webMethods Adapter for Documentum

Installation and User's Guide

Version 10.2

April 2018

WEBMETHODS

This document applies to webMethods Adapter 10.2 for Documentum and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2018 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at

http://softwareag.com/licenses.

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at http://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

DOCUMENT ID: ADAPTER-DOCUMENTUM-ECM-IUG-102-20180814

CONTENTS

1 About this Guide

This guide describes how to configure and use webMethods Adapter for Documentum ECM. It contains information for administrators and application developers who want to exchange data with Doumentum ECM system.

To use this guide effectively, you should be familiar with:

- The basic concepts and tasks for working with Documentum
- Creating flow or Java services
- Terminology and basic operations of your operating system
- The setup and operation of webMethods Integration Server.
- How to perform basic tasks with Software AG Designer.

1.1 Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies storage locations for services on webMethods Integration Server, using the convention *folder.subfolder:service*.
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
Italic	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
Monospace font	Identifies text you must type or messages displayed by the system.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

1.2 Online Information

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at http://documentation.softwareag.com. The site requires Empower credentials. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

You can find product information on the Software AG Empower Product Support website at https://empower.softwareag.com.

To submit feature/enhancement requests, get information about product availability, and download products, go to Products.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the Knowledge Center.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at http://techcommunity.softwareag.com. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology

Data Protection

With respect to the EU General Data Protection Regulation (GDPR), our product does not store, collect or process any personal information.

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

2 Overview of the Adapter

2.1 About Adapter - webMethods Adapter for Documentum

Documentum is an ENTERPRISE CONTENT MANAGEMENT platform, now owned by OPENTEXT, as well as the name of the software company that originally developed the technology. EMC acquired Documentum for $1.7 billion in December, 2003.[1] The Documentum platform was part of EMC's Enterprise Content Division (ECD) business unit, one of EMC's four operating divisions.

On September 12, 2016, OPENTEXT, a Canadian technology firm based in WATERLOO, ONTARIO, CANADA that specializes in ENTERPRISE CONTENT MANAGEMENT, announced a definitive agreement to acquire Documentum from Dell EMC for $1.6B USD.[2]

Documentum Content Server is the core of the Documentum content management platform. Content Server governs the content repository and enables a set of content management services for controlling content and processes throughout distributed enterprises. Content Server lets you store, manage, and deploy all types of content, including HTML and XML, graphics, and multimedia. Content Server provides services such as the following:

- Integrated workflow
- Lifecycle and process automation
- Version control
- Security
- Data dictionary for capturing and configuring business rules

With Content Server, users can share and reuse trusted content on demand within and between business units. Administrators can define, organize, automate, and monitor all the functions and tasks of complex business processes.

2.2 Architecture Overview

Adapter for Documentum ECM provides a set of user interfaces, services, and templates that enable you to create integrations with Documentum using Document Foundation Services (DFS). The adapter is provided as a single package that must be installed on Integration Server. For detailed installation instructions see Installing, Upgrading, and Uninstalling Adapter for Documentum ECM. For software requirements, see *webMethods Adapters System Requirements*.

Because Adapter Documentum ECM uses Documentum Foundation Services as a means to communicate with the Documentum system, the adapter requires installing the DFS java libraries that you can download from Documentum partner portal.

Adapter for Documentum ECM enables you to configure the following components:

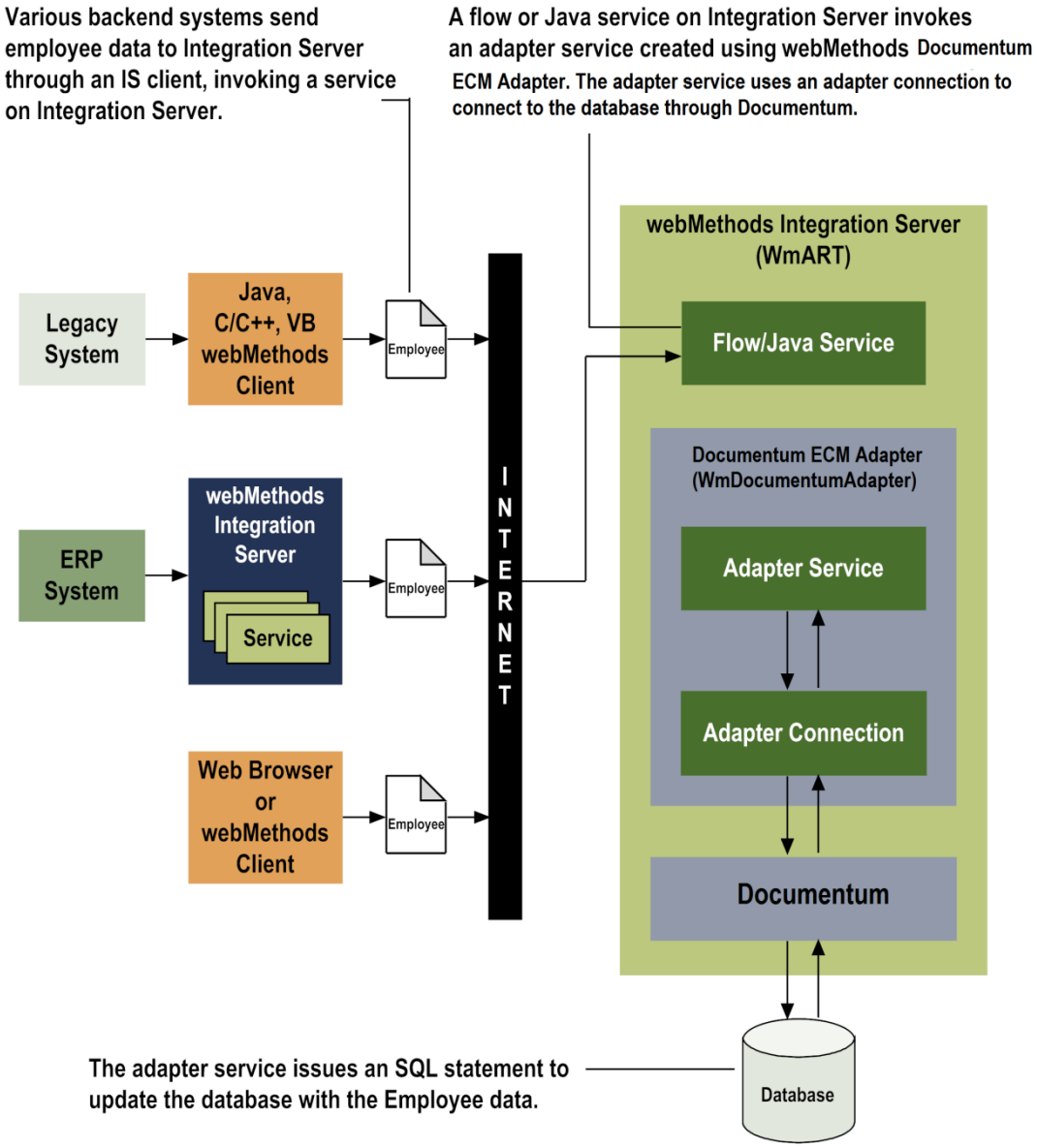
- **Adapter connections.** Enable Integration Server to connect to Documentum systems at run time. You must configure an adapter connection before you can con-

figure adapter services or adapter notifications. For a detailed description of adapter connections, see "Adapter Connections".

- **Adapter services.** Enable Integration Server to initiate and perform operations against a Documentum system. For example, an adapter service could upload an invoice document into Documentum and make it available for a trading partner to its invoices later on and see the original invoice document in PDF format. You configure adapter services using adapter services templates, which are provided with Adapter for Documentum ECM. For a detailed description of adapter services, see "Adapter Services".

- **Adapter notifications.** Monitor a Documentum system and notify Integration Server when an action (not initiated by Integration Server) occurs on a particular document type. For example, an adapter notification could notify Integration Server when an upload of a particular document type has occurred. For a detailed description of adapter notifications, see "**Error! Reference source not found.**".

The following diagram shows business integration where an adapter service is used to update a database with employee data. Several different types of external Integration Server (IS) clients could provide the employee data.



The architecture for integrations using adapter notifications is similar to the architecture for integrations using adapter services shown above, but it varies according to the type of notification. The primary difference between these types of integrations is that notifications are initiated by events that occur on Documentum, not by actions that occur on Integration Server.

With adapter notifications, you can capture event data from the Documentum system and use it to initiate another action within Integration Server. For example, you could create an adapter notification to monitor an invoice document type within Documentum and whenever an invoice document is uploaded to Documentum, you could post that employee data to a webMethods Broker. Broker clients could then subscribe to that notification's publishable document.

For more information about the architecture for the different types of adapter notifications, see "Notifications support component architecture"

2.3 Package Management

Adapter for Documentum ECM is provided as a package called WmDocumentumAdapter that you manage like any package on Integration Server.

There are several considerations regarding how you set up and effectively manage your packages on Integration Server:

- You must create user-defined packages for your connections, adapter services, and notifications. For details, see "Adapter for Documentum Package Management".
- You should understand how package dependencies work so you make the best decisions regarding how you manage your adapter services and notifications. For details, see "Package Dependency Requirements and Guidelines".

2.4 Adapter Connections

Adapter for Documentum connects to the Documentum system through Documentum Foundation Services API at run time. You create one or more connections at design time to use in integrations. Number of connections you create depend on how many Documentum system repositories are connecting to and your integration needs.

For example, if you are connecting to different Documentum repositories, you will need to create connections that are unique to those repositories. Additionally, if you have multiple installations of the same Documentum repositories in different stages (Development, Test, QA), you access each using different connections.

Adapter for Documentum connections contain parameters that Integration Server uses to manage connections to the database so that they can be used by the adapter to provide services. You configure connections using Integration Server Administrator. You must have Integration Server Administrator privileges to access Adapter for Documentum's administrative screens.

For instructions on configuring, viewing, editing, enabling, and disabling Adapter for Documentum connections, see "Adapter for Documentum Connections" . For information about setting user privileges, see *webMethods Integration Server Administrator's Guide*.

2.4.1 Connection Pools

Integration Server includes a connection management service that dynamically manages connections and connection pools based on configuration settings that you specify for the connection. All adapter services use connection pooling.

A connection pool is a collection of connections with the same set of attributes. Integration Server maintains connection pools in memory. Connection pools improve

performance by enabling adapter services to re-use open connections instead of opening new connections.

2.4.1.1 *Run-Time Behavior of Connection Pools*

When you enable a connection, Integration Server initializes the connection pool, creating the number of connection instances you specified in the connection's **Minimum Pool Size** field when you configured the connection.

Whenever an adapter service needs a connection, Integration Server provides a connection from the pool. If no connections are available in the pool, and the maximum pool size has not been reached, the server creates one or more new connections (according to the number specified in the **Pool Increment Size** field) and adds them to the connection pool. If the pool is full (as specified in **Maximum Pool Size** field), the requesting service will wait for Integration Server to obtain a connection, up to the length of time specified in the **Block Timeout** field, until a connection becomes available.

Periodically, Integration Server inspects the pool and removes inactive connections that have exceeded the expiration period that you specified in the **Expire Timeout** field.

If initialization of the connection pool fails because of a network connection failure or some other type of exception, you can enable the system to retry the initialization any number of times, at specified intervals. For information about configuring connections, see "Configuring Adapter for Documentum Connections".

2.4.1.2 *Built-In Services for Connections*

Integration Server provides built-in services that enable you to programmatically control connections. You can use them to enable and disable a connection, and to return usage statistics and the current state (Enabled or Disabled) and error status for a connection. These services are located in the WmART package, in the pub.art.connection folder.

The setAdapterServiceNodeConnection and setPollingNotificationNodeConnection built-in services enable you to change the connection associated with an adapter service or notification respectively.

For details, see Integration Server built-in services reference guide for your release.

2.5 Adapter Services

To use Adapter for Documentum, you create adapter services. Adapter services allow you to connect to the adapter's resource and initiate an operation on the resource from Integration Server.

You call adapter services from flow or Java services to interact with Documentum document types. The adapter services perform repository operations by calling Documentum Foundation Service APIs. Integration Server then uses adapter.

Adapter services are based on templates provided with Adapter for Documentum. Each template represents a specific technique for doing work on a resource, such as using the Create Document template to upload a new document into Documentum system.

An adapter service template contains all the code necessary for interacting with the resource but without the data specifications. You provide these specifications when you create a new adapter service.

Creating a new service from an adapter service template is straightforward. Using Software AG Designer, you assign the service a default adapter connection.

After you select the connection for the adapter service, you select the adapter service template and supply the data specifications using Designer. Some familiarity with using Designer is required. For more information, see the Designer Service Development online help for your release. See "About this Guide" for specific document titles.

Adapter for Documentum provides the following adapter service templates:

Adapter Service Type	Adapter Service Template	Description
Create	Create Document	Creates a document in Documentum and allows • specifying meta data (attributes) according to a document type selected. • specifying a folder location where to place this document in Documentum (incl. cabinet). • specify a content URL so that the content behind it is streamed and uploaded together with the document. Provides the client an object identifier to later find the document.
Create	Create Folder Path	Creates a folder path in Documentum passing back its object identifier.
Update	Update Document	Updates meta data and content of documents already available in Documentum.
Delete	Delete Document	Deletes a document specifying an object identifier.
Delete	Delete Folder Path	Deletes a folder path in Documentum allowing the client to specify if to delete subfolders and children.

Get	Get Document	Gets information about a Documentum document, passing back meta data and contents.
Search	Search Document	Searches documents in Documentum.

2.5.1 Using Adapter Services

The following table lists the tasks required to use adapter services.

For this task...	Use these tools...
1. Create an adapter connection. For details, see "Adapter for Documentum Connections".	Integration Server Administrator
2. Select the appropriate adapter service template and configure the adapter service. Depending on the type of adapter service, you specify: ■ The adapter connection ■ The document type ■ The query expression used to modify or select data, attributes to choose from a document type. ■ The input fields and types as needed ■ The output fields and types as needed For more information about configuring adapter services, see Adapter Services.	Designer
3. If you plan to use an Integration Server flow or Java service to invoke the adapter service, design the flow or Java service to use this adapter service.	Designer
4. Manage the adapter service. For details, see "Package Management", and "Adapter Services".	Designer and Integration Server Administrator

2.5.2 Changing the Connection Associated with an Adapter Service at Run Time

Integration Server enables you to dynamically select the connection a service uses to interact with the adapter's resource. This feature enables one service to interact with multiple, similar backend resources.

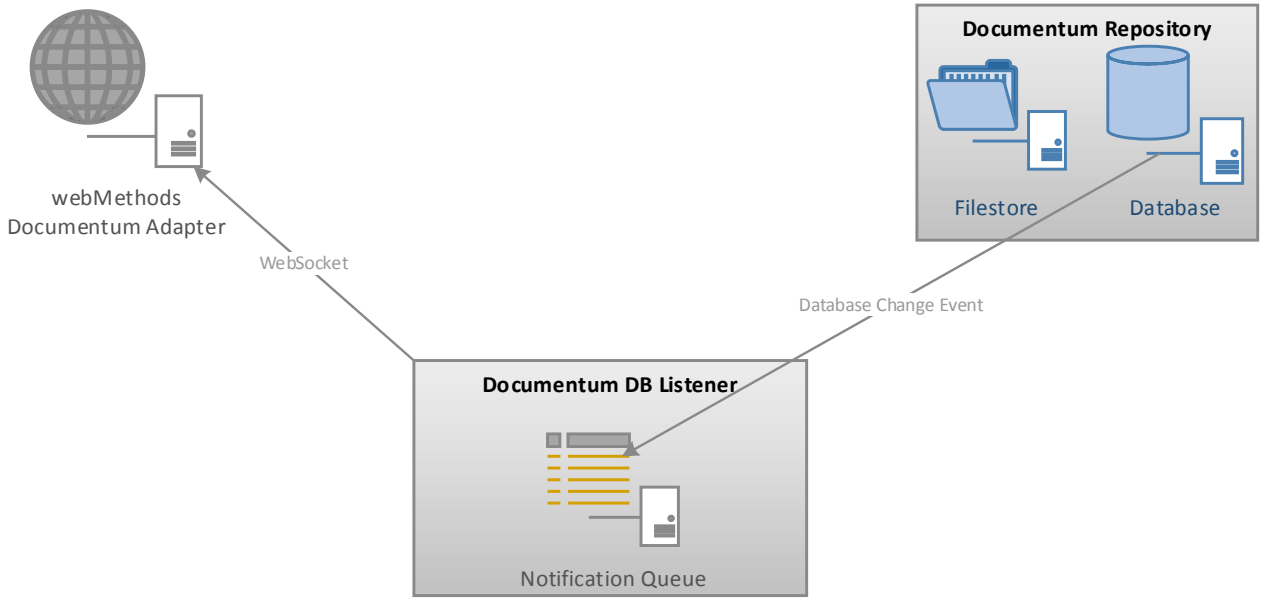
For example, a service can be defined to use a default connection that interacts with your company's production Documentum system. However, at run time you can override the default connection and instead use another connection to interact with the company's test Documentation system.

2.6 Real Time Notifications Support

If you want the webMethods Documentum adapter to receive the real-time notifications from Documentum about the changes in objects (document gets created, updated, versioned, deleted, etc.) you will need to deploy the Documentum DB Listener component.

2.6.1 Notifications support component architecture

Following diagram describes the architecture of the solution supporting the real-time notifications capability between the webMethods Documentum Adapter and a Documentum repository.



Essentially, a listener is registered in the underlying Documentum's database catching events and pushing them into an internal message queue, messages arriving into the

queue are serialized into JSON and sent over to the adapter via a WebSocket connection. Adapter converts this JSON back into object model and passes it on to the webMethods queue for the registered listener adapters to consume the notifications.

Different solutions are available depending on the database backend used by Documentum Content Server. For more details about the available solutions and how to deploy and enable them, refer to the „Installing Documentum DB Listener Solution“ section.

3 Installing, Upgrading, and Uninstalling Adapter for Documentum ECM

3.1 Overview

This chapter explains how to install, upgrade, and uninstall webMethods Adapter for Documentum ECM 10.. The instructions use the Software AG Installer and the Software AG Uninstaller wizards. For complete information about the wizards or other installation methods, or to install other webMethods products, see the webMethods installation guide for your release.

3.2 Requirements

For a list of operating systems, RDBMSs, and webMethods products supported by Adapter for Documentum ECM, see *webMethods Adapters System Requirements*.

Adapter for Documentum ECM has no hardware requirements beyond those of its host Integration Server.

3.3 The Integration Server Home Directory

Beginning with Integration Server 9.6, you can create and run multiple Integration Server instances under a single installation directory. Each Integration Server instance has a home directory under *Integration Server_directory*\instances*instance_name* that contains the packages, configuration files, log files, and updates for the instance.

For more information about running multiple Integration Server instances, see the Integration Server administration guide for your release. See “About this Guide” for specific document titles.

If you are using Integration Server 9.5 and lower, the Integration Server home directory is *Integration Server_directory*. For example, on Integration Server 9.5 the adapter package is installed in the *Integration Server_directory*\packages directory.

This guide uses the *packages_directory* as the home directory in Integration Server classpaths. For Integration Server 9.6 and above, the *packages_directory* is

Integration Server_directory\instances*instance_name*\packages directory. For Integration Server 9.5 and lower, the *packages_directory* is *Integration Server_directory*\packages directory.

3.3.1 Installing Adapter for Documentum ECM

Note: If you are installing Adapter for Documentum ECM in a clustered environment, you must install the adapter on each Integration Server in the cluster, and each installation must be identical. For more information about working with Adapter for Documentum ECM in a clustered environment, see

webMethods Adapter for JDBC Guide.

To install Adapter for Documentum ECM

1 Download Installer from the Empower Product Support website.

2 If you are installing the adapter on an existing Integration Server, shut down the Integration Server.

3 Start the Installer wizard.

4 Choose the webMethods release that includes the Integration Server on which you want to install the adapter. For example, if you want to install the adapter on Integration Server 9.8, choose the 9.8 release.

5 Specify the installation directory as follows:

- If you are installing on an existing Integration Server, specify the webMethods installation directory that contains the host Integration Server.
- If you are installing both the host Integration Server and the adapter, specify the installation directory to use.

6 In the product selection list, select **Adapters >webMethods Adapter for Documentum ECM** 2.0.

If you are using Integration Server 9.6 and above, you can choose to install the package in the default instance. In this case, Software AG Installer installs the adapter in both locations, *Integration Server_directory*\packages and the default instance packages directory located in

Integration Server_directory\instances\default\packages.

7 To download the documentation for the adapter, go to Software AG Documentation website.

8 After the installation completes, close the Installer and start the host Integration Server.

9 See "Installing Documentum Foundation Service Libraries on Integration Server".

10 See "Oracle based DB Listener Solution Installation"

3.3.2 Installing Documentum DB Listener Solution

There are two versions of this solution, so the first step is to find out which one should you install. The proper solution to be installed depends mostly on the database being used by Documentum's Content Server. For Oracle based systems, a web application is provided (war file) which is deployed to an application server. Microsoft SQL server based systems are provided with a Windows service and related scripts for installation.

3.3.2.1 *Oracle based DB Listener Solution Installation*

Database listener solution for Oracle includes two major phases. First, the database itself needs to be updated with a simple configuration table and appropriate privileges must be provided to the user account being used by this solution.

There are two SQL scripts provided inside resources folder of WmDocumentumAdapter. Execute the *01_init_config.sql* script to create and initialize the configuration table and sample configuration.

The table must be created in schema of Oracle user that will be used for this data source. The table structure is as follows:

Column Name	Data Type	Nullable	Data Default
OBJECT_TYPE	VARCHAR2(64 BYTE)	No	(null)
SYNC_EVENTS	VARCHAR2(256 BYTE)	No	(null)

This configuration does not support object inheritance, so every object type which needs to provide notifications to the Integration Server must be explicitly added to this table. Sync events column holds the comma separated list of audit trail events for the type which should be passed as notification events to the Integration Server.

As a sysdba execute *02_grant_notification_priv.sql* to give the appropriate privileges to the user account which will be used to connect to the database backend.

3.3.2.1.1 *Deploying DB Listener Solution to TomEE*

DB listener application is shipped as a war file. Before deploying it to TomEE application server there are several parameters to be set. Make sure to update the parameters from this table in web.xml to reflect your environment.

Parameter Name	Parameter Value
jdbc_url	JDBC URL for connecting to the Oracle database.
jdbc_user	Username for the account used to make connection.
jdbc_pass	Password for the account used to make connection.
dctm_repository	Documentum repository name.

After updating the web.xml use the manager app or simply place the war file into webapps folder and start the server.

Make sure that the port on which the application server is running is not blocked by the firewall.

Deploying the solution on TomEE is recommended for older Documentum installations (version 6.x) as they don't come with a compatible application server with all the necessary features for using WebSockets to send notifications. If your Documentum system is newer (version 7.x) and uses an Oracle database for keeping metadata, you can deploy the db listener application on Documentum's Method Server's JBoss instance.

3.3.2.1.2 *Deploying DB Listener Solution to JBoss AS 7.1.1*

Below are the steps for deploying the Documentum72OracleAuditMonitor.war on JBoss 7.1.1.

1. Add db connection module in JBoss

    a. Inside the Method Server JBoss locate modules folder (i.e. `c:\Documentum\jboss7.1.1\modules`)

b. Create the following path underneath:
com\oracle\ojdbc6\main

c. Inside the main folder place the ojdbc6.jar (you can get it from Oracle's official site).

d. Inside main folder create module.xml with following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.oracle.ojdbc6">        <re-
sources>
              <resource-root path="ojdbc6.jar"/>
      </resources>
      <dependencies>
              <module name="javax.api"/>
              <module name="javax.transaction.api"/>
      </dependencies>
</module>
```

2. Add a datasource and the corresponding driver

a. Locate the standalone.xml of the JMS (i.e. c:\Documentum\jboss7.1.1\server\DctmServer_MethodServer\configuration\standalone.xml)

b. Add a datasource node with a connection URL and username and password for connection

```
<datasource jndi-name="java:/dctm_db_listener" pool-name="dctm_db_listener" ena-
bled="true" use-java-context="true">
       <connection-url>jdbc:oracle:thin:@localhost:1521:orcl</connection-url>
       <driver>com.oracle</driver>
              <pool>
                     <min-pool-size>3</min-pool-size>
                     <max-pool-size>5</max-pool-size>
              </pool>
              <security>
                     <user-name>rep01</user-name>
                     <password>password</password>
              </security>
              <validation>
                     <exception-sorter class-
name="org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter"/>
              </validation>
              <timeout>
                     <blocking-timeout-millis>5000</blocking-timeout-millis>
                     <idle-timeout-minutes>5</idle-timeout-minutes>
              </timeout>
</datasource>
```

c. Add driver node specifying the oracle JDBc driver

```
<driver name="com.oracle" module="com.oracle.ojdbc6">
      <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
</driver>
```

Be careful here, module attribute must match the name of the module specified in step 1.d

3. Update EJB subsystem, under <subsystem xmlns="urn:jboss:domain:ejb3:1.2"> node add the following after `session-bean` tag:

```
<mdb>
      <resource-adapter-ref resource-adapter-name="hornetq-ra"/>
      <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
</mdb>
```

4. Update web subsystem, set native attribute to true

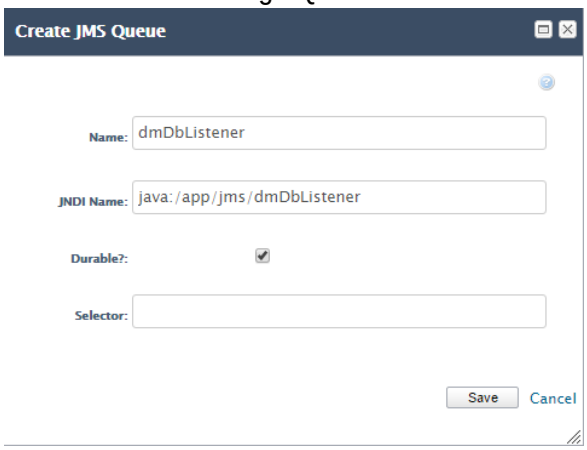
```
<subsystem xmlns="urn:jboss:domain:web:1.1" default-virtual-server="default-host" native="true">
```

5. Add message queue using JBoss admin console

   a. Log into the JBoss admin console

   b. Navigate to Profile > Messaging > Messaging Provider > JMS Destinations

   c. Create a JMS Message Queue



6. Before deploying the war file itself, update the `web.xml` to match your environment
   - Set the `web_socket_port` context parameter to the port number which will be used to accept WebSocket connections (must be a free port, not used by any other process)
   - Set the `dctm_repository` servlet parameter to match the name of Documentum repository

7. Place the war file in deployments folder of Documentum's JBoss instance and start the server (i.e. c:\Documentum\jboss7.1.1\server\DctmServer_MethodServer\deployments\)

3.3.2.1.3 *Deploying DB Listener Solution to WildFly 9.0.1*

Below are the steps for deploying the Documentum72OracleAuditMonitor.war on WildFly 9.0.1

8. Add db connection module in WildFly

   a. Inside the Method Server WildFly locate modules folder (i.e. `c:\Documentum\shared\wildfly9.0.1\modules`)

b. Create the following path underneath:
/system/layers/base/com/oracle/ojdbc6/main

c. Inside the main folder place the ojdbc6.jar (you can get it from Oracle's official site).

d. Inside main folder create module.xml with following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="com.oracle.ojdbc6">        <re-
sources>
            <resource-root path="ojdbc6.jar"/>
      </resources>
      <dependencies>
            <module name="javax.api"/>
            <module name="javax.transaction.api"/>
      </dependencies>
</module>
```

9. Add a datasource and the corresponding driver

a. Locate the standalone.xml of the JMS (i.e. `c:\Documentum\shared\wildfly9.0.1\server\DctmServer_MethodServer\configuration\standalone.xml`)

b. Add a datasource node with a connection URL and username and password for connection

```
<datasource jndi-name="java:/dctm_db_listener" pool-name="dctm_db_listener" ena-
bled="true" use-java-context="true">
      <connection-url>jdbc:oracle:thin:@localhost:1521:orcl</connection-url>
      <driver>com.oracle</driver>
            <pool>
                  <min-pool-size>3</min-pool-size>
                  <max-pool-size>5</max-pool-size>
            </pool>
            <security>
                  <user-name>rep01</user-name>
                  <password>password</password>
            </security>
            <validation>
                  <exception-sorter class-
name="org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter"/>
            </validation>
            <timeout>
                  <blocking-timeout-millis>5000</blocking-timeout-millis>
                  <idle-timeout-minutes>5</idle-timeout-minutes>
            </timeout>
</datasource>
```

c. Add driver node specifying the oracle JDBc driver

```
<driver name="com.oracle" module="com.oracle.ojdbc6">
      <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
</driver>
```

Be careful here, module attribute must match the name of the module specified in step 1.d

10. Update EJB subsystem, under <subsystem xmlns="urn:jboss:domain:ejb3:3.0"> node add the following after `session-bean` tag:

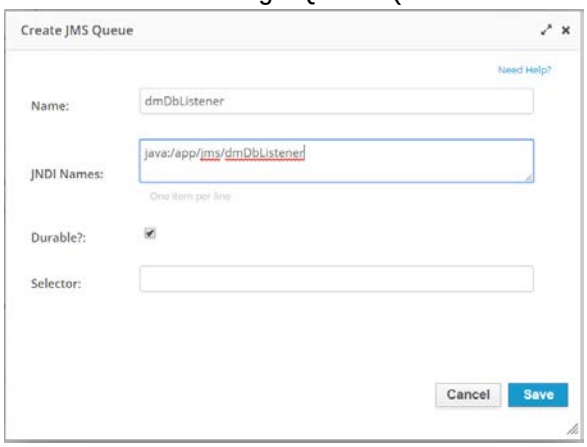
```
<mdb>
      <resource-adapter-ref resource-adapter-name="hornetq-ra"/>
      <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
</mdb>
```

11. Add message queue using WildFly admin console

    a. Log into the WildFly admin console

    b. Navigate to Configuration > Subsystem > Messaging > Messaging Provider > default
    c. Click on "destinations"

    d. Create a JMS Message Queue (click on "Add" button)



12. Before deploying the war file itself, update the `web.xml` to match your environment
    - Set the `dctm_repository` servlet parameter to match the name of Documentum repository

13. Place Documentum73OracleAuditMonitor.war file in deployments folder of Documentum's WildFly instance and start the server (i.e. `c:\Documentum\shared\wildfly9.0.1\server\DctmServer_MethodServer\deployments\`) or use WildFly admin console (Deployments > Add > Upload a new deployment, select Documentum73OracleAuditMonitor.war, keep default values for "Name" and "Runtime Name" fields, check "Enable" checkbox and click "Finish" button)

Important Note: Provided war files for TomEE, JBoss and WildFly application servers are not identical, the implementation inside it is different. It is not enough just to change the web.xml settings, correct war file must be deployed.

