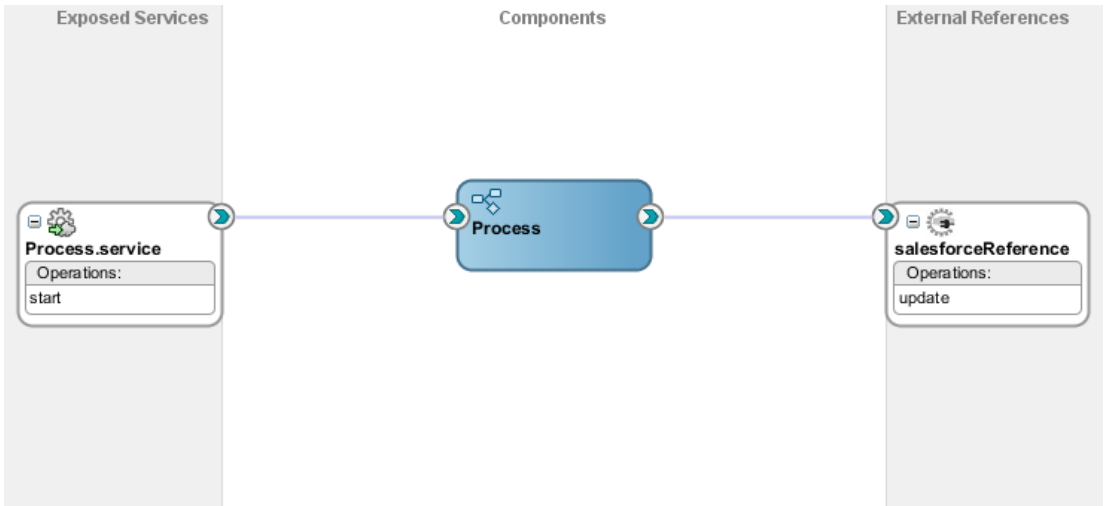
29. A new **Data Association** page opens. Map the **dataObject_Output** of **Process** to the **argument_Output** of **End** and click **OK** as shown in the screenshot below.

Figure 11 - 64 Data – Association Mapping



30. This completes the project creation. The composite.xml looks as shown in the screenshot below.

Figure 11 - 65 Composite.xml

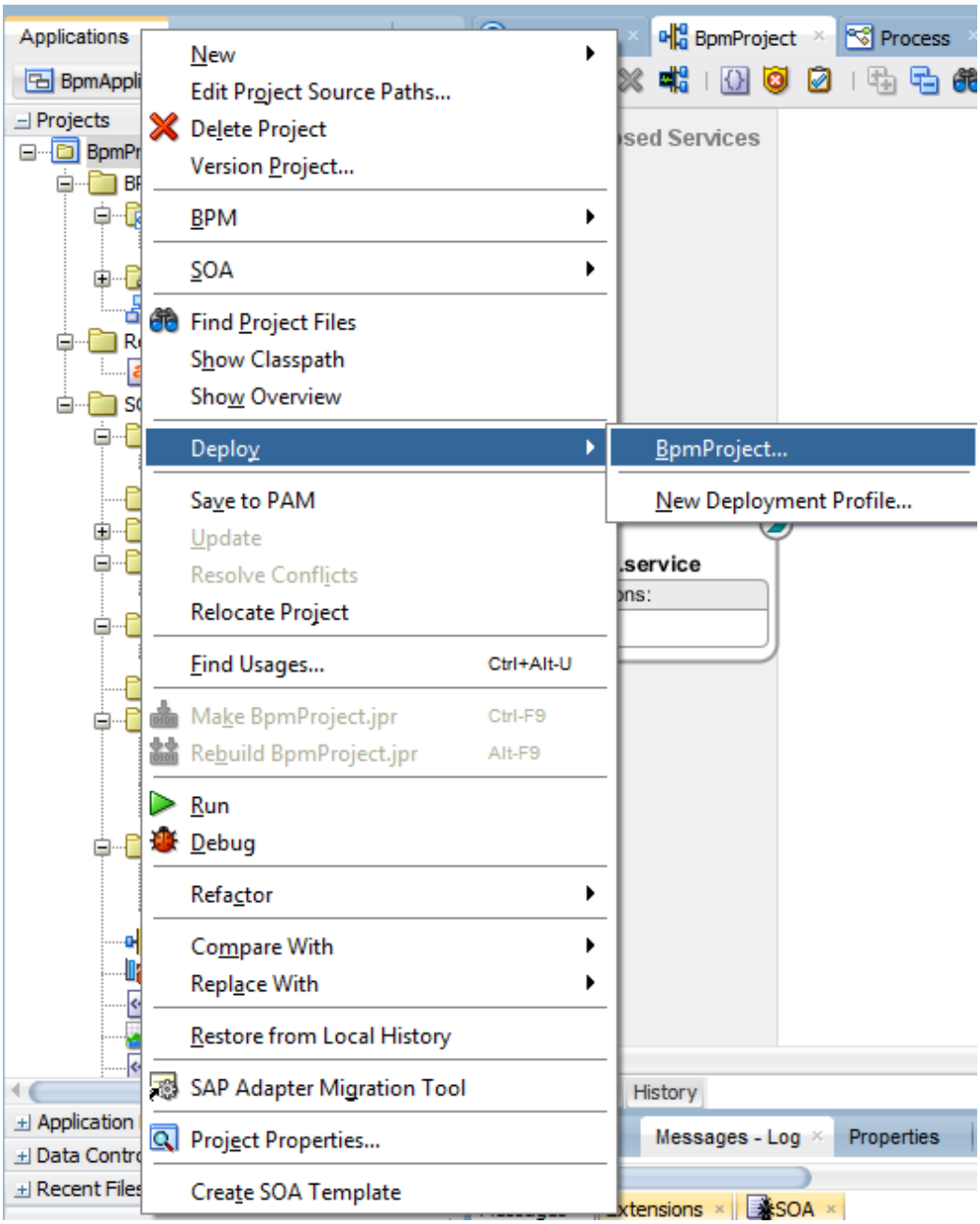


11.9. Deploy the Composite

Perform the following steps to deploy the composite

1. In the **Application Navigator** pane, right-click **BpmProject** and select **Deploy → BpmProject**, as shown in the screenshot below.

