

Developing OTDs for Oracle® Java CAPS Communication Adapters

Copyright © 2009, 2011, Oracle and/or its affiliates. All rights reserved.

License Restrictions Warranty/Consequential Damages Disclaimer

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

Warranty Disclaimer

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

Restricted Rights Notice

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

U.S. GOVERNMENT RIGHTS

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

Hazardous Applications Notice

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

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group in the United States and other countries.

Third Party Content, Products, and Services Disclaimer

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

Contents

Developing OTDs for Communication Adapters	5
Understanding the HTTPS OTD	5
HTTPS Client OTD	5
HTTPS Server OTD	6
Creating a New COM and DCOM OTD	10
▼ To Create a COM OTD	10
Relaunching OTDs	15
▼ To Relaunch an Existing OTD	16
File Adapter Components	16
File OTD Operations	16
SNA Object Type Definitions (OTDs)	17
▼ To associate the standard SNA Adapter OTD to a new Java Collaboration:	17

Developing OTDs for Communication Adapters

The following sections provide information and instructions on how to develop OTDs for communication adapters.

- “Understanding the HTTPS OTD” on page 5
- “Creating a New COM and DCOM OTD” on page 10
- “Relaunching OTDs” on page 15
- “File Adapter Components” on page 16
- “SNA Object Type Definitions (OTDs)” on page 17

Batch Adapter OTDs are described in a separate document, *Oracle Java CAPS Adapter for Batch User's Guide*.

The HL7 Message Library and OTDs are described in a separate document, *Oracle Java CAPS Message Library for HL7 User's Guide*.

Understanding the HTTPS OTD

The following topics describe the functionality of the HTTPS OTD:

- “HTTPS Client OTD” on page 5
- “HTTPS Server OTD” on page 6

HTTPS Client OTD

The HTTPS OTD is specific to the HTTPS Adapter. It is used as an inbound or outbound OTD in a Collaboration.

OTDs have a hierarchical data structure composed of fields containing methods and properties. The top root element of the OTD is the HTTPClientApplication interface, and the fields underneath contain Java methods. You can use these Java methods to create Business Rules that specify the HTTP message format and invoke messaging to and/or from an HTTP server.

To access other Java classes and methods, you can use the Collaboration Editor (Java) to utilize the entire contents available for HTTPClientApplication.

HTTP OTD Method Descriptions

The HTTP OTD includes the following methods used in HTTP data exchange:

get	The method called in the Collaboration (Java) to send an HTTP get request to an HTTP server.
post	The method called in the Collaboration (Java) to send an HTTP post request to an HTTP server.
getRequest	The method called in the Collaboration (Java) for other “request” related helper methods, such as to set the URL, to add properties, and so on.
getResult	The method called in the Collaboration (Java) for other “respond” related helper methods, such as, to obtain the respond code, respond result, text result, and so on.

For more information on methods available in the HTTP OTDs, see the HTTPS Adapter’s Javadoc.

HTTPS Server OTD

The HTTPS Server input OTD has two nodes, Request and Response. The Request node contains the data that the HTTPS Server adapter receives from an HTTP client, while the Response node is used to set the HTTP response data that will be sent back to the HTTP client.

FIGURE 1 Input Server OTD

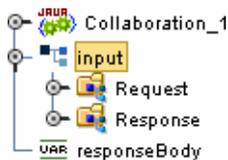


FIGURE 2 Input Server Request Node

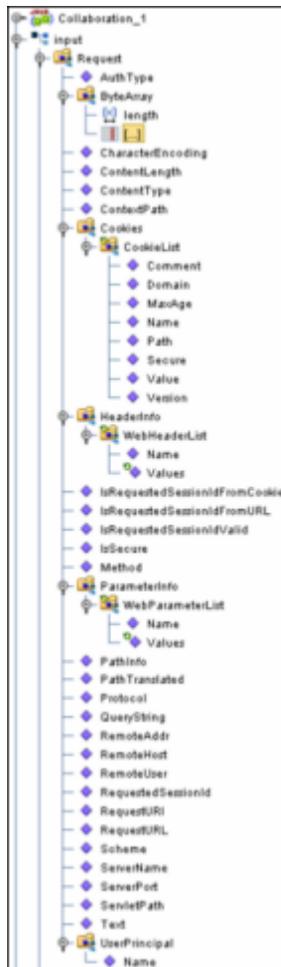
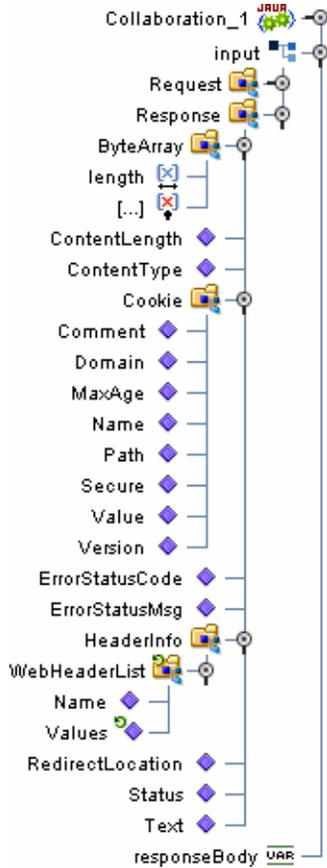


