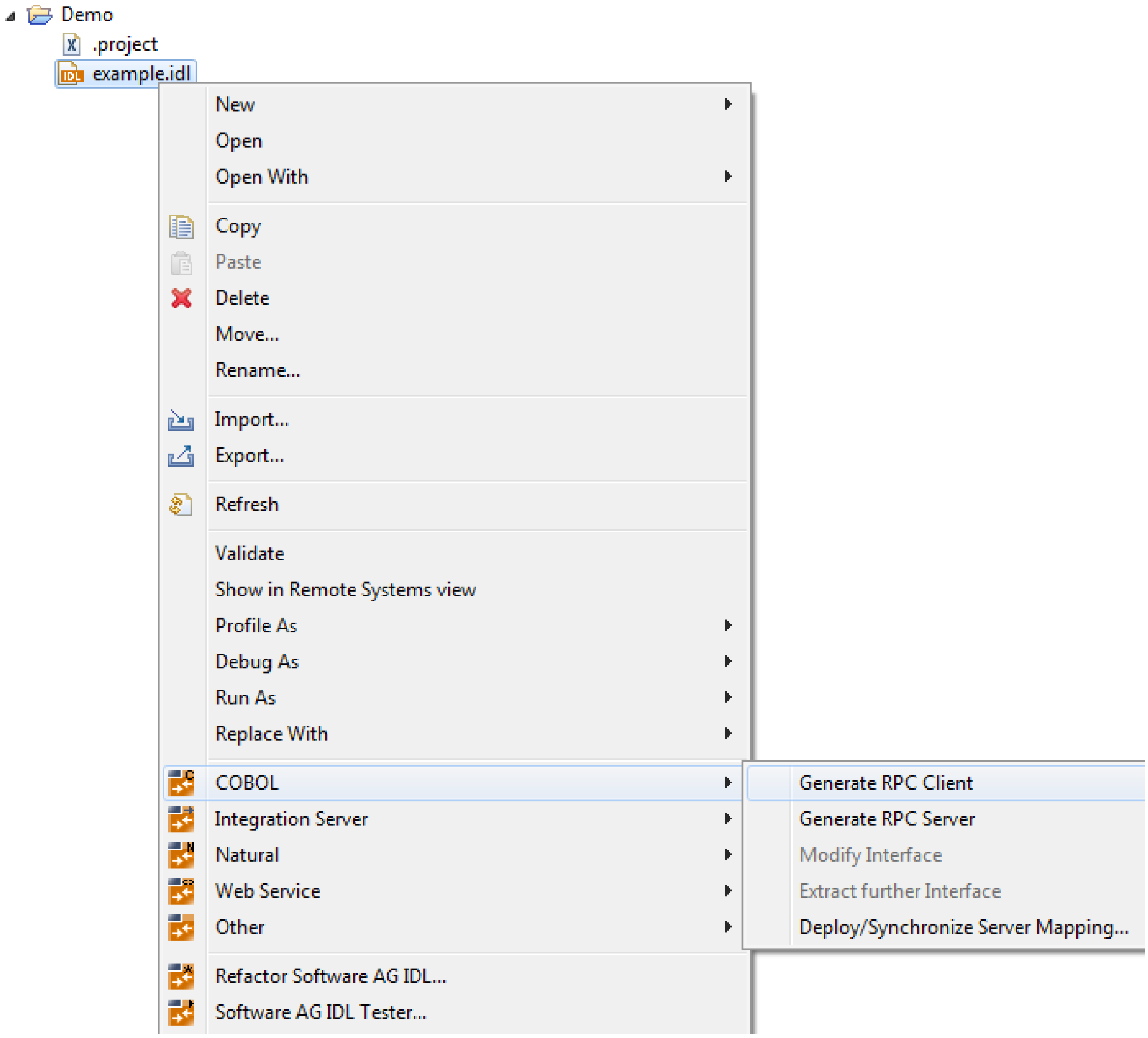
Generating COBOL Source Files from Software AG IDL Files

This chapter describes how to generate COBOL source files from Software AG IDL files. It covers the following topics:

- Select an IDL File and Generate RPC Client or RPC Server
- Generation Settings - Properties
- Generation Settings - Preferences

Select an IDL File and Generate RPC Client or RPC Server

From the context menu, choose **COBOL > Generate RPC Client** and **Generate RPC Server** to generate the COBOL source files.



Note:
In command-line mode, use command `-cobol:client` or `-cobol:server`. See *Using the COBOL Wrapper in Command-line Mode*. Note that existing files will always be overwritten.

Results for **RPC client**:

- The folders *client* and *include* are created as subfolders to the **IDL-Specific Output Folder** defined in the *Generation Settings - Properties*.
- The *client* folder contains the client interface objects, and optionally the generic RPC service module. See *Generic RPC Services Modules*.
- The folder *include* contains the associated copybooks, the RPC communication area copybook `ERXCOMM` and optionally the copybooks `COBINIT` and `COBEXIT`.

