webMethods EntireX

Software AG IDL Extractor for PL/I

Version 9.6

April 2014

webMethods EntireX

This document applies to webMethods EntireX Version 9.6.

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

Document ID: EXX-EEXXPLIEXTRACTOR-96-20140628

Table of Contents

Software AG IDL Extractor for PL/I

The Software AG IDL Extractor for PL/I extracts a Software AG IDL file from a PL/I source. The PL/I source can be located in the file system or accessed remotely within a PL/I RPC environment definition.

Using	Describes how to use the Software AG IDL Extractor for PL/I.
RPC Environment Manager	Using the RPC Environment Manager to create, edit and remove RPC environments.
RPC Environment Monitor	using the RPC Environment Monitor to check the availability of each RPC environment.
Command-line Mode	Describes how to use the IDL Extractor for PL/I in command-line mode.
PL/I to IDL Mapping	Describes the subset of PL/I data types that can be mapped to Software AG IDL data types.

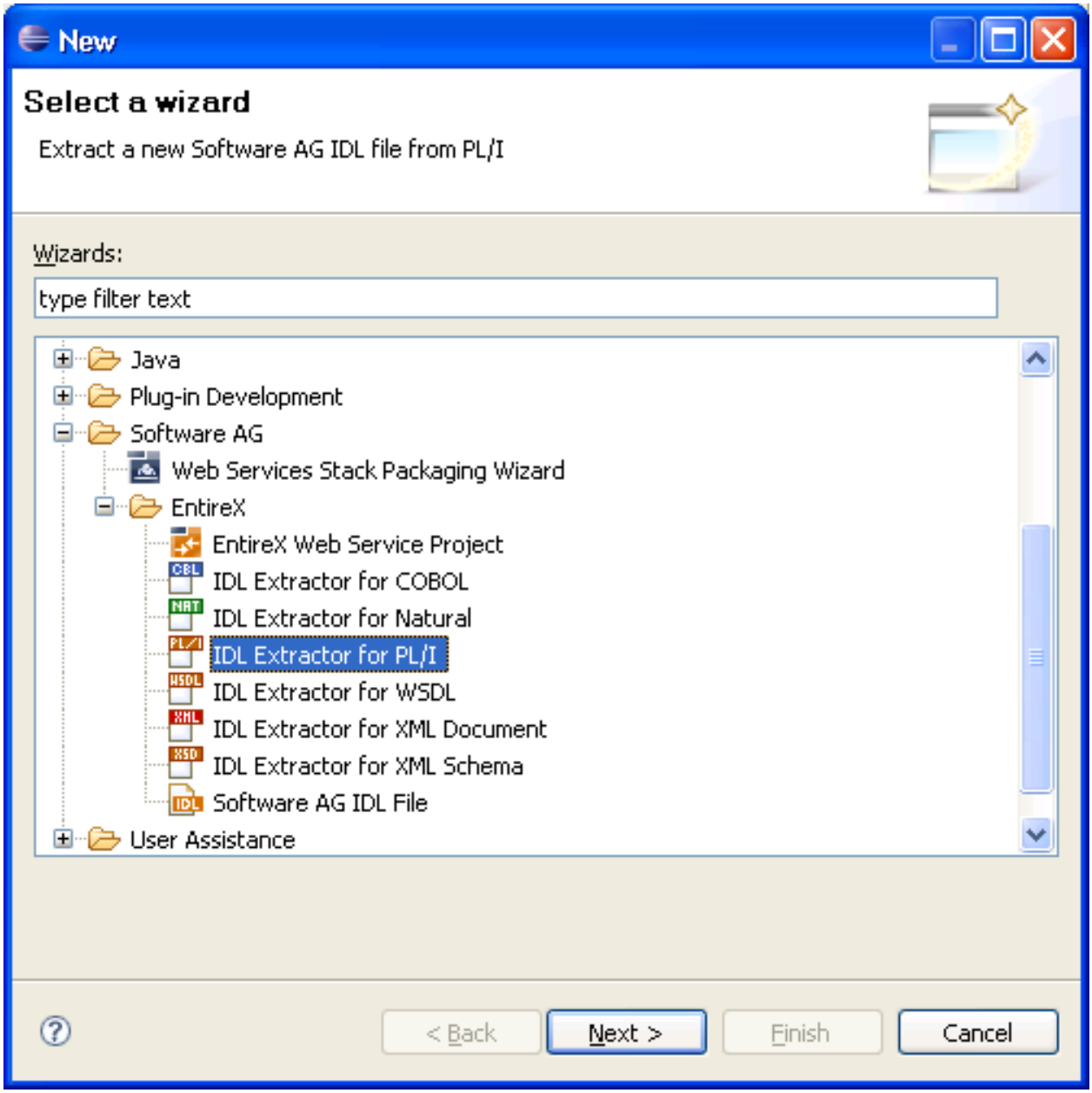
1 Using the Software AG IDL Extractor for PL/I

This chapter describes how to use the IDL Extractor for PL/I.