Required war (**Documentum72OracleAuditMonitor.war, Documentum73OracleAuditMonitor.war** and **DocumentumOracleAuditMonitor.war**) files can be found under **resources** folder inside WmDocumentumAdapter package.

3.3.2.1.4 *WebSocket URL format*

Depending on the application server used for deploying the listener application, different Notification WebSocket Url should be provided in the adapter connection settings.

When deploying the application on TomEE the Url form looks like this:

```
ws://<hostname>:<websocket_port>/DocumentumOracleAuditMonitor/notifications
```

For Jboss 7.1.1 the Url is sligthly different, as it does not contain the war file name:

```
ws://<hostname>:<websocket_port>/notifications
```

When deploying the application on WildFly 9.0.1 the Url form looks like this:

```
ws://<hostname>:<websocket_port>/Documentum73OracleAuditMonitor/notifications
```

3.3.2.2 *MSSQL based DB Listener Solution Installation*

This approach is applicable only for Documentum server that is installed on top of Microsoft SQL server. Required packages (AuditMonitorMSSQLServiceRelease.zip and SAG Notification Service.zip) can be found in **resources** folder inside WmDocumentumAdapter package.

This solution is working in following way:

1. One windows service (*MonitorAuditService*) is monitoring *dm_audittrail_s* table in MSSQL DB. When new audittrail is created, windows service will check if this event is configured for propagation to MSMQ private queue. If so, new MSMQ item (in specific queue) will be created with all necessary data (see table below in section 3.3.2.3). Service installation directory will contain *AuditMonitor.exe.config* XML file where we can configure the following:

- Logging
- Connection string to MSSQL DB with user authentication
- Repository name (will be just passed unchanged in Notification Message)
- Repository type (will be just passed unchanged in Notification Message)
- MSMQ name (this MSMQ will be used in communication between two services)
- Per type configuration which events should be subscribed for notification.

2. Another windows service (*SAG Notification Service*) will monitor MSMQ items of specific queue. When new item is created, this queue item info will be propagated to SAG Documentum adapter running on Integration server, configured to listen specific WebSocket URL and port. Service installation directory will contain *SAG Notification Service.exe.config* XML file where we can configure the following:

- Logging
- Web Socket server URL to be used (same URL must be used for Notification WebSocket Url parameter in wmDocumentumAdapter connection configuration)
- MSMQ name (needs to be with same name as in previous service XML).

3. Documentum adapter will receive notifications via WebSocket connection and process them accordingly.

Both services should be installed in the same Windows machine, which could be the machine where Documentum Content server is installed or some other Windows machine that is able to connect to MSSQL via connection string.

3.3.2.2.1 *MSSQL server changes*

From Documentum Server Manager stop docbroker and repositories.

Login into Microsoft SQL server Management studio to Documentum DB as sysdba.

Under Databases node, locale database containing dbo.dm_audittrail_s table.

Alter *SQL SERVER SCRIPT.sql* file and change *DM_rep01_docbase* string with name of database found above.

Execute script.

From Documentum Server Manager start docbroker and repositories.

Note: This script is doing following:

- *Creating user with its own schema and sufficient privileges in proper database to perform table monitoring (with SqlTableDependency)*
- *Enabling service broker*

3.3.2.2.2 *MSMQ (Message Queueing)*

MSMQ must be enabled on machines where services are activated. This link explains how to activate it:

https://msdn.microsoft.com/en-us/library/aa967729%28v=vs.110%29.aspx

After activating the MSMQ, new private queue should be created. This is done in *Computer Management/Services and Applications/Message Queueing/Private Queues*.

The name of the new queue must be *sagnotifier* (the full name is: *.\private$\sagnotifier*).

In the properties pane of the 'sagnotifier' on the Security tab, give full control to some user that will be later used in service installation.

3.3.2.2.3 *Installation and configuring of MonitorAuditService*

This service will perform monitoring of auditrail database table. When new audit trail is created, it will be analyzed and if matches a configured type, new MSMQ item will be created.

Installation files are located in *„AuditMonitorMSSQLServiceRelease.zip"* file. Extract files to some location (*C:\[ServiceLocation]*), and start installation with the following command (run command prompt with admin privileges):

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\InstallUtil
"C:\[ServiceLocation]\AuditMonitor.exe"
```

Note: Close services window while performing this step of the setup.

During the installation, the user will be prompted with the form for entering the details about the identity that will be used to run the service. The suggested identity is the same that is set with full control on MSMQ created private queue.

After the installation go to Service, find new installed service and in Properties / Log On check if not already checked *Allow service to interact with desktop.*

After the installation, the service must be configured, by editing the .config file: *C:\[ServiceLocation]\ AuditMonitor.exe.config*

Configuration holds following properties:

- Logging configuration (level, output file, size…)

- Connection-string – holding connection string to MSSQL server (database name we changed in previous step, security, authentication – user defined here must be same as the one we created with script against MSSQL server).

  Note: Connection string can be obtained from Microsoft SQL server Management studio from database properties (database we altered in first step). It is important to use Integrated Security=false; otherwise, passed authentication will be overridden by local user.
- Repository – holding repository name (this string will be passed unchanged in Notification Message)
- Repository type – holding repository type. In this case, just string "documentum". It will be passed unchanged in Notification Message
- Msmq name – Name of the private MSMQ (the one we created in previous steps).
- Per type configuration of monitored events. We add as key documentum type prefixed with "track_" and as value we define comma separated list of events for which we want to create notification. Type hierarchy is not applied here, so, if we must specify exact documentum type and its events to be tracked.

Note: This configuration will filter audits happening in Documentum system. To configure audit events being tracked in Documentum, use Documentum Administrator / Audit management. After that, this configuration will just narrow which audits are being passed to MSMQ and later to SAG Documentum Adapter listener

Example of configuration:

```
<add key="connection-string" value="Data Source=TRANSNET;Initial
Catalog=DM_rep01_docbase;Persist Security Info=false;Integrated Security=false;User
Id=startUser;Password=startUser"/>
<add key="repository" value="rep01"/>
<add key="repository-type" value="documentum"/>
<add key="msmq-name" value=".\Private$\sagnotifier"/>
<add key="track_dm_document"
value="dm_save,dm_destroy,dm_checkin,dm_checkout,dm_lock,dm_setfile,dm_unlock"/>
<add key="track_dm_folder" value="dm_save,dm_destroy"/>
```

Note: In service directory, new Logs folder will be created where service will log its activity. Default configuration for log file is:

```
<file value="Logs\notificationservice.log"/>
```

3.3.2.2.4 *Installation and configuring of WebSocket Service*

The component that sends the actual notification messages to the Integration Server is the windows service **SAG Notification Service**. It should be installed on the machine where Monitor Audit Service is installed.

The service internally runs the WebSocket server, that accepts the connection(s) from the Integration Server Documentum adapter (which acts as the WebSocket client) and sends the notification messages that come from the MSMQ.

Installation files for the service are located in „*SAG Notification Service.zip*“ file. All the files should be extracted to some location (*C:\[ServiceLocation]*), and installation should be started with the following command (run command prompt with admin privileges):

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\InstallUtil
"C:\[ServiceLocation]\SAG Notification Service.exe"
```

Note: Close services window while performing this step of the setup.

During the installation, the user will be prompted with the form for entering the details about the identity that will be used to run the service. The suggested identity is the same that is used in installation of previous windows service that also have full control of MSMQ created private queue.

After the installation go to Service, find new installed service and in Properties / Log On check if not already checked *Allow service to interact with desktop.*

After the installation, the service should be configured, by editing the .config file: *C:\[ServiceLocation]\SAG Notification Service.exe.config*

The config file should contain name of MSMQ private queue

<add key="sag-msmq-servers" value=".\private$\sagnotifier"/>

Also, the url and the port that will be used by the web socket server instance must be configured. It is important that firewall must be configured so it doesn't block the traffic on the configured port.

<add key="sag-websocket-server-url" value="http://localhost:4649"/>

This will reflect WebSocket URL: ws://<hostname>:<websocket_port>/SAGNotifier

For example: ws://localhost:4649/SAGNotifier

Note: Configuration changes (if any) on both services will be visible only when services are restarted!

3.3.2.3 *Message format*

The format of the Notification Message, with the explanation of every field is listed in the following table:

Field name	Field description
MessageId	Object ID of audit trail object.
RepositoryId	Repository Name.
HostName	Host machine.
ObjectName	The name of the document.
ObjectId	The Object Id of the document.
ObjectPath	Currently not used, “n/a” will be passed.

RepositoryType	String "documentum".
Username	The username of the user that committed the operation.
EventDate	The date and time of the operation.
ActionType	The type of the committed operation (action).
ItemType	The name of the object type.
BeforeProperties	The map of the properties that were changed during the operation and their values before the operation.
AfterProperties	The map of the properties that were changed during the operation and their values after the operation.

3.4 Uninstalling Documentum ECM Adapter

To uninstall Documentum ECM Adapter

1. Shut down the host Integration Server. You do not need to shut down any other webMethods products or applications that are running on your machine.
2. Start Software AG Uninstaller, selecting the webMethods installation directory that contains the host Integration Server.
3. In the product selection list, select **Adapters > webMethods Adapter for Documentum**. You can also choose to uninstall documentation.
4. After Uninstaller completes, restart the host Integration Server.

Uninstaller removes all Documentum ECM Adapter-related files that were installed. However, Uninstaller does not delete files created after you installed the adapter (for example, usercreated or configuration files), nor does it delete the adapter directory structure.

You can go to the *Integration Server_directory*\packages directory and *Integration Server_directory**instances**default**packages* directory.

Delete the WmDocumentumAdapter directory.

4 Package Management

4.1 Overview

The following sections describe how to set up and manage your Adapter for Documentum ECM packages, set up Access Control Lists (ACLs), and use the adapter in a clustered environment.

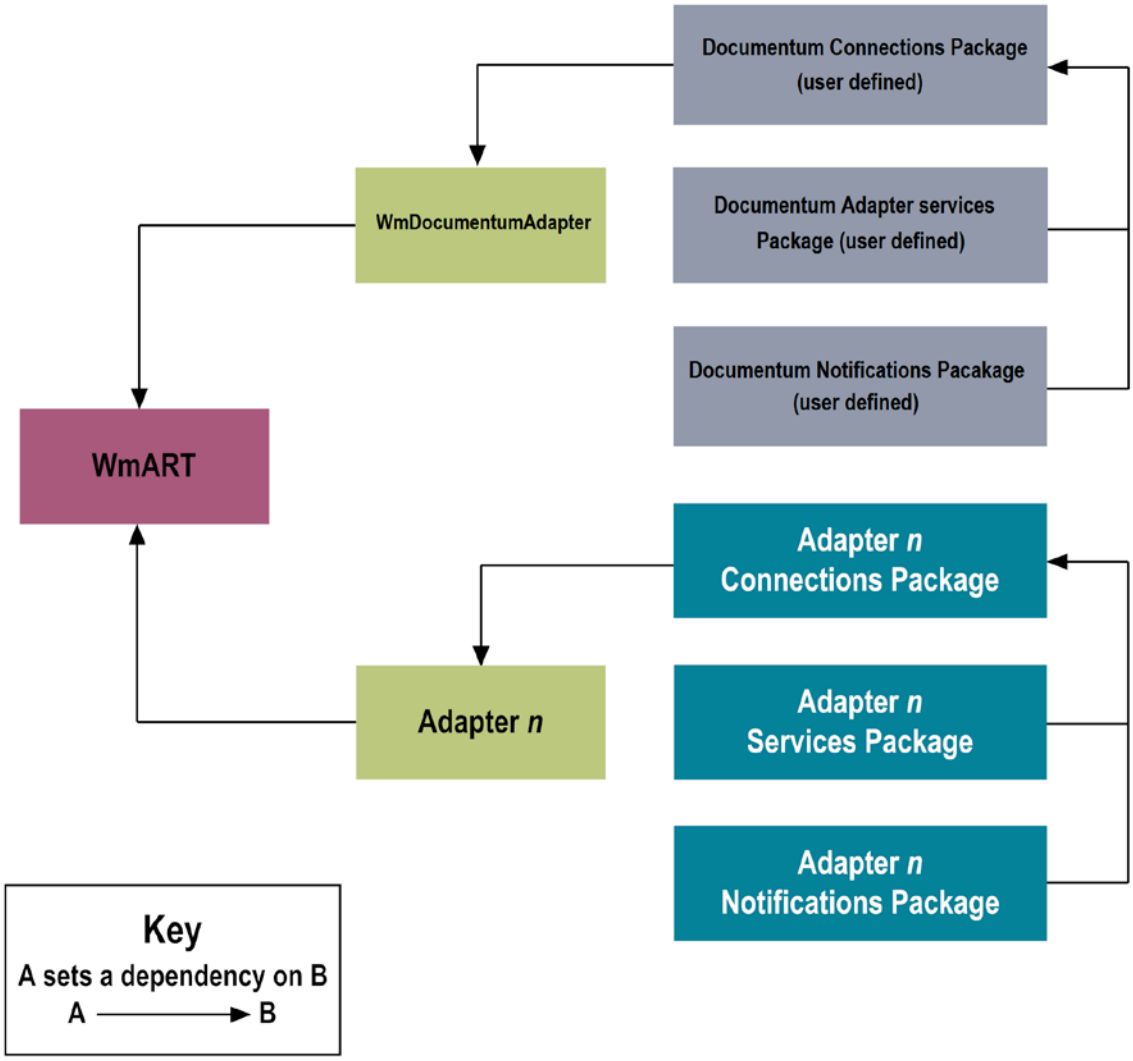
4.2 Adapter for Documentum Package Management

Adapter for Documentum ECM is provided as a package called WmDocumentumAdapter.

You manage the WmDocumentumAdapter package as you would manage any package on webMethods Integration Server.

When you create connections, adapter services, and adapter notifications, define them in user-defined packages rather than in the WmDocumentumAdapter package. Doing so will allow you to manage the package more easily.

As you create user-defined packages in which to store connections, adapter services, and adapter notifications, use the package management functionality provided in Software AG Designer and set the user-defined packages to have a dependency on the WmDocumentumAdapter package. That way, when the WmDocumentumAdapter package loads or reloads, the user-defined packages load automatically. See the following diagram:



Package management tasks include:

- Setting package dependencies (see "Package Dependency Requirements and Guidelines")
- "Enabling Packages"
- "Importing and Exporting Packages"
- "Group Access Control"

4.2.1 Package Dependency Requirements and Guidelines

This section contains a list of dependency requirements and guidelines for user-defined packages. For instructions for setting package dependencies, see the *Designer Service Development Help* for your release.

- A user-defined package must have a dependency on its associated adapter package, WmDocumentumAdapter. (The WmDocumentumAdapter package has a dependency on the WmART package.)
- Package dependencies ensure that at startup the Integration Server automatically loads or reloads all packages in the proper order: the WmART package first, the adapter package next, and

the user-defined packages last. The WmART package is automatically installed when you install Integration Server. You should not need to manually reload the WmART package.

- If the connections and adapter services of an adapter are defined in different packages, then:
  - A package that contains the connections must have a dependency on the adapter package.
  - Packages that contain adapter services must have a dependency on their associated connection package.
- Keep connections for different adapters in separate packages so that you do not create interdependencies between adapters. If a package contains connections for two different adapters, and you reload one of the adapter packages, the connections for both adapters will reload automatically.
- Integration Server will not allow you to enable a package if it has a dependency on another package that is disabled. That is, before you can enable your package, you must enable all packages on which your package depends. For information about enabling packages, see "Enabling Packages".
- Integration Server will allow you to disable a package even if another package that is enabled has a dependency on it. Therefore, you must manually disable any userdefined packages that have a dependency on the adapter package before you disable the adapter package. For information about disabling packages, see "Disabling Packages".
- You can name connections, adapter services, and notifications the same name provided that they are in different folders and packages.

4.2.2 Enabling Packages

All packages are automatically enabled by default. Use the following procedure when you want to enable a package that was previously disabled.

To enable a package

1. Open Integration Server Administrator if it is not already open.
2. In the **Packages** menu of the navigation area, click **Management**.
3. Click **No** in the **Enabled** column. The server displays a ✔ and **Yes** in the **Enabled** column.

Note: Enabling an adapter package will not cause its associated user-defined packages to be reloaded. For information about reloading packages, see *webMethods Service Development Help*.

Important! Important! Before you manually enable a user-defined package, you must first enable its associated adapter package (WmDocumentumAdapter).

4.2.3 Disabling Packages

When you want to temporarily prohibit access to the elements in a package, disable the package. When you disable a package, the server unloads all of its elements from memory. Disabling a package prevents Integration Server from loading that package at startup.

Important! If your adapter has multiple user-defined packages, and you want to disable some of them, disable the adapter package first (WmDocumentumAdapter). Otherwise, errors will be issued when you try to access the remaining enabled user-defined packages.

To disable a package

1. Open Integration Server Administrator if it is not already open.
2. In the **Packages** menu of the navigation area, click **Management**.
3. Click **Yes** in the **Enabled** column for the package that you want to disable. The server issues a prompt to verify that you want to disable the package. Click **OK** to disable the package. When the package is disabled, the server displays **No** in the **Enabled** column.

A disabled adapter will:

- Remain disabled until you explicitly enable it using Integration Server Administrator.
- Not be listed in Designer.

4.2.4 Importing and Exporting Packages

You import and export packages using Designer. Exporting allows you to export the package to a .zip file and save it to your hard drive. The .zip file can then be imported for use by another package.

Important! Do not rename packages you export; the rename function is comparable to moving a package, and when you import the renamed package, you lose any triggers, connections, and notifications associated with this package.

For details about importing and exporting packages, see *webMethods Service Development Help*.

4.3 Group Access Control

To control which groups have access to which adapter services, use access control lists (ACLs). For example, you can use ACLs to prevent one development group from inadvertently updating the work of another group, or to allow or deny access to services that are restricted to one group but not to others.

For information about assigning and managing ACLs, see *webMethods Service Development Help*.

5 Adapter for Documentum ECM Connections

5.1 Overview

This chapter describes how to configure and manage Documentum ECM Connections. For more information about how adapter connections work, *see* Adapter Connections.

5.2 Before Configuring or Managing Adapter Connections

Perform the following steps before configuring or managing adapter connections.

To prepare to configure or manage adapter connections

1. Install webMethods Integration Server and Adapter for Documentum on the same machine.
2. Install Documentum Foundation Service API in the Adapter for Documentum ECM package.
3. Install Documentation Foundation Services on Documentum System.
4. Install Documentum Foundation Service API Libraries on Integration Server

5.3 Configuring Adapter for Documentum Connections

You can configure an adapter's connections from the Integration Server Administrator and Designer.

Perform the following steps to configure the adapter's connections in Integration Ser-vice Administrator:

1. In the Adapters menu in Integration Server Administrator's navigation area, click **Documentum Adapter**.
2. The Connections screen displays all connections that have been created. If no connections have been created for the current package, **No connections found** is displayed.
3. Click **Configure New Connection**
4. The Configure Connection Type is displayed. Enter the parameters that are required for this connection. See "Run-Time Behavior of Connection Pools" for information of the connection settings.
5. Once the information for the new connection is complete, click **Save Connection.**
6. The new connection is saved and can be used when creating a new service from the service template. See "Adapter Services"

5.4 Connection Properties

The section above shows you how to create a new connection to your Documentum system. This section describes the parameters for the defining of a connection.

Adapter Properties

Attribute	(Example) Value	Description
Package	DocumentumAdapterSamples	The Integration Server package you want to use.
Folder Name	connections	The folder within the IS package.
Connection Name	sample	The name of the connection acting as default.
Service Context Root Url	http://documentum.local:8080/emc-dfs/services	The DFS service context root URL this is where all the DFS services are hosted via HTTP/HTTPS.
Repository Name	rep01	The name of the repository in Documentum server.
Username	username	The username for connecting.
Password	secret	The user's password.
Retype Password		The user's password retyped.
webSocketUrl	ws://<documentum server host name>:<port>/<notifications context>	The address of the web socket url that sends the real time notifications about updated content in Documentum repo. Depending on Documentum version and configuration, see WebSocket URL format.

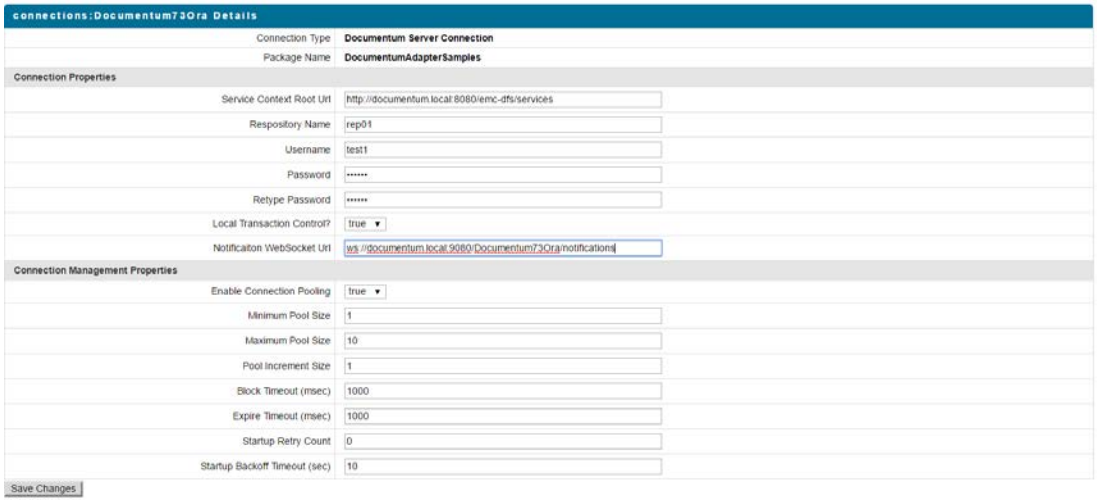
Connection Management Properties

Field	Description/Action
Enable Connection Pooling	Enables the connection to use connection pooling. For more information about connection pooling see the Integration Server connection pool documentation. Note: If you plan to enable connection pooling in a clustered environment, consider the connection pool size.
Minimum Pool Size	If connection pooling is enabled, this field specifies the number of connections to create when the connection is enabled. The adapter will keep open the number of connections you configure here regardless of whether these connections become idle.
Maximum Pool Size	If connection pooling is enabled, this field specifies the maximum number of connections that can exist at one time in the connection pool.
Pool Increment Size	If connection pooling is enabled, this field specifies the number of connections by which the pool will be incremented if connections are needed, up to the maximum pool size.
Block Timeout	If connection pooling is enabled, this field specifies the number of milliseconds that the Integration Server will wait to obtain a connection with Documentum before it times out and returns an error. For example, you have a pool with Maximum Pool Size of 20. If you receive 30 simultaneous requests for a connection, 10 requests will be waiting for a connection from the pool. If you set the Block Timeout to 5000, the 10 requests will wait for a connection for 5 seconds before they time out and return an error. If the services using the connections require 10 seconds to complete and return connections to the pool, the pending requests will fail and return an error message stating that no connections are available. If you set the Block Timeout value too high, you may encounter problems during error conditions. If a request contains errors that delay the response, other requests will not be sent. This setting should be tuned in conjunction with the Maximum Pool Size to accommodate such bursts in processing.
Expire Timeout	If connection pooling is enabled, this field specifies the number of milliseconds that an inactive connection can remain in the pool before it

Field	Description/Action
	is closed and removed from the pool. The connection pool will remove inactive connections until the number of connections in the pool is equal to the Minimum Pool Size. The inactivity timer for a connection is reset when the connection is used by the adapter. If you set the Expire Timeout value too high, you may have a number of unused inactive connections in the pool. This consumes local memory and a connection on your backend resource. This could have an adverse effect if your resource has a limited number of connections. If you set the Expire Timeout value too low, performance could degrade because of the increased activity of creating and closing connections. This setting should be tuned in conjunction with the Minimum Pool Size to avoid excessive opening/closing of connections during normal processing.
Startup Retry Count	The number of times that the system should attempt to initialize the connection pool at startup if the initial attempt fails. The default is 0.
Startup Backoff Timeout	The number of seconds that the system should wait between attempts to initialize the connection pool.

5.5 Viewing Adapter Connection Parameters

You can view a connection's parameters from Integration Server Administrator and Designer.



5.5.1 Using Integration Server Administrator to View Adapter Connection Parameters

Perform the following steps to view adapter connection parameters in Integration Server Administrator.

To view the parameters for a connection using Integration Server Administrator

1 In the **Adapters** menu in Integration Server Administrator's navigation area, click **Documentum Adapter**.

When using the adapter with Integration Server 8.0 and later, you can sort and filter the list of connections that appears on the Connections screen.

- To sort information on the Connections screen, click the **Up** and **Down** arrows at the top of the column you want to sort.
- To filter the list of connections:

  i On the Connections screen, click **Filter Connections**.

  ii Type the criterion by which you want to filter into the **Filter criteria** box. Filtering is based on the node name, not the connection alias. To locate all connections containing specific alphanumeric characters, use asterisks (*) as wildcards. For example, if you want to display all connections containing the string "abc", type `*abc*` in the **Filter criteria** box.

  iii Click **Submit**. The Connections screen displays the connections that match the filter criteria.

  iv To re-display all connections, click **Show All Connections**.

  The Connections screen appears, listing all the current connections. You can control the number of connections that are displayed on this screen.

2 On the Connections screen, click the icon for the connection you want to see.

The View Connection screen displays the parameters for the connection. For descriptions of the connection parameters, see "Configuring Adapter for Documentum Connections".

3 Click **Return to Documentum Adapter connections** to return to the main connections screen.

5.5.2 Using Designer to View Adapter Connection Parameters

Perform the following steps to view adapter connection parameters in Designer.

To view the parameters for a connection using Designer

1 From the Designer navigation area, open the package and folder in which the connection is located.

2 Double-click the connection you want to view.

The parameters for the connection appear on the **Connection Information** tab. For descriptions of the connection parameters, see "Configuring Adapter for Documentum Connections".

5.6 Editing Adapter Connections

If the login information for a database changes, or if you want to redefine parameters that a connection uses when connecting to a database, you can update a connection's parameters using Integration Server Administrator.

To edit a connection

1. In the **Adapters** menu in Integration Server Administrator's navigation area, click **Documentum Adapter**.
2. Make sure that the connection is disabled before editing it. For instructions, see "Disabling Adapter Connections".
3. On the Connections screen, click the  icon for the connection you want to edit.

   The Edit Connection screen displays the current parameters for the connection. Update the connection's parameters by typing or selecting the values you want to specify.

   For descriptions of the connection parameters, see "Configuring Adapter for Documentum Connections".
4. Click **Save Changes** to save the connection and return to the Connections screen.

5.7 Copying Adapter Connections

You can copy an existing Adapter for Documentum connection to configure a new connection with the same or similar connection properties without having to re-type all of the properties for the connection. You copy adapter connections using Integration Server Administrator.

To copy a connection

1. In the **Adapters** menu in Integration Server Administrator's navigation area, click **Adapter for Documentum**.
2. On the Connections screen, click the  icon for the connection you want to copy.

   The Copy Connection screen displays the current parameters for the connection you want to copy. Name the new connection, specify a package name and folder name, and edit any connection parameters as needed by typing or selecting the values you want to specify.

   Note: When you copy a connection, the new connection does not save the password of the original connection. You must enter and then retype the password before you can save the new connection.

   For descriptions of the connection parameters, see "Configuring Adapter for Documentum Connections".
3. Click **Save Connection Copy** to save the connection and return to the Connections screen.

5.8 Deleting Adapter Connections

If you no longer want to use a particular Adapter for Documentum connection, you can delete it. You delete adapter connections using Integration Server Administrator.

If you delete a Adapter for Documentum connection, the adapter services or notifications that are defined to use the connection will no longer work. However, you can assign a different connection to an adapter service and re-use the service. To do this, use the setAdapterServiceNodeConnection built-in service.