Notes:

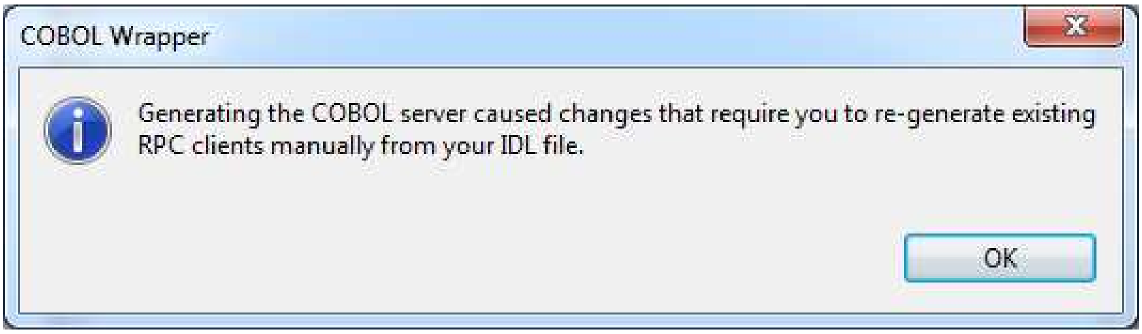
1. The generic RPC service module `COBSRVI` is only generated if the option **Generate Generic RPC Service Module COBSRVI** is set, see *Generate Generic RPC Service for Module COBSRVI*.
2. For further information on the purpose and usage of associated copybooks, see *Using the Generated Copybooks*.
3. For further information on the purpose and usage of the RPC communication area copybook `ERXCOMM` , see *Using the RPC Communication Area*.
4. The copybooks `COBINIT` and `COBEXIT` are only generated if **Copybook** has been selected as *RPC Communication Area*.

Results for **RPC server**:

- The folder *server* is created as a subfolder to the **IDL-Specific Output Folder** defined in the *Generation Settings - Properties*. It contains the RPC server skeletons.

  Warning:
  Take care not to overwrite an existing RPC server implementation with an RPC server skeleton. We recommend moving your RPC server implementation to a different folder.

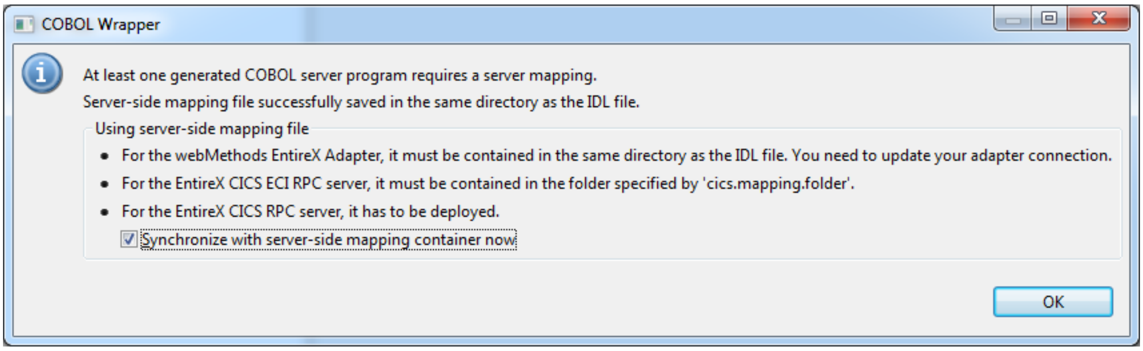
- If required, a server mapping file is generated, too. See *When is a Server Mapping File Required?* in the EntireX Workbench documentation. The server mapping file is of type client-side (extension .cvm) or server-side (.svm). See *How to Set the Type of Server Mapping Files*.
  - If you are using client-side mapping files, the following dialog is displayed.



Generating the COBOL server caused changes that require you to re-generate existing RPC clients manually from your IDL file.

    You need to rebuild all RPC clients communicating with this RPC server program. Select the appropriate wrapper (see *EntireX Wrappers* in the EntireX Workbench documentation) and re-generate the client interface objects. For connections with the webMethods EntireX Adapter you need to update your Adapter connection. See *Step 3: Select the Connection Type* in the Integration Server Wrapper documentation.

  - If you are using server-side mapping files, the dialog below is displayed:



At least one generated COBOL server program requires a server mapping.
Server-side mapping file successfully saved in the same directory as the IDL file.

Using server-side mapping file

- For the webMethods EntireX Adapter, it must be contained in the same directory as the IDL file. You need to update your adapter connection.
- For the EntireX CICS ECI RPC server, it must be contained in the folder specified by 'cics.mapping.folder'.
- For the EntireX CICS RPC server, it has to be deployed.

☑ Synchronize with server-side mapping container now

The generated server-side mapping file need to be synchronized with the server-side mapping container of the target RPC server, except for IMS Connect and CICS ECI connections with the EntireX Adapter, where they are wrapped into the Integration Server connection - the same as client-side mapping files, see *Integration Server Wrapper*.

- Check the option **Synchronize with server-side mapping container now** for the following RPC servers:
  - z/OS (CICS, Batch, IMS) | Micro Focus | BS2000/OSD | z/VSE (CICS, Batch)
- Uncheck the option **Synchronize with server-side mapping container now** for
  - EntireX Adapter and IMS Connect and CICS ECI connections
  - the following RPC servers: CICS ECI | IMS Connect
  - later synchronization of other RPC servers

To quit the COBOL Wrapper and deploy the server-side mapping file

1. Check the option **Synchronize with server-side mapping container now** and choose **OK**. This calls the Deployment Wizard. See *Server Mapping Deployment Wizard* in the EntireX Workbench documentation.
   - If you are using the Server Mapping Deployment Wizard for first time with no predefined deployment environment preferences, continue with *Step 2a: Create a New Deployment Environment* in the EntireX Workbench documentation.
   - If deployment environments are already defined, you may also continue with *Step 3: Select and Existing Deployment Environment and Deploy*.
2. Continue with *Using the COBOL Wrapper for the Server Side*.