Figure 11 - 66 Deploy the Composite



2. Select the **Deploy to Application Server** option and follow the instructions. Using this option, you can deploy the composite on the Application server after providing the details of the server.

11.10. Test the Composite

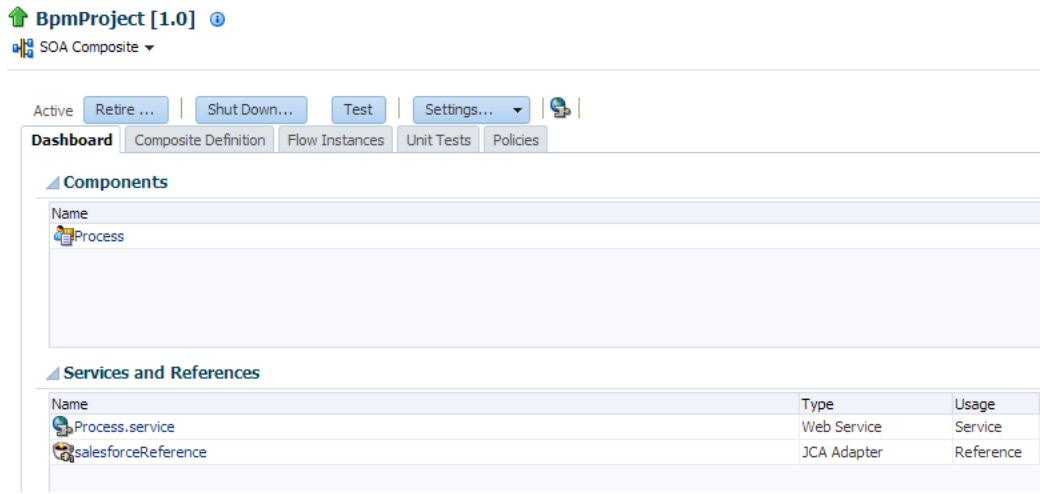
You can run and test the instances of deployed BPM composite applications from Oracle Enterprise Manager Grid Control Console. This enables you to manage a composite application, initiate and track an instance of a composite and to view detailed component instance audit trails. Perform the following process to test the composite.

11.10.1. Test the Outbound Process

Perform the following steps to test the Outbound process:

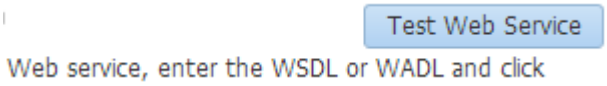
1. Login to Enterprise Manager Console for the server on which you have deployed the project.
2. Open **BpmProject** under the **Default** partition.
3. Click the **Test** button to test the Web service, as shown in the screenshot below.

Figure 11 - 67 Test the Web Service



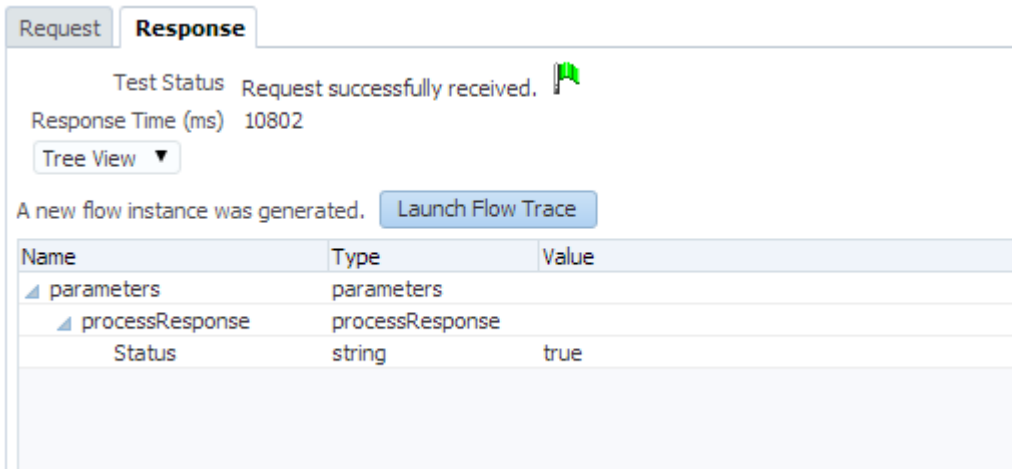
4. Provide the input payload and click the **Test Web Service** button, as shown in the screenshot below.

Figure 11 - 68 Test Web Service



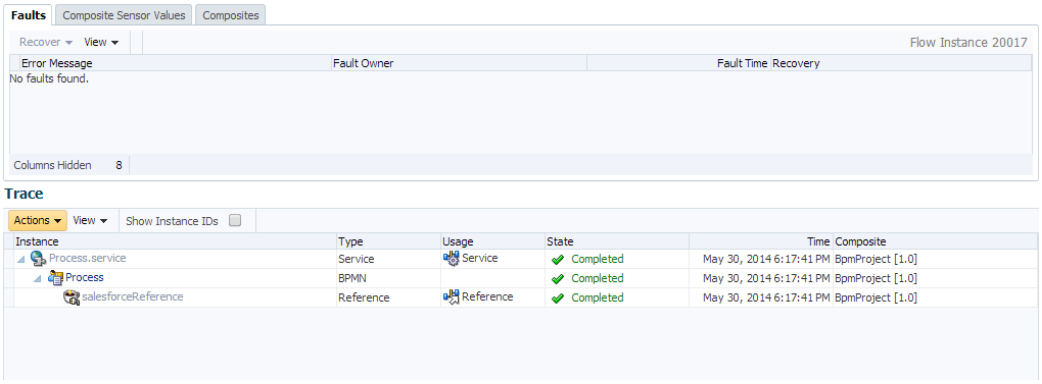
5. After successful execution, the response contains **Status** as true for Account update, as shown in the screenshot below.