To delete a connection

1. In the **Adapters** menu in the Integration Server Administrator navigation area, click **Adapter for Documentum**.
2. Make sure that the connection is disabled before deleting. To disable the connection, click **Yes** in the **Enabled** column and click **OK** to confirm. The **Enabled** column now shows **No** (Disabled) for the connection.
3. On the Connections screen, click  for the connection you want to delete.

   Integration Server deletes the adapter connection.

5.9 Enabling Adapter Connections

An Adapter for Documentum connection must be enabled before you can configure any adapter service using the connection, or before an adapter service can use the connection at run time. You enable adapter connections using Integration Server Administrator.

Note: When you reload a package that contains enabled connections, the connections will automatically be enabled when the package reloads. If the package contains connections that are disabled, they will remain disabled when the package reloads.

To enable a connection

1. In the **Adapters** menu in the Integration Server Administrator navigation area, click **Adapter for Documentum**.
2. On the Connections screen, click **No** in the **Enabled** column for the connection you want to enable.

   Integration Server Administrator enables the adapter connection and displays a  and **Yes** in the **Enabled** column.

5.10 Disabling Adapter Connections

Note: Adapter for Documentum connections must be disabled before you can edit or delete them. You disable adapter connections using Integration Server Administrator.

To disable a connection

1. In the **Adapters** menu in the Integration Server Administrator navigation area, click **Adapter for Documentum**.
2. On the Connections screen, click **Yes** in the **Enabled** column for the connection you want to disable.

The adapter connection becomes disabled and you see a **No** in the **Enabled** column.

6 Adapter Services

6.1 Overview

The following services are available with the Documentum Adapter.

Service	Description
Create Document	Creates a new document in Documentum
Get Document	Gets metadata of a document in Documentum
Delete Document	Deletes a document in Documentum
Update Document	Updates metadata of a document in Documentum
Search Documents	Queries documents in Documentum according to the parameters defined in the service template.
Create Folder Path	Creates a folder path in Documentum
Delete Folder Path	Deletes a folder path in Documentum

6.2 Create Document

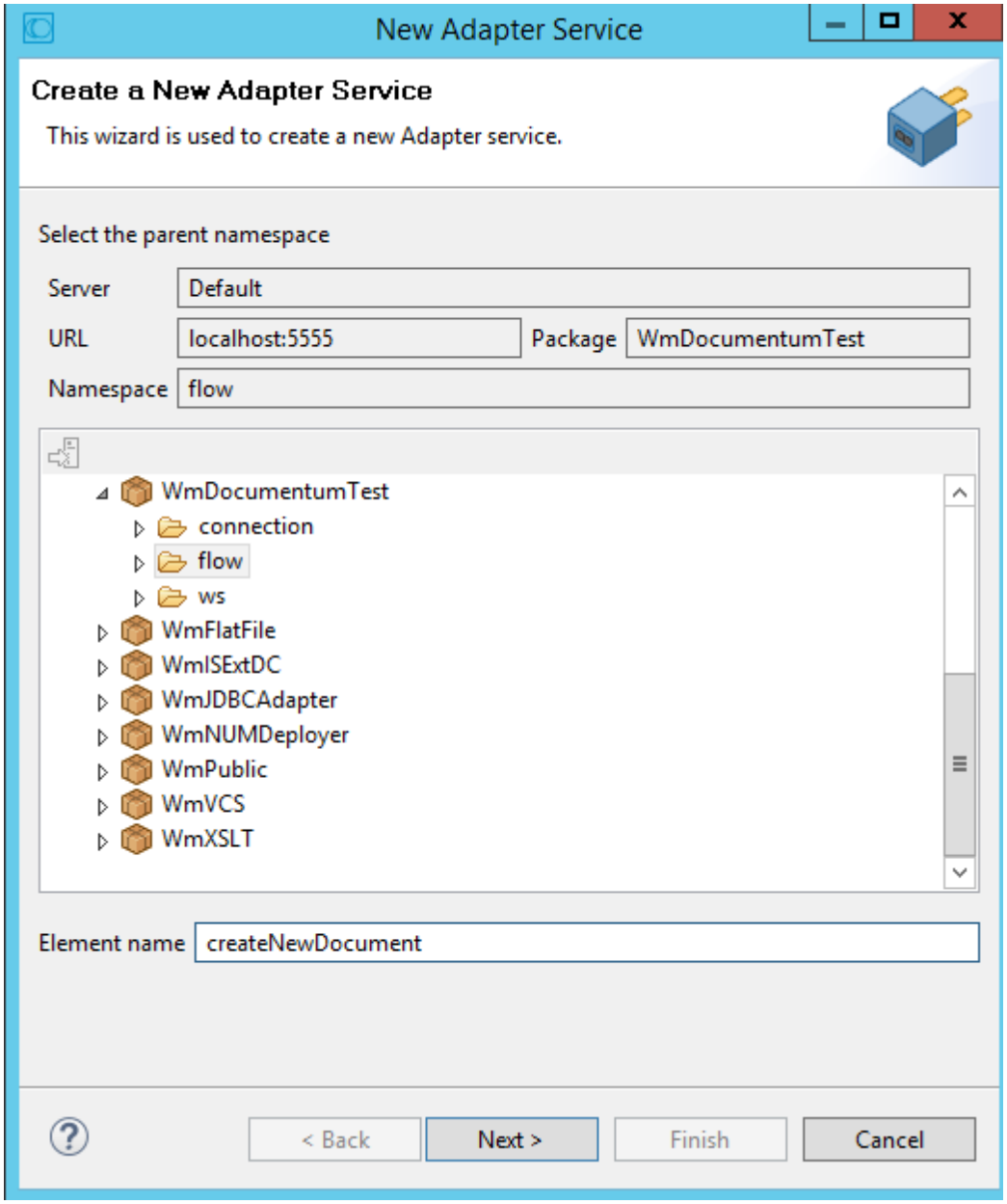
6.2.1 Description

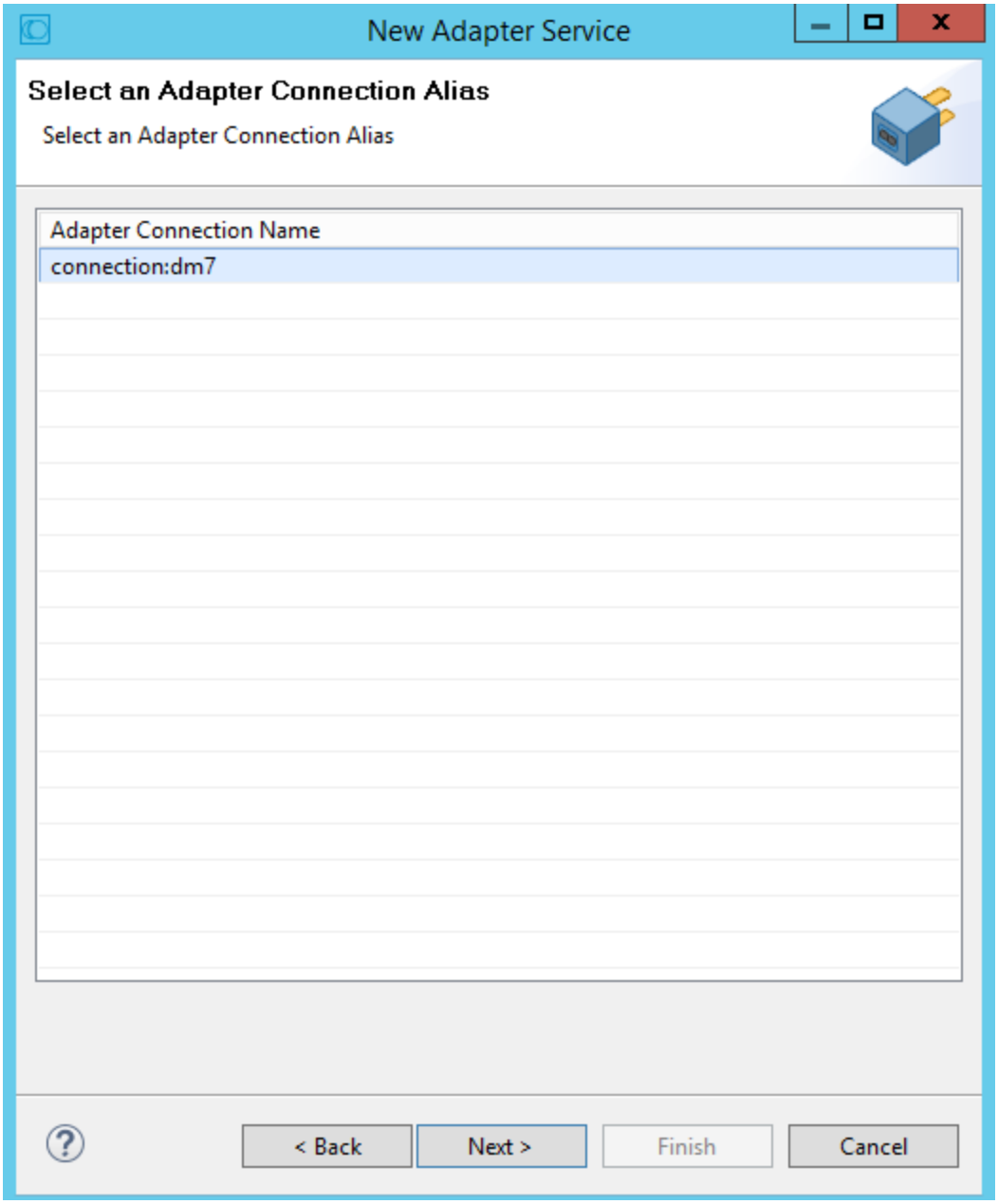
Creates a new document in Documentum providing its attributes (meta-data), a (optional) folder location and a (optional) document URL.

6.2.2 Configure the Service

To create services folder in your DocumentumAdapterSamples sample package

1. Right click->**New Adapter Service**

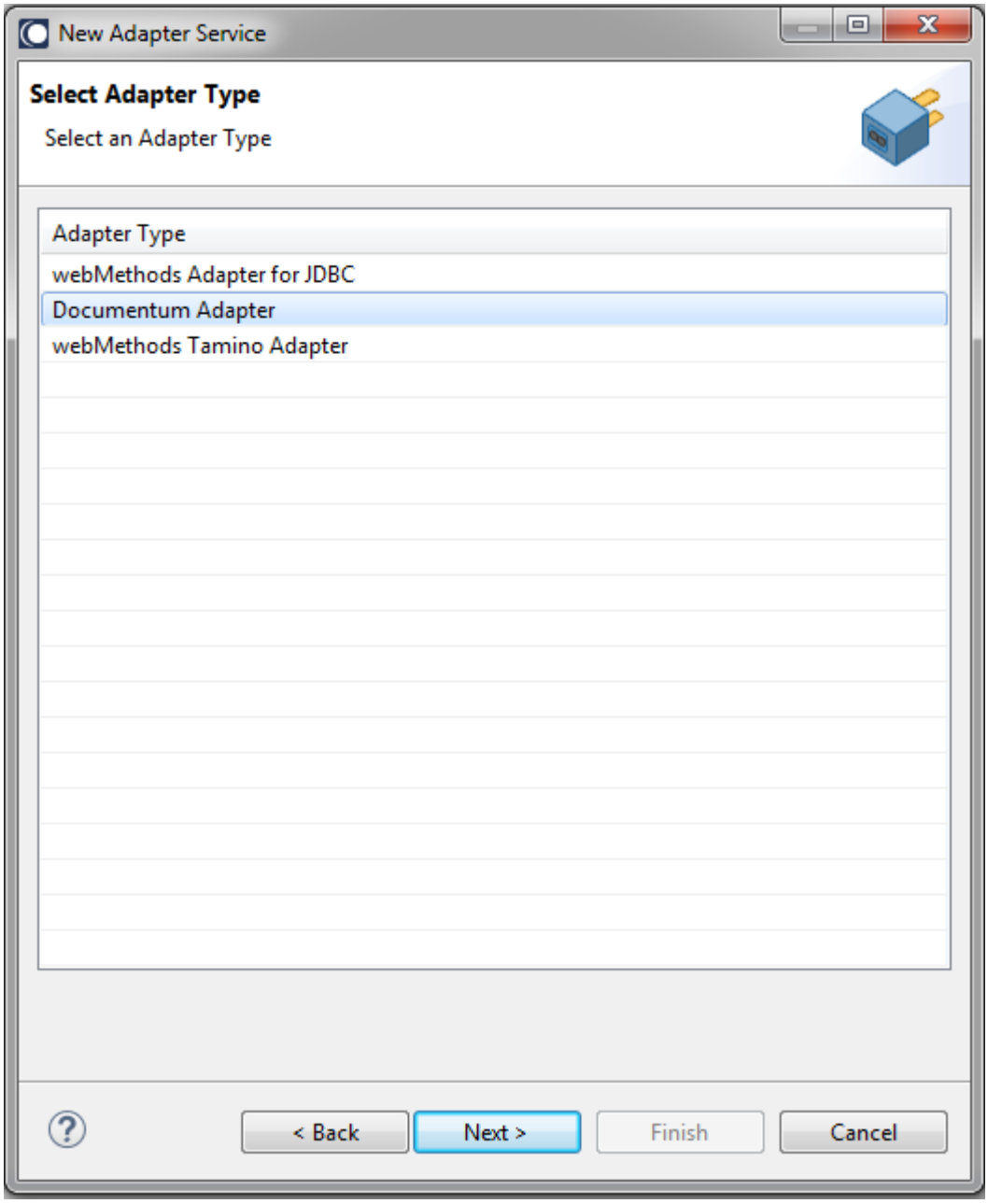

New Adapter Service
Create a New Adapter Service
This wizard is used to create a new Adapter service.
Select the parent namespace
Server
Default
URL
localhost:5555
Package
WmDocumentumTest
Namespace
flow
WmDocumentumTest
connection
flow
ws
WmFlatFile
WmISExtDC
WmJDBCAdapter
WmNUMDeployer
WmPublic
WmVCS
WmXSLT
Element name
createNewDocument
< Back
Next >
Finish
Cancel



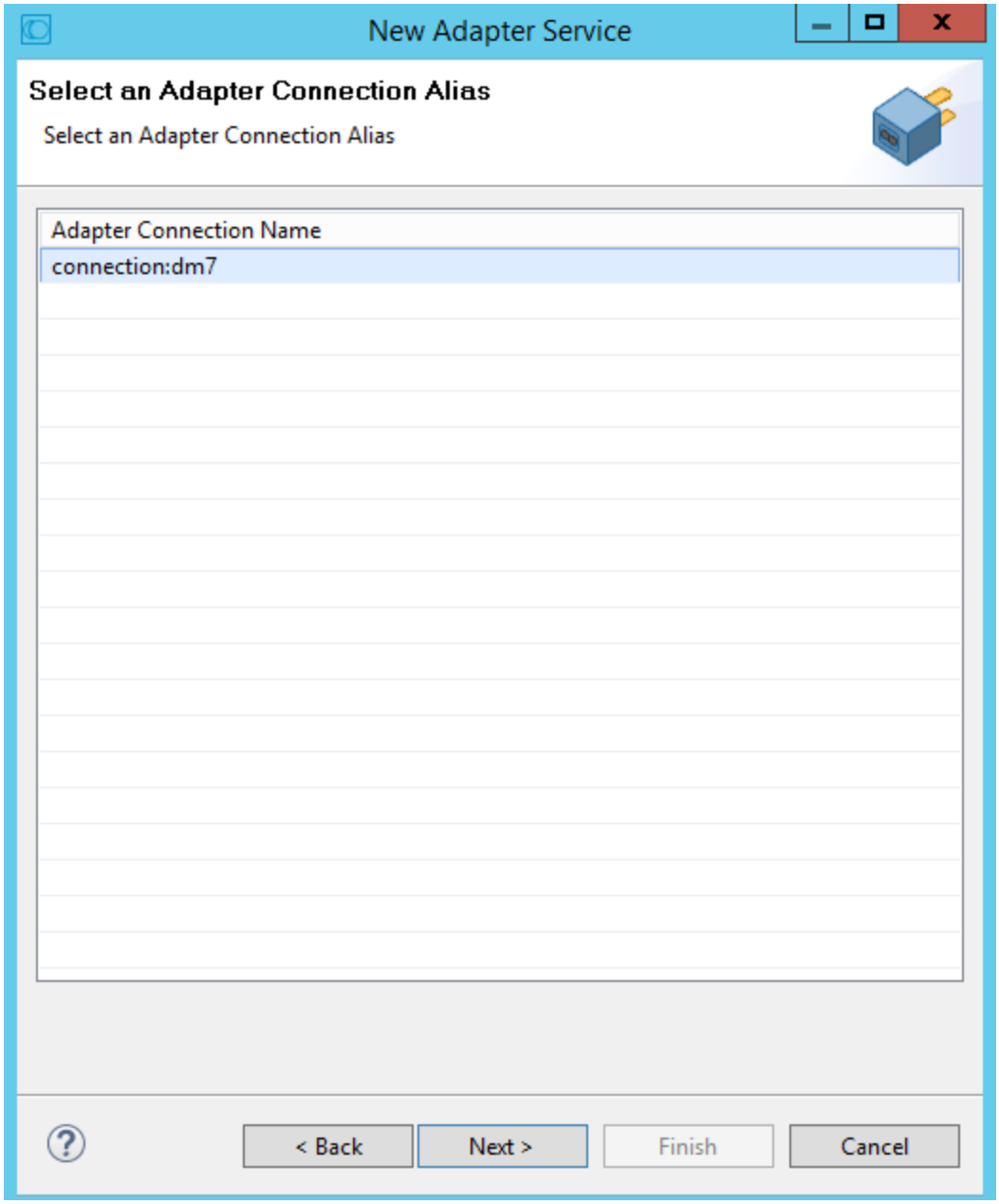
2. Enter the name of the service you want to create in the **Element name** field.

Note: The created service names should be unique within the folder. Example, the name given for the service in **Element name** field can be `ExampleService`.

3. Select your adapter type(Documentum Adapter) and click **Next**.

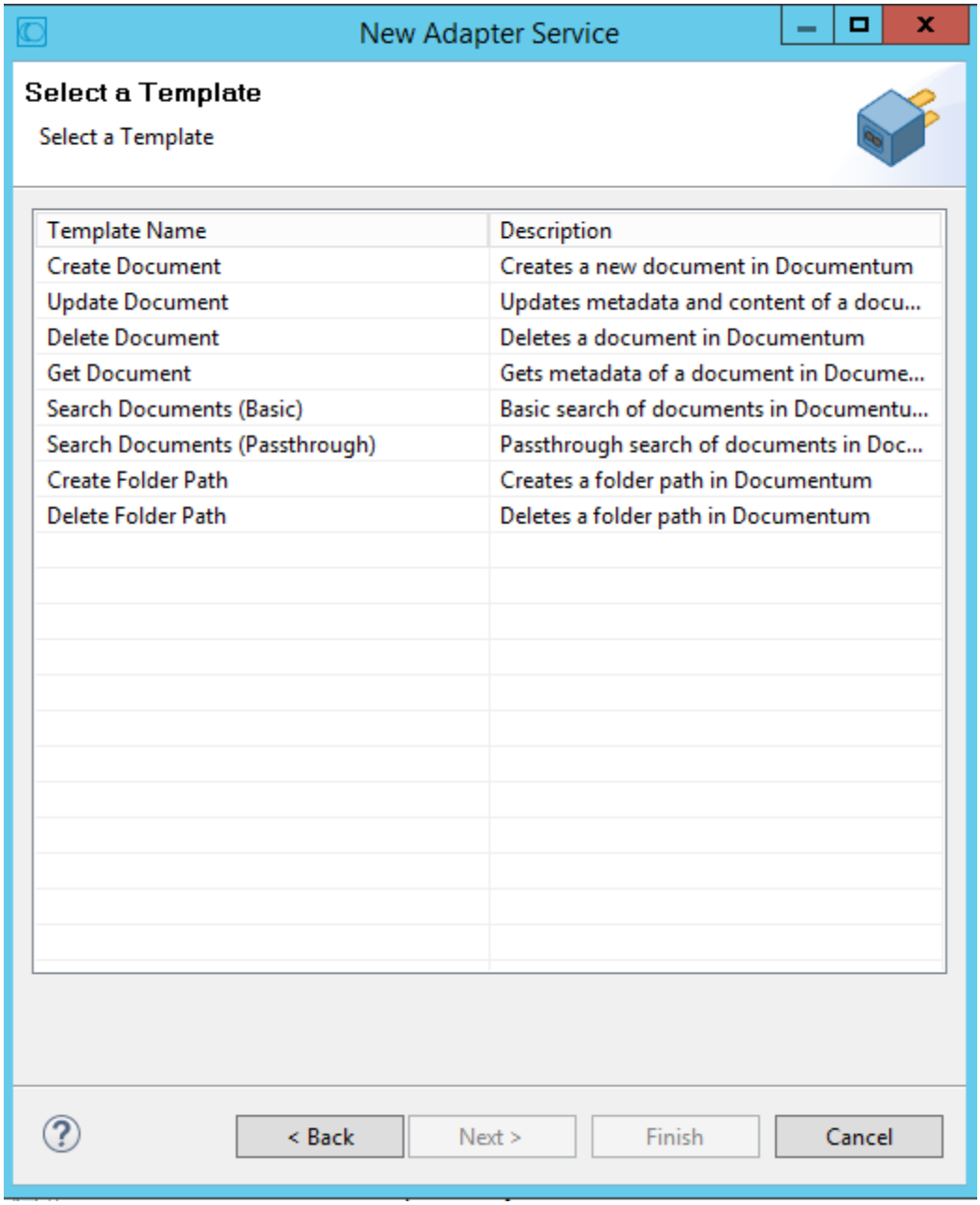


4. Select your preconfigured connection that should be associated with the service and click **Next**.



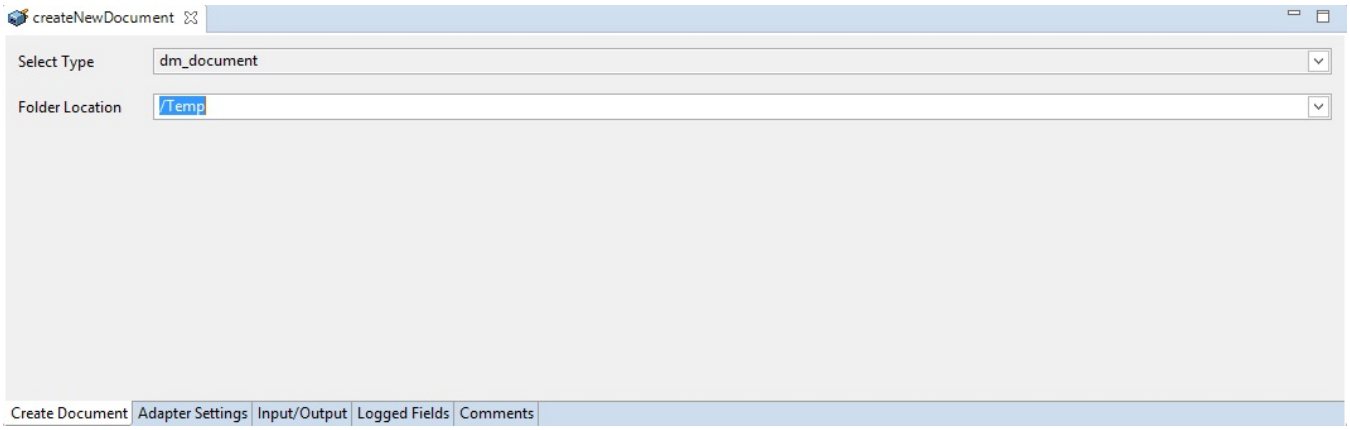
5. Finally, the service templates screen appears, showing the templates that are contained in the package. Each service has a corresponding template.

The next section describes how to create each service provided by the Documentum Adapter.



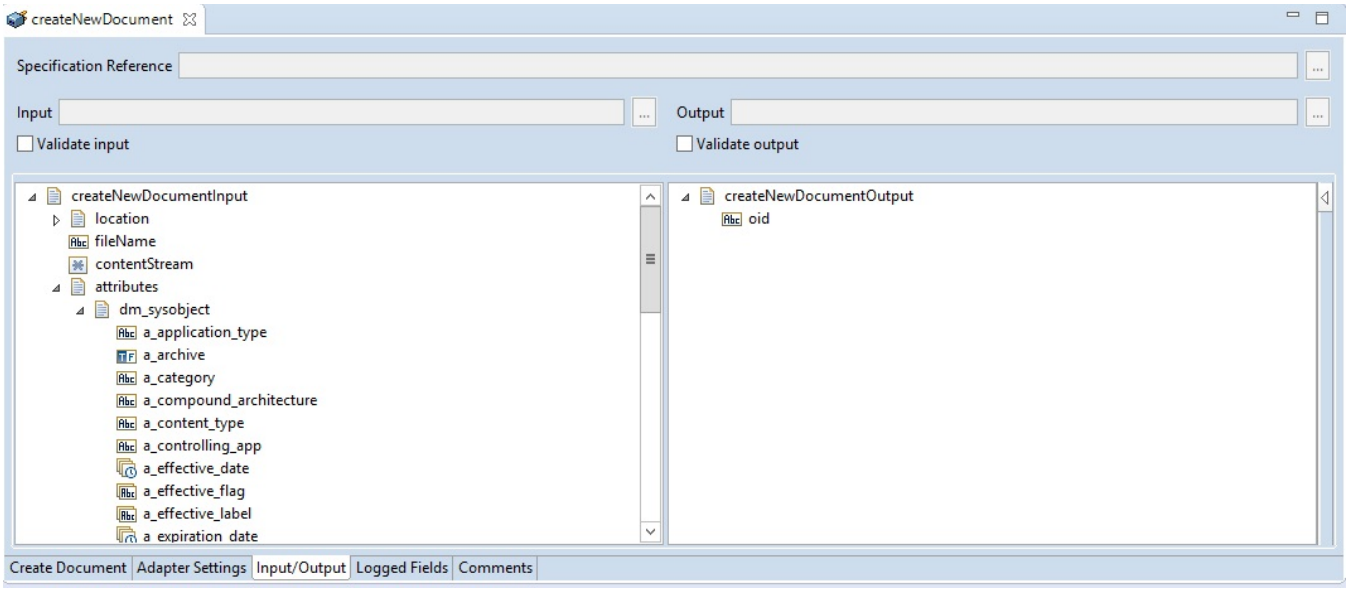
Click **Finish**.

The Documentum adapter queries all the available Documentum document types and provides you a following list:



The first entry selected and all its attributes being provided. Choose the document type you want to create.