To quit the COBOL Wrapper without deploying the server-side mapping file

1. Clear the option **Synchronize with server-side mapping container now** and choose **OK**.
   - Synchronize the server-side mapping container of the target RPC server later. See *Deploying Server-side Mapping Files to the RPC Server* in the RPC server documentation for z/OS (CICS, Batch, IMS) | Micro Focus | CICS ECI | IMS Connect | BS2000/OSD | z/VSE (CICS | Batch).

- For the webMethods EntireX Adapter and IMS Connect or CICS ECI connections, update your Adapter connection. See *Step 3: Select the Connection Type* in the Integration Server Wrapper documentation.

2. Continue with *Using the COBOL Wrapper for the Server Side*.

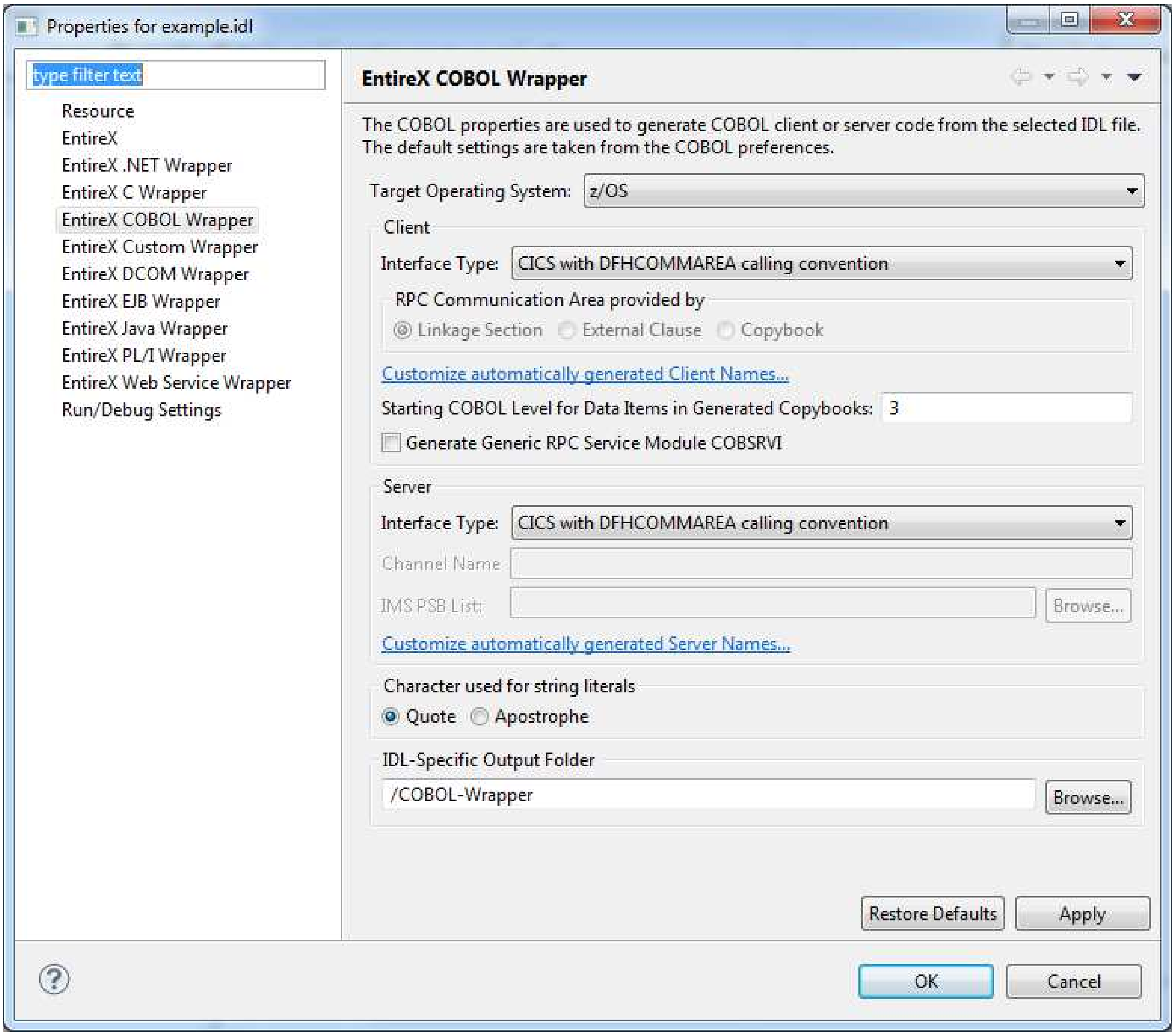
Generation Settings - Properties

This section covers the following topics:

- Introduction
- Target Operating System
- Characters Used for String Literals
- IDL-Specific Output Folder
- Client Interface Types
- Customize Automatically Generated Client Names
- Starting COBOL Level for Data Items in Generated Copybooks
- RPC Communication Area
- Generate Generic RPC Service for Module COBSRVI
- Customize Automatically Generated Server Names
- Server Interface Types
- IMS PSB List
- Channel Name

Introduction

Whenever a new IDL file is created, defaults for the properties are copied from the preferences. See *Generation Settings - Preferences*. To set individual properties per IDL file for COBOL Wrapper generation, use the **Properties** wizard of the IDL file. The **Target Operating System**) and the **Interface Type** are essential. They determine if other parameters such as **RPC Communication Area provided by** can be set or have to remain fixed. The parameter **IDL-Specific Output** defines the location to store the source file subfolders. **Target Operating System** determines whether file extensions are generated or not.