Figure 11 - 69 Test Status



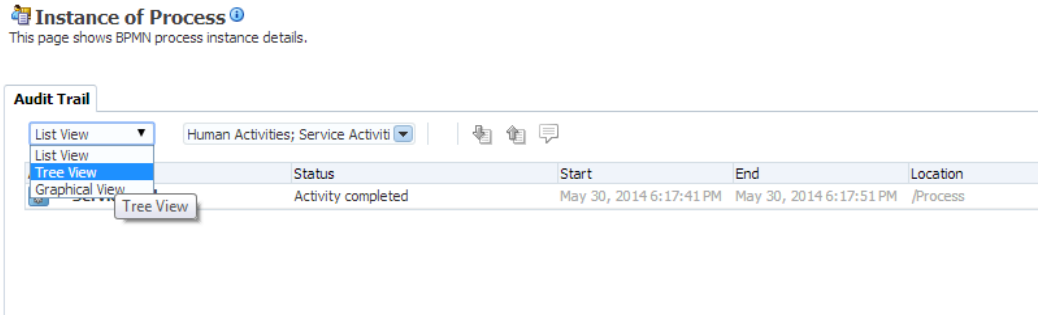
6. Click the **Launch Flow Trace** button to view the **Audit Trail**, as shown in the screenshot below.

Figure 11 - 70 Launch Flow Trace



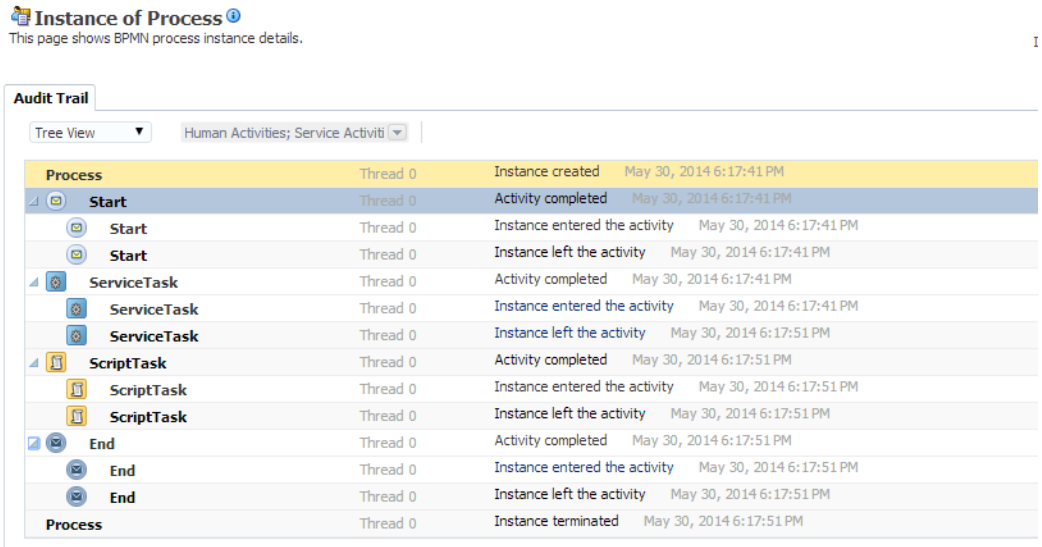
7. Click the List view and select Tree view to see the Audit trail in detail, as shown in the screenshot below.

Figure 11 - 71 Audit Trail



8. The **Audit Trail** looks like as shown in the screenshot below.

Figure 11 - 72 Audit Trail



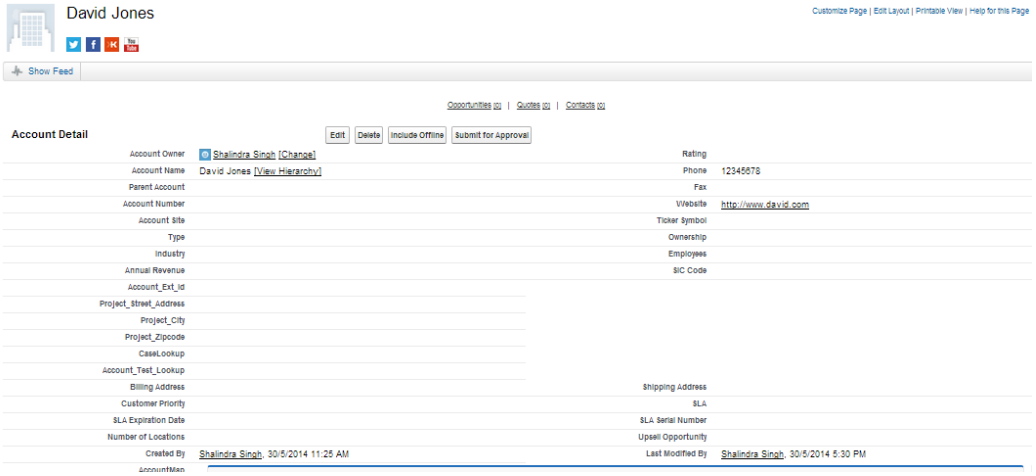
9. The Service Task of the process is shown in the screenshot below.

Figure 11 - 73 Service Task

```
<auditQueryPayload auditId="3370" ciKey="20265" xmlns="http://xmlns.oracle.com/bpmn/engine/audit">
   <serviceOutput>
      <element name="parameters" isBusinessIndicator="false">
         <value>
            <![CDATA[<tns:updateResponse xmlns:tns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/Bpm
   <tns:SaveResult>
      <id xmlns="urn:enterprise.soap.sforce.com">0019000000vG4LhAAK</id>
      <success xmlns="urn:enterprise.soap.sforce.com">true</success>
   </tns:SaveResult>
</tns:updateResponse>
]]>
         </value>
      </element>
   </serviceOutput>
   <dataState>
      <dataObject name="dataObject_Response" isBusinessIndicator="false">
         <value>
            <![CDATA[<updateResponse xmlns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/BpmApplicat
   <tns:SaveResult xmlns:tns="http://xmlns.oracle.com/pcbpel/adapter/salesforce/BpmApplication/BpmProject
      <id xmlns="urn:enterprise.soap.sforce.com">0019000000vG4LhAAK</id>
      <success xmlns="urn:enterprise.soap.sforce.com">true</success>
   </tns:SaveResult>
</updateResponse>
]]>
         </value>
      </dataObject>
   </dataState>
</auditQueryPayload>
```

10. This completes the Update Account scenario. The success of the task can be verified on Salesforce.com using the ID given by you for updating Account, as shown in the screenshot below.

Figure 11 - 74 Updated Account



A

Appendix

This appendix outlines the steps for generating enterprise WSDL for your Salesforce.com organization and how to configure the Credential Store Key in your Enterprise Manager. These steps are pre-requisites for proper adapter functioning and need to be performed only once, provided you do not make any object level changes in your organization. These sections have been referenced several times in the document and need to be followed exactly as described.