Extracting Software AG IDL File from Local PL/I Source File

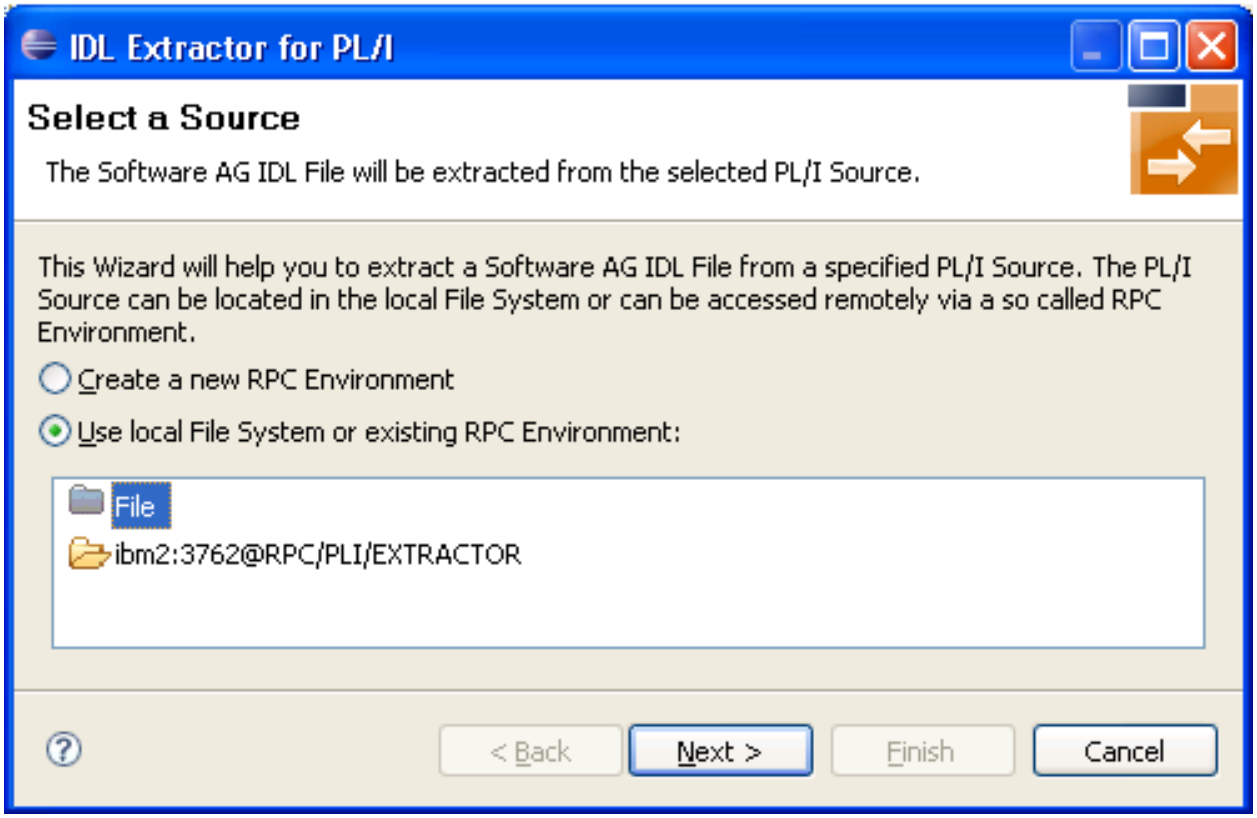
Start the Wizard

Start the IDL Extractor for PL/I Wizard. When the PL/I source file to extract is available in your workspace and you have selected it, the file location will be entered in the wizard automatically.

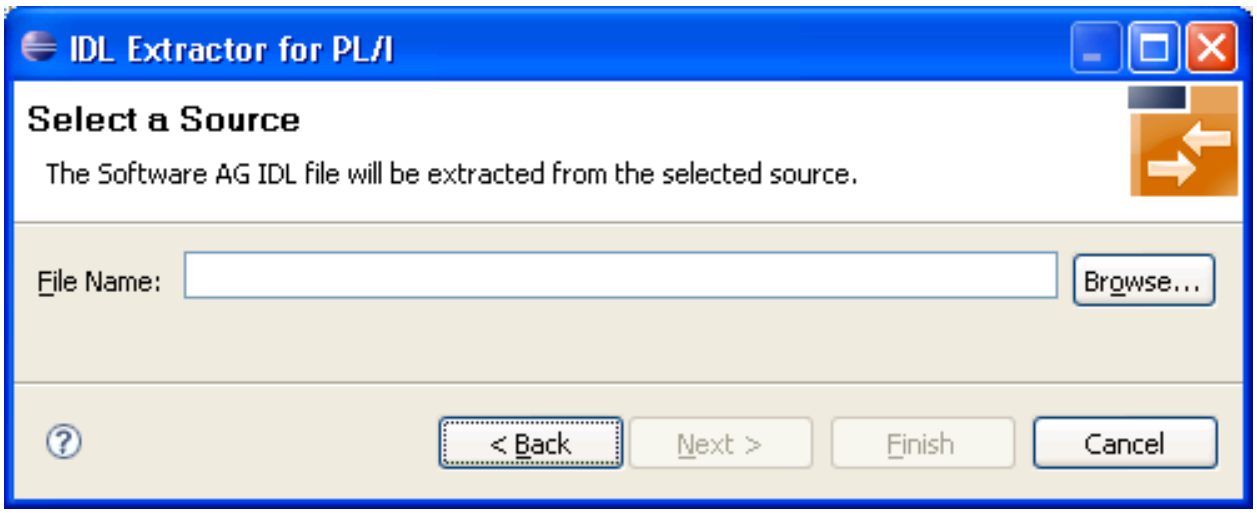


Select a Source

Select **Use local File System or existing RPC Environment**, choose **File** and press **Next**.



If you selected the PL/I source file before you started the wizard, the file location is already present, otherwise press Enter or click **Browse** for the PL/I source file.