EntireX COBOL Wrapper

The COBOL properties are used to generate COBOL client or server code from the selected IDL file. The default settings are taken from the COBOL preferences.

Target Operating System: z/OS

Client

Interface Type: CICS with DFHCOMMAREA calling convention

RPC Communication Area provided by

● Linkage Section ○ External Clause ○ Copybook

Customize automatically generated Client Names...

Starting COBOL Level for Data Items in Generated Copybooks: 3

□ Generate Generic RPC Service Module COBSRVI

Server

Interface Type: CICS with DFHCOMMAREA calling convention

Channel Name:

IMS PSB List:

Customize automatically generated Server Names...

Character used for string literals

● Quote ○ Apostrophe

IDL-Specific Output Folder

/COBOL-Wrapper

In the following, we give a detailed description of the properties that need to be set for each type of generation:

- **For client and server generation:**
  - *Target Operating System*
  - *Characters Used for String Literals*
  - *IDL-Specific Output Folder*
- **For client generation only:**
  - *Client Interface Types*
  - *Customize Automatically Generated Client Names*
  - *Starting COBOL Level for Data Items in Generated Copybooks*

  - *RPC Communication Area*
  - *Generate Generic RPC Service for Module COBSRVI*
- **For server generation only:**
  - *Server Interface Types*
  - *Customize Automatically Generated Server Names*
  - *IMS PSB List*
  - *Channel Name*

Target Operating System

Select the target operating system for which COBOL code is to be generated. See *Platform Coverage* for a full list of supported operating system versions.

Value	Description
z/OS	IBM z/OS operating system.
z/VSE	IBM z/VSE operating system.
BS2000	Fujitsu Siemens BS2000/OSD operating system.
IBM i	IBM i operating system.
Windows	Microsoft Windows operating system.
UNIX	UNIX operating system.

Characters Used for String Literals

With this option you can specify how string literals are specified in the generated COBOL code. See your COBOL compiler documentation for information on how string literals are enclosed.

Value	Description
Quote	String literals will be enclosed in double quotes in the generated COBOL code.
Apostrophe	String literals will be enclosed in apostrophes (single quotes) in the generated COBOL code.

IDL-Specific Output Folder

This field specifies the folder where the COBOL files will be stored, by default in the same folder as the IDL file. For a non-default location, enter another folder name or choose **Browse...**.

Client Interface Types

Interface Type	Target Operating System	Description	RPC Communication Area Usage
CICS with DFHCOMMAREA calling convention	z/OS, z/VSE	Use this option if you want to build a CICS RPC client application that calls the client interface object(s) with the DFHCOMMAREA interface. Follow the steps under *Using the COBOL Wrapper for CICS with `DFHCOMMAREA` Calling Convention (z/OS and z/VSE)*.	The RPC communication area is passed as described in *Using the RPC Communication Area with `EXEC CICS LINK`*. See also *RPC Communication Area*.

Interface Type	Target Operating System	Description	RPC Communication Area Usage
CICS with standard linkage calling convention	z/OS, z/VSE	Use this option if you want to build a CICS RPC client application that calls the client interface object(s) with a standard linkage interface. Follow the steps under *Using the COBOL Wrapper for CICS with Call Interfaces (z/OS and z/VSE)*.	The RPC communication area is passed with one of the options as described in *Using the RPC Communication Area with a Standard Call Interface*. See also *RPC Communication Area*.
Batch with standard linkage calling convention	z/OS, z/VSE, BS2000/OSD, IBM i	Use this option if you want to build a batch RPC client application that calls the client interface object(s) with a standard linkage interface. Follow the steps under *Using the COBOL Wrapper for Batch (z/OS, BS2000/OSD, z/VSE and IBM i)*.	
IMS BMP with standard linkage calling convention	z/OS	Use this option if you want to build an IMS RPC client application that calls the client interface object(s) with a standard linkage interface for IMS BMP mode. Follow the steps under *Using the COBOL Wrapper for IMS (z/OS)*.	
IMS MPP with standard linkage calling convention	z/OS	Use this option if you want to build an IMS RPC client application that calls the client interface object(s) with a standard linkage interface for IMS MPP mode. Follow the steps under *Using the COBOL Wrapper for IMS (z/OS)*.	
IDMS/DC with standard linkage calling convention	z/OS	Use this option if you want to build an IDMS/DC client application that calls the client interface object(s) with a standard linkage interface for IDMS/DC. Follow the steps under *Using the COBOL Wrapper for IDMS/DC with Call Interfaces (z/OS)*.	
Micro Focus with standard linkage calling convention	UNIX, Windows	Use this option if you want to build a Micro Focus client application that calls the client interface object(s) with a standard linkage interface. Follow the steps under *Using the COBOL Wrapper for Micro Focus (UNIX and Windows)*.	