This appendix contains the following topics:

- Section A.1, "Generating the Enterprise WSDL"
- Section A.2, "CSF Key in Enterprise Manager"

A.1 Generating the Enterprise WSDL

To download the Web Services Description Language (WSDL) file, required by the adapter to integrate different applications with Salesforce.com, you must have “Modify All Data” permission. (The System Administrator profile has this permission.)

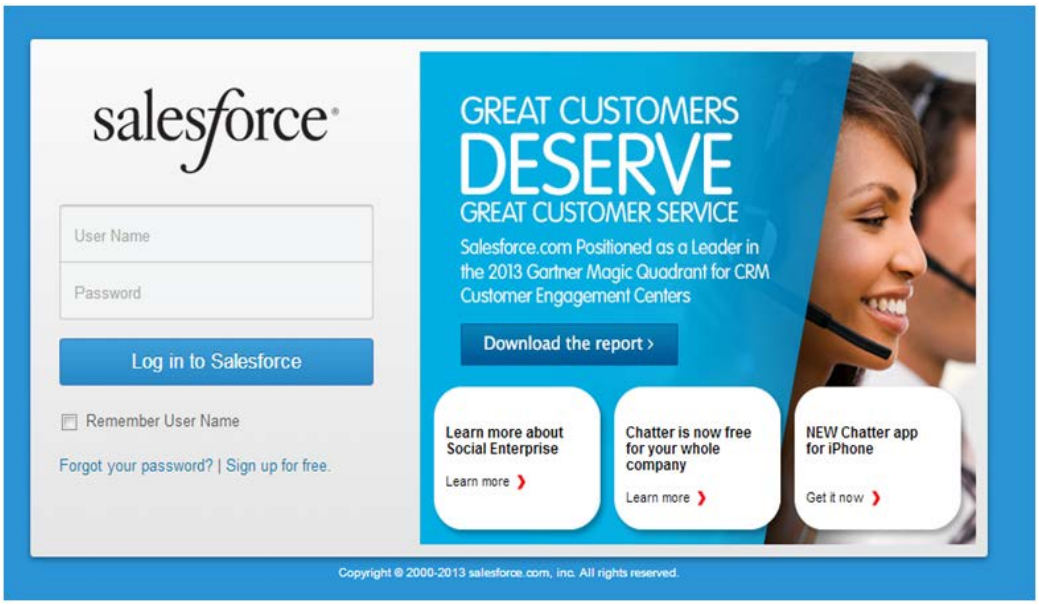
The generated WSDL defines all of the API calls, objects (including standard and custom objects), and fields that are available for API access for the organization.

Perform the following steps to generate the WSDL file for the organization:

1. Log in to your Enterprise, Unlimited, or Developer Edition Salesforce.com account. Open the Web browser and enter the following URL: `www.salesforce.com`

The login window is displayed, as shown in Figure A–1.

Figure A- 1 Login Window

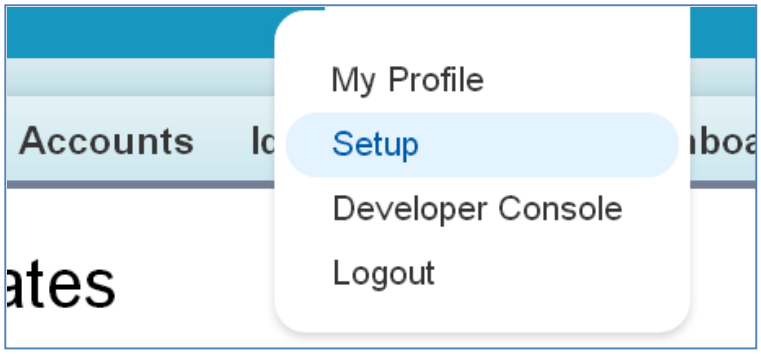


2. Log in to the **Salesforce.com** using a valid user name and password.

Note that you must login as an administrator or as a user who has the **Modify All Data** permission. Logins are checked to ensure they are from a known IP address.

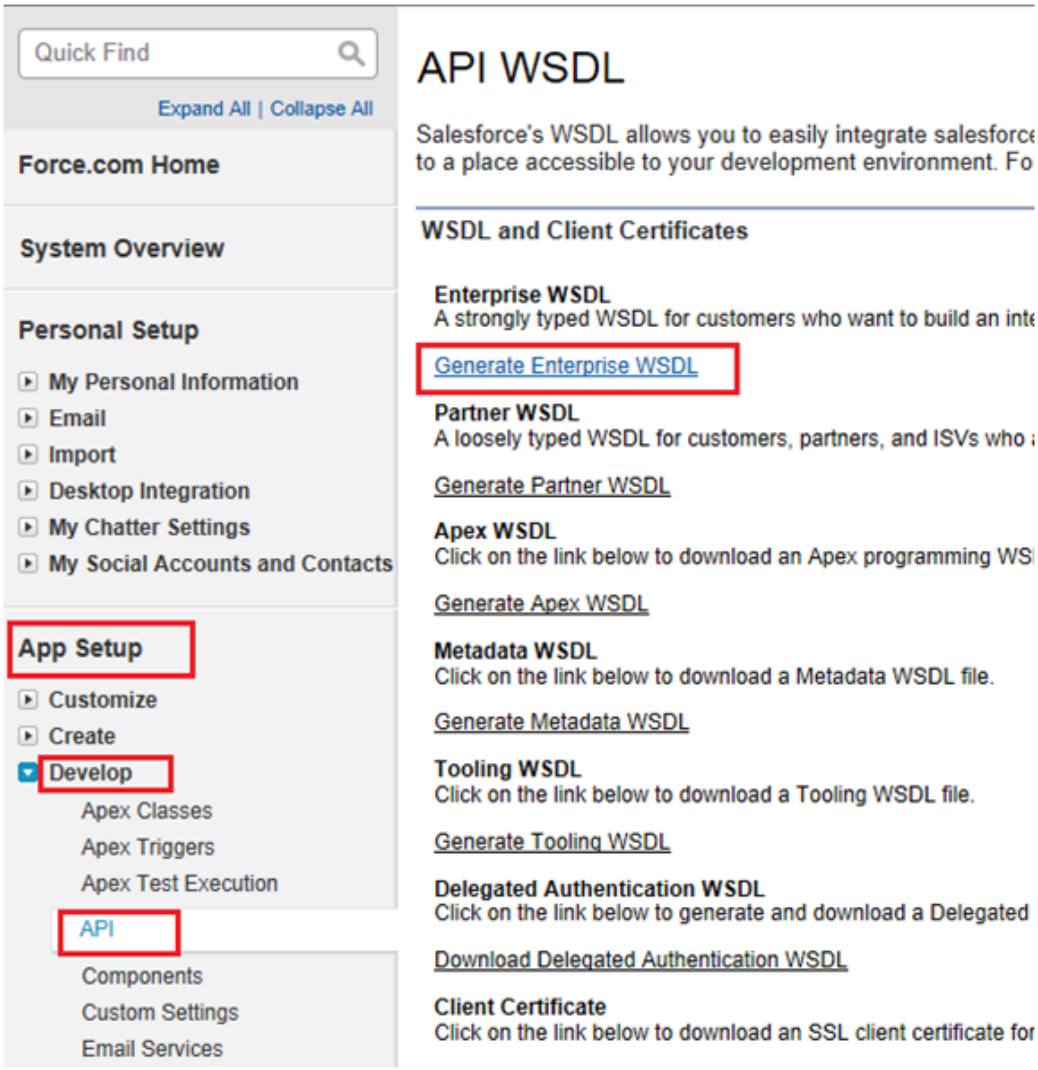
3. Click on **User menu for** user name, and select **Setup**, as shown in Figure A–2.