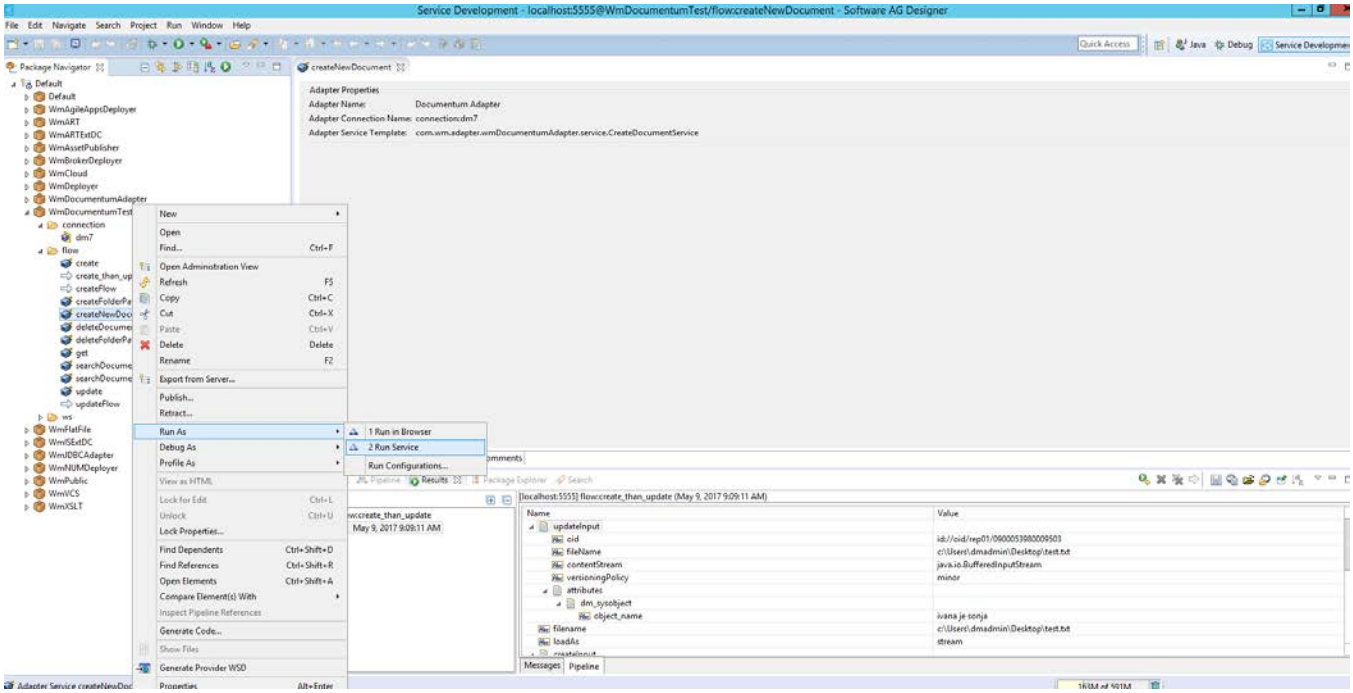
In the **Input/Output** tab you can see the webMethods docType that will be generated from the document type selected:

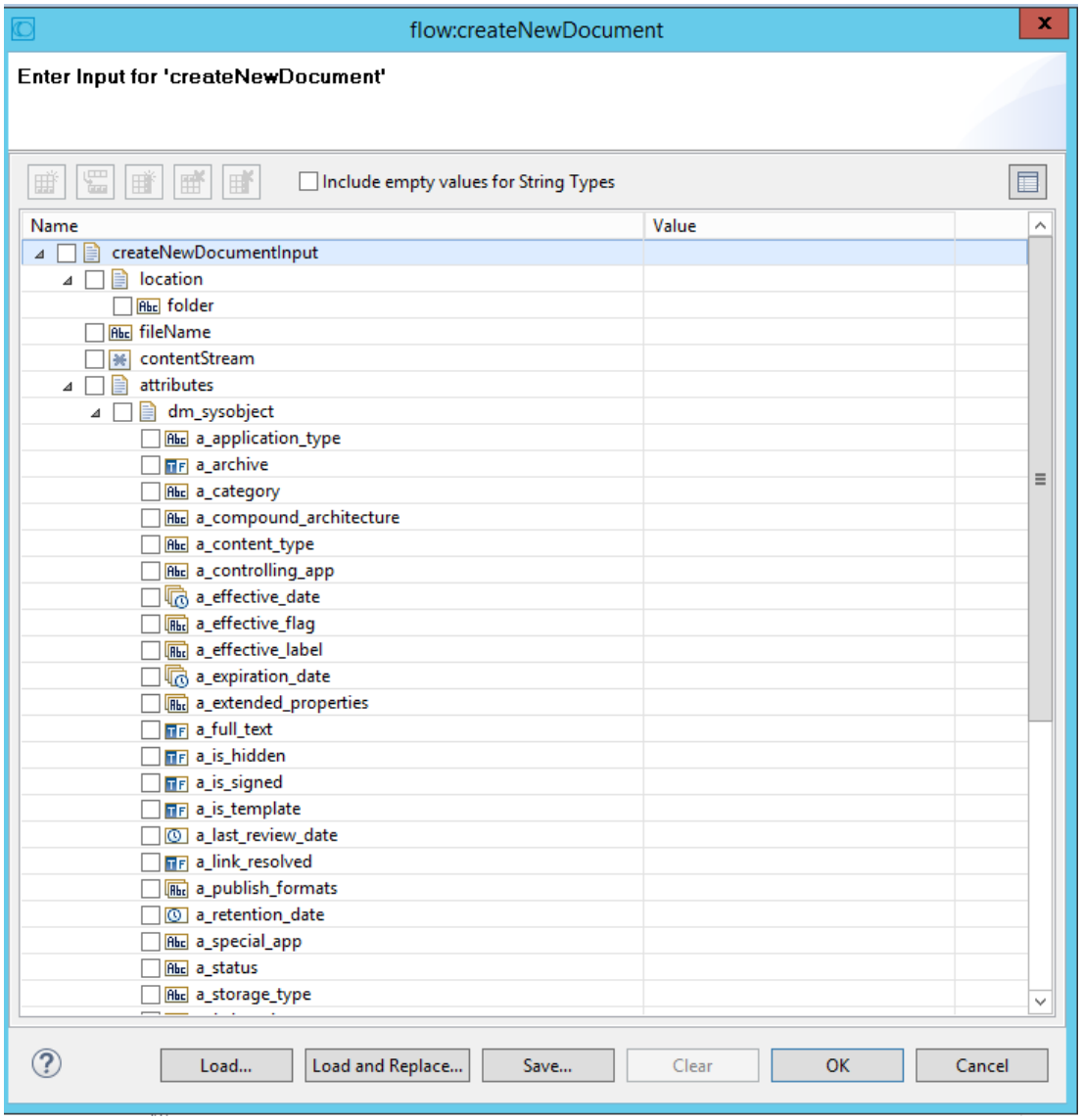


Save the new service.

6.2.3 Execute the service

Select the newly created service and Right-click , **Run As->Run as Service,** to get the service execution dialog box.


Service Development - localhost:5555@WmDocumentumTest/flow:createNewDocument - Software AG Designer
File Edit Navigate Search Project Run Window Help
Package Navigator
createNewDocument
Adapter Properties
Adapter Name: Documentum Adapter
Adapter Connection Name: connection:dm7
Adapter Service Template: com.wm.adapter.wmDocumentumAdapter.service.CreateDocumentService
Default
WmAgileAppsDeployer
WmART
WmARTExtDC
WmAssetPublisher
WmBrokerDeployer
WmCloud
WmDeployer
WmDocumentumAdapter
WmDocumentumTest
connection
dm7
flow
create
createFlow
createNewDoc
deleteDocume
get
searchDocume
update
updateFlow
ws
WmFlatFile
WmISExtDC
WmJDBCAdapter
WmNUMDeployer
WmPublic
WmVCS
WmXSLT
New
Open
Find... Ctrl+F
Open Administration View
Refresh F5
Copy Ctrl+C
Cut Ctrl+X
Paste Ctrl+V
Delete Delete
Rename F2
Export from Server...
Publish...
Retract...
Run As
Debug As
Profile As
View as HTML
Lock for Edit Ctrl+L
Unlock Ctrl+U
Lock Properties...
Find Dependents Ctrl+Shift+D
Find References Ctrl+Shift+R
Open Elements Ctrl+Shift+A
Compare Element(s) With
Inspect Pipeline References
Generate Code...
Show Files
Generate Provider WSD
Properties Alt+Enter
1 Run in Browser
2 Run Service
Run Configurations...
Pipeline
Results
Package Explorer
Search
[localhost:5555] flow:create_than_update (May 9, 2017 9:09:11 AM)
wcreate_than_update
May 9, 2017 9:09:11 AM
Name
Value
updateInput
oid
id://oid:/rep01/0900053980009503
fileName
c:\Users\dmadmin\Desktop\test.txt
contentStream
java.io.BufferedInputStream
versioningPolicy
minor
attributes
dm_sysobject
object_name
ivana je sonja
filename
c:\Users\dmadmin\Desktop\test.txt
loadAs
stream
Messages
Pipeline



- **Attributes (under attributes/<document type>)**

Fill in Documentum attributes like *object_name*, title, *firstname*, *surname* etc.

Note: If you do not fill the *object_name* and *title*, it will not appear in the Documentum web client.

- **Folder Name (under location/folder)**

You can select a folder from the location where this document will be imported in Documentum (start the path with the Documentum cabinet). This is an optional process.

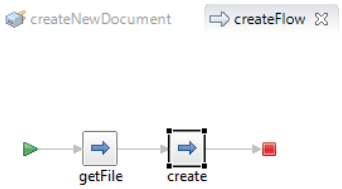
For example in Documentum web client you can see the folders as follows:

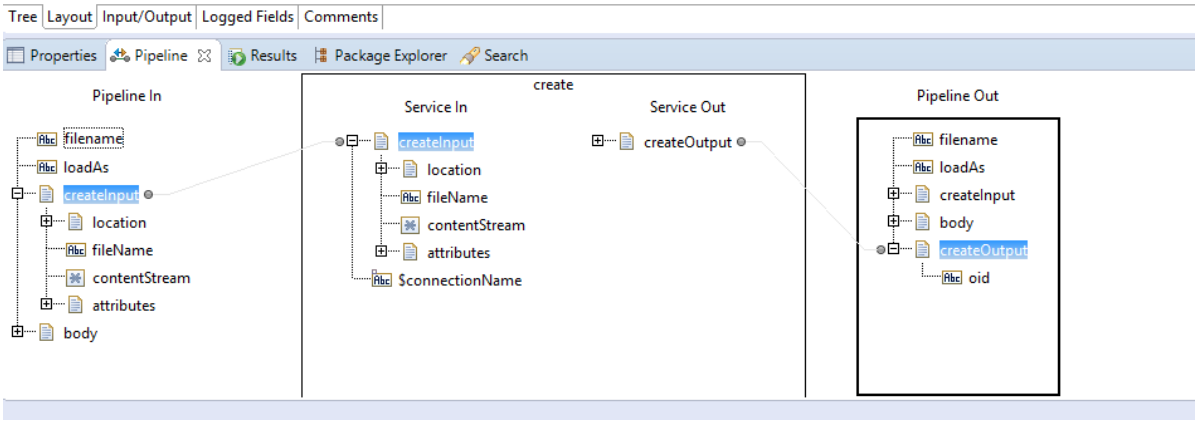
Select the Temp cabinet as location for new document in this folder /Temp/A/B.

- **ContentStream**
- **FileName**

You can also attach content to the newly created document by passing some content stream to input of this service. If stream is passed to this service FileName as input is also required (used to get file extension of passed stream). This is an optional process. So, either both parameters are optional, or if contentStream is passed, then also filename needs to be passed.

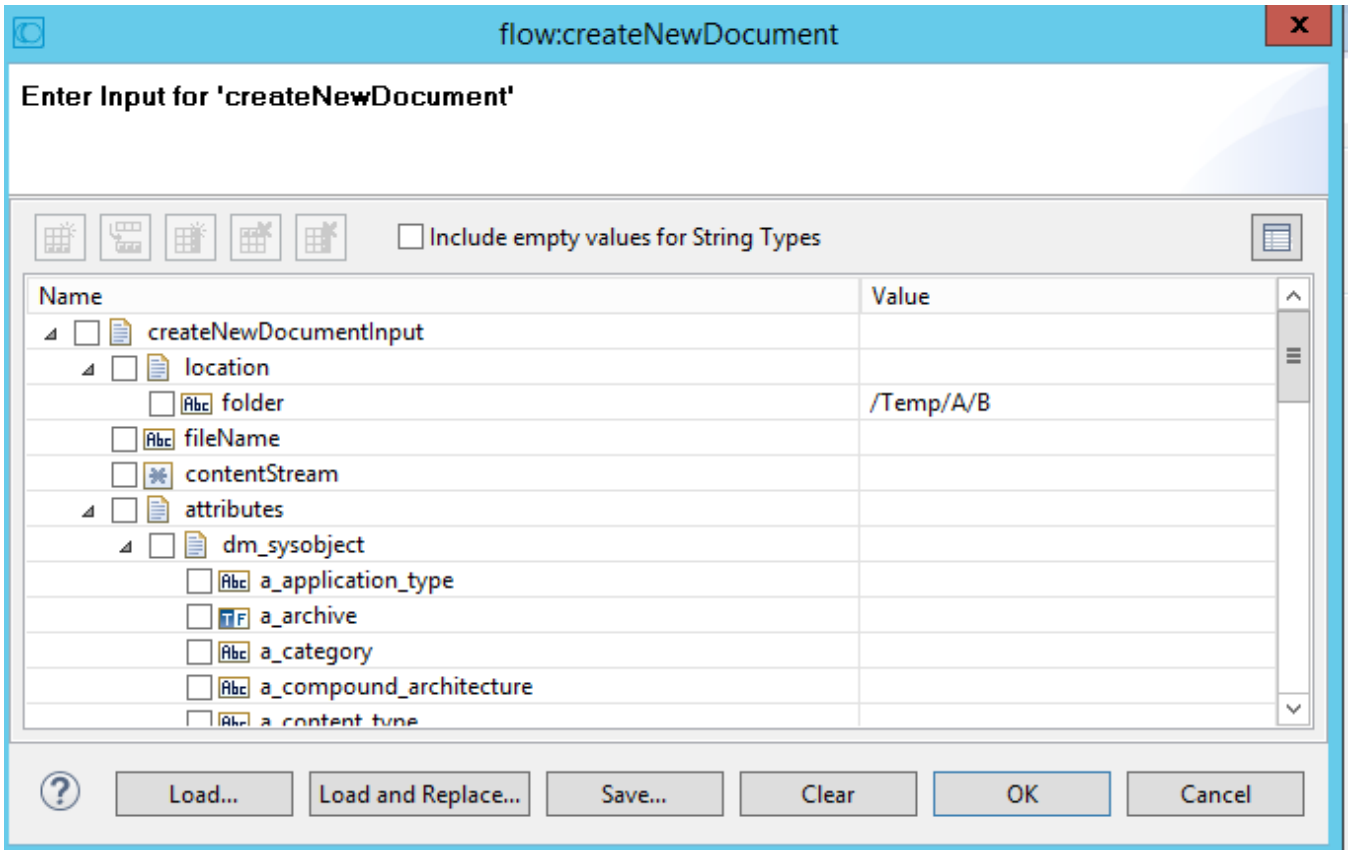
Note: New Action Flow service can be created with two included services getFile and createNewDocument to properly test case when contentStream is also passed. First service will read file from file system and send its stream and filename to another service.

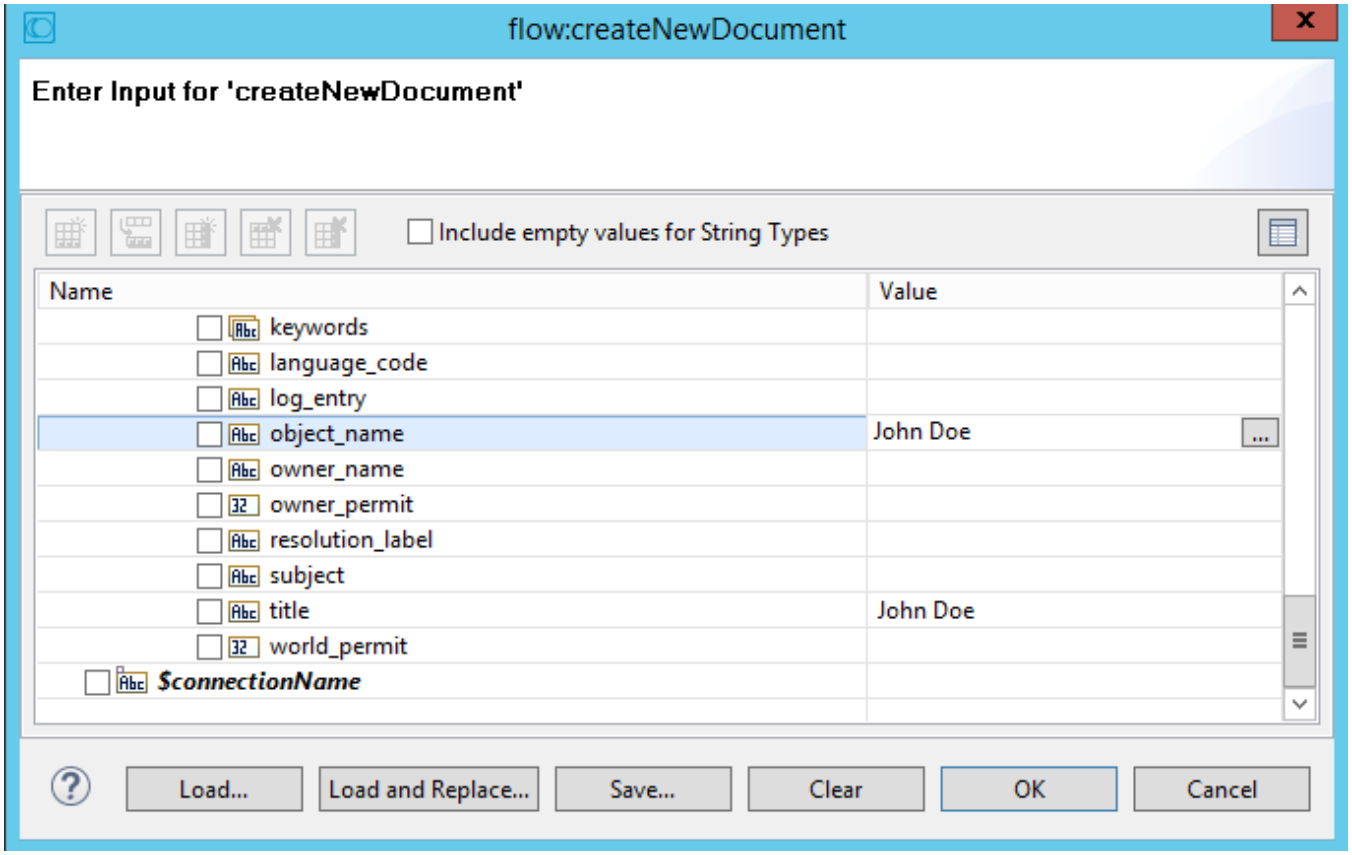




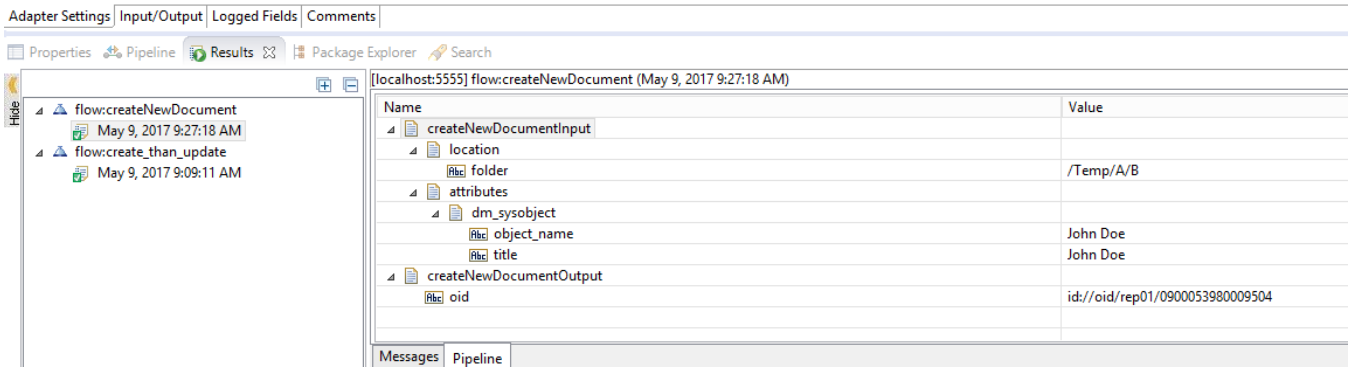
Optionally you can configure an alternative connection (non-default) under `$connectionName`.

Having filled in the information, click **OK.**





Click **OK** to execute and you will get the following:



In the Results tab containing an ObjectIdentifier (OID) of Documentum of the newly created document

id://oid/rep01/0900053980009504

If you change into Documentum web client you will see that a new document has been created.

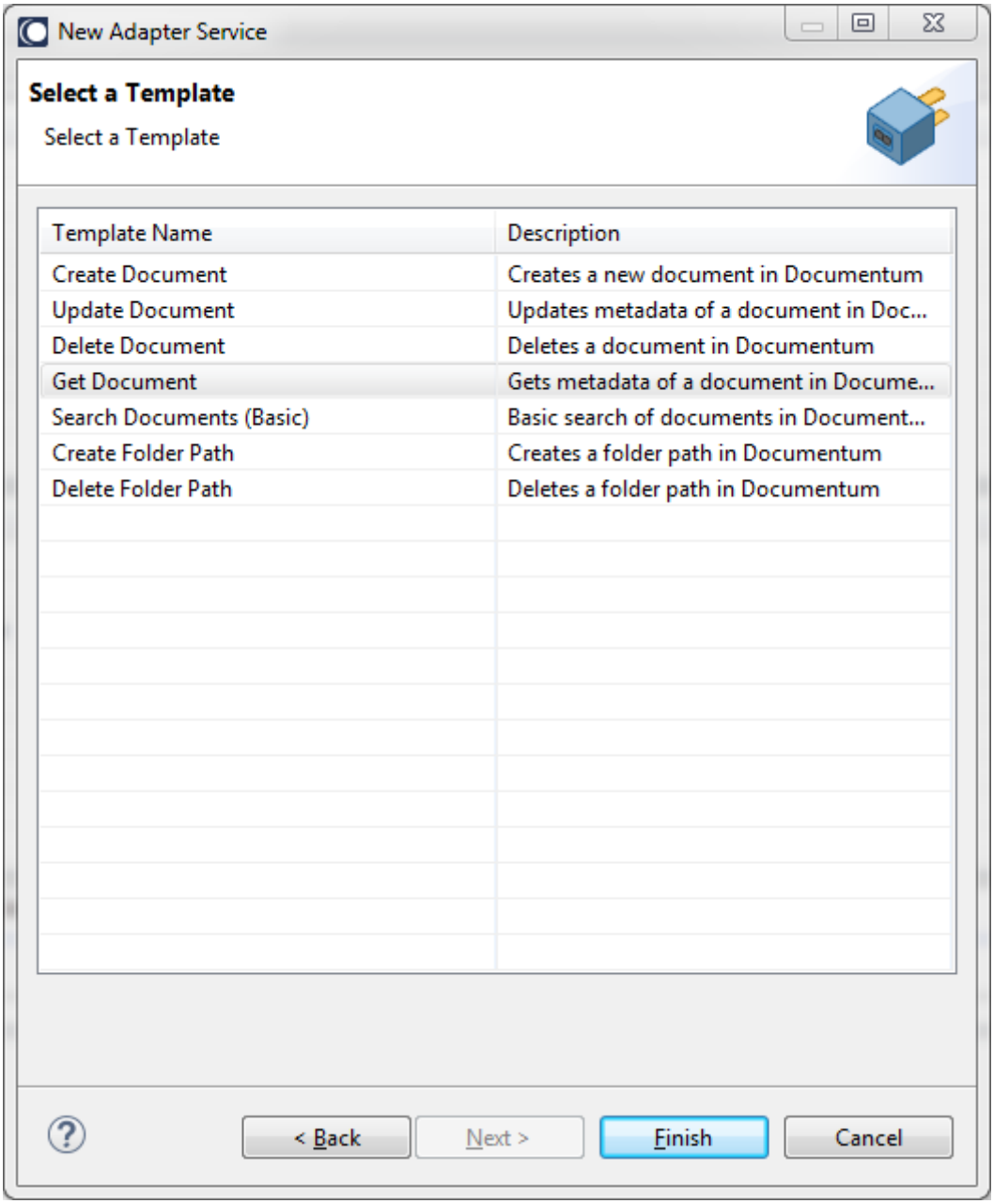
6.3 Get Document

6.3.1 Description

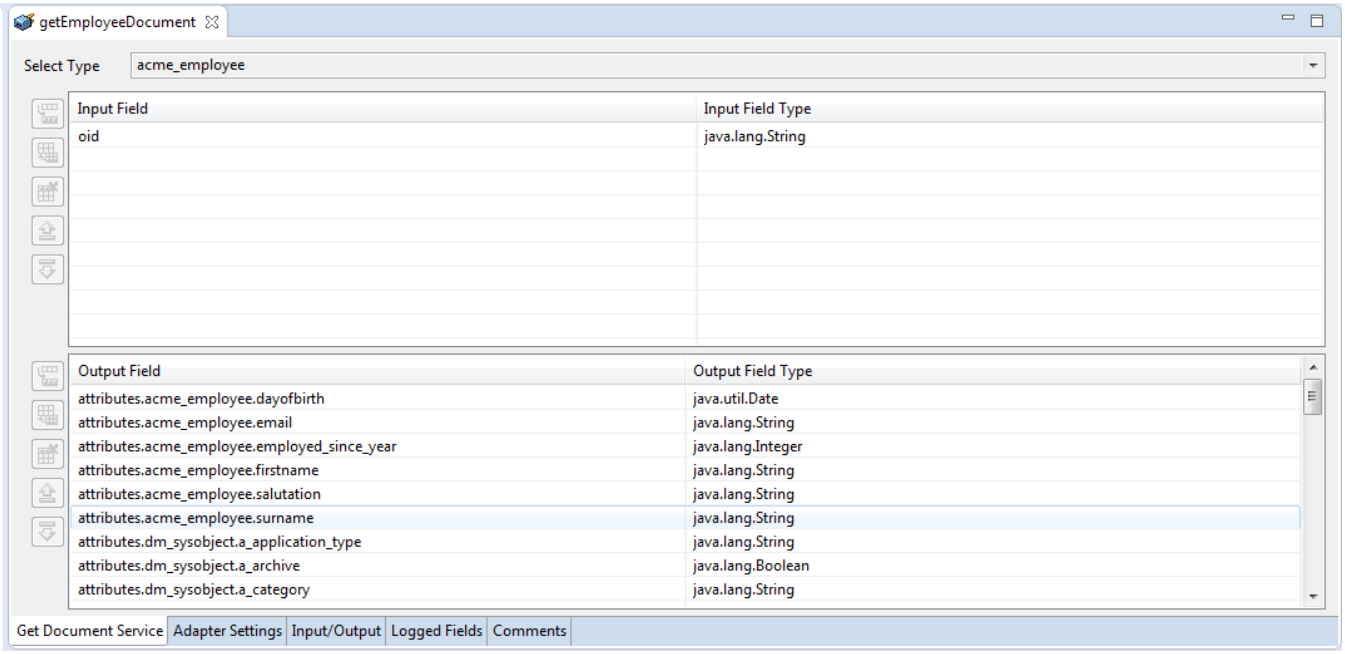
This service allows you to get back meta data and contents associated with a document in Documentum by specifying an object identifier.

6.3.2 Configure the service

Create the service as described in create adapter services by naming it `getEmployeeDocument` and choose the **Get Document** service template .