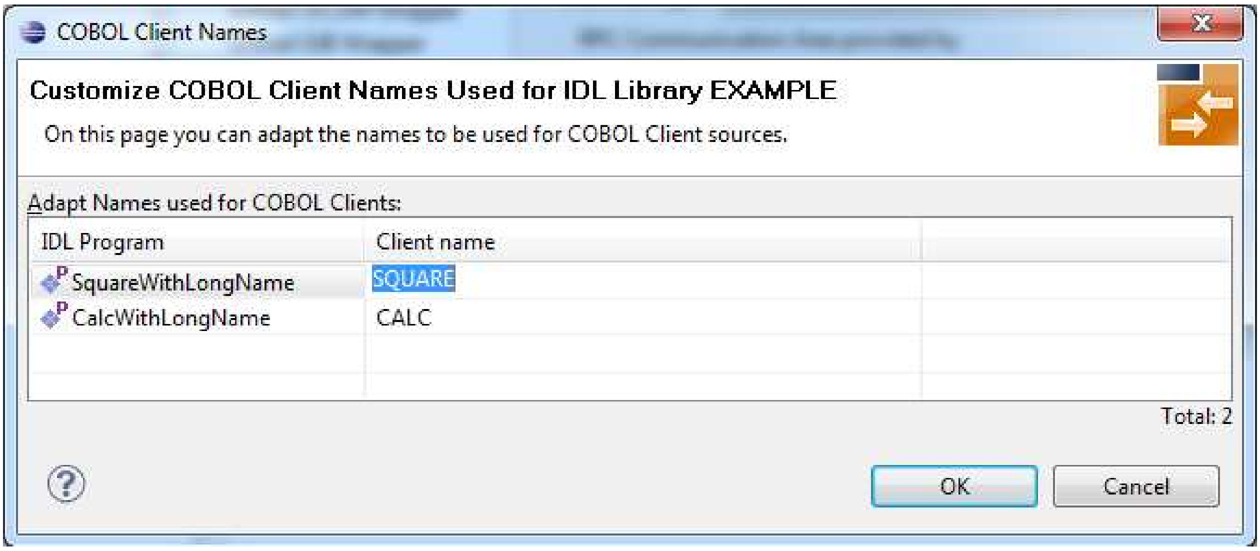
Customize Automatically Generated Client Names

If you open the link **Customize automatically generated Client Names** on the **Properties** page you can adapt the names for the COBOL client interface objects (subprograms). When you call the page the first time, COBOL names are suggested based on the IDL program (`program-definition`) or IDL program alias names. The page varies, depending on whether the target COBOL environment supports long COBOL names or not:

- z/OS and z/VSE
- IBM i
- UNIX and Windows with Micro Focus
- BS2000/OSD

z/OS and z/VSE

Max. 8 characters (short names) are supported as COBOL names:


COBOL Client Names
Customize COBOL Client Names Used for IDL Library EXAMPLE
On this page you can adapt the names to be used for COBOL Client sources.
Adapt Names used for COBOL Clients:

IDL Program	Client name
SquareWithLongName	SQUARE
CalcWithLongName	CALC

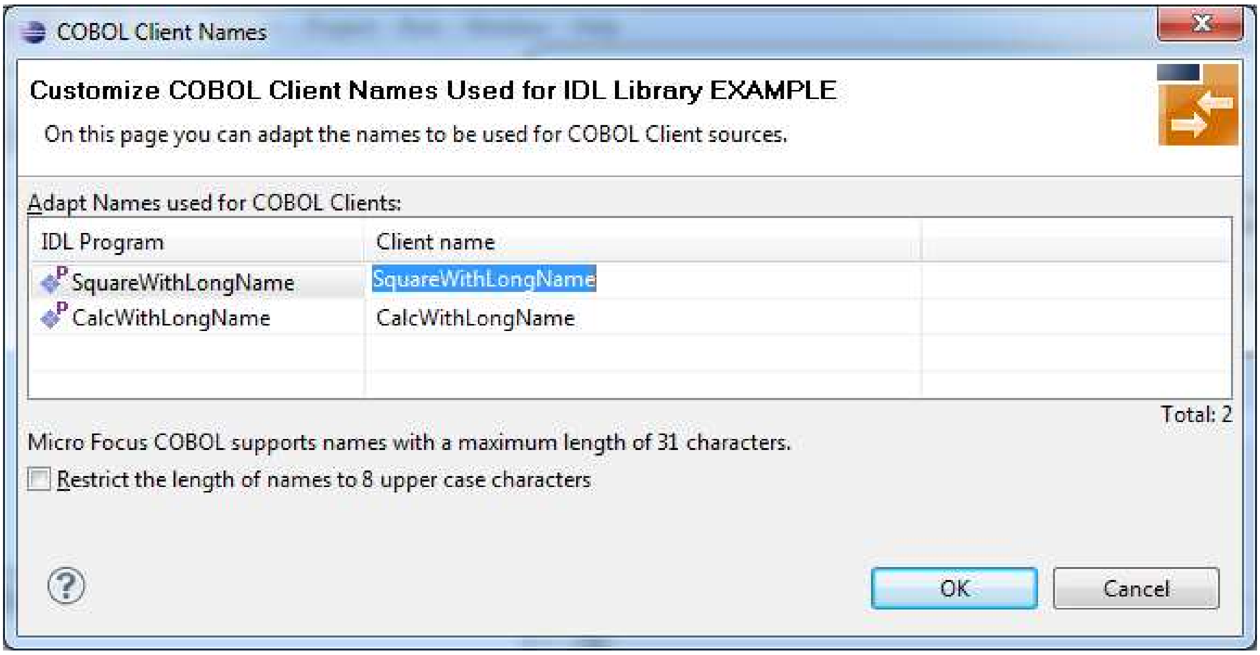
Note:
If your IDL file contains more than one IDL library, the additional column **IDL Library** is displayed.

IBM i

Customization of client names for IBM i is the same as for z/OS and z/VSE. See *z/OS and z/VSE*.

UNIX and Windows with Micro Focus

Max. 31 characters are supported as COBOL names. By default, names are generated with a maximum of 8 characters (short names).



COBOL Client Names

Customize COBOL Client Names Used for IDL Library EXAMPLE

On this page you can adapt the names to be used for COBOL Client sources.