Figure A- 2 Setup Window



4. Under **App Setup**, Expend **Develop** and click on **API** to display the WSDL download page, as shown in Figure A–3.

Figure A- 3 WSDL Download Page



5. If the organization has managed packages installed in the organization, click **Generate Enterprise WSDL**, Salesforce.com will prompt you to select the version of each installed package to include in the generated WSDL.

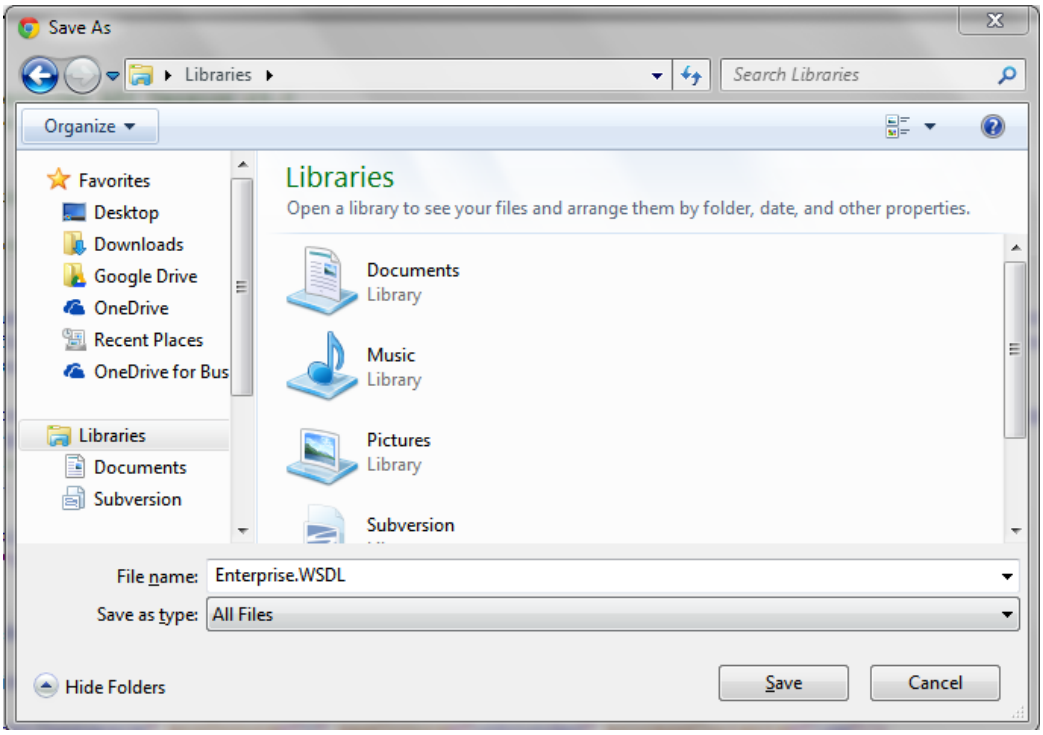
OR

Right-click on **Generate Enterprise WSDL** and save it to a local directory.

Note that in the Right-click menu, Internet Explorer users can choose **Save Target As**, while Mozilla Firefox users can choose **Save Link As** to save it to the local directory.

6. The **Save** dialog is displayed, as shown in Figure A – 4.

Figure A- 4 Save Dialog Box



7. Provide a name for the WSDL file and a location to save the WSDL file on your file system.
8. Click **Save**.

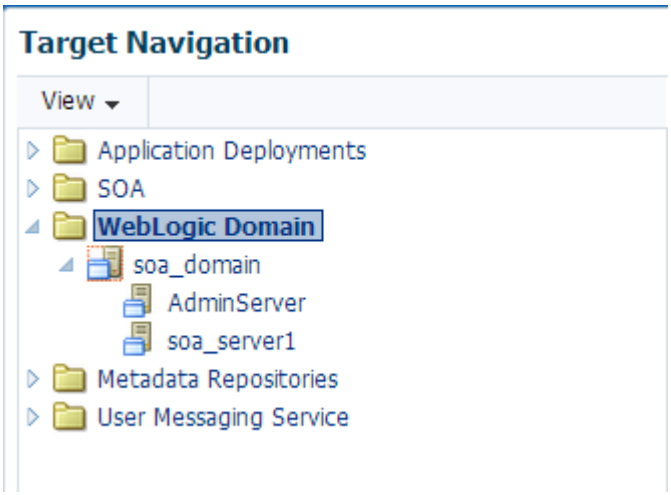
A.2 CSF Key in Enterprise Manager

The Oracle Cloud Adapter for Salesforce.com provides enhanced security through the Credential Store Framework (CSF) Key. This key needs to be presented on the WebLogic server where the adapter is being deployed, and has to be configured in the Enterprise Manager for the Cloud Adapter for Salesforce.com to be successfully able to retrieve the salesforce.com login credentials.