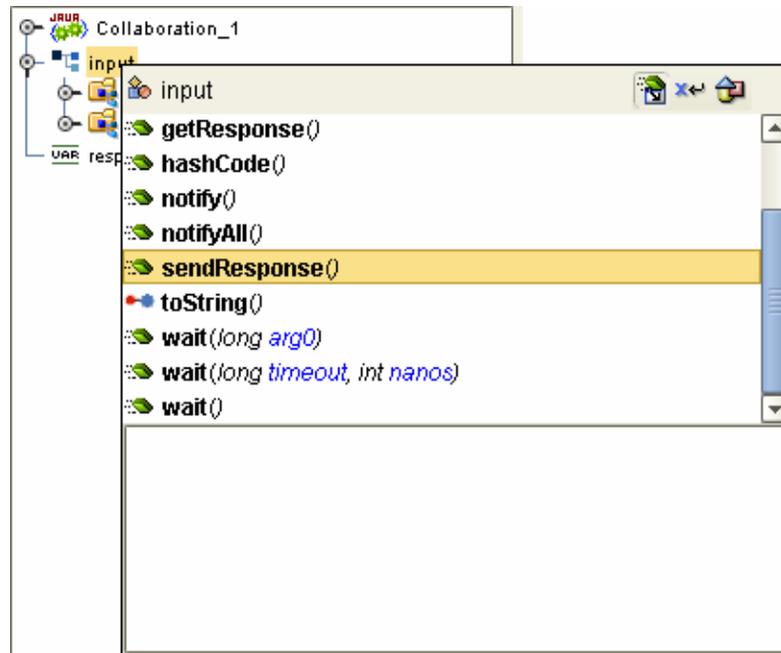
FIGURE 3 Input Server Response Node



Working With the Server OTD

Use the OTDs Request and Response nodes to build the logic in your HTTPS Collaborations. The HTTP response is not sent back to the HTTP client until `sendResponse()` method is called on the HTTP server input OTD.