Adapt Names used for COBOL Clients:

IDL Program	Client name
SquareWithLongName	SquareWithLongName
CalcWithLongName	CalcWithLongName

Total: 2

Micro Focus COBOL supports names with a maximum length of 31 characters.

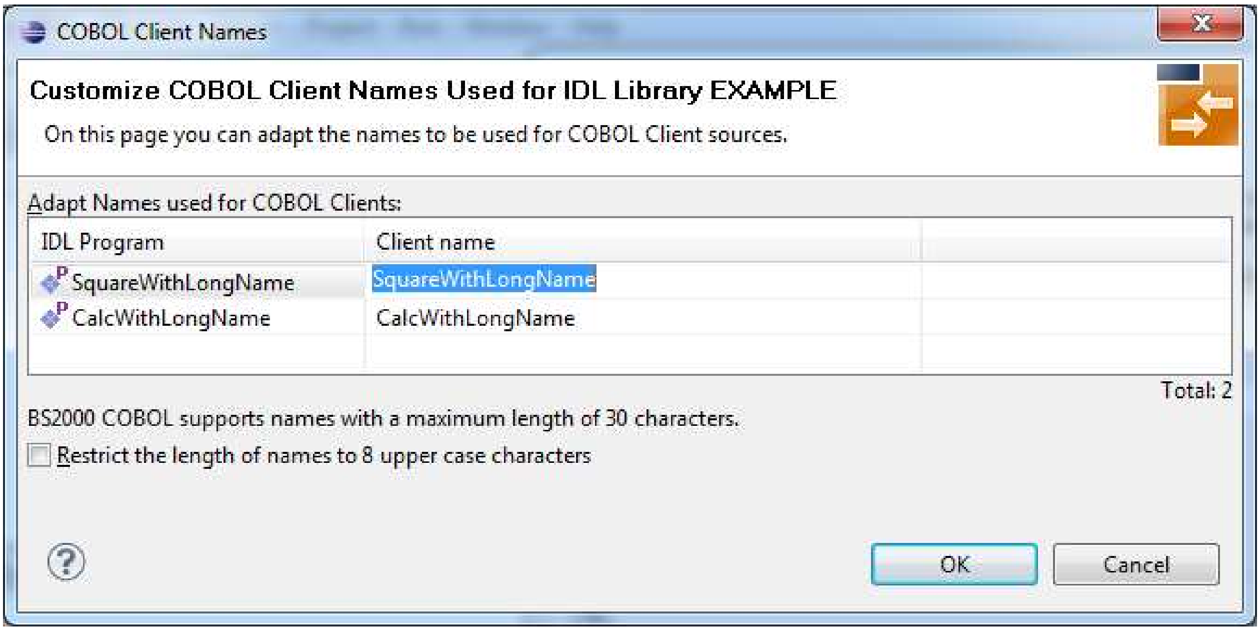
□ Restrict the length of names to 8 upper case characters

Notes:

1. If your IDL file contains more than one IDL library, the additional column **IDL Library** is displayed.
2. With the check box **Restrict the length of names to 8 characters** you can flip between short names and long names. Both sorts of names (short and long) are stored in the property file. For generation you have to decide if short or long names are to be used.

BS2000/OSD

Max. 30 characters are supported as COBOL names. By default, names are generated with a maximum of 8 characters (short names).



COBOL Client Names

Customize COBOL Client Names Used for IDL Library EXAMPLE

On this page you can adapt the names to be used for COBOL Client sources.

Adapt Names used for COBOL Clients:

IDL Program	Client name
SquareWithLongName	SquareWithLongName
CalcWithLongName	CalcWithLongName

Total: 2

BS2000 COBOL supports names with a maximum length of 30 characters.

□ Restrict the length of names to 8 upper case characters

Notes:

1. If your IDL file contains more than one IDL library, the additional column **IDL Library** is displayed.
2. With the check box **Restrict the length of names to 8 characters** you can flip between short names and long names. Both sorts of names (short and long) are stored in the property file. For generation you have to decide if short or long names are to be used.

Starting COBOL Level for Data Items in Generated Copybooks

With this option you can specify the starting COBOL level used in the generated copybooks for COBOL data items.

See *Using the Generated Copybooks* for syntax examples.

Specify a valid COBOL level in the range 1-49. The COBOL programming language maximum of 49 subtracted by the specified level must provide enough levels to hold all IDL levels. Note that IDL types may consume more than one COBOL level, for example:

- IDL unbounded groups require a COBOL level for every dimension. If they are defined on IDL level 1, an extra COBOL level is required
- IDL unbounded arrays require a COBOL level for every dimension plus one extra COBOL level
- some basic (scalar) IDL data types need extra COBOL levels

Notes:

1. Do not specify a level too deep because you may exceed the COBOL programming language maximum of 49 and the generated copybook cannot be compiled.
2. For compatibility with *Client and Server Examples for z/OS CICS*, the level must be 3 or above.
3. For compatibility with all other delivered examples, the level must be 2 or above.

RPC Communication Area

The RPC communication area is used to specify parameters that are needed to communicate with the broker and are not specific to client interface objects. These are for example the broker ID, client parameters such as `userID` and `password` and the server address such as *class/servername/service* etc.