Follow the steps below to configure the CSF key in the Enterprise Manager:

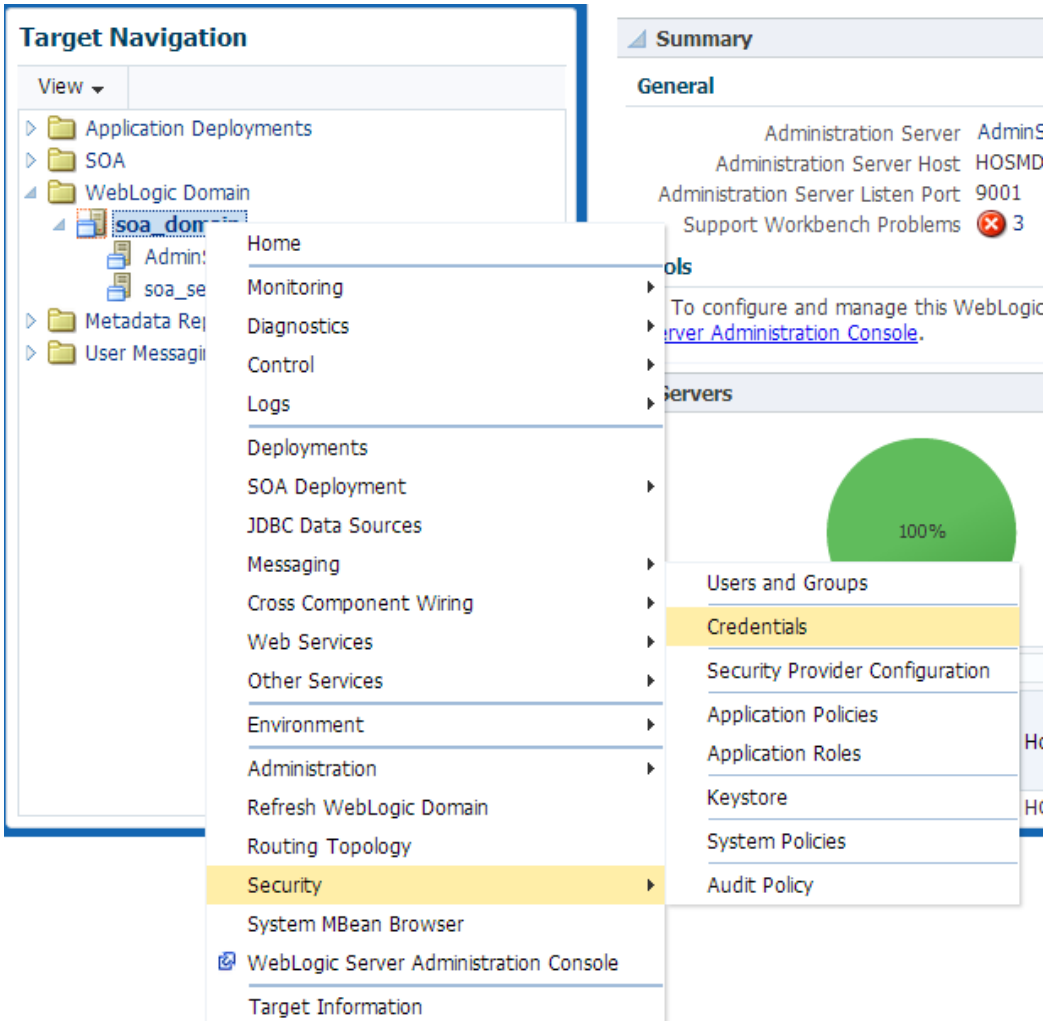
1. Login to **Enterprise Manager** and navigate to **WebLogic Domain**, as shown in Figure A-5.

Figure A- 5 Enterprise Manager



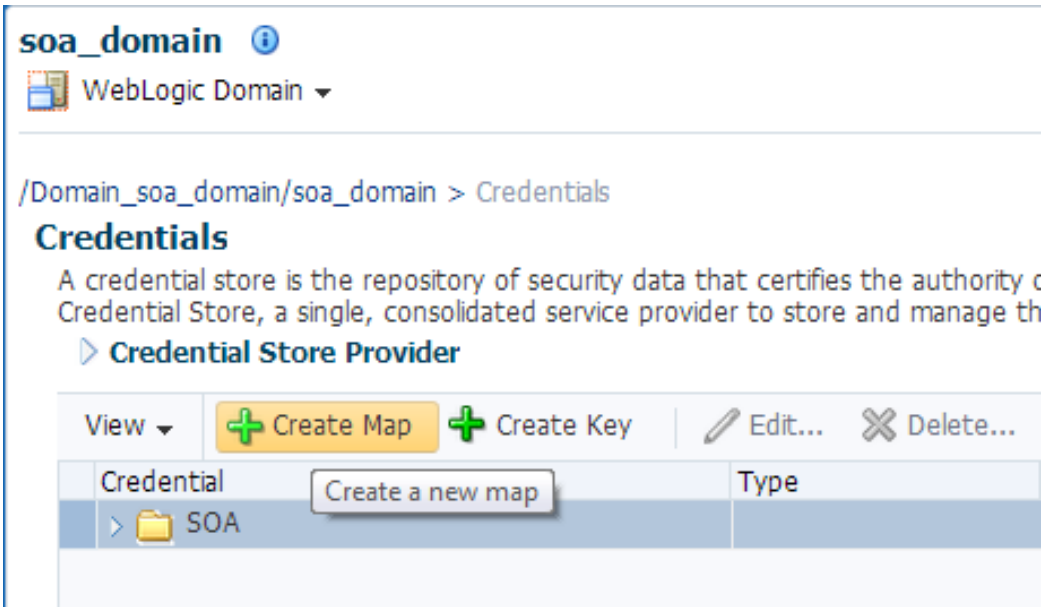
2. Right-click the managed server you are using for deployment and navigate to **Credentials** under **Security**, as shown in Figure A-6.