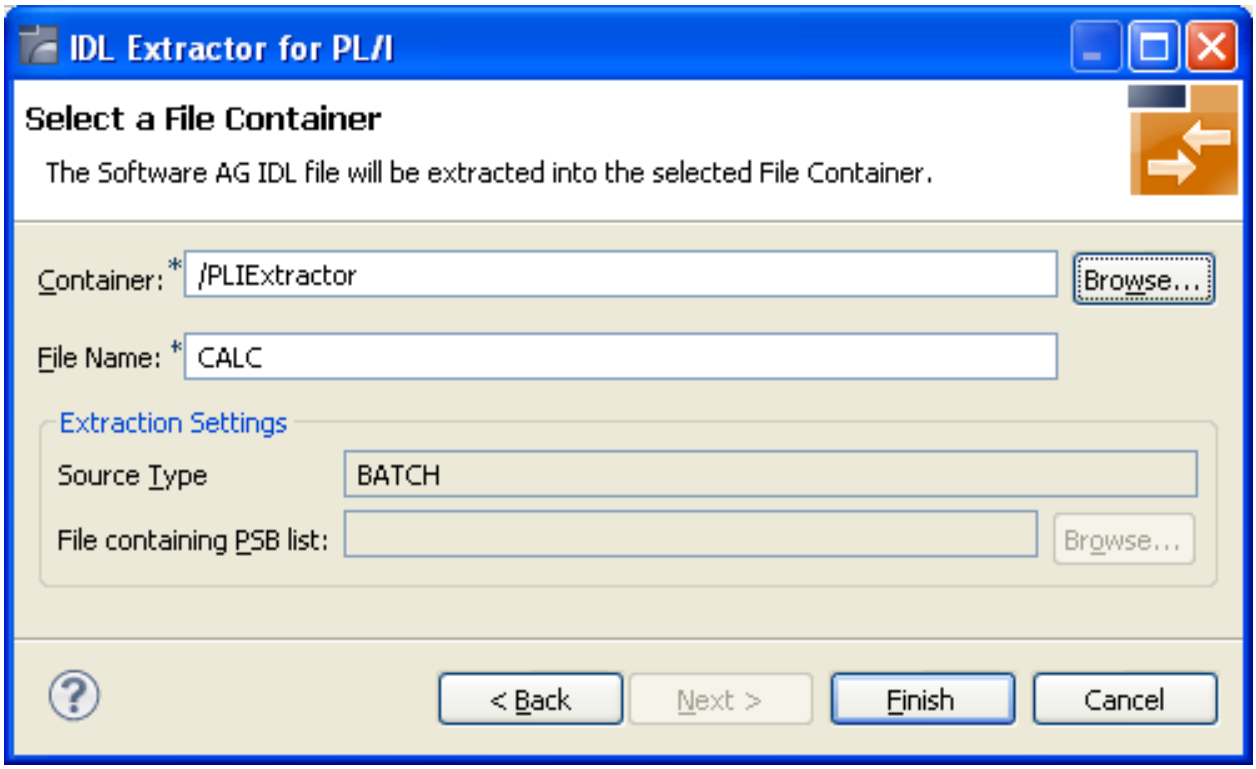


Select the File Container

Select the **Container** where the Software AG IDL file will be stored. Enter the **File Name** of the new IDL file.

The following rules apply to the **Extraction Settings**:

- The **Source Type** must match the PL/I programs you are extracting, otherwise extractions fails. Adjust the source type in the Preferences; see *Preferences*.
- For the source type IMS, optionally IMS-specific PCB pointers can be provided in a so-called PSB List in the field **File containing PSB List**. See *PSB List*. The IDL Extractor for PL/I then marks these parameters with the IMS attribute. See `attribute-list` under *Software AG IDL Grammar* in the *IDL Editor* documentation. This is required to create RPC clients correctly calling IMS BMP programs with PCB pointers successfully.



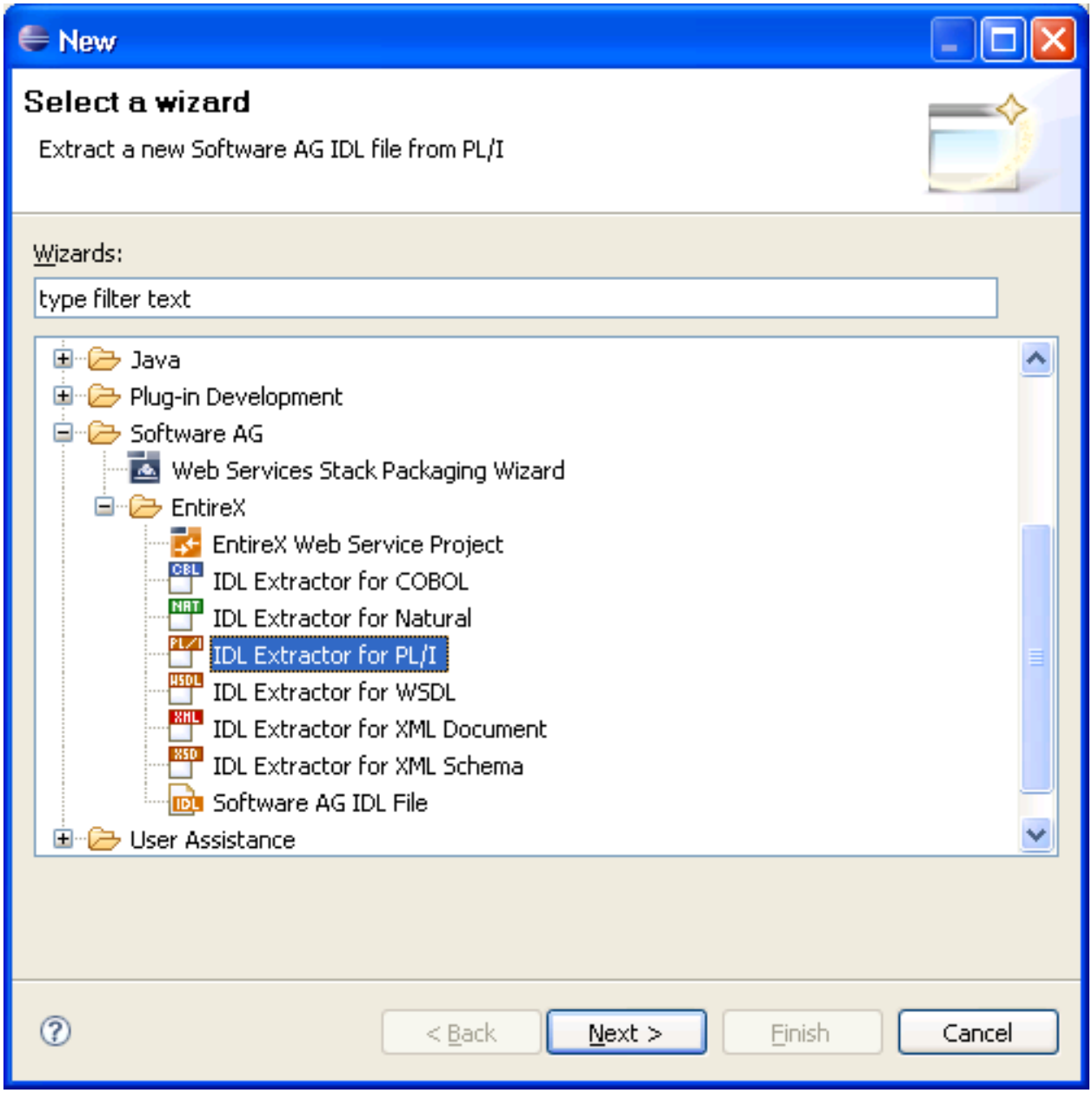
Press **Finish** to extract. For more information see *Extraction Result*.