FIGURE 4 sendResponse() Method

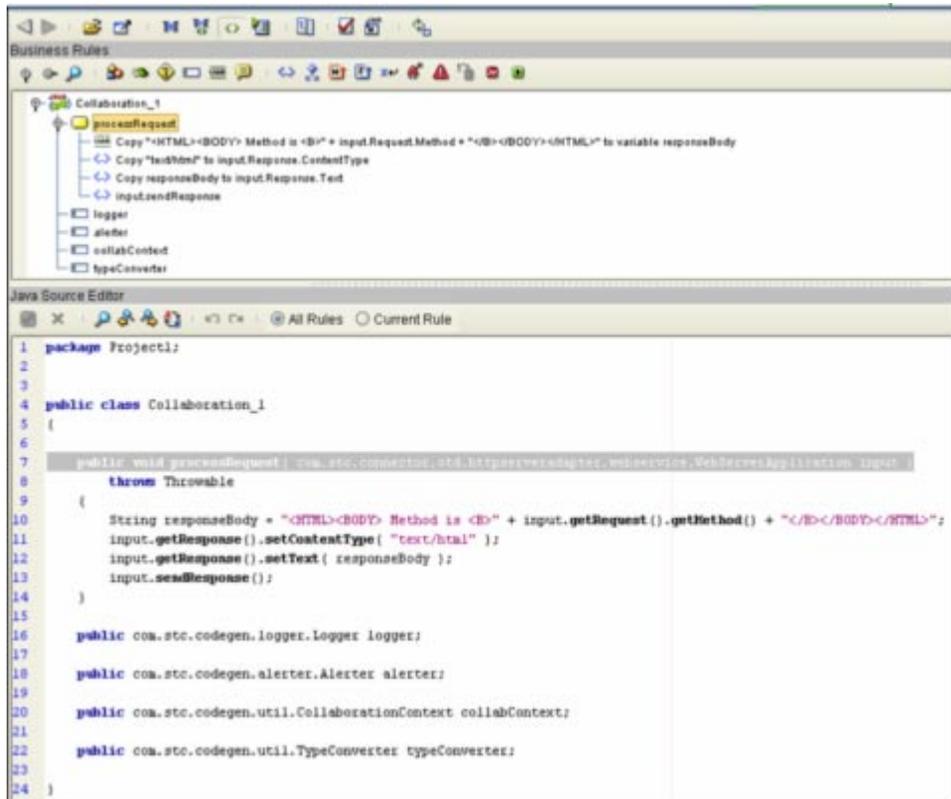


It is critical that you use this method to send the response back to the client. Otherwise, the client will wait indefinitely for the response. HTTP requires that a response be sent to the client whether the response is a valid application response or an application error response.

Collaboration Example

The following example shows a simple Java Collaboration that retrieves the HTTP method from the Request node through the Method property. This creates an HTML response indicating the HTTP method. This method is retrieved from the request, sets the ContentType property as “text/html” on the Response node. It then sets the Text property with the HTML response, and then calls the sendResponse() method. This is called on the HTTP server input OTD to send the constructed response to the HTTP client.

FIGURE 5 sendResponse() Example



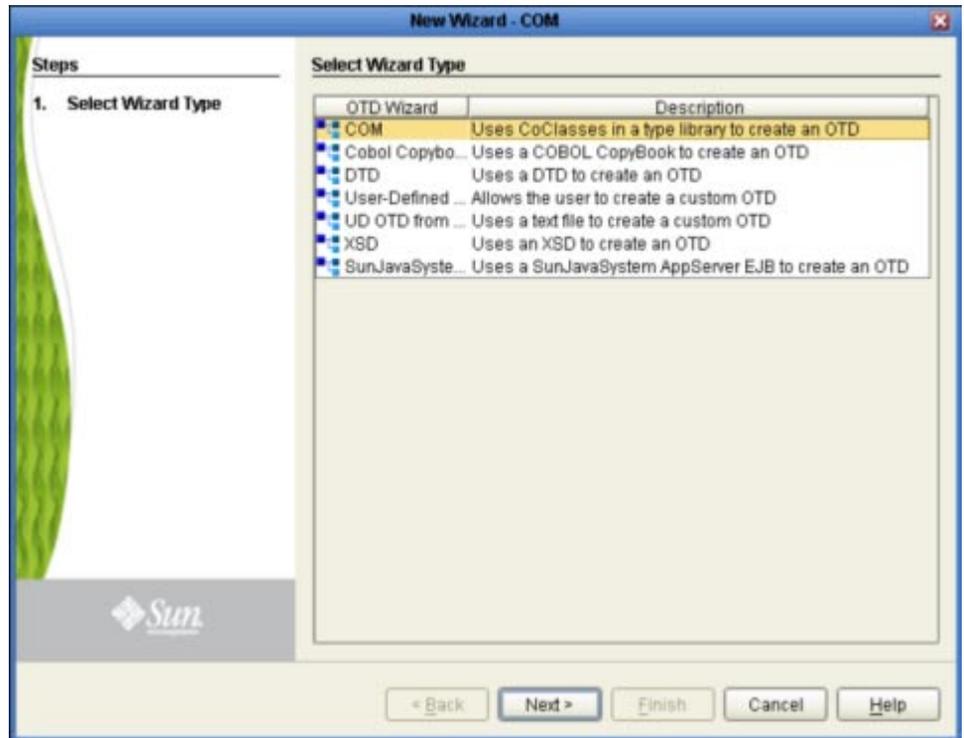
Creating a New COM and DCOM OTD

The COM OTD Wizard generates an OTD from a COM automation-compatible component's Type Library files. COM Type Library files describe the methods and properties exposed from an automation-compatible component. COM type libraries may have the file extension .tlb or .olb, however, most components typically embed the type library file in the DLL, OCX, or EXE file that contains the component.