Value	Description
`External Clause`	The RPC communication area is provided as a global area to the RPC client application and the generated client interface object(s). For more information, see option `External Clause` under *Using the RPC Communication Area with a Standard Call Interface*. The COBOL external clause is an extension to COBOL 85 standards and might not be supported by every COBOL compiler. Check your COBOL compiler documentation.
`Linkage Section`	The RPC communication area is provided via an additional parameter between your RPC client application and the generated client interface object(s). For more information, see option `Linkage Section` under *Using the RPC Communication Area with a Standard Call Interface* and *Using the RPC Communication Area with* *`EXEC CICS LINK`*.
`Copybook`	The RPC communication area is provided inside the generated client interface object(s). It is not visible in the RPC client application. Default values are retrieved from EntireX workbench preferences or IDL-specific properties and can be overwritten in the copybook `COBINIT` (see folder *include*). For more information, see option `Copybook` under *Using the RPC Communication Area with a Standard Call Interface*.

Generate Generic RPC Service for Module COBSRVI

The generic RPC service module COBSRVI is generated in the folder *client*. See *Generic RPC Services Modules*. Use this option to control the generation of this module.

If you are using the COBOL Wrapper for the first time:

z/OS and z/VSE

- *Clear* this option for the interface type "CICS with DFHCOMMAREA calling convention". The generic RPC server module is not needed because it is already installed with your z/OS and z/VSE mainframe installation, see *Delivered Modules* for z/OS | z/VSE.

- *Check* this option for all other interface types to generate the generic RPC server module.

BS2000/OSD

- *Clear* this option for the interface type "BATCH with standard linkage calling convention". The generic RPC server module is already installed with your BS2000/OSD mainframe installation, see *Delivered Modules for BS2000/OSD*.

All Other Operating Systems

- *Check* this option for all other interface types to generate the generic RPC server module.

If you are an experienced user of the COBOL Wrapper:

- *Clear* this option if you can reuse the generic RPC server module from a previous COBOL Wrapper project. This will speed up generation time. It is important that *Target Operating System*, *Client Interface Types* and *Characters Used for String Literals* are the same.
- *Check* this option if you need an update of the generic RPC server module because of a newer COBOL Wrapper version (Eclipse update without mainframe installation) to generate the generic RPC server module.

Customize Automatically Generated Server Names

If you open the link **Customize automatically generated Server Names** on the properties page you can, adapt the names for the COBOL server (subprograms). When you call the page the first time, COBOL names are suggested based on the IDL program (`program-definition`) or IDL program alias names. For further details on customizing names for the server side, see the platform-specific section under *Customize Automatically Generated Client Names*; the information here also applies to server names:

- *z/OS and z/VSE*
- *UNIX and Windows with Micro Focus*
- *BS2000/OSD*

Notes:

1. Customization of server names is not supported under IBM i.
2. If the server names (automatically generated or customized) differ from the IDL program names, a server mapping file is required. A server mapping file is an EntireX Workbench file with extension .svm or .cvm. It is generated during generation of RPC server and has to be used in subsequent steps. See *Server Mapping Files for COBOL* and *Using the COBOL Wrapper for the Server Side*.

Server Interface Types

Interface Type	Target Operating System	Description
CICS with DFHCOMMAREA calling convention	z/OS, z/VSE	Use this option if you want to build a CICS RPC server application with a DFHCOMMAREA interface. Follow the steps under *Using the COBOL Wrapper for CICS with `DFHCOMMAREA` Calling Convention (z/OS and z/VSE)*.
CICS with Channel Container calling convention	z/OS	Use this option if you want to build a CICS RPC server application with a channel container interface. To specify a channel name, see *Channel Name*. Follow the steps under *Using the COBOL Wrapper for CICS with Channel Container Calling Convention (z/OS)*.
CICS with DFHCOMMAREA large buffer interface	z/OS, z/VSE	Use this option if you want to build a CICS RPC server application with a large buffer interface. Follow the steps under *Using the COBOL Wrapper for CICS with `DFHCOMMAREA` Large Buffer Interface (z/OS and z/VSE)*.
Batch with standard linkage calling convention	z/OS, z/VSE, BS2000/OSD, IBM i	Use this option if you want to build a batch RPC server application. Follow the steps under *Using the COBOL Wrapper for Batch (z/OS, BS2000/OSD, z/VSE and IBM i)*.
IMS BMP with standard linkage calling convention	z/OS	Use this option if you want to build an IMS RPC server application for IMS BMP mode (no MPP) with standard call interfaces. If your server uses PCB pointers, see *IMS PSB List* below. Follow the steps under *Using the COBOL Wrapper for IMS BMP (z/OS)*.
Micro Focus with standard linkage calling convention	UNIX, Windows	Use this option if you want to build a Micro Focus RPC server application with standard linkage interface(s). Follow the steps under *Using the COBOL Wrapper for Micro Focus (UNIX and Windows)*.

IMS PSB List

IMS PSB List applies to the server interface type "IMS BMP with standard linkage calling convention" only. If your server uses PCB pointers and requires that they are passed through the linkage section, an IMS PSB list is required. Your IDL must comply with the rules under *IMS PCB Pointer IDL Rules*. If no PCB pointers are required, omit the IMS PSB list. See *Server Interface Types* for more information.

Channel Name

Channel Name applies to the server interface type "CICS with Channel Container calling convention" only.

If a channel name is specified, the server is

- called with the given channel name
- generated with COBOL code to check for channel name validity.