Extract Software AG IDL File from a Remote PL/I RPC Environment

This section covers the following topics:

Start the Wizard

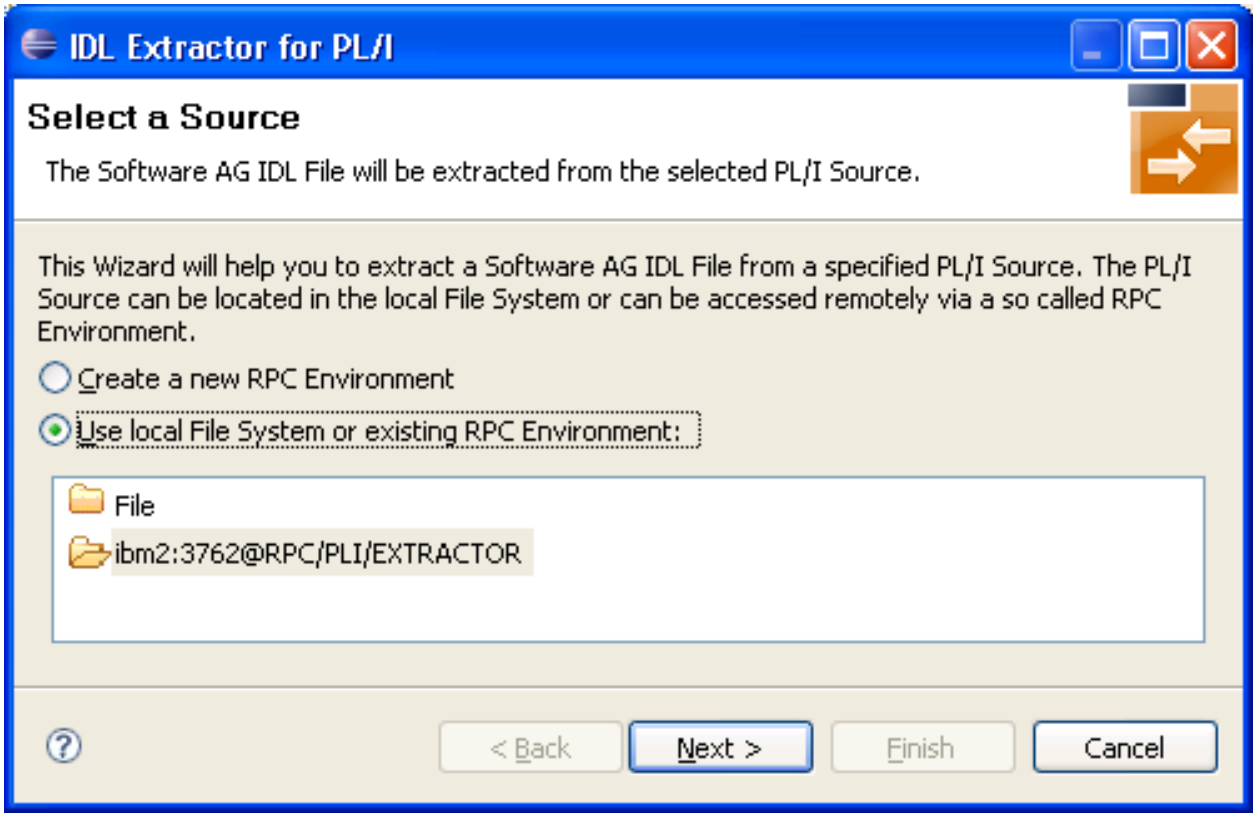
Start the **IDL Extractor for PL/I Wizard**.



Select an RPC Environment

Select **Create a new RPC Environment** and press **Next** if no RPC environment exists or you want to create a new RPC environment. Continue with instructions under *Create RPC Environment (Optional)*.

Select **Use local File System or existing RPC Environment**, choose the RPC environment from the list below and press **Next**. Continue with instructions under *Select Data Set (Optional)*.



Create RPC Environment (Optional)