In the Software AG Designer, the result will look like:

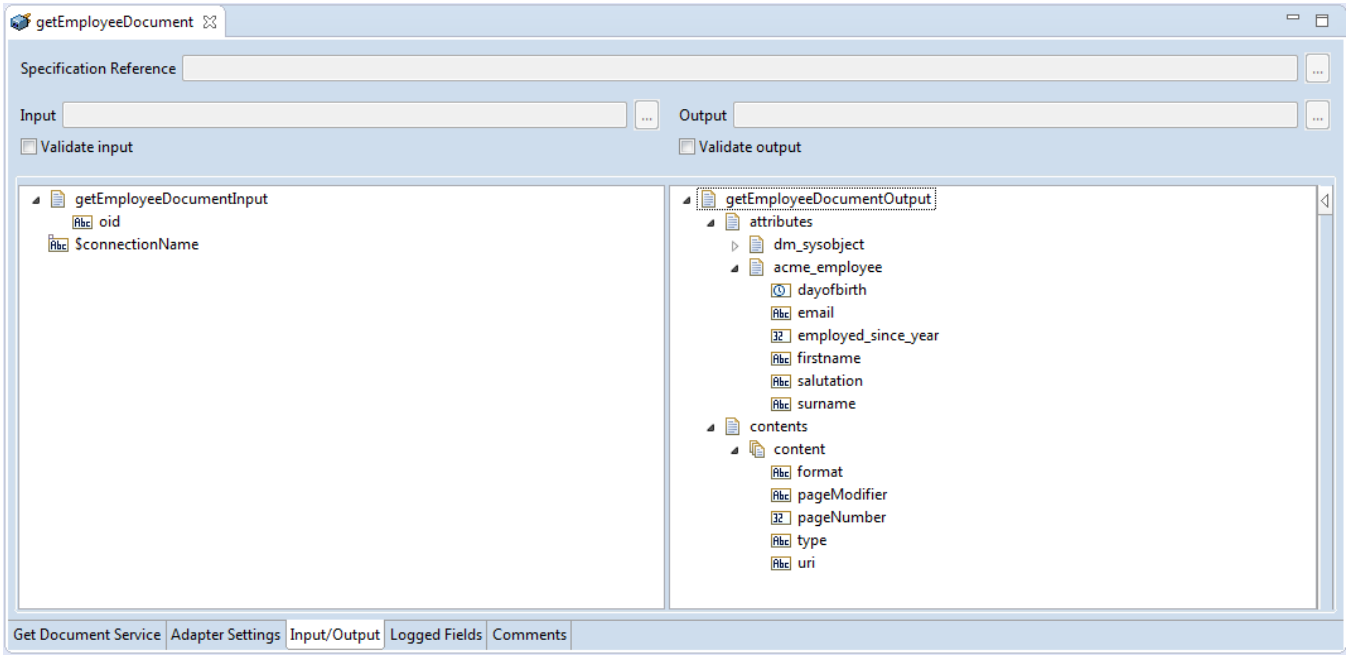


The screenshot below provides you with the following options:

Configuration Tab	Input	Description
Get Document Service	Select Type	Allows you to select the *document type* being retrieved from Documentum in the list box; in our example, we have chosen the type acme_employee. Once you choose a document type the input/output signature of the service is generated.
Input/Output	Input	The only input is an object identifier (attribute `oid`) which is the result of a create document service.
	Output	The output of the service is a node tree attributes <document type> <attribute> contents content[] format pageModifier

Configuration Tab	Input	Description
		pageNumber type uri pageNumber

In our example we have chosen document type acme_employee and this will give you the following service signature in the Input/Output tab

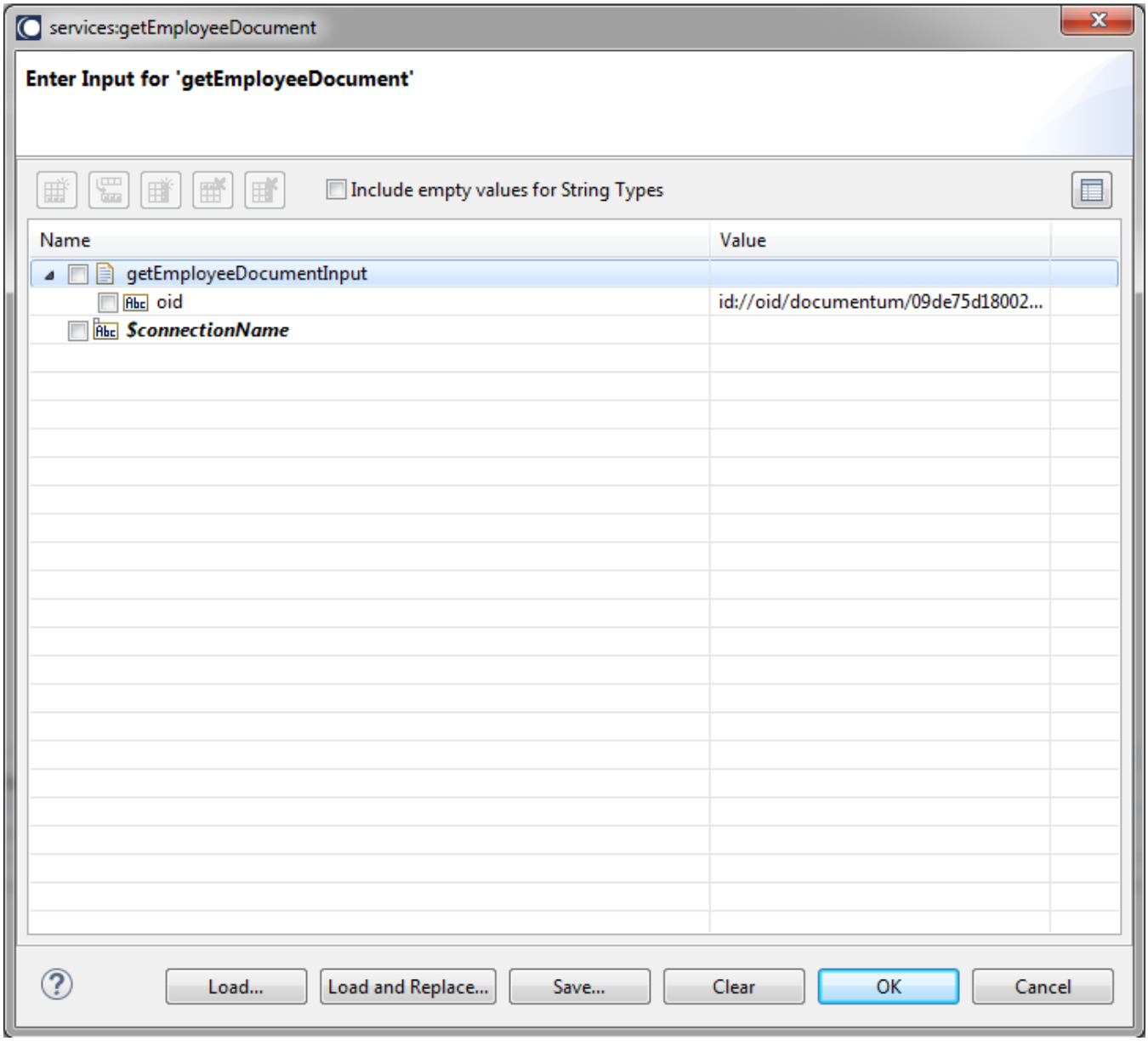


As mentioned the only input here is oid (the object identifier) and the output shows you

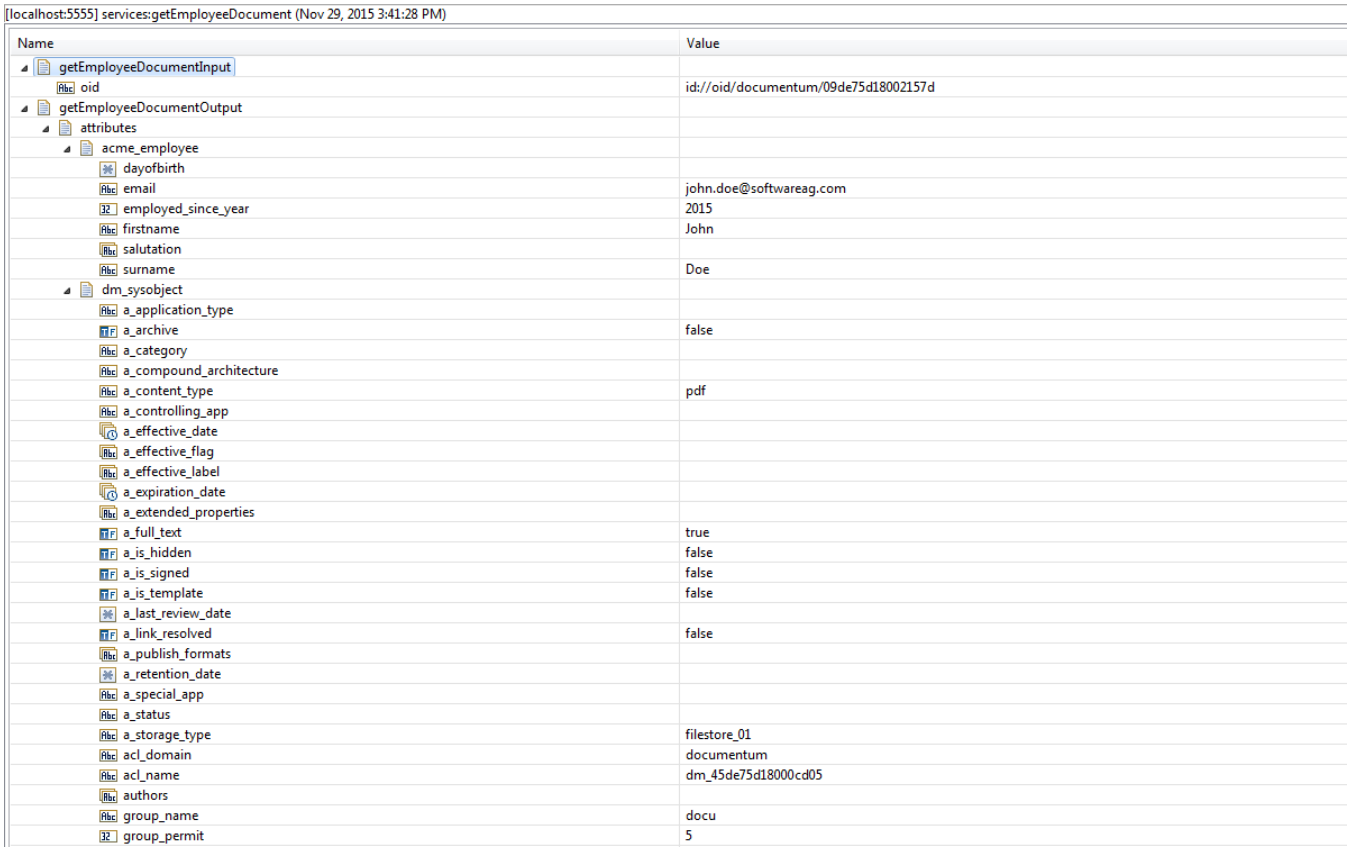
- Under attributes two document types, `dm_sysobject` and `acme_employee` where `acme_employee` is the document type we have choosen to get meta data and `dm_sysobject` is the supertype and it inherits attributes from (if there were alos supertypes of dm_sysobject having additional properties they would have been shown here too as separate document type)
- Under contents a list of content nodes detailling the contents associated with the document (if any)

6.3.3 Execute the service

To run this service do a Right-click, **Run as**->**Run As Service** and provide an existing object identifier for input oid.



The result could appear as follows:


[localhost:5555] services:getEmployeeDocument (Nov 29, 2015 3:41:28 PM)

Name	Value
getEmployeeDocumentInput	
oid	id://oid/documentum/09de75d18002157d
getEmployeeDocumentOutput	
attributes	
acme_employee	
dayofbirth	
email	john.doe@softwareag.com
employed_since_year	2015
firstname	John
salutation	
surname	Doe
dm_sysobject	
a_application_type	
a_archive	false
a_category	
a_compound_architecture	
a_content_type	pdf
a_controlling_app	
a_effective_date	
a_effective_flag	
a_effective_label	
a_expiration_date	
a_extended_properties	
a_full_text	true
a_is_hidden	false
a_is_signed	false
a_is_template	false
a_last_review_date	
a_link_resolved	false
a_publish_formats	
a_retention_date	
a_special_app	
a_status	
a_storage_type	filestore_01
acl_domain	documentum
acl_name	dm_45de75d18000cd05
authors	
group_name	docu
group_permit	5

- Under `acme_employee` you can see some data of the employee, also
- we have meta data in `dm_sysobject` that is automatically created by Documentum when a document is created
- Contents you see that one PDF content is associated with this employee document and that it has an URI specified where you could retrieve the document

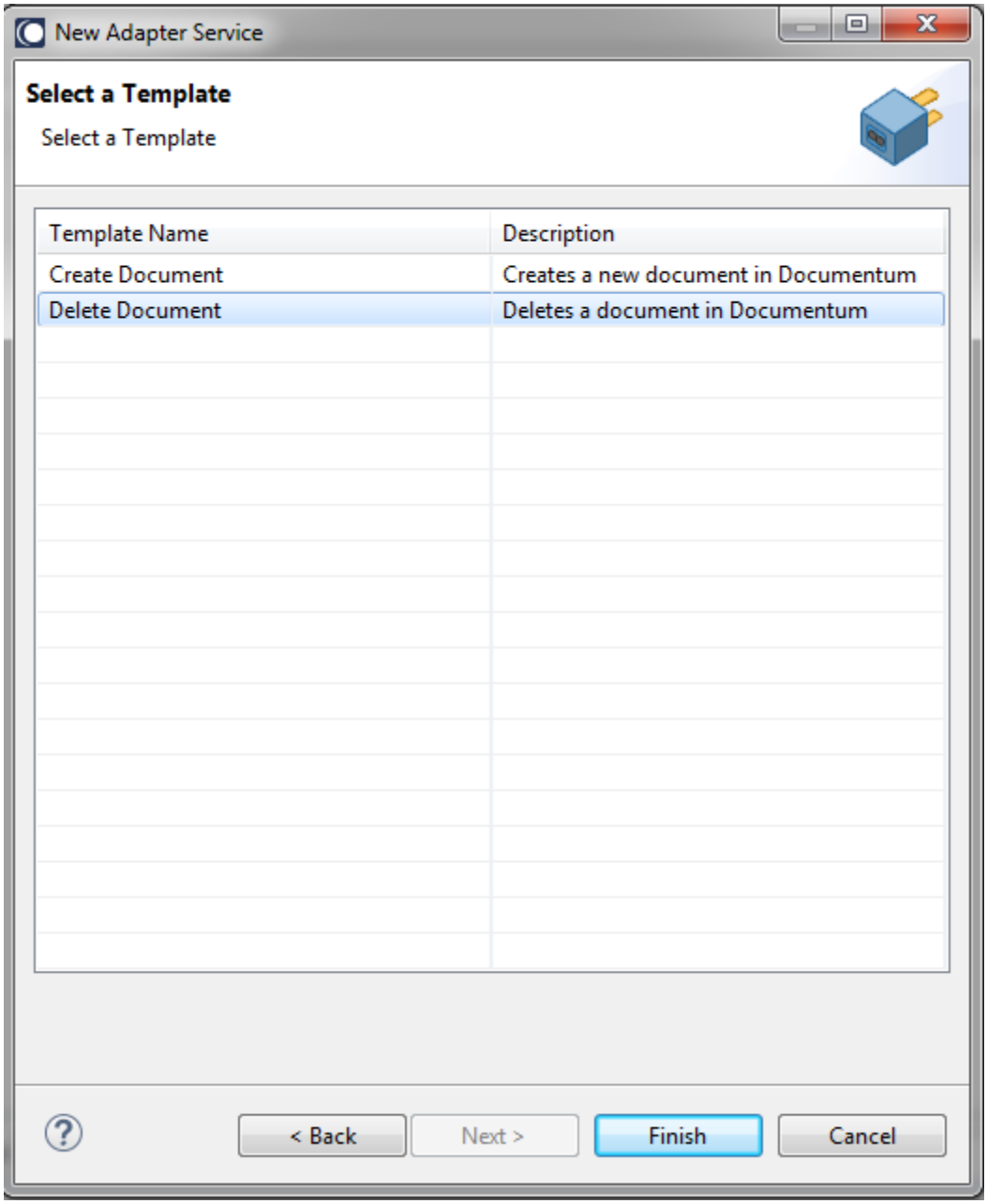
6.4 Delete Document

6.4.1 Description

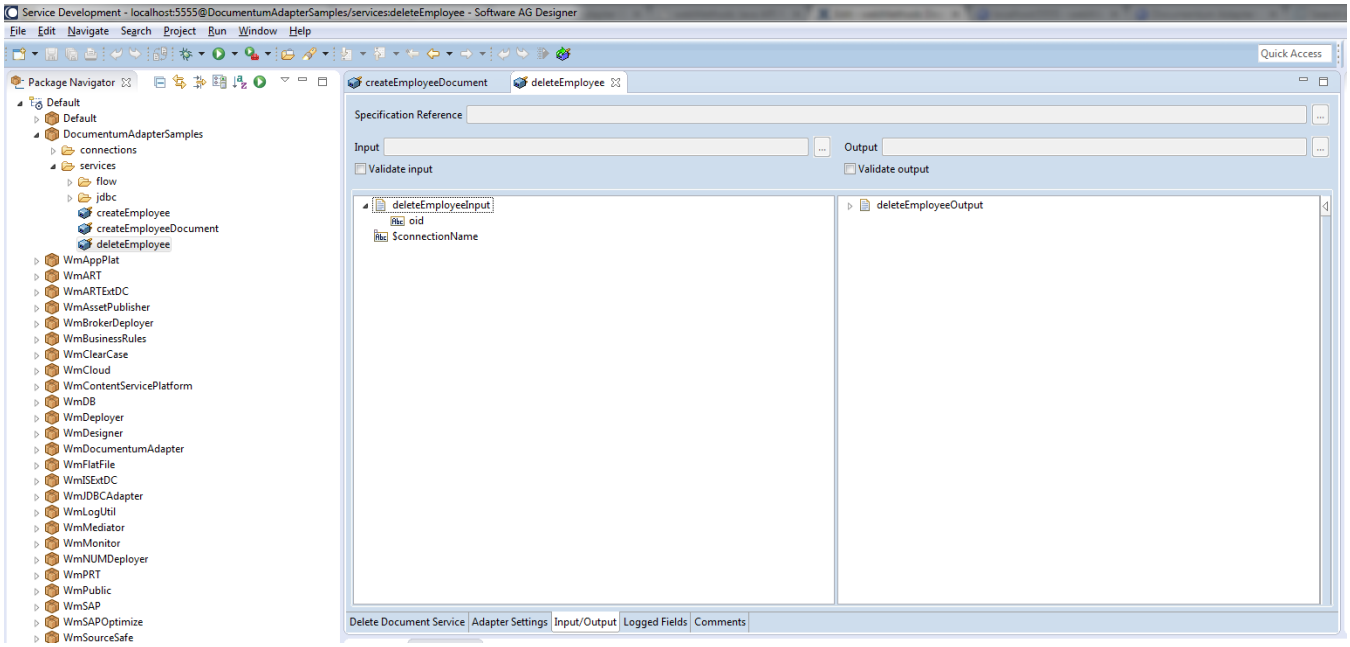
This service enables you to delete a document in Documentum, provided an object identifier like id://oid/documentum/09de75d180021539.

6.4.2 Configure the service

Create the service as described in create adapter services by naming it `deleteEmployee` and choose the **Delete Document** service template.



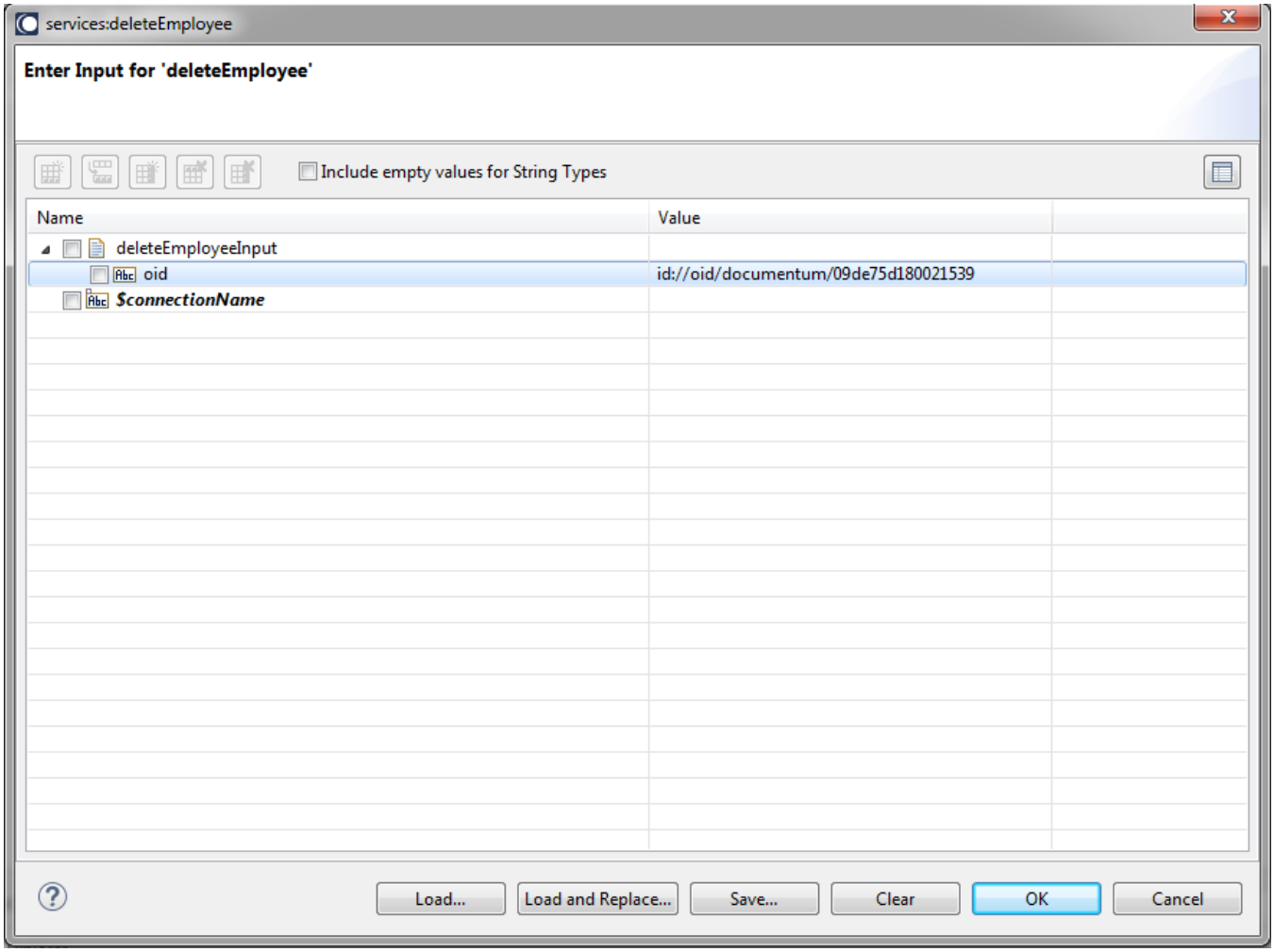
The resulting service is depicted in the following screenshot:



As you can see the only input the adapter service expects is an object identifier that is shown in the input field `oid` and so there is nothing more you have to configure.

6.4.3 Execute the service

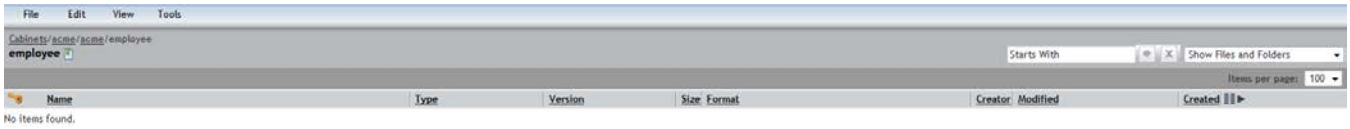
To run this service do a Right-click , **Run As**->**Run as Service** and provide some object identifier of an existing document.



If the deletion succeeds then you will get not receive any specific result and if the deletion fails then an exception is generated.

The document, which is missing from the location, is created before in the Documentum web client.

.



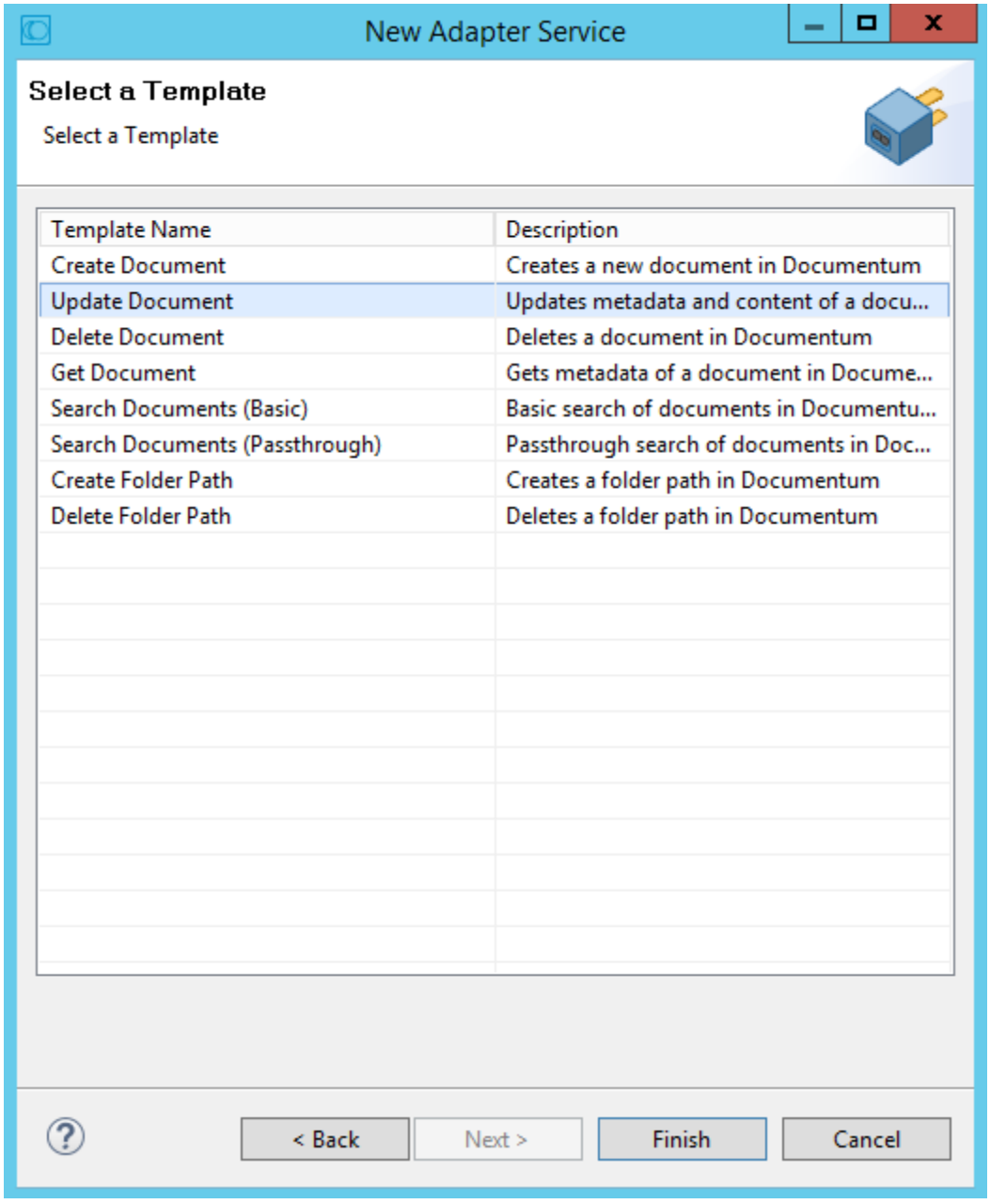
6.5 Update Document

6.5.1 Description

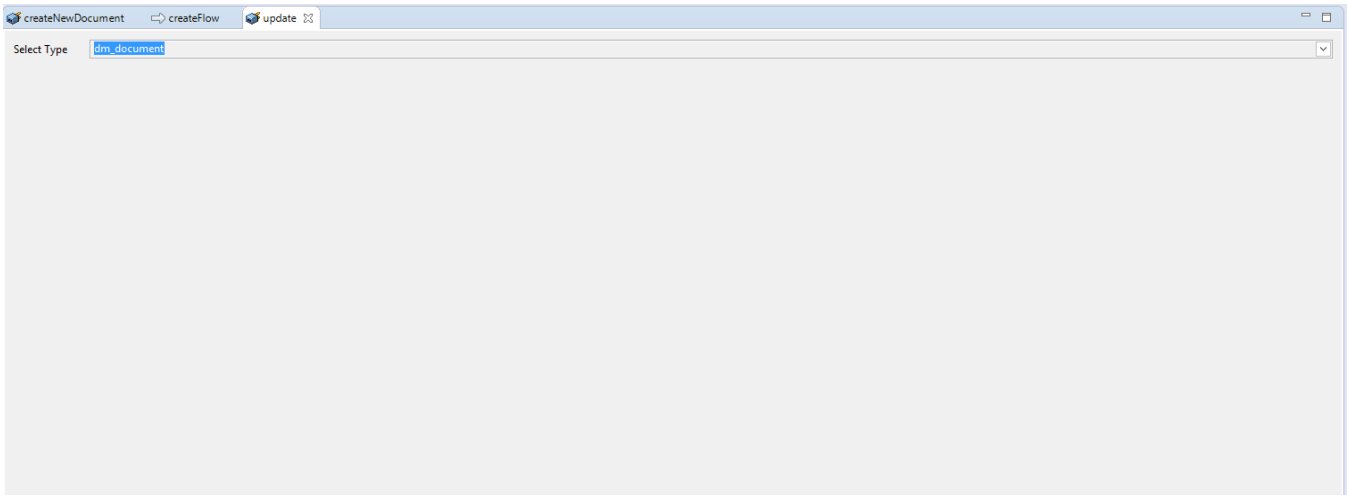
This service allows you to update the properties of a document already stored in Documentum.

6.5.2 Configure the service

Create the service as described in create adapter services by naming it `update` and choose the **Update Document** service template .



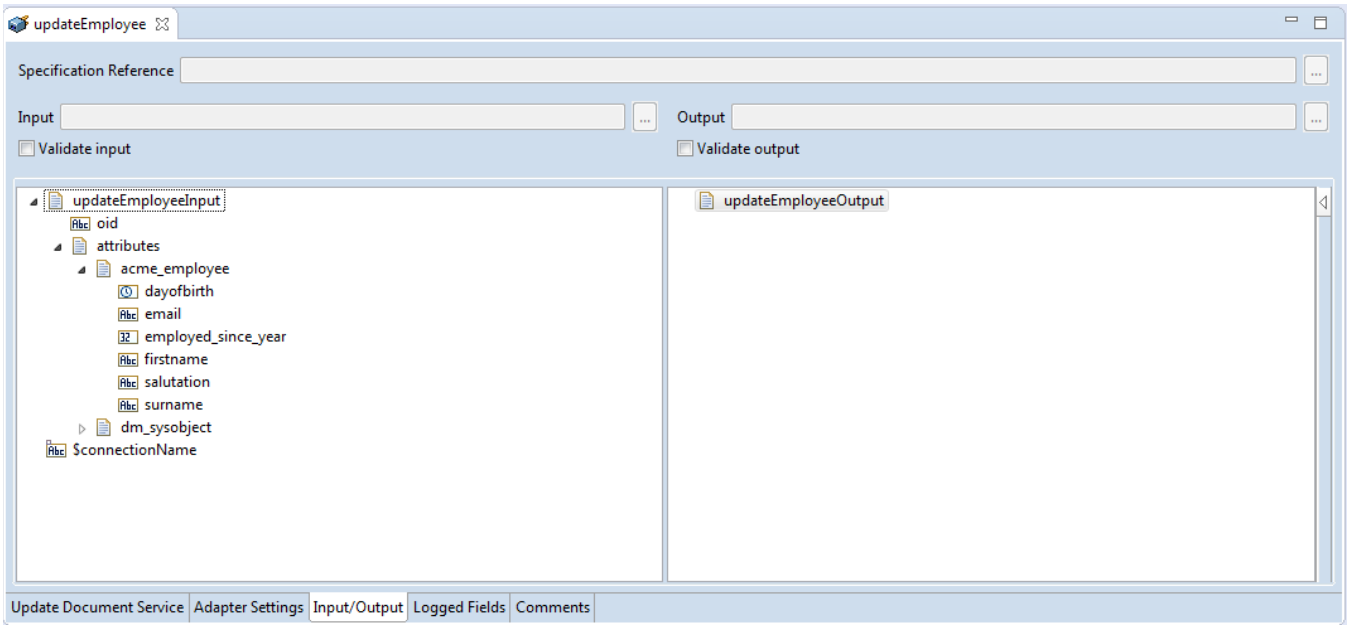
In the Software AG Designer the result looks as follows:



The screenshot below provides you with the following options:

- Oid – Mandatory parameter, there we pass object id of document being updated.
- ContentStream and FileName have same purpose as in create, they are optional, but if content stream is passed, then filename also needs to be passed. Document will be created with new content if content stream is passed.
- Versioning policy – Valid inputs here are strings MAJOR, MINOR or empty string (input is key insensitive). If MAJOR is passed, new major version of document will be created than updated with new attribute values and content if present. If MINOR is passed, new minor version of document will be created than updated. If empty string is passed or this input parameter remains empty, no versioning will be performed. Version with passed object id will be updated with new attribute values and content.