Figure A- 6 WebLogic Domain in Enterprise Manager



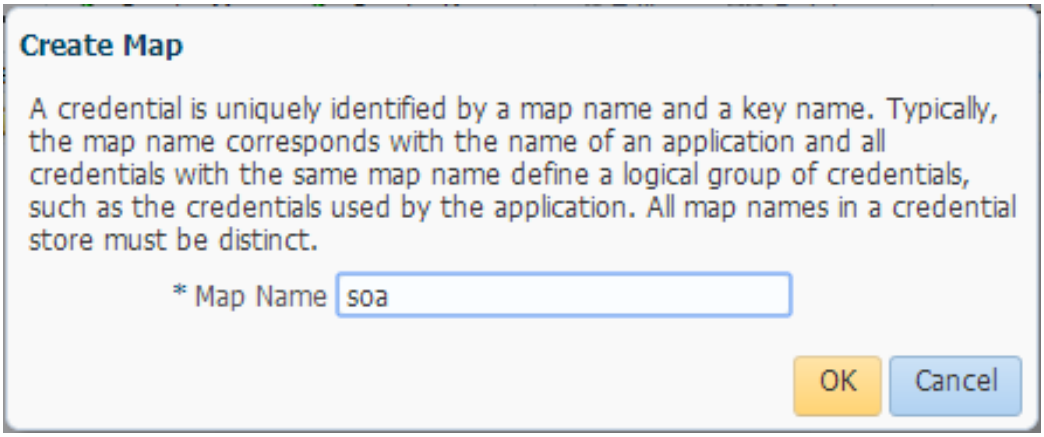
3. The Credentials page is displayed. Click on **Create Map**, as shown in Figure A-7.

Figure A- 7 Create Map Screen



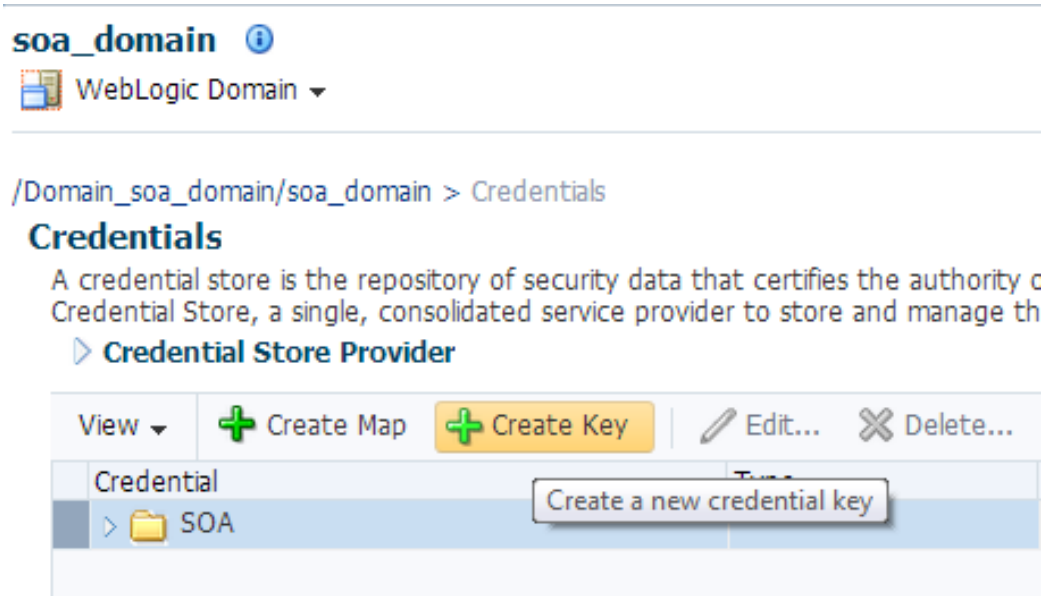
4. Create a map with the name **SOA** and click **OK**, as shown in Figure A-8.

Figure A- 8 Map Name



5. The created SOA map is displayed in the **Credential** list, Select **SOA** map from credential list and click on **Create Key**, as shown in Figure A-9.

Figure A- 9 Create Key



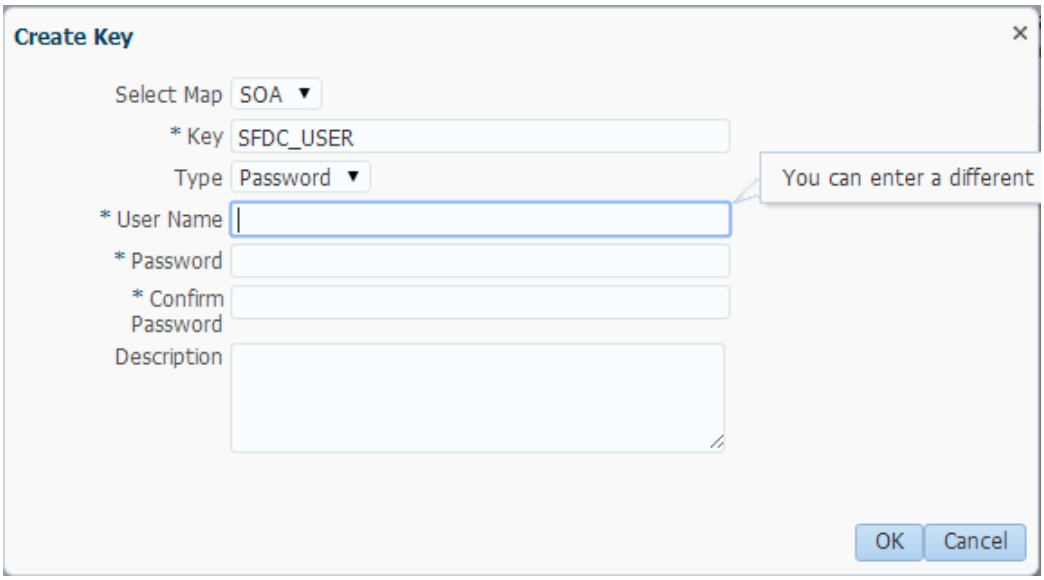
6. Enter a unique CSF key to be associated with a user account, as shown in Figure A-10.

Note that this key will have to be configured in exact same form and characters in the JDeveloper Adapter configuration wizard.