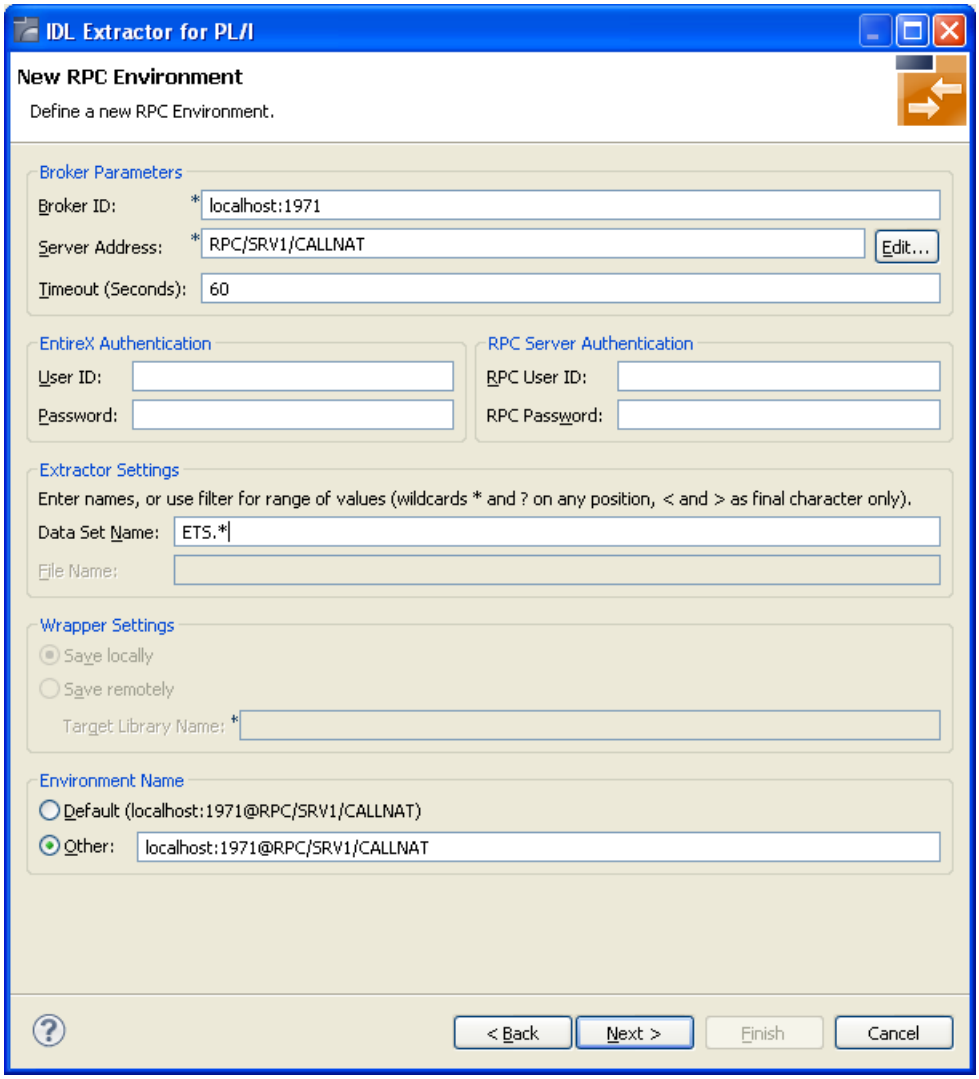
The connection to the Extractor Service to browse for PL/I programs is defined on the **RPC Environment** page. See *Extractor Service*.

In the **Broker Parameters**, required fields are **Broker ID** and **Server Address**, which have the default format "brokerID@serverAddress". The given Timeout value must be in the range from 1 to 9999 seconds (default: 60).

The **EntireX Authentication** describes the settings for the Broker, the RPC Server Authentication the settings for the RPC Server.

The following rules apply to the **Extractor Settings**:

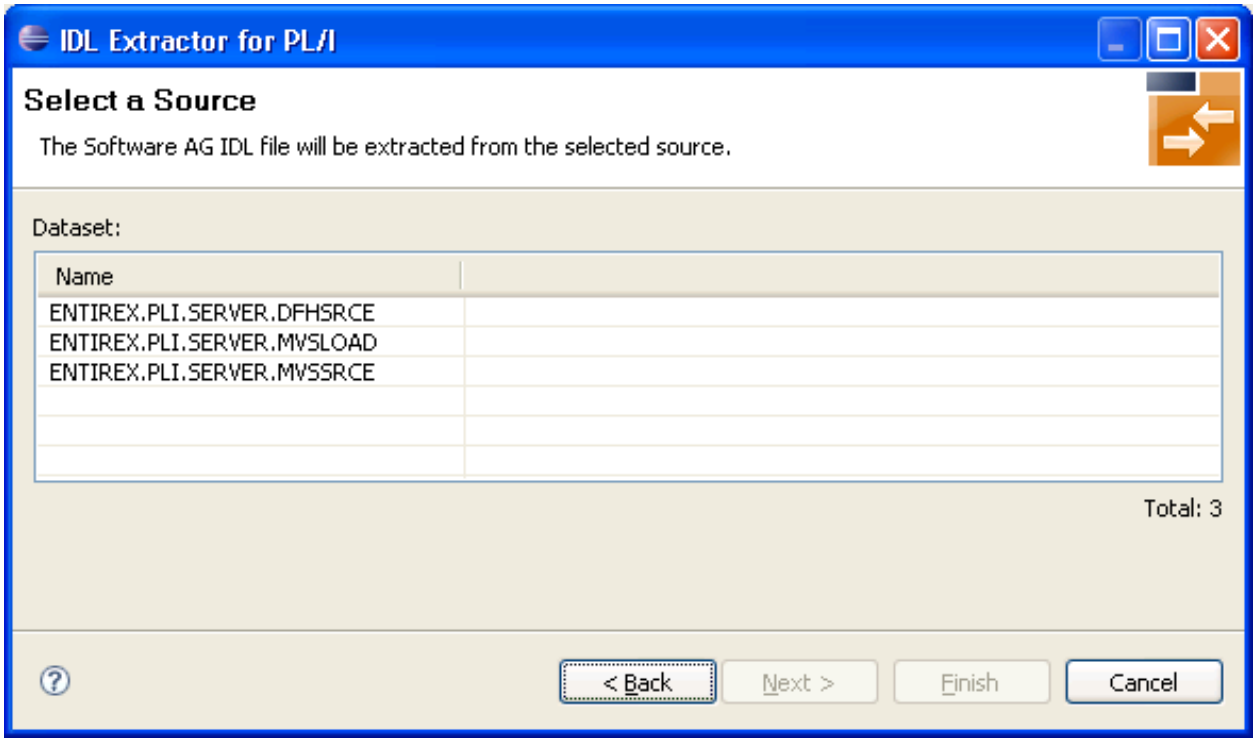
- A high-level qualifier is required in the **Data Set Name** or **HLQ** field. The Extractor Service will then offer only data sets with this high-level qualifier.
- In the **Member Name** field you can provide a prefix for the partitioned data set or CA Librarian members. The Extractor Service will then offer only members beginning with this prefix.



The RPC environments are managed in the Preferences. See *RPC Environment Manager*.

Select Data Set (Optional)

The following page offers all data sets starting with the high-level qualifier defined in the **Filter Settings** of the remote PL/I RPC environment. Select the data set from the list and press **Next**.