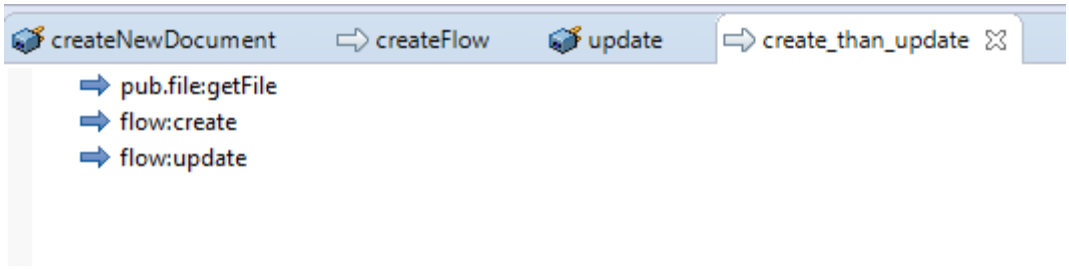
In our example we have chosen document type `dm_document` and this will give you the following service signature in the **Input/Output** tab



Having now the document type and attributes in the input signature, you should now provide the updated properties.

6.5.3 Execute the service

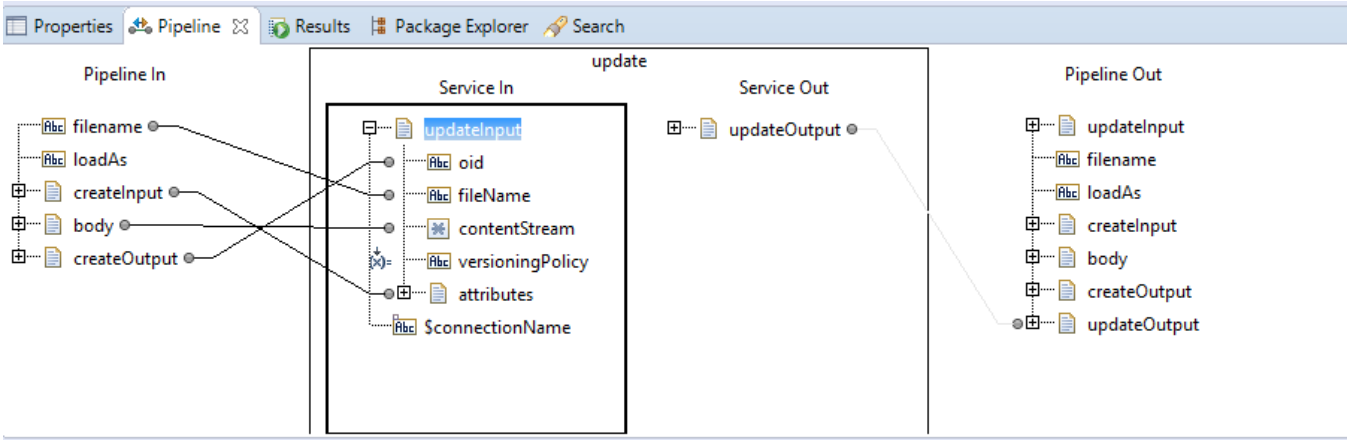
As this service mandates an existing document we cannot execute it right away using right click, **Run as->Run as Service**.But you should implement a small flow service, refer webMethods Integration Server administrator's Guide which could look as follows:



Also executes the following three Documentum adapter services:

- Reads file from file system and sends stream to create service.
- Creates a new document as described in Create Document Service
- Updates the employee document with new property values

So, after the document is created you have the oid in the pipeline and you can map it together with some property updates as shown in the following picture:



Execute the service using right click - **Run as**->**Run Flow Service**, and it should proceed without errors.

Ensure to set a breakpoint on the service lines saying the following:

- services:createDocument

Else you cannot see the property changes in Documentum (e.g. via webtop).

6.6 Search Documents

6.6.1 Description

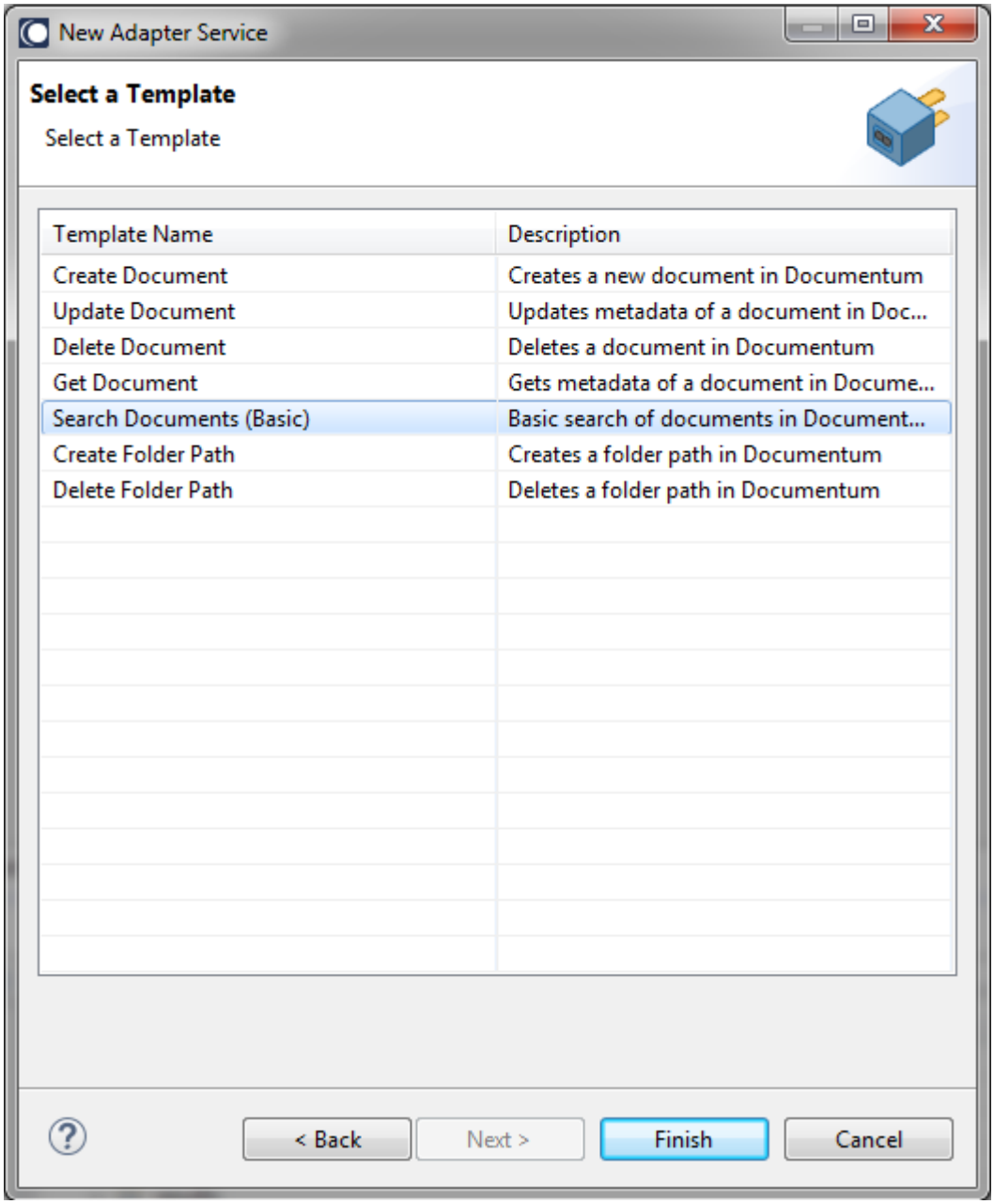
You can use this search service to do basic searches in Documentum. To define a search you have to select a document type you want to search, then you can choose which properties you want to have returned in the search and make some basic search configuration (number of records returned, do a conjunction or a disjunction search)

Finally, you can provide a list of search conditions that have to be applied.

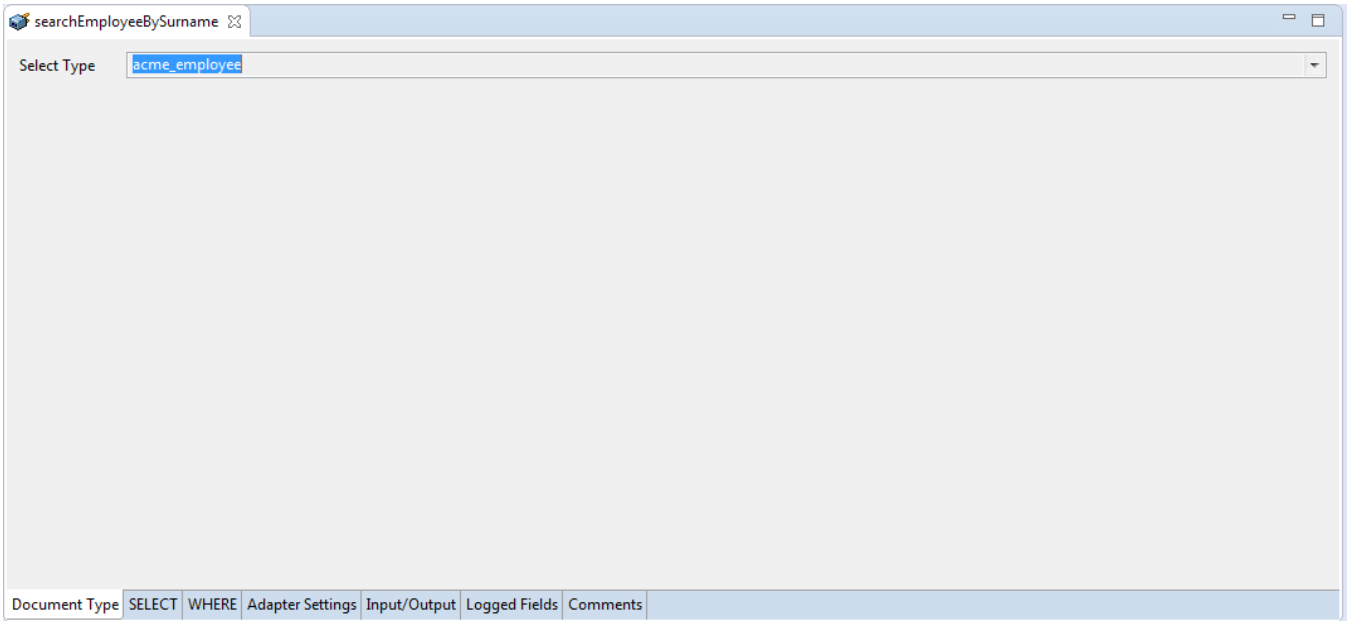
As the name of the search template implies this is a basic search in the sense that you can only specify the logical root operator conjunction or disjunction, whichhas to be applied to a list of property expressions but you are not able to specify nested expressions (this will be provided in a special search called structured query search).

6.6.2 Configure the service

Create the service as described in create adapter services by naming it `searchEmployeeBySurname` and choose the **Search Documents (Basic)** service template **.**

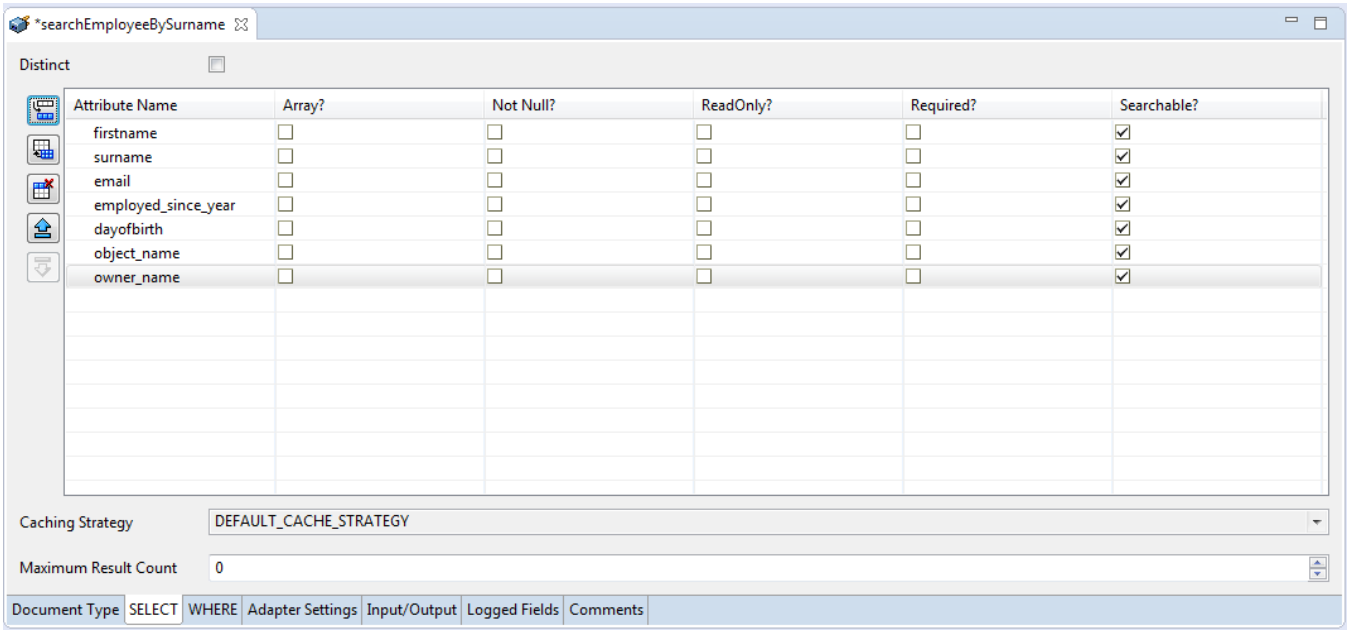


In Software AG Designer the search template is depicted as follows:



The only thing you have to select here is the document type you want to do a search. Depending on the selected document types the input/output signatures will provide you with different properties you can choose.

So, click **SELECT** tab and select the properties you want to have returned from the search as follows:



You can use the controls on the left to view.

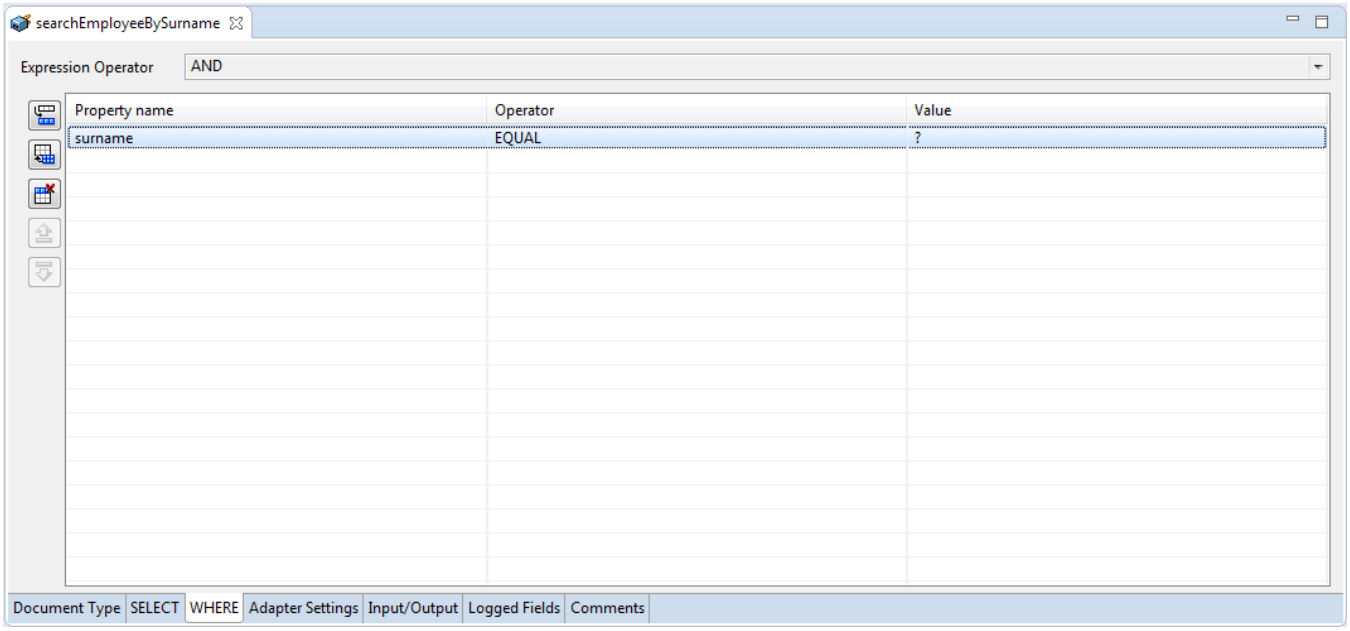
Action	Description
	Inserts a new property: the first property inserted is the first property that is defined in the choosen document type (or in one of its supertypes), sorted in alphabetical order.

Action	Description
	If you click in the Attribute Name table cell and choose a different property for this row then the property next inserted is the next one in the property list after the one choosen
	Creates one row per property available on the selected document type (or in one of its supertypes). **be careful** Icon In Documentum a document type can have dozens of properties inherited and in webMethods ADK there is no button to remove all records of the table, so you have to remove them one by one!
	Removes the selected row
	Can be used to re-order the properties in the table. **Note that** Icon Reordering does not affect the output signature.

Also, in this view you can specify the following

Attribute	Description
Distinct	Do a distinct search?
Caching Strategy	The caching strategy used in Documentum
Maximum Result Count	Limit the number of results

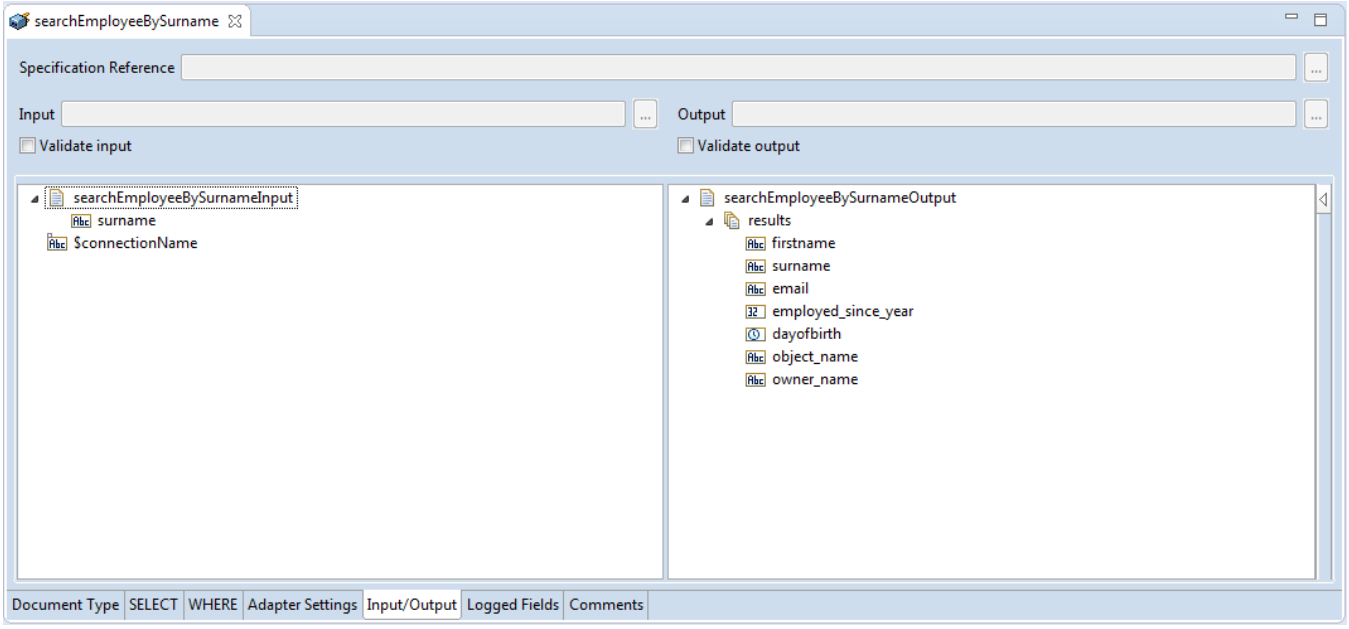
If you have selected the properties you want to search, then go to **WHERE** tab and specify the expressions that should apply to your search.



In our example, we have added one expression to filter for property surname EQUALS ?, so we search for all employees that have a surname matching the one we provide in the service template when executing the service.

Here you also have the option to specify the root expression operator which can be AND or OR and that is applied to the list of property expressions you configure here.

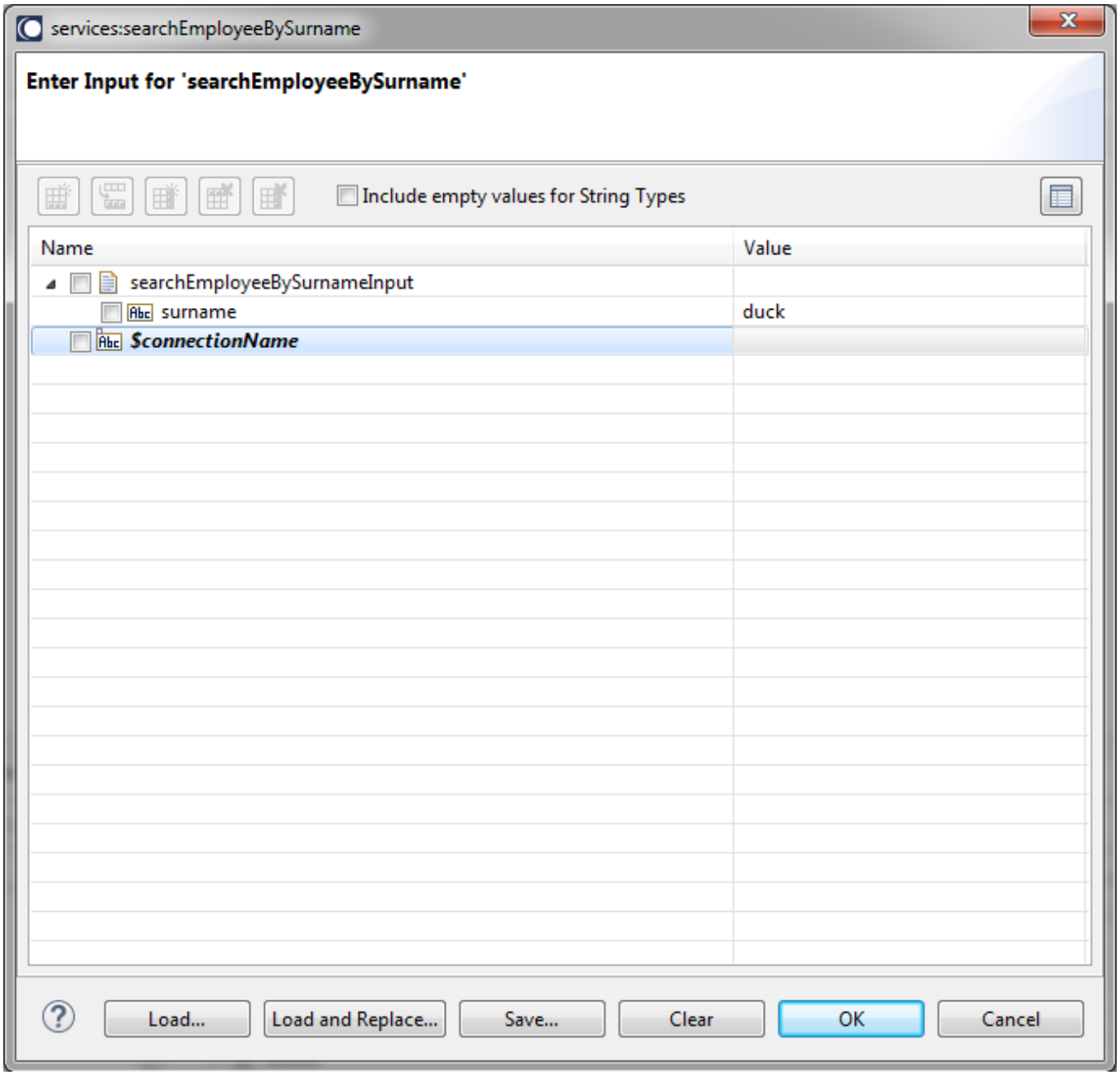
Once you have configured your service template , it looks like the following depicted in the below screenshot:



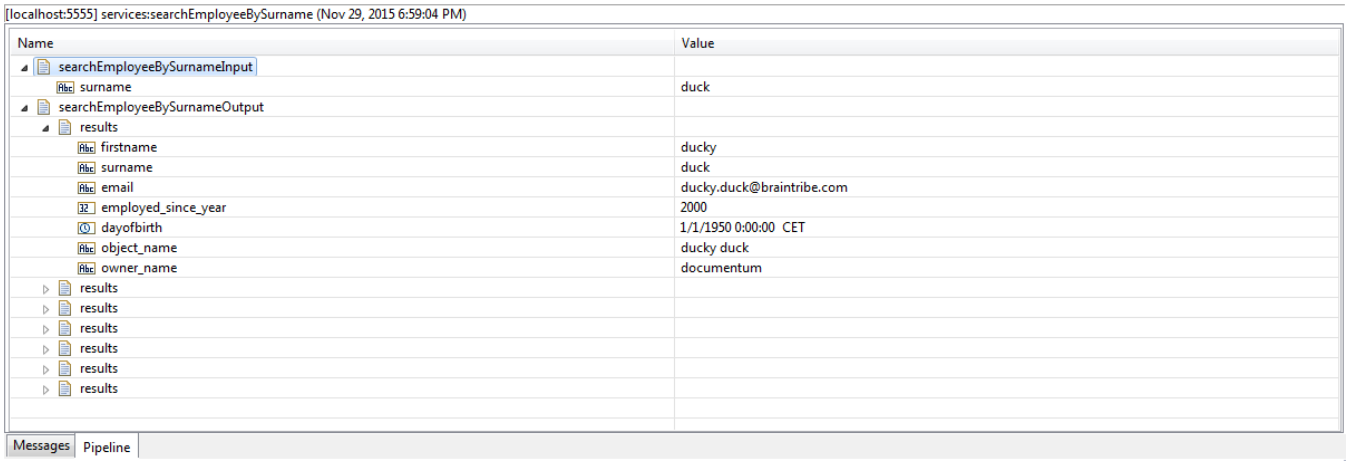
where you have in the input signature on the left all the properties you want to specify a operator expression for (e.g. the value of the surname to filter) and on the right you have under the results node the list of properties you have configured in the SELECT tab.

6.6.3 Execute the service

To run this service do a Right-click, **Run as**->**Run as Service** and specify a value for the surname property you want to search for. Example,



Executing the search will give you something like



and you can see that only employees named duck are passed back and that only the properties you wanted to include are passed back with the search.

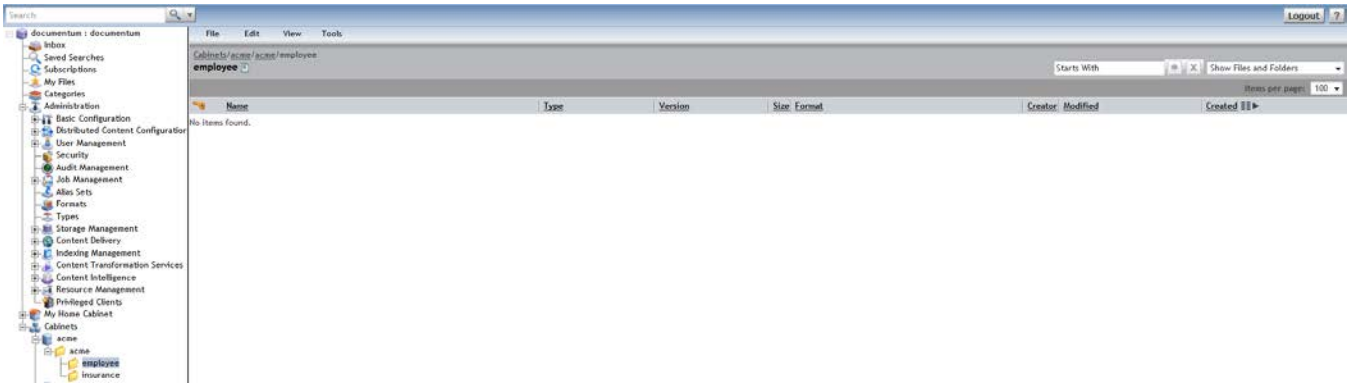
6.7 Create Folder Path

6.7.1 Description

This service allows you to create a folder path in Documentum.

Folders in the Documentum are documents of type `dm_folder`. In this case, it describes the means of structuring the location where the documents are created.