▼ To Create a COM OTD

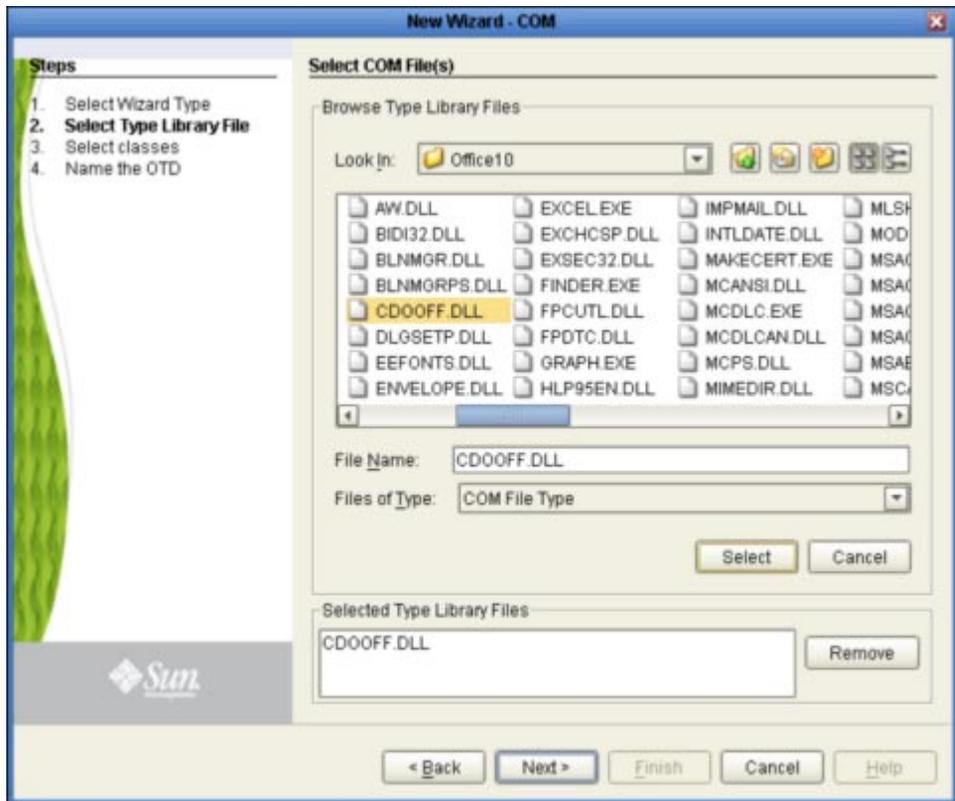
- 1 Right click your Project from the Projects window tree, point to New, and then select Object Type Definition.

- 2 Select the COM Wizard from the Select Wizard Type window of the New Object Type Definition Wizard and click Next).

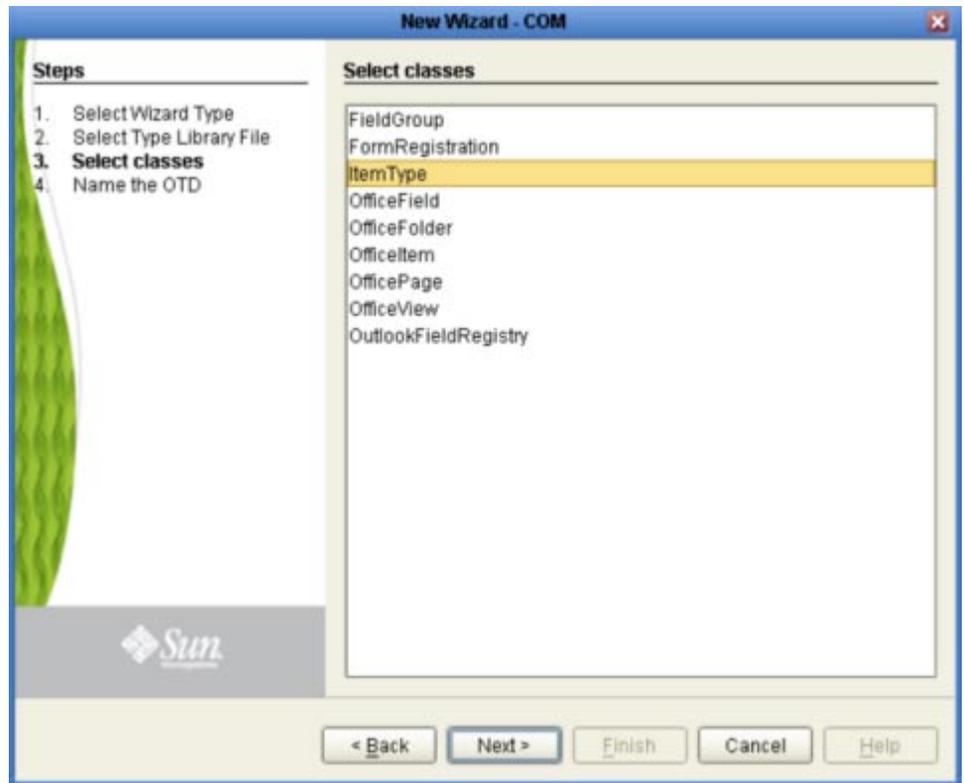


- 3 Browse to the directory that contains the type library file from which the OTD will be created. You can only select one type library file at a time.