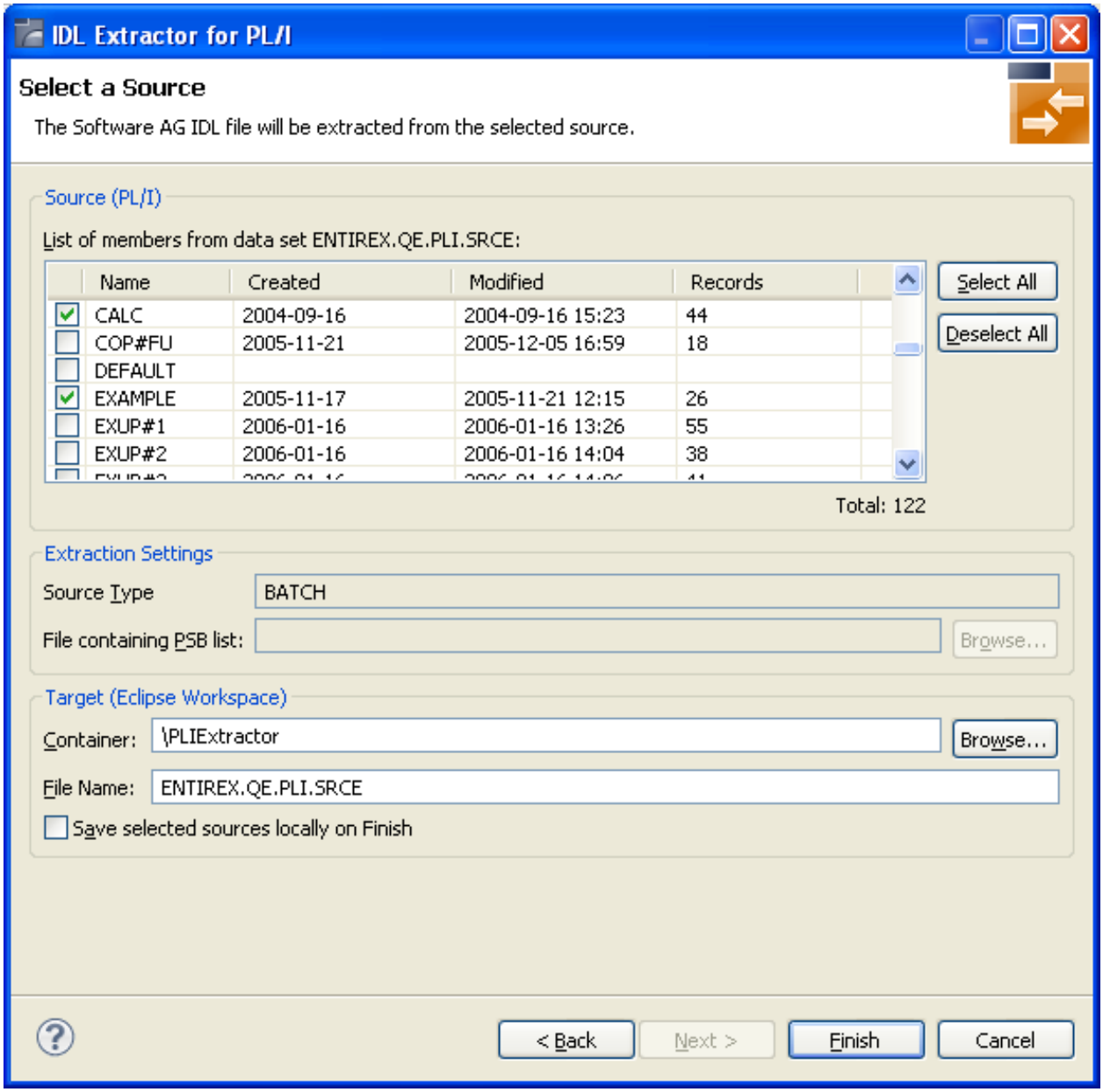


Select Source and Extract

In the **Source** pane, select at least one PL/I file from the **Member** list. The buttons on the right allow you to **Select All** or **Deselect All** members from the list.

The following applies to the **Extraction Settings** pane:

- The Source Type must match the PL/I programs you are extracting, otherwise extractions fails. Adjust the source type in the Preferences; see ***Preferences***.
- For the source type IMS, optionally IMS-specific PCB pointers can be provided in a so-called PSB List in the field **File containing PSB List**. See *PSB List*. The IDL Extractor for PL/I then marks these parameters with the IMS attribute. See `attribute-list` under *Software AG IDL Grammar* in the *IDL Editor* documentation. This is required to create RPC clients correctly calling IMS BMP programs with PCB pointers successfully.
- In the **Target (Eclipse Workspace)** pane, select the Container where the IDL file will be stored. Enter the name of the new IDL file.



Press **Finish** to extract. For more information see *Extraction Result*.

Extraction Result

When the operation is completed, the IDL file is opened with the IDL Editor.

If the PL/I source contains *IMS-specific PCB Pointers* as described in the IMS RPC Server documentation, the extracted IDL contains those pointers marked with the attribute "`IMS`" see `attribute-list` under *Software AG IDL Grammar* in the *IDL Editor* documentation. As a preceding step, use the PL/I Wrapper to generate server interface object(s) and provide them to the IMS RPC Server. See *Using the PL/I Wrapper for IMS BMP*.