Figure A- 10 Create Key Screen

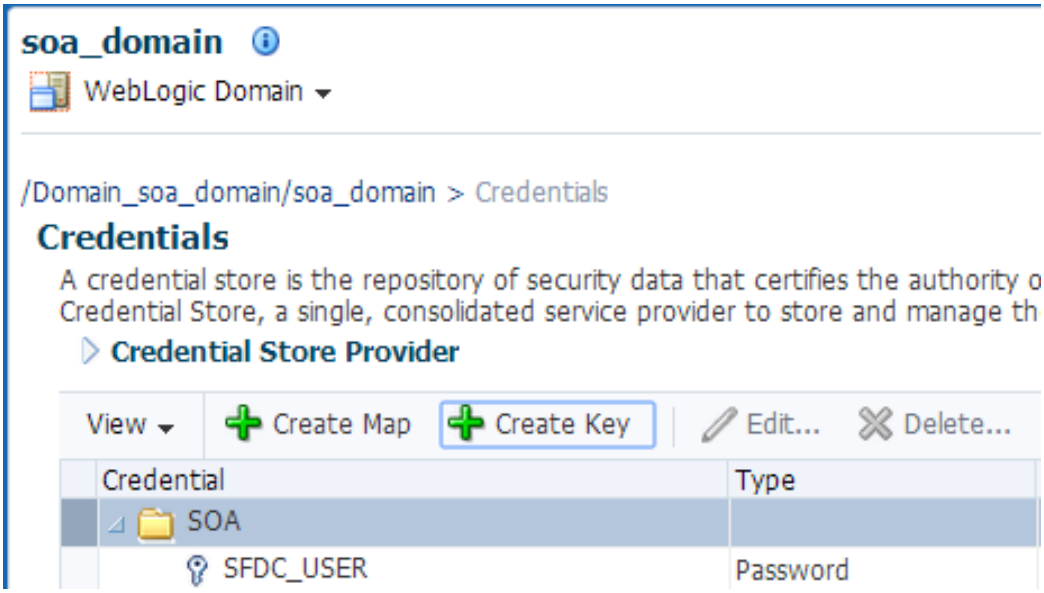
7. Keep the **Type** of key as **Password** and enter the remaining credentials, the optional description and click **OK**, as shown in Figure A-11.

Figure A- 11 Enter Credentials



8. The added CSF key is displayed under the SOA map, as shown in Figure A-12.

Figure A- 12 Credentials Store Provider



Glossary

Adapter

An adapter is a software that an application client or an application server uses to connect to a specific Enterprise Information System (EIS).

Business Service

Also known as a Web service. A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity.

Integration WSDL

The non-polymorphic WSDL generated by the Oracle Cloud Adapter for Salesforce.com during design-time; contains the request and response structures for the selected operation and objects.

Enterprise WSDL

The enterprise WSDL file is a strongly typed representation of your organization's data. It provides information about your schema, data types, and fields to your development environment, allowing for a tighter integration between it and the Force.com Web service. This WSDL changes if custom fields or custom objects are added to, renamed, or removed from, your organization's Salesforce.com configuration directly from Salesforce.com
- http://www.salesforce.com/us/developer/docs/api/Content/sforce_api_quickstart_intro.htm

MDS

An application server and Oracle relational database that keep metadata in these areas: a file-based repository data, dictionary tables (accessed by built-in functions) and a metadata registry. One of the primary uses of MDS is to store customizations and persisted personalization for Oracle applications.

CSF

Credential Store Framework (CSF) stores the login information of the user matched to a unique key called the CSF Key.

CSF Key

Credential Store Framework Key; used to externalize the login credentials of the user from design-time and run-time of the Oracle Cloud Adapter for Salesforce.com. The Oracle Cloud Adapter for Salesforce.com uses this key to retrieve the login information of the Salesforce.com user from the Credential Store.

OPatch

OPatch is the Oracle database's Interim (one-off) Patch Installer. If OPatch is not installed into your Oracle Home ($ORACLE_HOME/OPatch), you may need to download it from Metalink and install it yourself. All patches that are installed with Oracle's OPatch Utility can be listed by invoking the **opatch** command with the **lsinventory** option.

Cloud SDK

Cloud Software Development Kit is a set of development tools that allows for the creation of the design-time and run-time of cloud adapters for Oracle SOA Suite.

Terms and Acronyms

CSF- Credential Store Framework

DT- Design-time

OSB – Oracle Service Bus

RT – Run-time

SFDC – Salesforce.com

SOA –Service-Oriented Architectures

SOSL – Salesforce Object Search Language

SOQL – Salesforce Object Query Language

WSDL –Web Services Description Language

Index

A

Adapter Configuration · 2
AllOrNoneHeader · 3-18
AllowFieldTruncationHeader · 3-19
Architecture · 1
AssignmentRuleHeader · 3-19
Audit Trail · 11-115
Authentication Key · 4-5
Authentication Key field · 10-6
Automatic Commit · 3-1

C

Certificate Export Wizard · 2-8
Clear Cache · 4-7
Component Palette · 4-1
convertLead · 3-2
ConvertLead · 2-5
Create XML Schema · 7-7
Create: · 2-5
Creating Outbound Processes Using Oracle Service Bus · 6-1
CSF Key · 2-2

D

Data Association · 7-38
Delete · 2-5
Design Phase · 3-22
Design-Time · 2-12
Destination Module · 11-141

E

Enterprise WSDL · 1-4
Execution Phase · 3-23

F

Filter Field · 4-9
Functional Overview · 1-1

G

getDeleted · 3-3
getUpdated · 3-4
getUserInfo · 3-11

K

Keystore · 2-11
Keytool · 2-10

L

LimitInfoHeader · 3-3, 3-4, 3-11

M

merge · 3-5
Merge · 2-5

O

Offline Configuration · 3-26
Oracle Metadata Services · 11-75

P

Password · 2-1

Q

Query · 2-5
Query Test · 3-28
Query Test Tool · 3-29
queryAll · 3-15
queryLocator · 11-50

R

Refresh Bind Parameters · 4-11
Retrieve · 2-5
Run-Time · 2-13

S

Salesforce Exceptions · 3-30
Salesforce Faults · 3-30
Salesforce Host Unreachable · 3-30
Search · 2-5
Service Bus Console · 6-24
SOAP API · 1-4, 3-1
Supported Versions · 1-3
Synchronous BPEL Process · 11-5

T

Test Connection Functionality · 4-7

U

undelete · 3-5
Update · 2-5
upsert · 3-6
Upsert · 2-5
User Name · 2-1

V

varDone · 11-51
varQueryLoc · 11-51

W

WSDL Location · 2-17