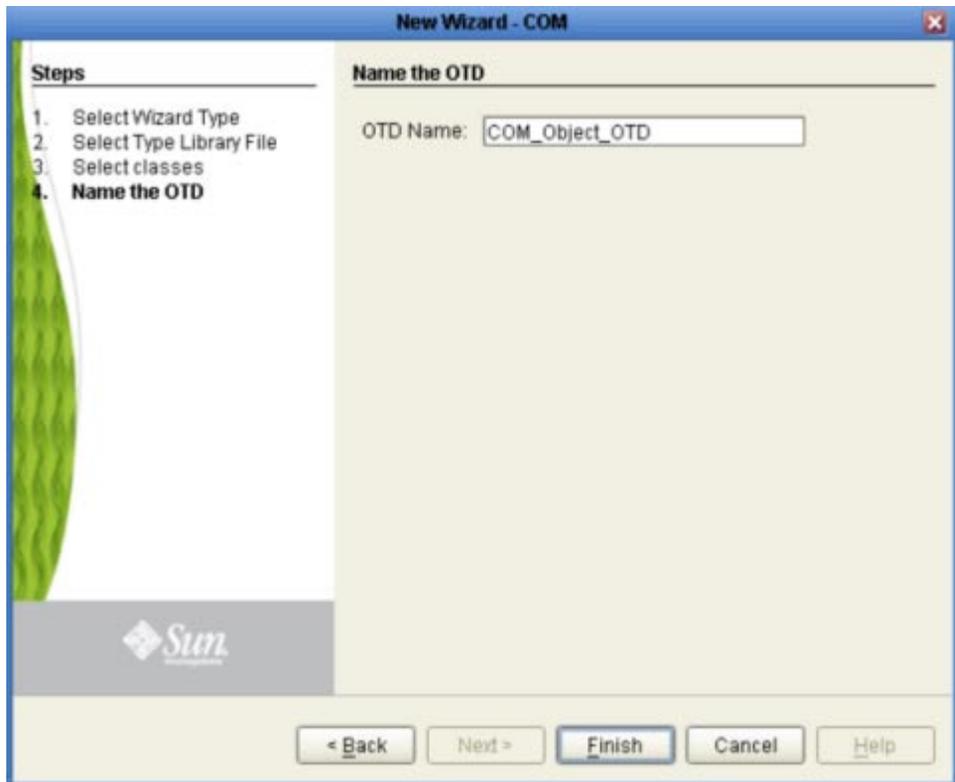
- 4 Select your type library file, click the Select button, then click Next.



- 5 Select one or more of the classes from the type library and click Next.



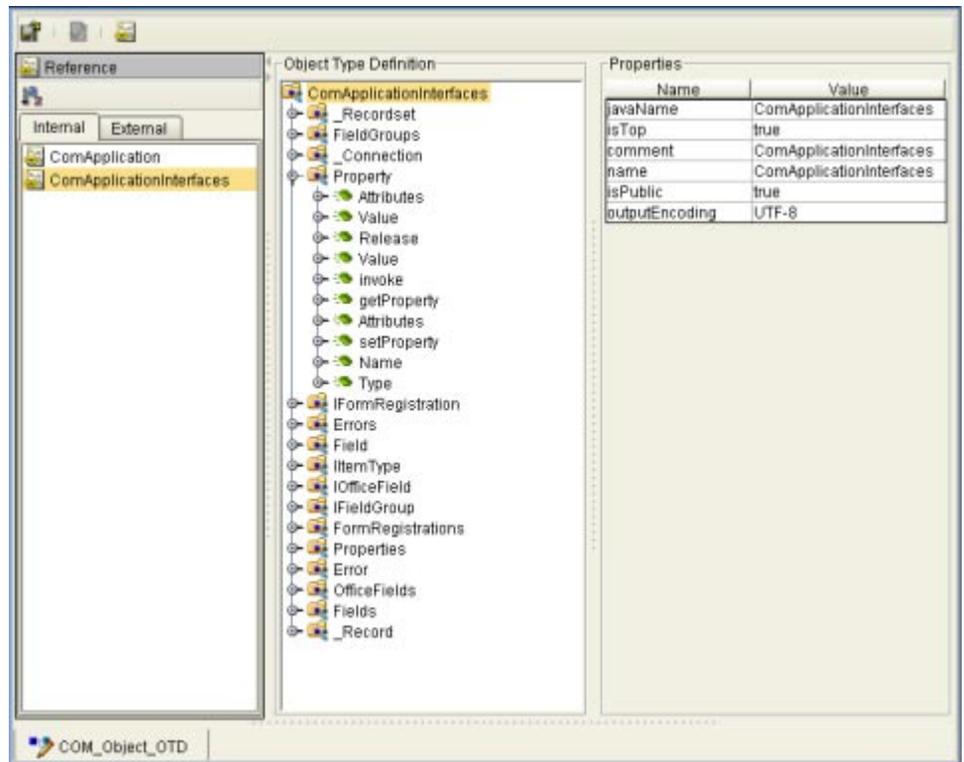
- 6 Enter a name for the new OTD in the OTD Name field and click Finish.