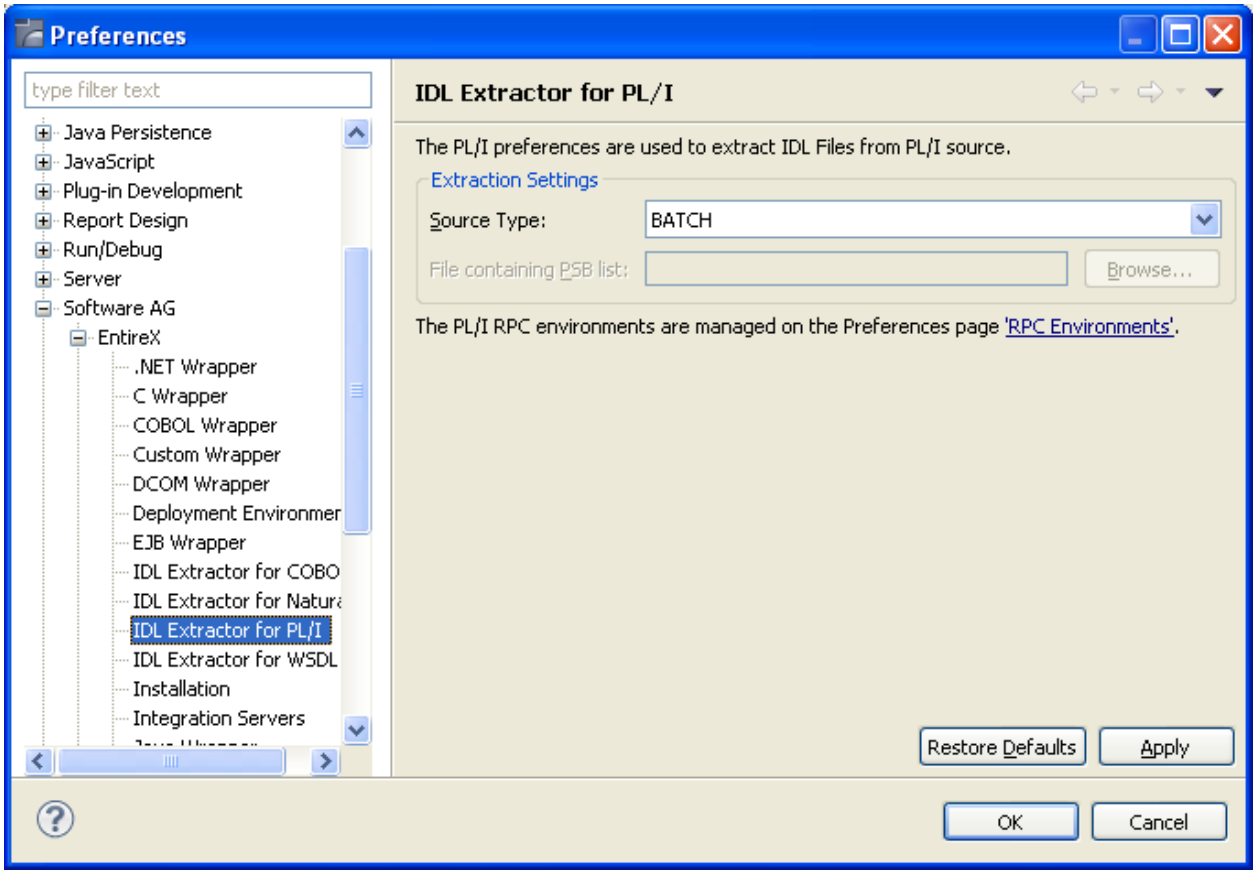
If the PL/I source file contains parameters that cannot be mapped to IDL parameters, an IDL file with incorrect IDL syntax is created. The unsupported parameters lead to IDL parameters of data type "Error", which is not supported. The **Problems View** of the PL/I source file contains markers for all unsupported parameters.

Preferences

The preference page for IDL Extractor for PL/I manages the default values for the IDL Extractor for PL/I Wizard.

The following applies to the **Extraction Settings** pane:

- The Source Type must match the PL/I programs you are extracting, otherwise extractions fails. Adjust the source type in the Preferences.
- For the source type IMS, optionally IMS-specific PCB pointers can be provided in a so-called PSB List in the field **File containing PSB List**. See *PSB List*. The IDL Extractor for PL/I then marks these parameters with the IMS attribute. See `attribute-list` under *Software AG IDL Grammar* in the *IDL Editor* documentation. This is required to create RPC clients correctly calling IMS BMP programs with PCB pointers successfully.


Preferences
type filter text
Java Persistence
JavaScript
Plug-in Development
Report Design
Run/Debug
Server
Software AG
EntireX
.NET Wrapper
C Wrapper
COBOL Wrapper
Custom Wrapper
DCOM Wrapper
Deployment Environmer
EJB Wrapper
IDL Extractor for COBO
IDL Extractor for Natura
IDL Extractor for PL/I
IDL Extractor for WSDL
Installation
Integration Servers
IDL Extractor for PL/I
The PL/I preferences are used to extract IDL Files from PL/I source.
Extraction Settings
Source Type:
BATCH
File containing PSB list:
Browse...
The PL/I RPC environments are managed on the Preferences page 'RPC Environments'.
Restore Defaults
Apply
OK
Cancel

2 Using the IDL Extractor for PL/I in Command-line Mode

Command-line Options

See *Using the EntireX Workbench in Command-line Mode* for the general command-line syntax.

Task	Command	Option	Description
Extract the PL/I source objects from an RPC Extractor Service.	`-extract:pli`	`-brokerpassword`	Password used for broker authentication.
		`-brokeruser`	User used for broker authentication.
		`-environment`	Name of the environment or an RPC server description.
		`-filter`	Filter the PL/I source objects. Show those objects which match the pattern.
		`-help`	Display this usage message.
		`-ims`	Name of the file with the names of the IMS psb parameters.
		`-project`	Name of the project or subfolder where the IDL file is stored.
		`-rpcpassword`	Password used for RPC server authentication.
		`-rpcuser`	User used for RPC server authentication.
		`-source`	A PL/I source object in the environment.
List the PL/I source objects on an RPC Extractor Service.	`-list:pli`	`-brokerpassword`	Password used for broker authentication.
		`-brokeruser`	User used for broker authentication.
		`-environment`	Name of the environment or an RPC server description.
		`-filter`	Filter the PL/I source objects. Show those objects which match the pattern.
		`-help`	Display this usage message.
		`-rpcpassword`	Password used for RPC server authentication.
		`-rpcuser`	User used for RPC server authentication.
		`-source`	A PL/I source object in the environment.

Example

```
<workbench> -extract:pli -environment pliBroker:2006@RPC/PLISRV1/EXTRACTOR -project ↩
/Demo -source PLI.DATA.SET -filter PLISRC1
```

where `<workbench>` is a placeholder for the actual Workbench starter as described under *Using the EntireX Workbench in Command-line Mode*.

The extracted Software AG IDL file will be stored in the project *Demo*.

If the environment name is not a defined RPC environment in the current workspace, the name will be interpreted as a Broker ID and RPC server address (*brokerID@serverAddress*).