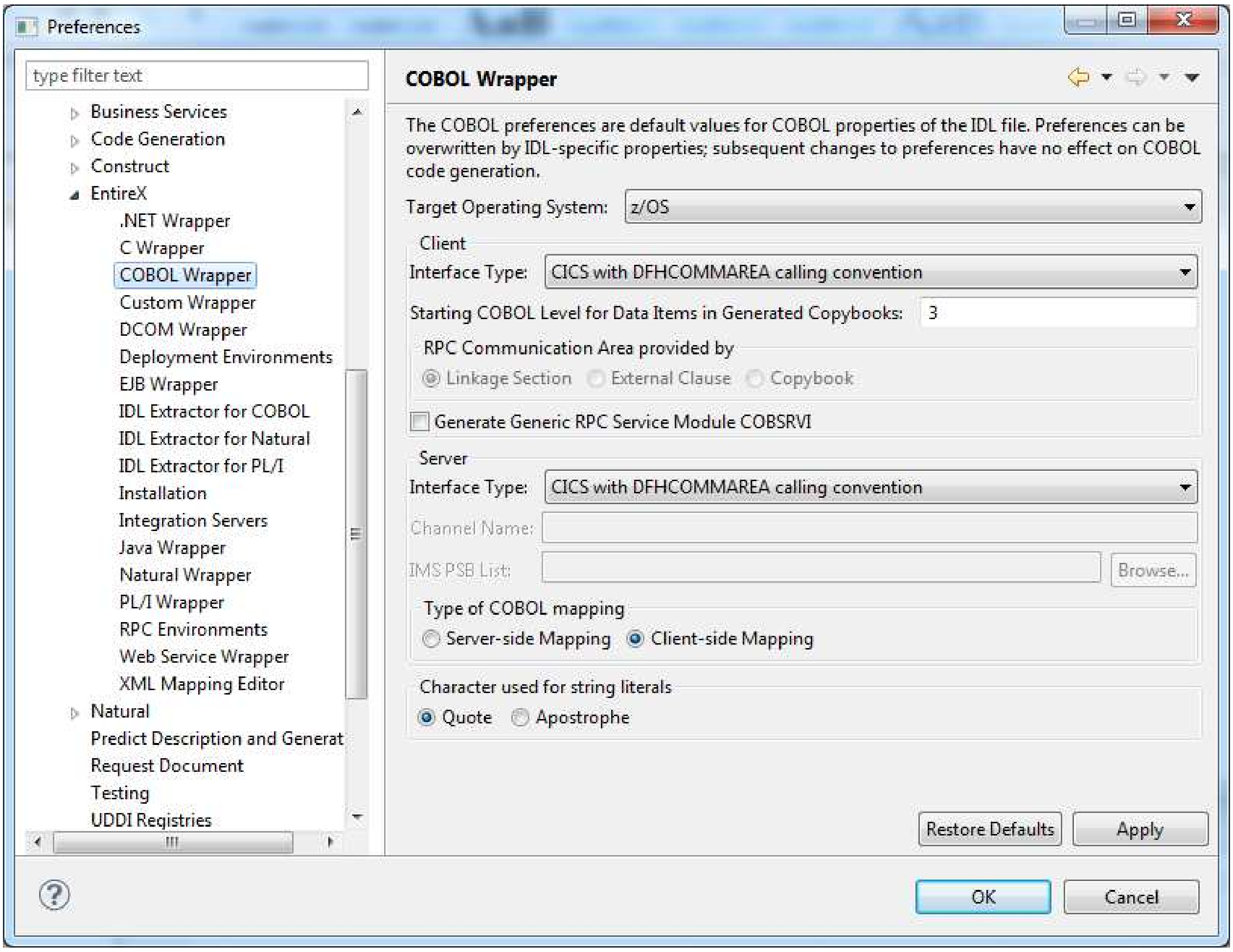
If no channel name is specified, the server is

- called with the "EntireXChannel" channel name
- generated without COBOL code to check for channel name validity.

Your IDL must comply with the rules described under *CICS Channel Container IDL Rules*. See *Server Interface Types* for more information.

Generation Settings - Preferences

Use the **Preferences** page of the COBOL Wrapper to set the workspace defaults for the target operating system, interface types etc. The settings (except **Type of COBOL mapping**) are used as the defaults for the IDL properties when a new IDL file is created; see *Generation Settings - Properties*.



COBOL Wrapper

The COBOL preferences are default values for COBOL properties of the IDL file. Preferences can be overwritten by IDL-specific properties; subsequent changes to preferences have no effect on COBOL code generation.

Target Operating System: z/OS

Client

Interface Type: CICS with DFHCOMMAREA calling convention

Starting COBOL Level for Data Items in Generated Copybooks: 3

RPC Communication Area provided by

● Linkage Section ○ External Clause ○ Copybook

□ Generate Generic RPC Service Module COBSRVI

Server

Interface Type: CICS with DFHCOMMAREA calling convention

Channel Name:

IMS PSB List: Browse...

Type of COBOL mapping

○ Server-side Mapping ● Client-side Mapping

Character used for string literals

● Quote ○ Apostrophe

- **Type of COBOL mapping**
  - Every EntireX Workbench (Eclipse) workspace is either in client-side mapping mode (generating EntireX Workbench server mapping files with extension .cvm) or server-side mapping mode (generating EntireX Workbench server mapping files with extension .svm). See *Server Mapping Files for COBOL* for an introduction. You can adjust the mode here, which will also set the mode of the IDL Extractor for COBOL to the same value. See *IDL Extractor for COBOL Preferences* in the IDL Extractor for COBOL documentation.
  - Server mapping files are generated automatically for RPC servers if required. See *When is a Server Mapping File Required? - COBOL Wrapper* in the *EntireX Workbench* documentation.
  - Server mapping files are not generated for RPC clients.

For a description of all other preferences, see *Generation Settings - Properties*.