If any of the selected CoClasses contain a method with an unsupported data type, an Information box appears.

The Information box indicates that some methods were not created in the OTD, and provides the location of the generated “Skipped Methods” log. This log provides a report of any methods that were skipped when the OTD was created (this information is also written to the IDE log file). If this information box appears, click OK to acknowledge and close the Information box.

The OTD Editor appears displaying the new OTD.



The resulting OTD is now available for use in your Collaborations.

Relaunching OTDs

A single OTD can consist of many lines of metadata. When a change to the metadata occurs in an OTD, it does not have to be recreated from scratch. Using the Relaunch function allows the OTD to be rebuilt and saved under the same name, then relaunched back to the same Java collaboration or BPEL.

▼ To Relaunch an Existing OTD

- 1 **Right-click on the OTD in the Projects window.**
- 2 **Click Relaunch from the submenu.**
The Select Files Wizard opens.
- 3 **Enter the File Name (or Browse and Select) that you wish to be relaunched and click Next.**
- 4 **Continue with the Wizard as described when creating the OTD.**
- 5 **Click the Finish button to save the changes.**

When relaunching an OTD, an existing collaboration will not be affected if:

- New columns are added
- Deleted columns are not used in the original collaboration

Note – Validation will fail if existing collaborations are not modified when columns are renamed or deleted.

File Adapter Components

The File adapter is used in most of the sample Projects provided with Oracle Java CAPS and the Intelligent Adapters. The File adapter includes the following components that are unique to this adapter:

- **File Adapter Configuration Properties File:** The properties files for the File Adapter defines how the adapter connects with external systems as well as how data (messages) are handled. These properties are configured using the Properties Editor. For more information about the Batch Adapter properties files and the Properties Editor.
- **FileClient OTD:** The File OTD provides functionality for receiving or sending files and triggering a Business Process or Collaboration. Object Type Definitions (OTDs) map input and output message segments at the field level.

File OTD Operations

The following File Adapter OTD attributes or operations are used in both BPEL and JCD Projects:

- **receive** (inbound File Adapter)

- **write** (outbound File Adapter)

The **write** (output) Business Process operation has an additional input function provided to receive and handle incoming error exception messages.

SNA Object Type Definitions (OTDs)

Unlike most other Adapters, the SNA Adapter does not consist of an OTD wizard. OTD wizards typically facilitate the creation of a Collaborations that are used with Adapter projects. When an OTD wizard is available, a skeleton Collaboration is created to provide minimal functionality that you must modify to suit your application's needs. Without the OTD wizard, as in the case of the SNA Adapter, you must create your Collaborations completely from scratch.

▼ **To associate the standard SNA Adapter OTD to a new Java Collaboration:**

- 1 From the Projects window, right-click the targeted project.
- 2 Point to New and select Collaboration Definition (Java)
- 3 Complete steps 1 and 2 of the Collaboration Definition Wizard (Java).
- 4 Select the OTD to use in the new Collaboration by traversing the Look In drop-down box: CAP Components Library/Adapters/SNALU62.
- 5 Highlight the desired OTD name and click the Add button.
- 6 Optionally, modify the instance name of the OTD that will be used in the Collaboration.
- 7 Click the Finish button.

The new Collaboration that implements the SNA Adapter OTD is created. For details about the SNA Adapter methods that may be used with Collaborations for the, refer to the associated Javadoc.