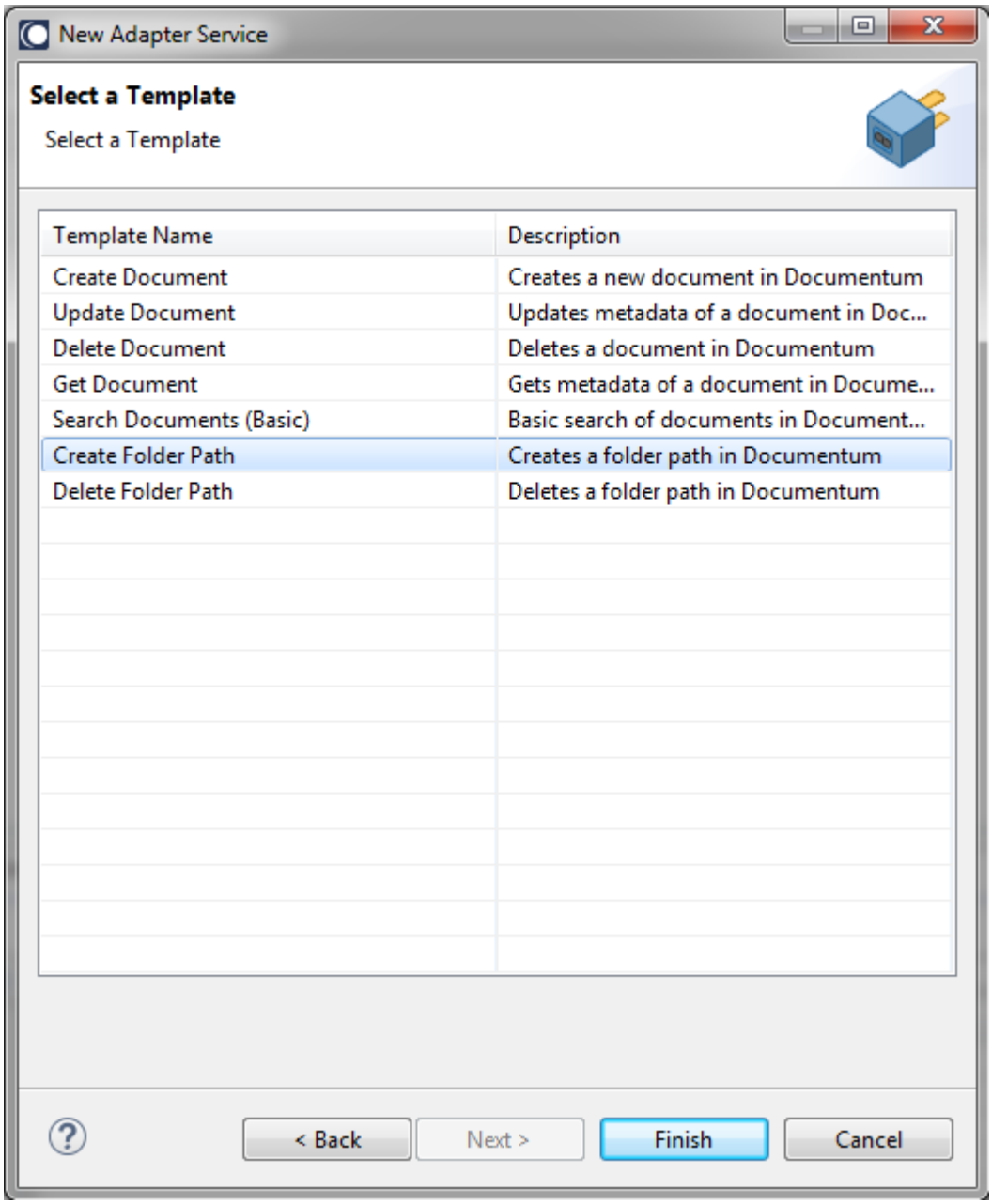
If you open Documentum webtop you can see the folders that have been created in a folder tree directly under the cabinets as follows:



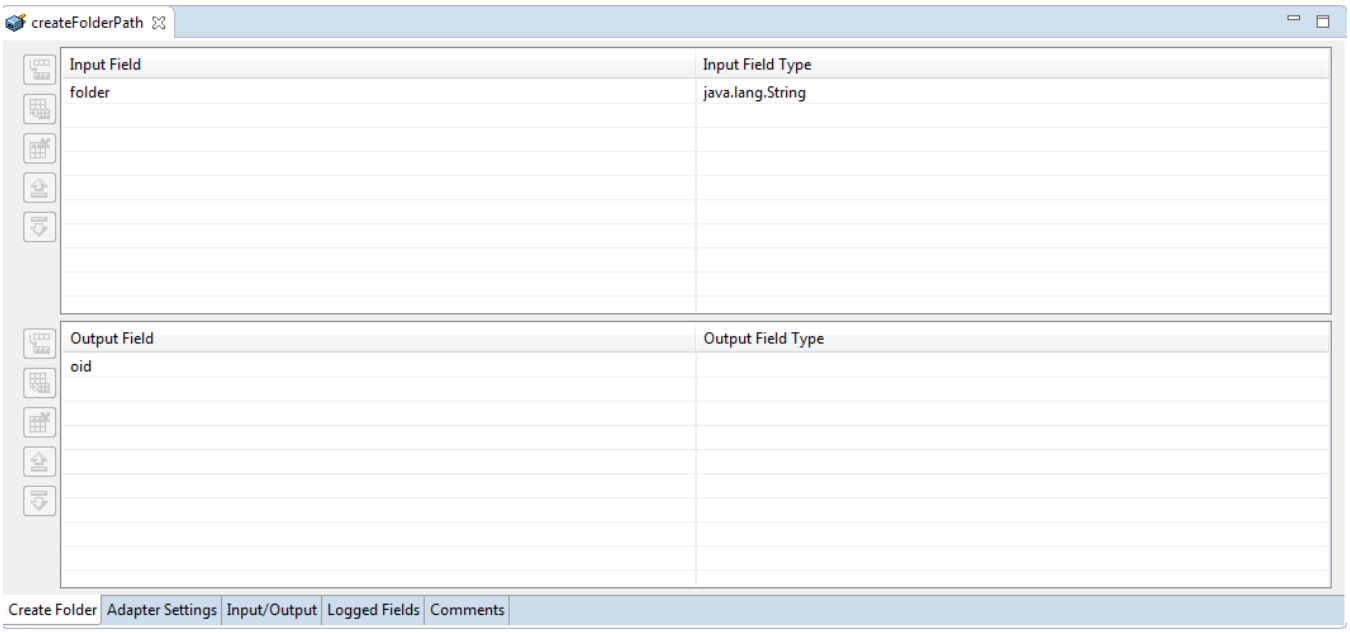
Folders unlike cabinets are just a way to structure the document locations where cabinets are the top level objects and also specify where in the storage system the physical objects
are to be placed.
It is optional to the have folders icon. If you create a document without specifying its folder path, then it is created by default in the user's home cabinet.

6.7.2 Configure the service

Create the service as described in create adapter services by naming it `createFolderPath` and choose the **Create Folder Path** service template**.**



In the Software AG Designe, the result appear as follows:



The screenshot depicted below provides you with the following options:

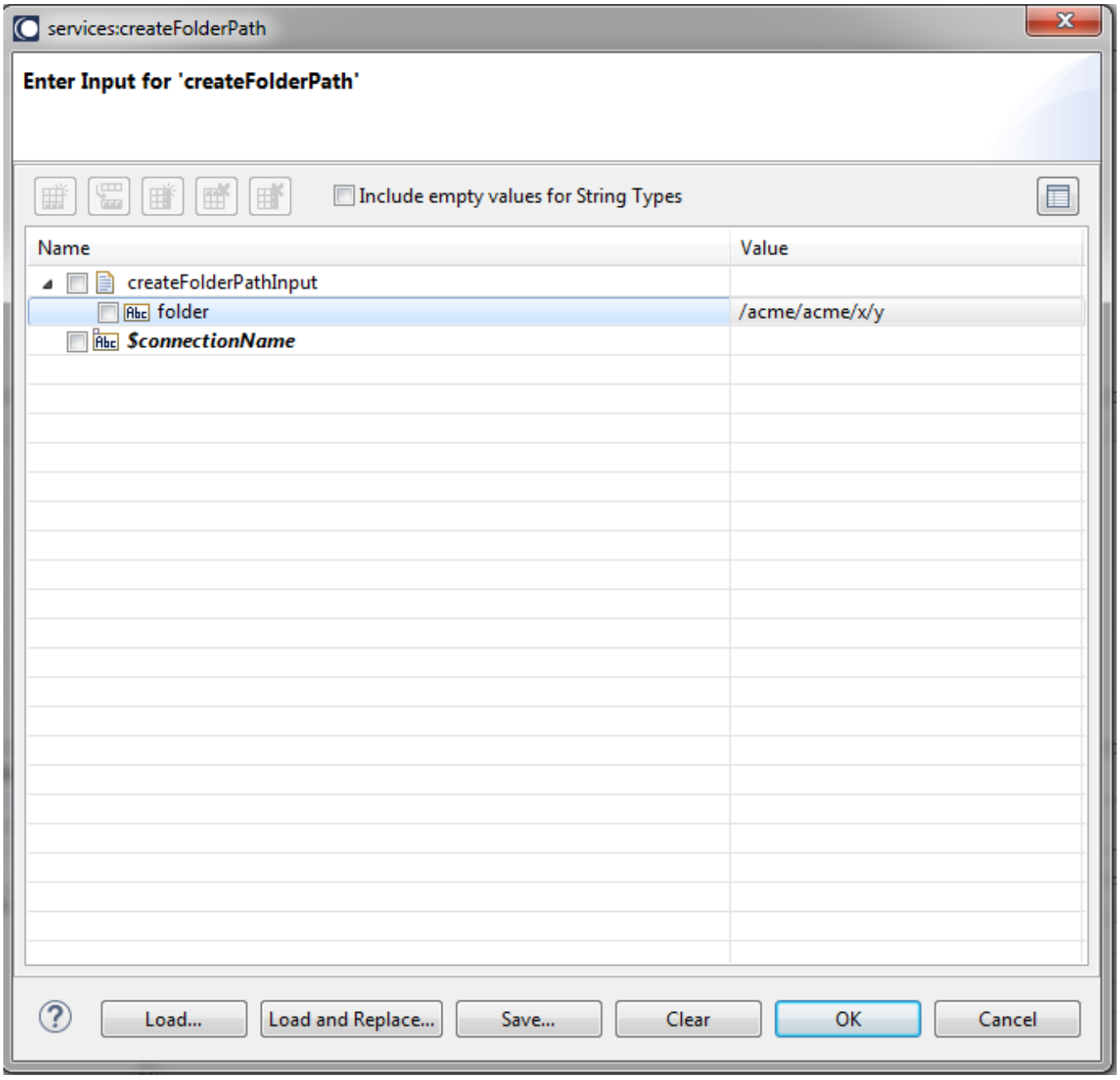
Configuration Tab	Input	Description
Create Folder	--	As we only have to specify the folder path to create there is no additional input required and everything we need is in the input/output signature
Input/Output	Input	The only input is a folder path (attribute folder) like /acme/acme/employee where the first part of the path is the cabinet under which to create the folder and the rest is the complete folder path to be created under the cabinet.
	Output	The output of the service is an object identifier (attribute oid) of the created (or existing) folder

If the folder already exists with the cabinet then only its `oid` is passed back and no error is thrown.

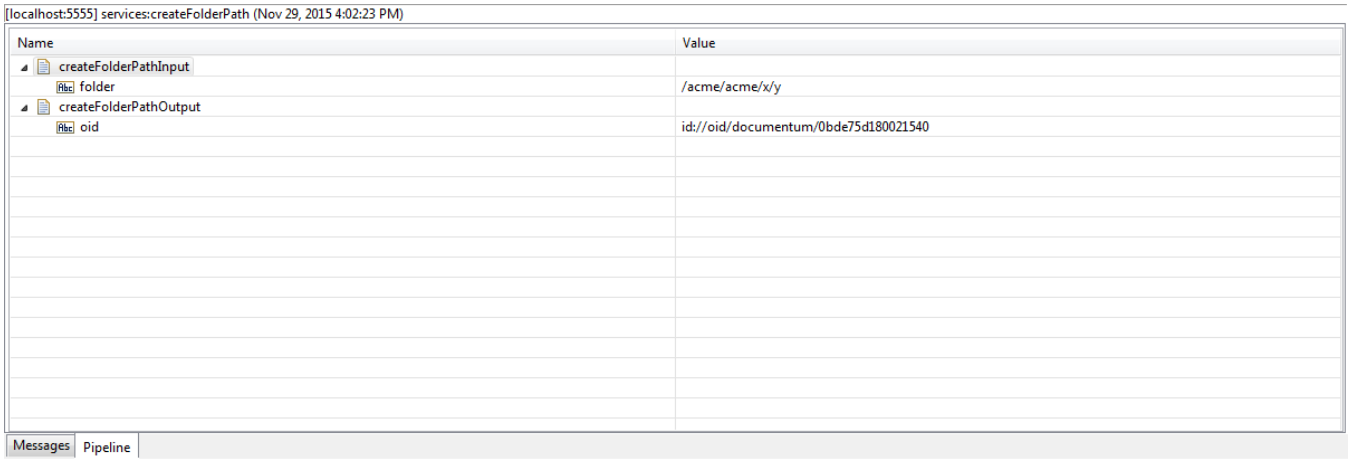
Note: You can specify arbitrary long folder paths and Documentum will create the complete path (that maybe contains a lot of subdirectories that all do not exist).

6.7.3 Execute the service

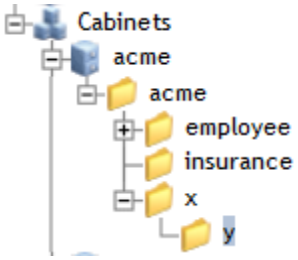
To run this service, Right-click **Run as** -> **Run as Service** and provide a folder path to be created such as, /acme/acme/x/y



The output will appears as follows:



After the service execution, the folders will also be visible in Documentum webtop.



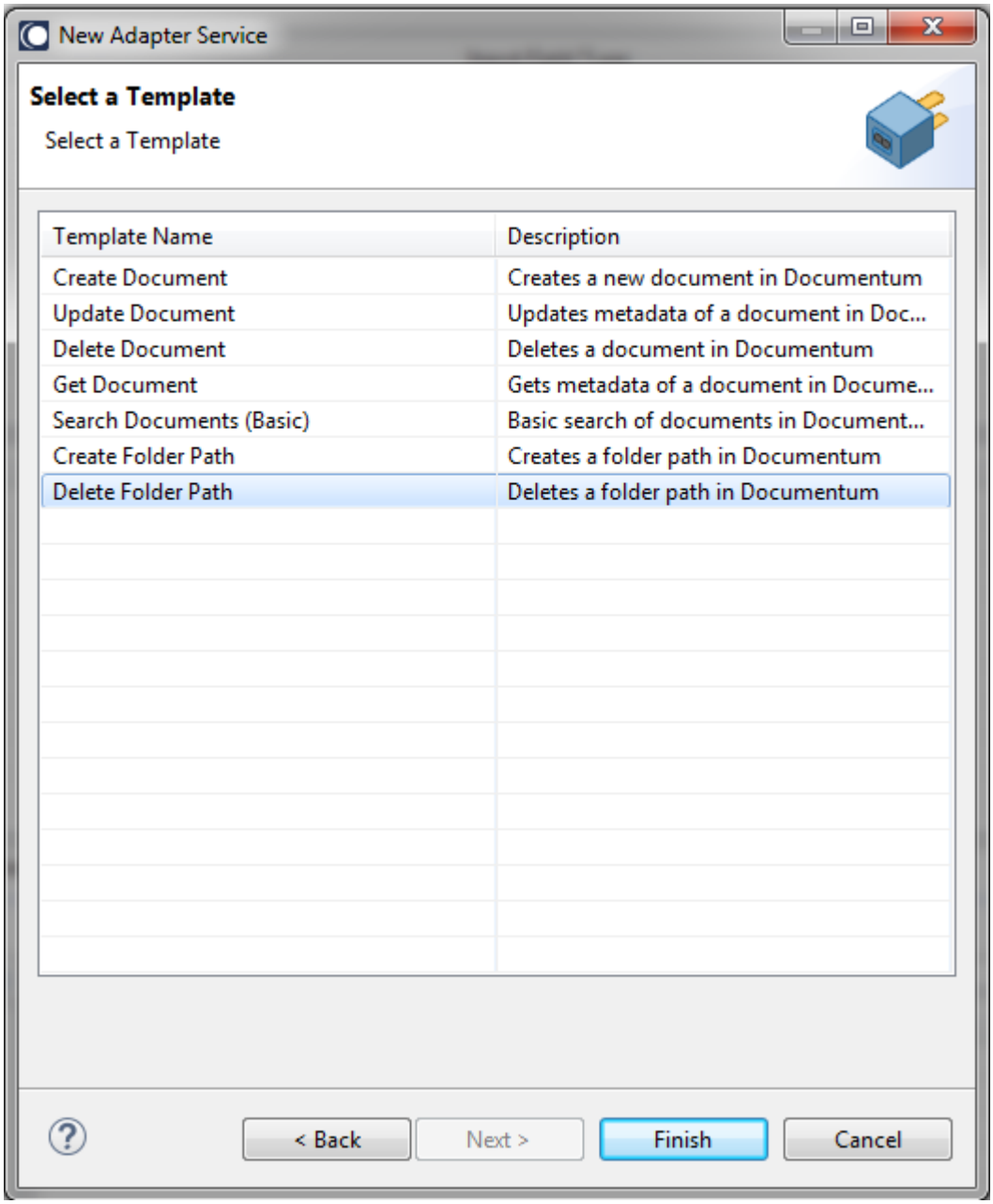
6.8 Delete Folder Path

6.8.1 Description

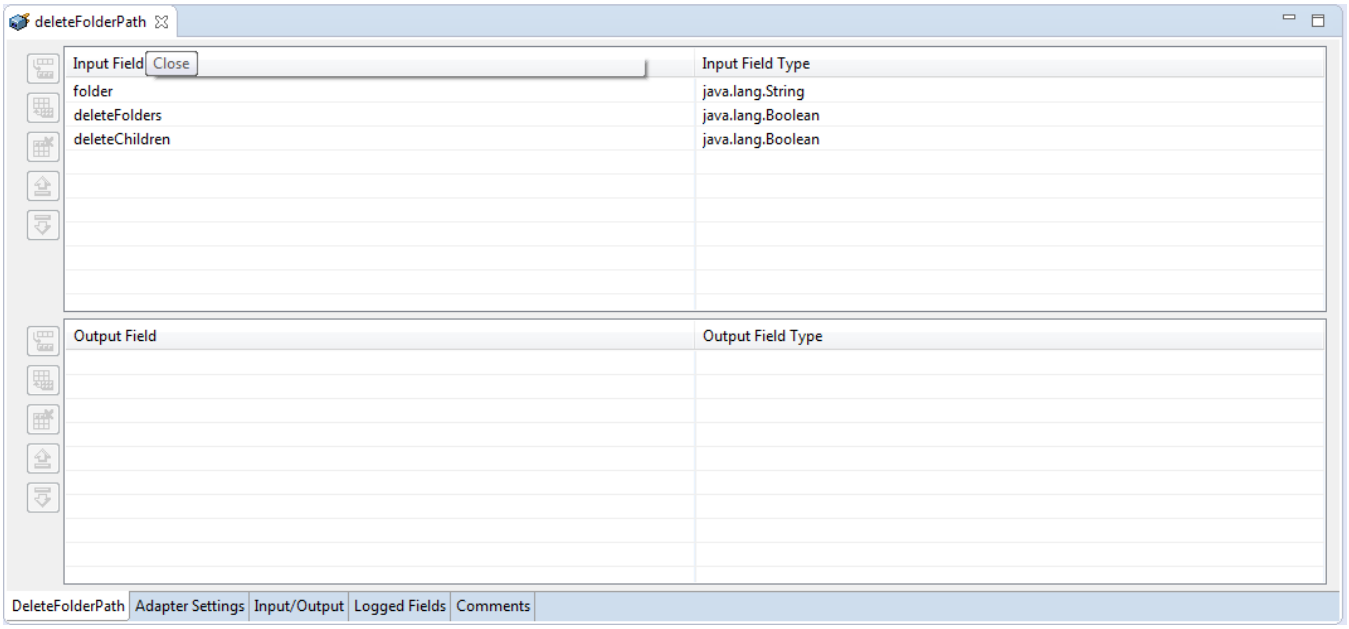
Using this service you can delete a folder path in Documentum.

6.8.2 Configure the service

Create the service as described in create adapter services by naming it `deleteFolderPath` and choose the **Delete Folder Path** service template.



In the Software AG Designer the result will look as follows:

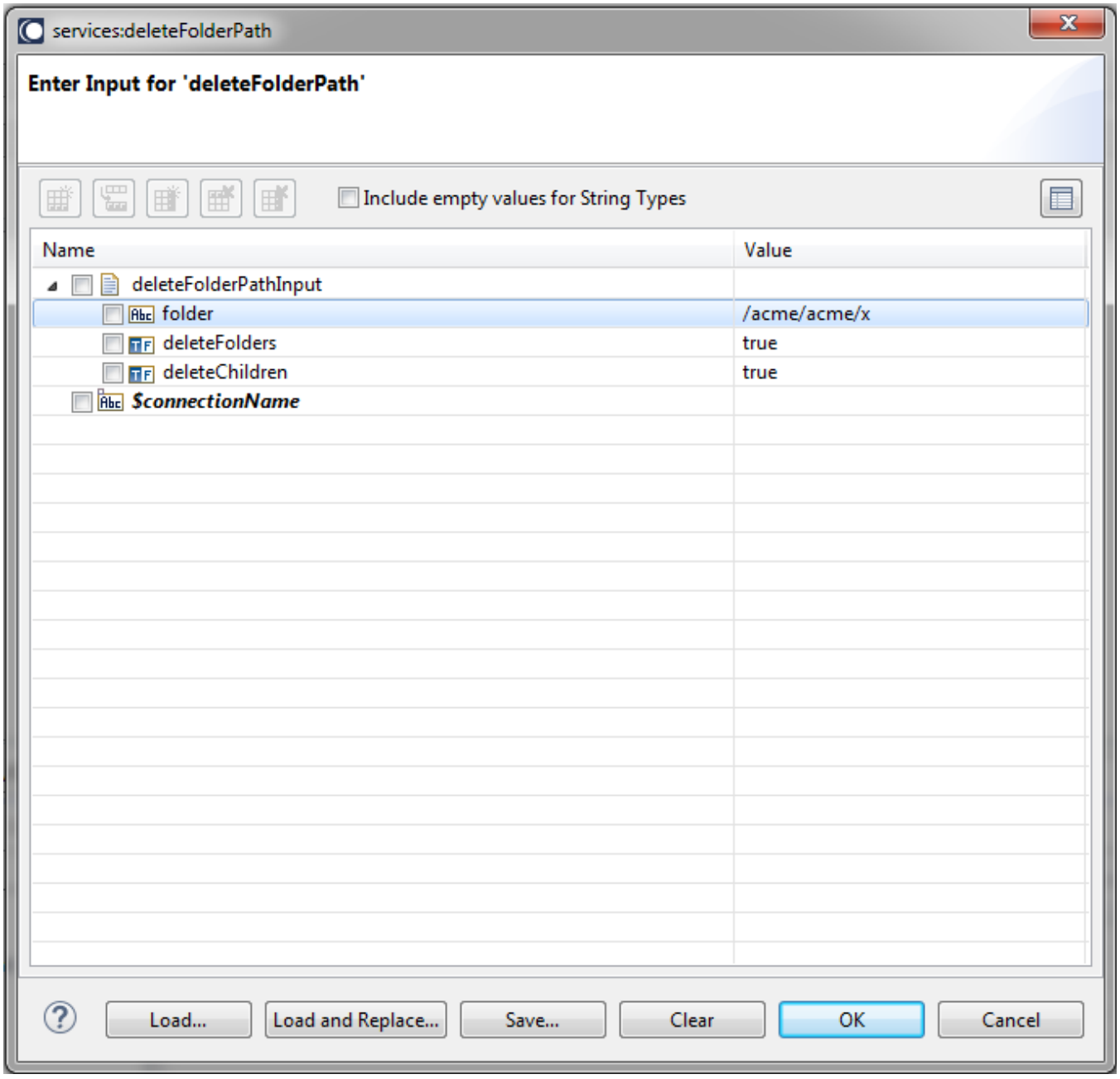


The screenshot provides you with the following options:

Configuration Tab	Input	Description
Delete Folder	Folder	The folder path to delete
	deleteFolders	If true, deletes all folders under a folder specified in folder. This setting does not specify whether none folder objects that are linked into other folders are deleted from the repository.
	deleteChildren	sets property 'deepDeleteChildrenInFolders' with the Documentum API, see Documentum API for details on it
Input/Output	Input	A tree can be, • folder • deleteFolders • deleteChildren
	Output	No output provided, if the peration fails an error is thrown

6.8.3 Execute the service

To run this service do a Right-click, **Run as** -> **Run as Service** and specify a folder path. Example, the one we used in Create Folder Path Service) and set `deleteFolders` to true to delete all sub folders.



Execute the call and if you check Documentum webtop for the parent folder */acme/acme*, you will see that the folders beginning with root x have been deleted as follows:

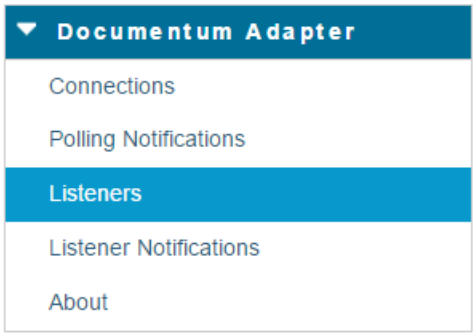
7 Adapter Notifications

Adapter notifications provide a powerful model for detecting and processing events that occur in the Documentum repository.

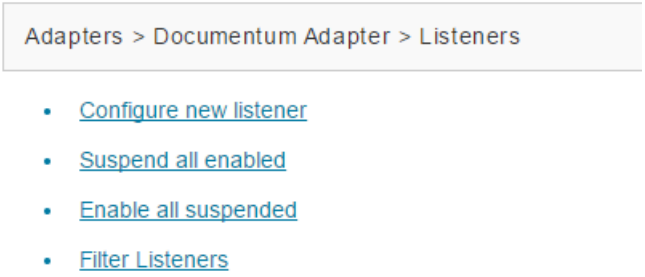
7.1 Adapter listener

If you have installed our add-on in Documentum repository and provided webSocketUrl in your connection, you will be able to configure a Listener.

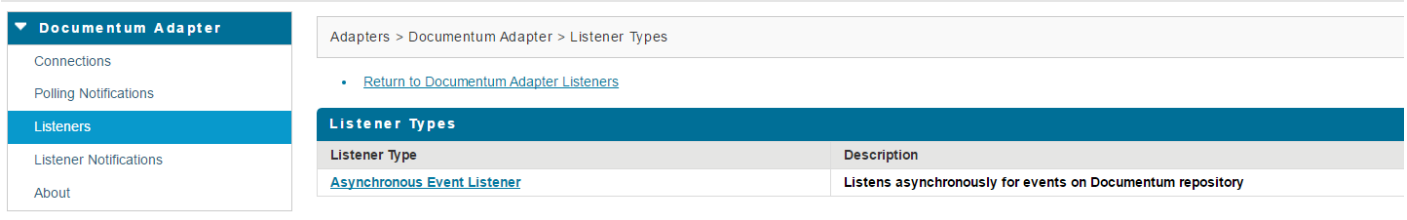
First, you must create a new Listener. Go to Adapters > Documentum adapter > Listeners,



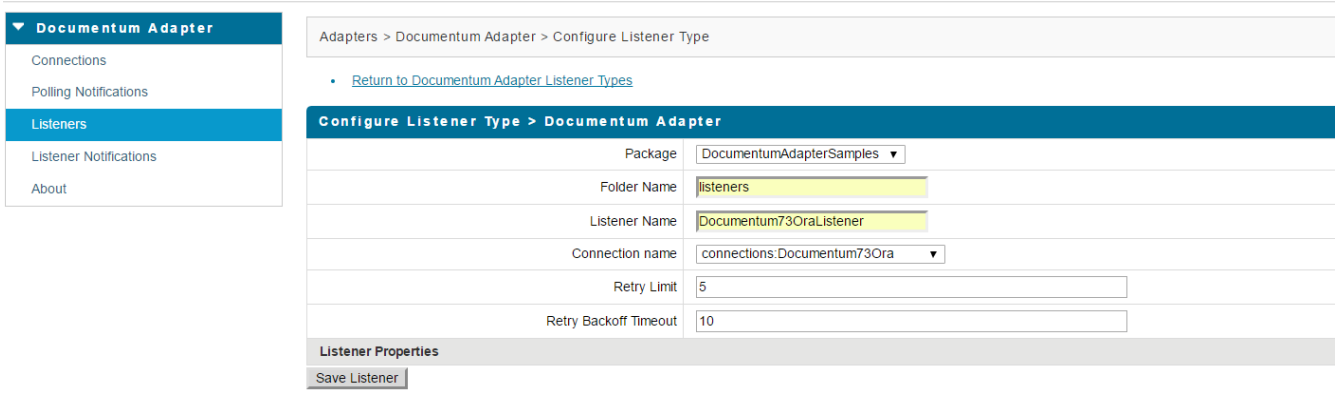
Click on “Configure new listener”



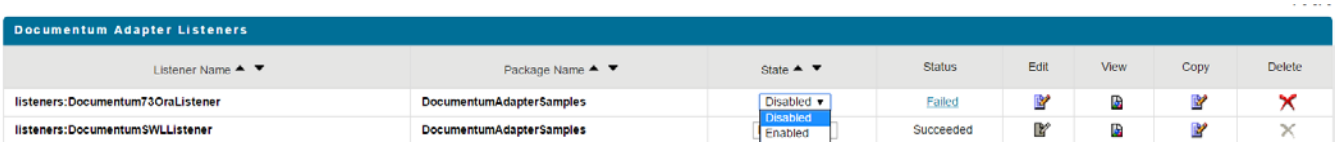
Select “Asynchronous Event Listener”



Select package and folder where you want the new listener to be created, provide listener name and select connection that you want to use for your listener.

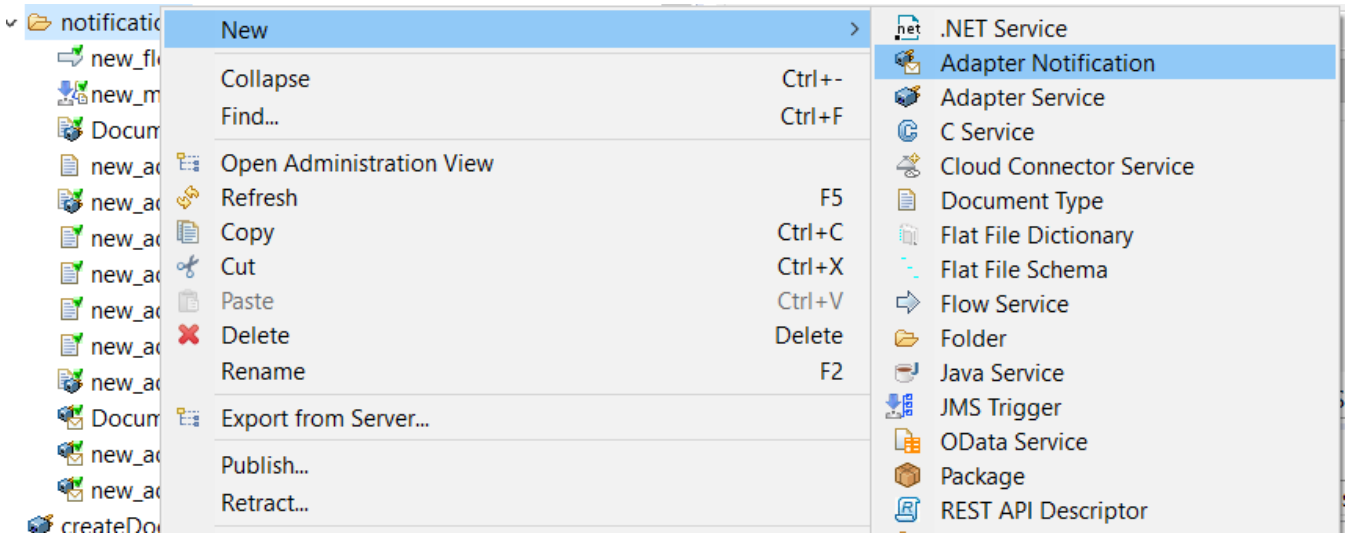


After new adapter listener is created, you must enable it.

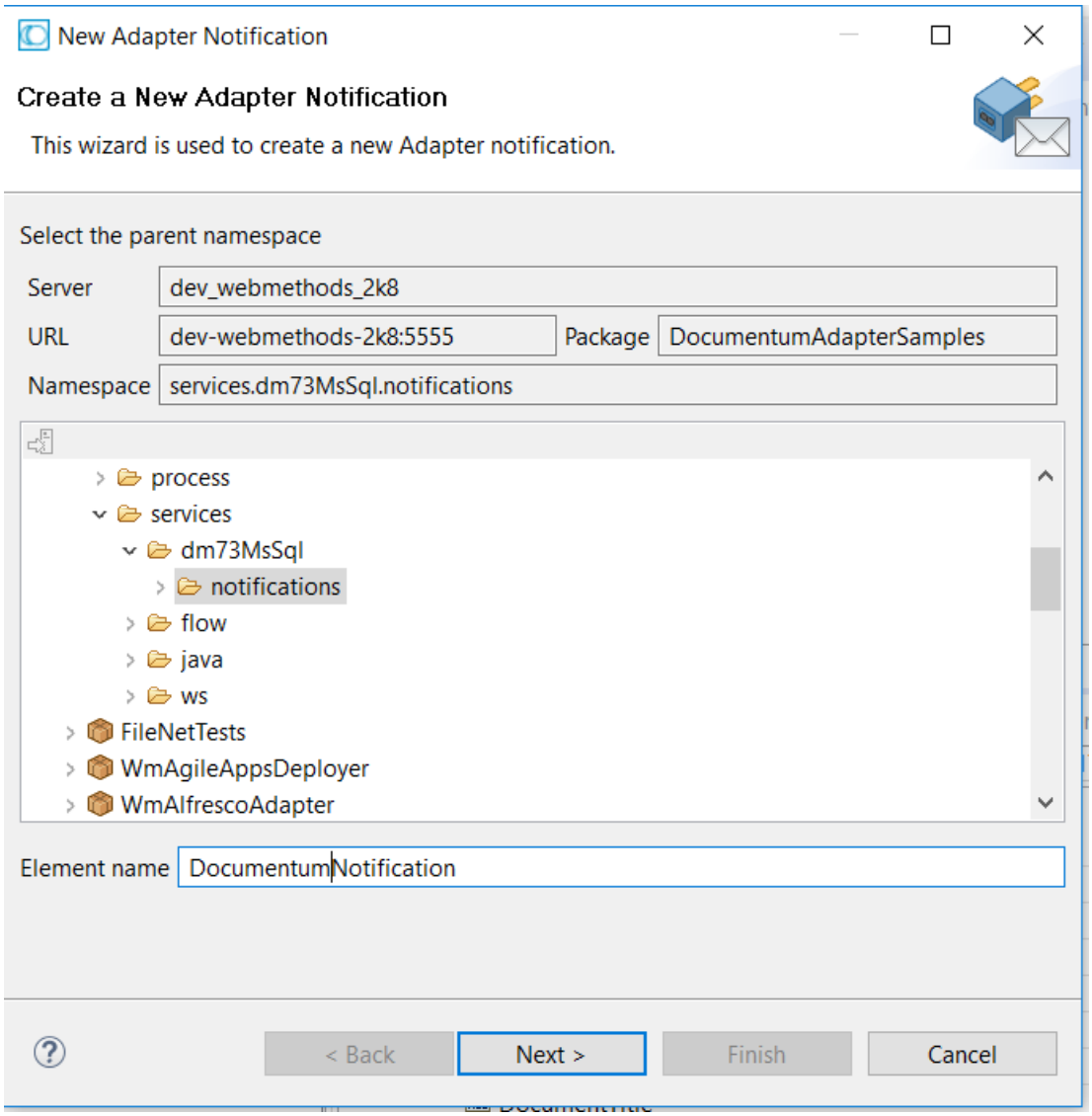


7.2 Create and configure notifications

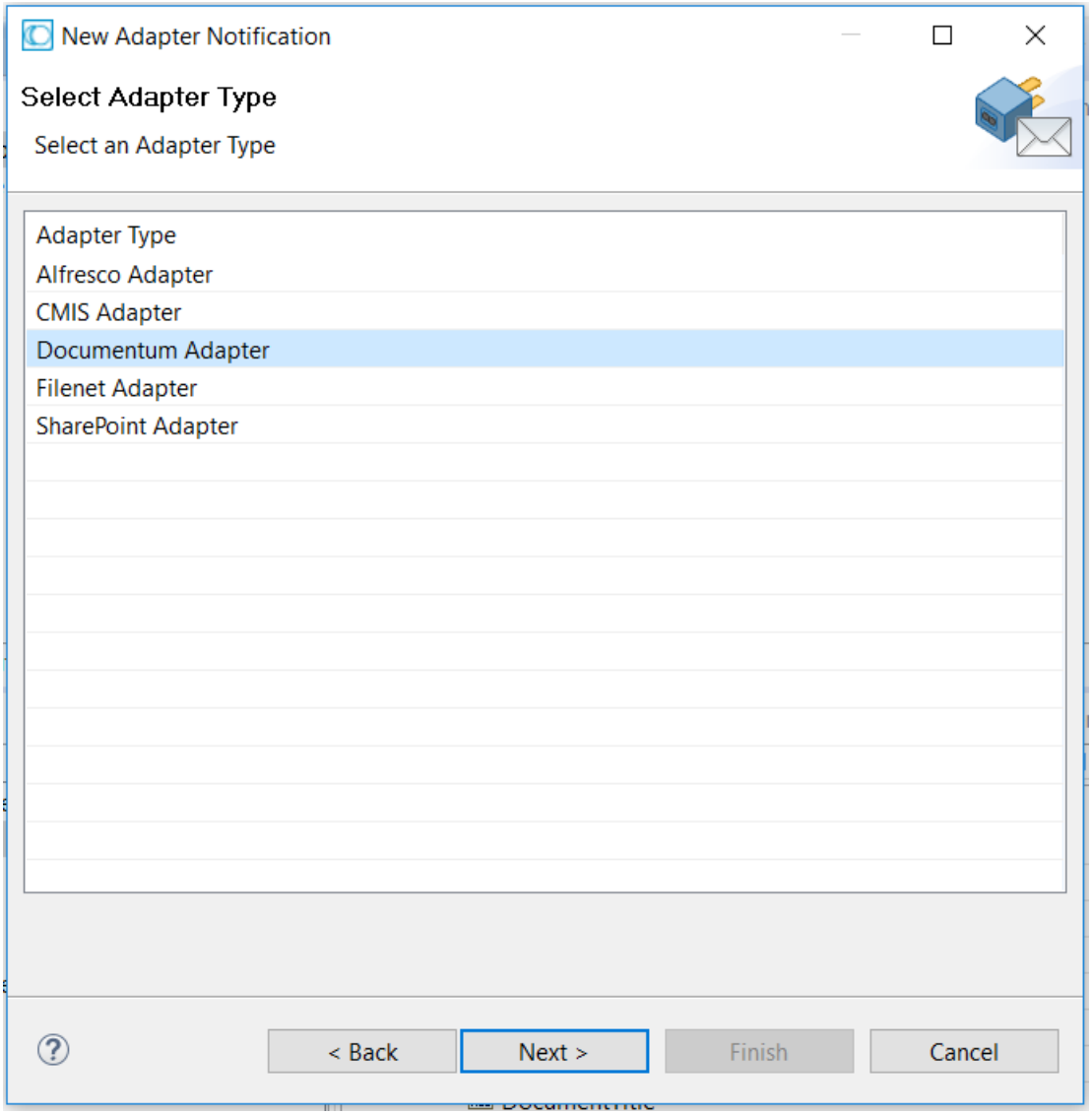
After the listener is enabled, you can create new Adapter Notification using Designer.



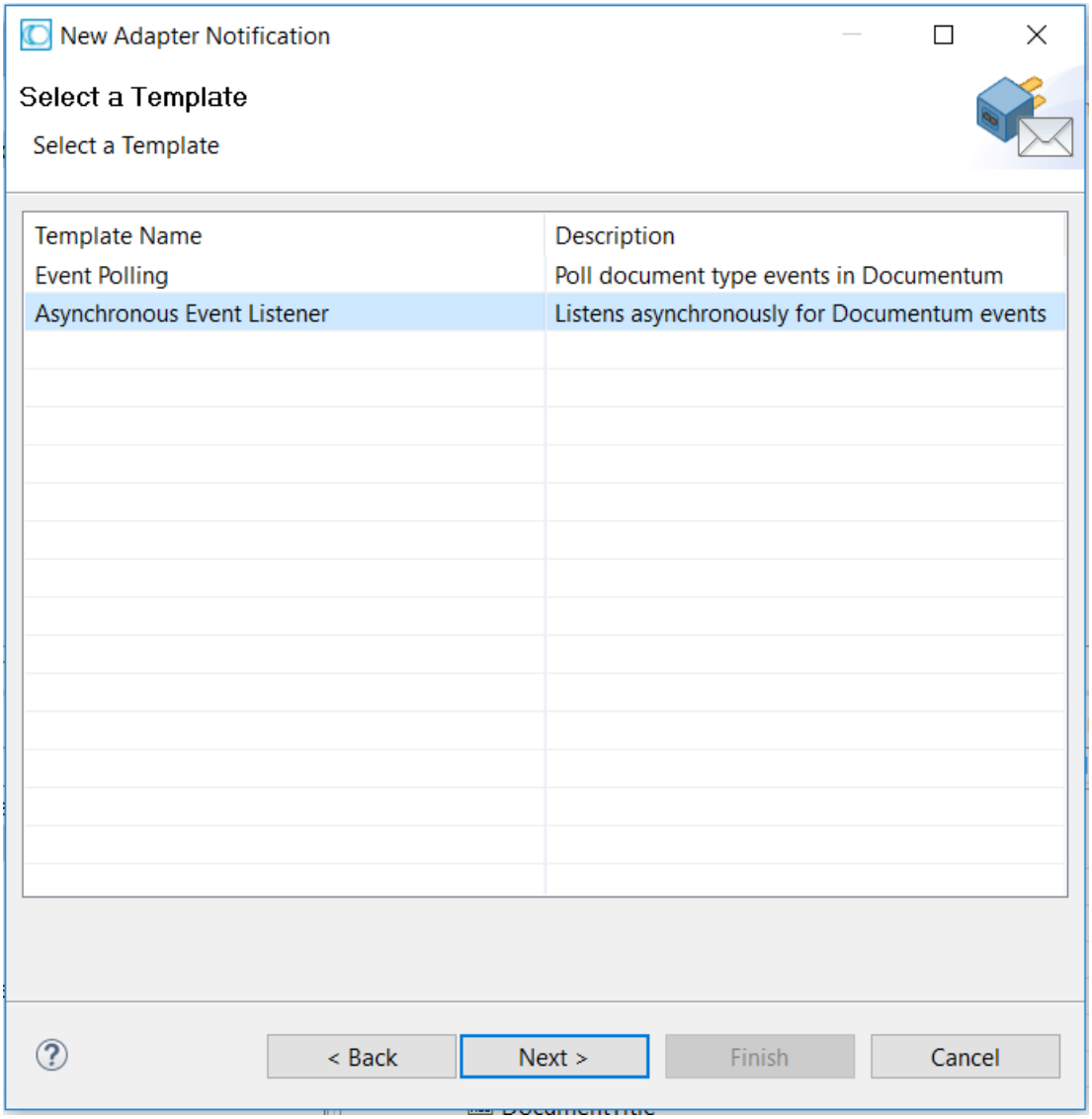
Select folder where you want new adapter notification to be created and provide name of your adapter notification.



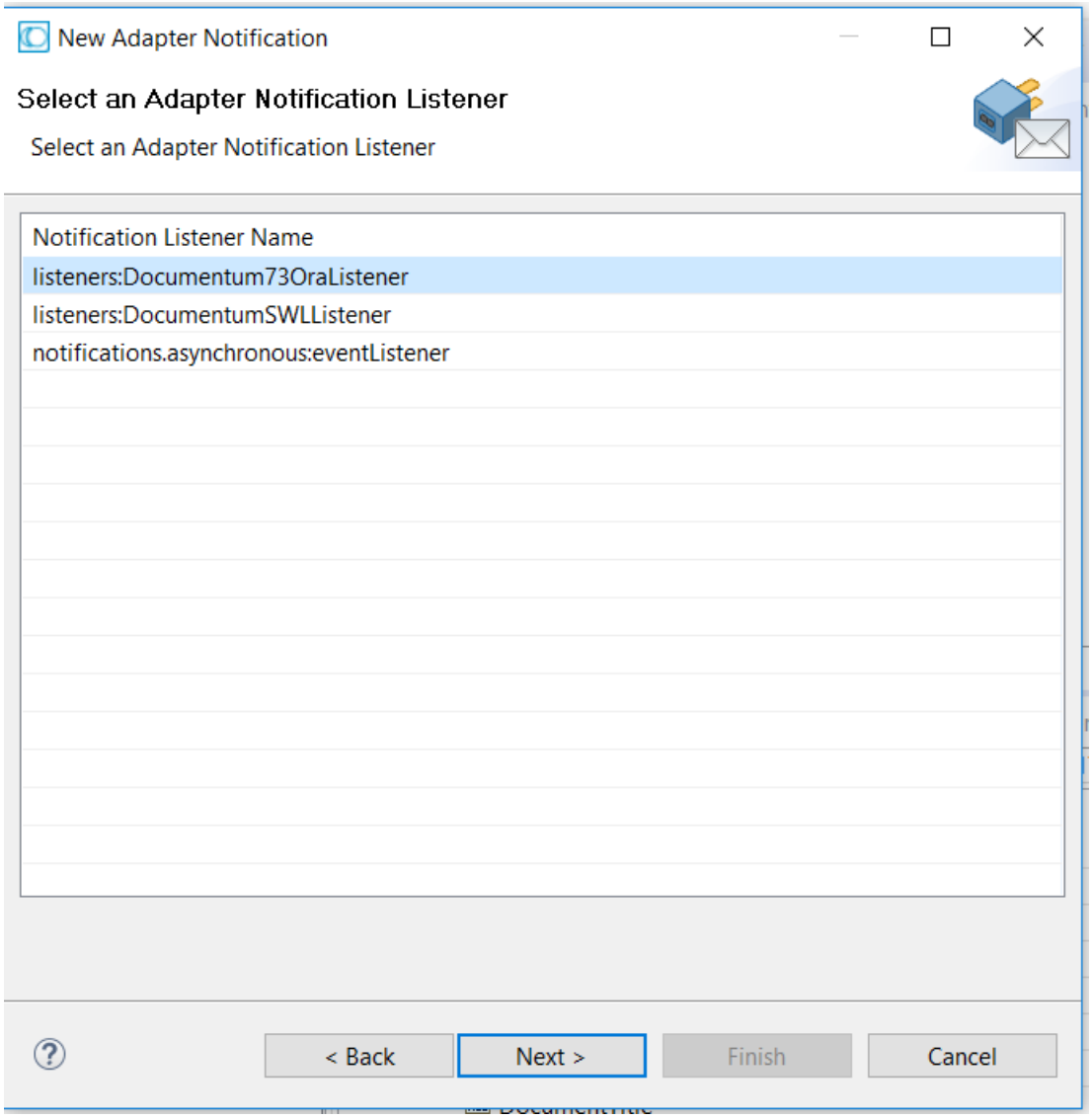
Select “Documentum Adapter” from list of available adapters.



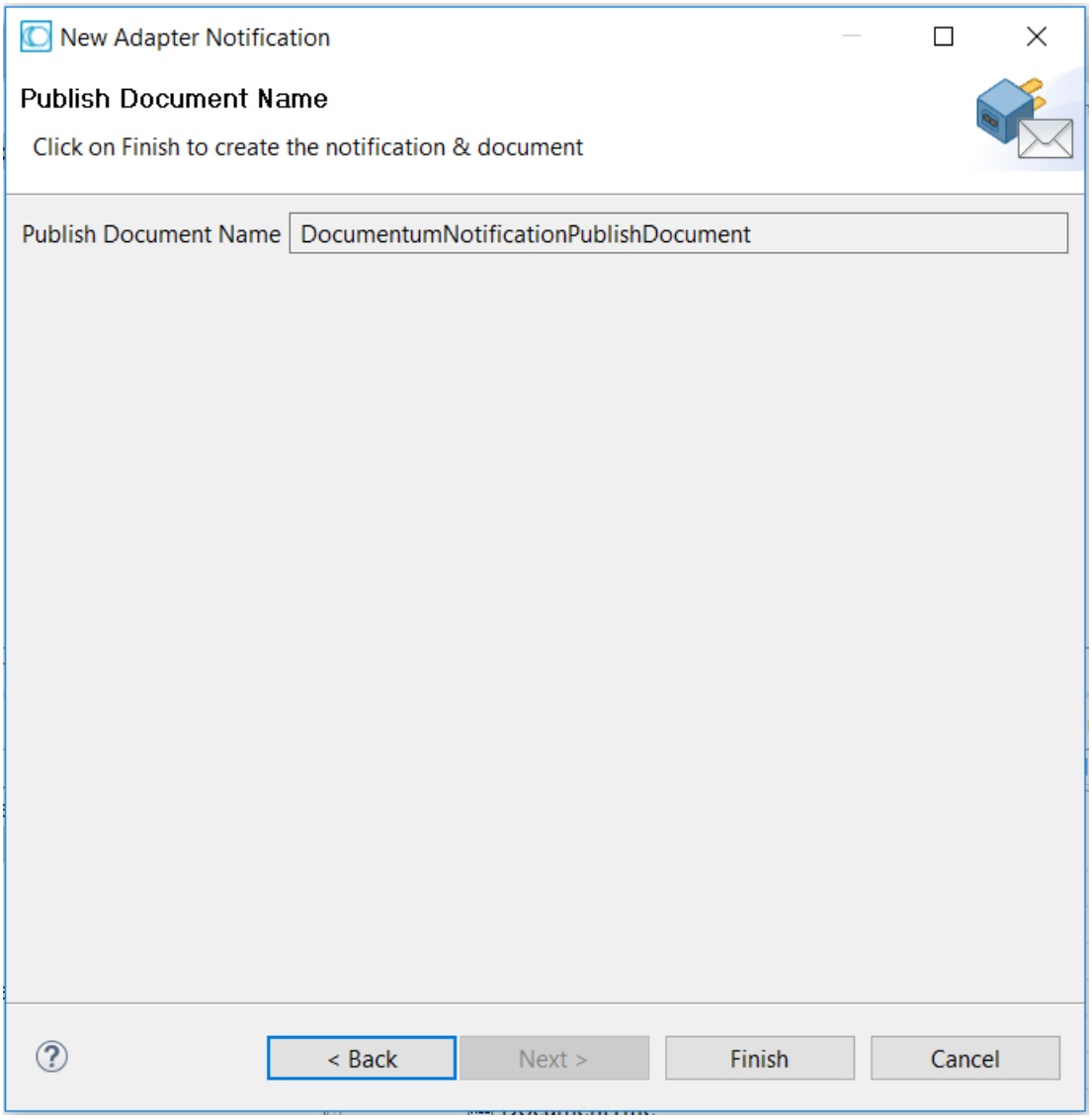
Select “Asynchronous Event Listener” from available notification templates.



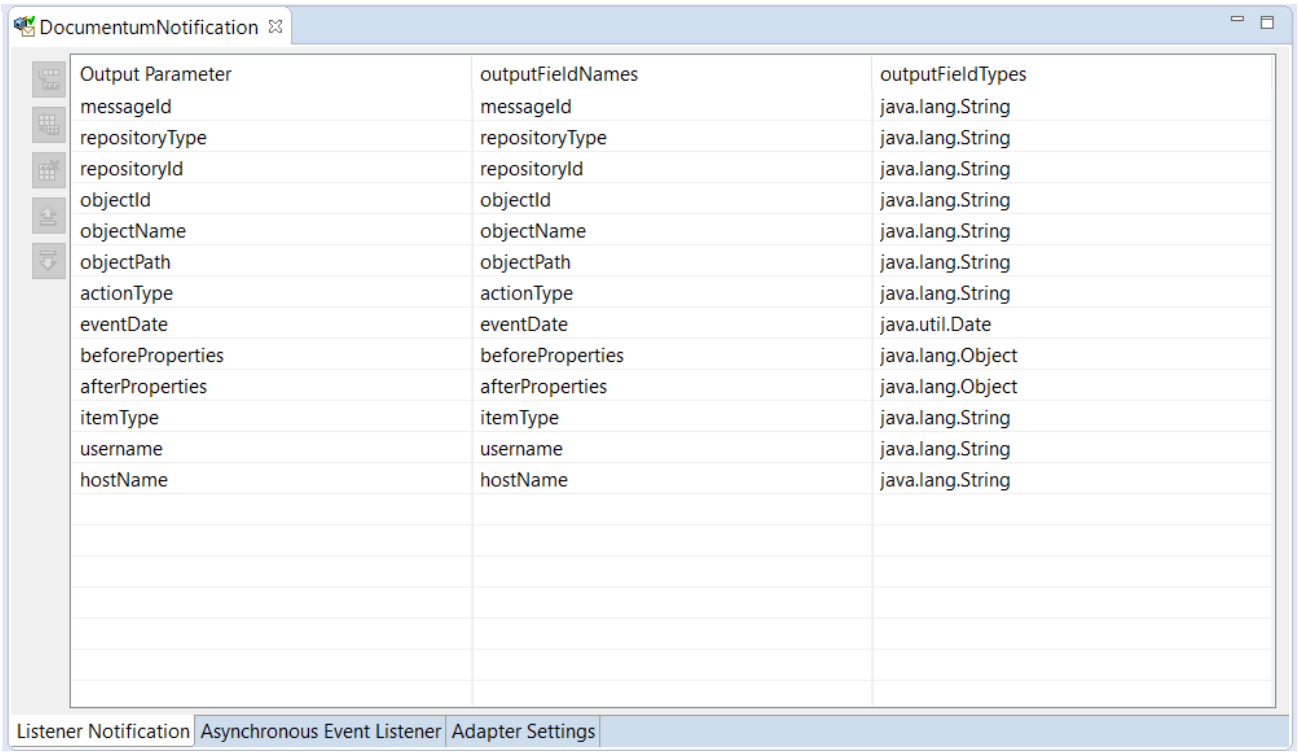
Select the listener that you have created in first step.



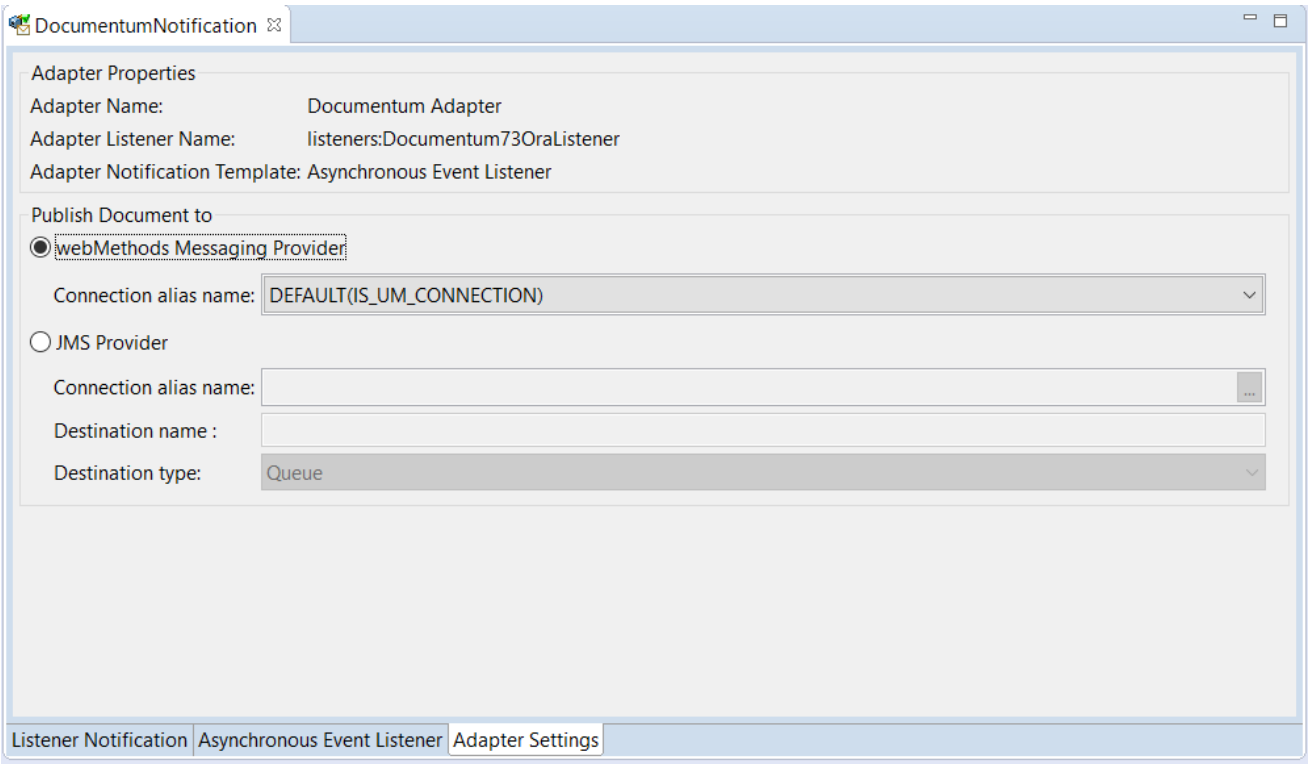
Click “Finish” button to finish creation of notification and document. Name of the document cannot be changed.



In next screen, you can see output fields and their corresponding types.



In “Adapter Settings” tab you can select messaging provider, destination name and destination type.



These steps enable saving documents published by Adapter Notifications to the messaging system. In order to process published documents, you can use webMethods Messaging Trigger or JMS Trigger, depending on the type of messaging provider you have selected in previous step.

You can use IS Administrator UI to enable or disable adapter notifications. In column “Enabled” click on No/Yes to enable/disable notifications.