The source specifies a data set name and the optional filter defines the member name. Simple wildcard notation with an asterisk (*) can be used at the end of these names.

Status and processing messages are written to standard output (stdout), which is normally set to the executing shell window.

3 PL/I to IDL Mapping

This chapter describes how PL/I data types, their attributes and related syntax are mapped to Software AG IDL data types by the IDL Extractor for PL/I.

IDL Extractor for PL/I Input

PL/I source code is the input for IDL generation. The IDL Extractor for PL/I inspects the parameter definition of PL/I procedures or PL/I functions and their `DECLARE` statements.

The sources

- must contain external PL/I procedures or PL/I functions;
- must be free of preprocessor statements;
- must be compiled with *no errors* and *no warnings*.

IDL Extractor for PL/I Output

The IDL Extractor for PL/I generates:

- the *Software AG IDL file name* by adding the extension ".idl" to the PL/I source file name without extension;
- the *Software AG IDL library name* from the PL/I source file name without extension;
- the *Software AG IDL program name* from the name of the PL/I external procedure or function.

Mapping PL/I Data Types to Software AG IDL

The IDL generator maps the following subset of PL/I data types to IDL data types, other PL/I data types as transfer parameters are *not* supported. If the PL/I source file contains parameters which cannot be mapped to IDL parameters, an IDL file with incorrect IDL syntax will be created.

The following metasymbols and informal terms are used for the Software AG IDL in the table below.

- The metasymbols [and] enclose optional lexical entities.
- The metasymbols (and) enclose numeric expressions which must be evaluated.
- The informal term `n` and `m` is a sequence of numeric characters, for example 123.
- The metasymbols * and / represent a numeric expression which must be evaluated for the real number.

PL/I Data Type	Software AG IDL	Description	Notes
CHARACTER (*n*)	A*n*	Alphanumeric	1
CHARACTER (*)	AV	Alphanumeric variable length	
CHARACTER (*n*) VARYING	AV*n*	Alphanumeric variable length with maximum length	1
GRAPHIC (*n*)	K(*n**2)	Kanji fixed length	2
GRAPHIC (*)	KV	Kanji variable length	
GRAPHIC (*n*) VARYING	KV(*n**2)	Kanji variable length with maximum length	2
BIT (*n*)	B(*n*/8)	Binary	3
BIT (*)	BV	Binary variable length	
FLOAT BINARY FLOAT BINARY (21) FLOAT DECIMAL FLOAT DECIMAL(6)	F4	Floating point (small)	
FLOAT BINARY (53) FLOAT DECIMAL (16)	F8	Floating point (large)	
FIXED BINARY FIXED BINARY (7)	I1	Integer (small)	
FIXED BINARY (15)	I2	Integer (medium)	
FIXED BINARY (31)	I4	Integer (large)	
BIT BIT (1)	L	Logical	
PIC 'S(*m*)9[V(*n*)9]'	N*m*[.*n*]	Unpacked decimal	4
PIC '(*m*)9[V(*n*)9]'	NU*m*[.*n*]	Unpacked decimal unsigned	4
FIXED DECIMAL (*m*,*n*)	P(*m*-*n*)[.*n*]	Packed decimal	5

Notes:

1. *n* is the number of graphic characters (DBCS).
2. *n* is the number of DBCS characters.
3. *n* is the number of bits and *n must* be a multiple of 8.
4. *m*,*n*, are numbers, where *n* <=7, *n*<=29 and *m*+*n* <=29.
5. *m*,*n*, are numbers, where *n* <=7 and *m* <= 29.

Functions

The function return value of a PL/I external function will be mapped to an additional parameter "`Function_Result`" with the direction `out`; this parameter will be appended to the last parameter of the procedure.

For example, the external function `R_CHAR`

```
R_CHAR:  PROCEDURE (p) RETURNS ( CHAR (20) ) ;
          PUT SKIP LIST('R_CHAR $dollar;Revision: n.n $');
          DCL p CHAR(10);
          RETURN (p);
 END R_CHAR;
```

will be mapped to the Software AG IDL program:

```
program 'R_FLOAT' is
      define data parameter
      1 p     (F4)      In Out
      1 Function_Result         (F4)      Out
      end-define
```

Structures

Structures are mapped to Software AG IDL groups. Asterisks as fillers or reserved items are *not* supported.

```
declare 1 Payroll,
          2 Name,
                       3 Last char(20),
                       3 First char(15),
          2 Hours,
                       3 Regular fixed dec(5,2),
                       3 Overtime fixed dec(5,2),
          2 Rate,
                       3 Regular fixed dec(3,2),
                       3 Overtime fixed dec(3,2);
```

will be mapped to Software AG IDL

```
1 Payroll In Out
 2 Name
  3 Last (A20)
  3 First (A15)
 2 Hours
  3 Regular (P3.2)
  3 Overtime (P3.2)
 2 Rate
  3 Regular (P1.2)
  3 Overtime (P1.2)
```

Arrays

Arrays are mapped to Software AG IDL arrays. The dimension of an array is restricted to 3.

```
DCL A CHAR(10) DIMENSION (100);
DCL B CHAR(10) DIMENSION (*);
DCL C CHAR(10) DIMENSION (-5:10);
DCL D CHAR(10) DIMENSION (10,10,10);
```

will be mapped to

```
1 A (A10/100)      In Out
1 B (A10/V)        In Out
1 C (A10/16)       In Out
1 D (A10/10,10,10) In Out
```

Aligned

The `ALIGNED` attribute will be mapped to the Software AG IDL attribute `aligned`. See `attribute-list` under *Software AG IDL Grammar* in the *IDL Editor* documentation.

PL/I to IDL Restrictions

The following table lists features, clauses and items that are not supported by the IDL Extractor for PL/I:

Item	Description
PL/I Preprocessor	MACRO preprocessor for PL/I source program alteration. It is executed prior to compilation
`UNALIGNED` attribute	The `UNALIGNED` attribute reduces to one byte the alignment requirements for halfwords, fullwords, and doublewords and it reduces to one bit the alignment requirement for bit strings.

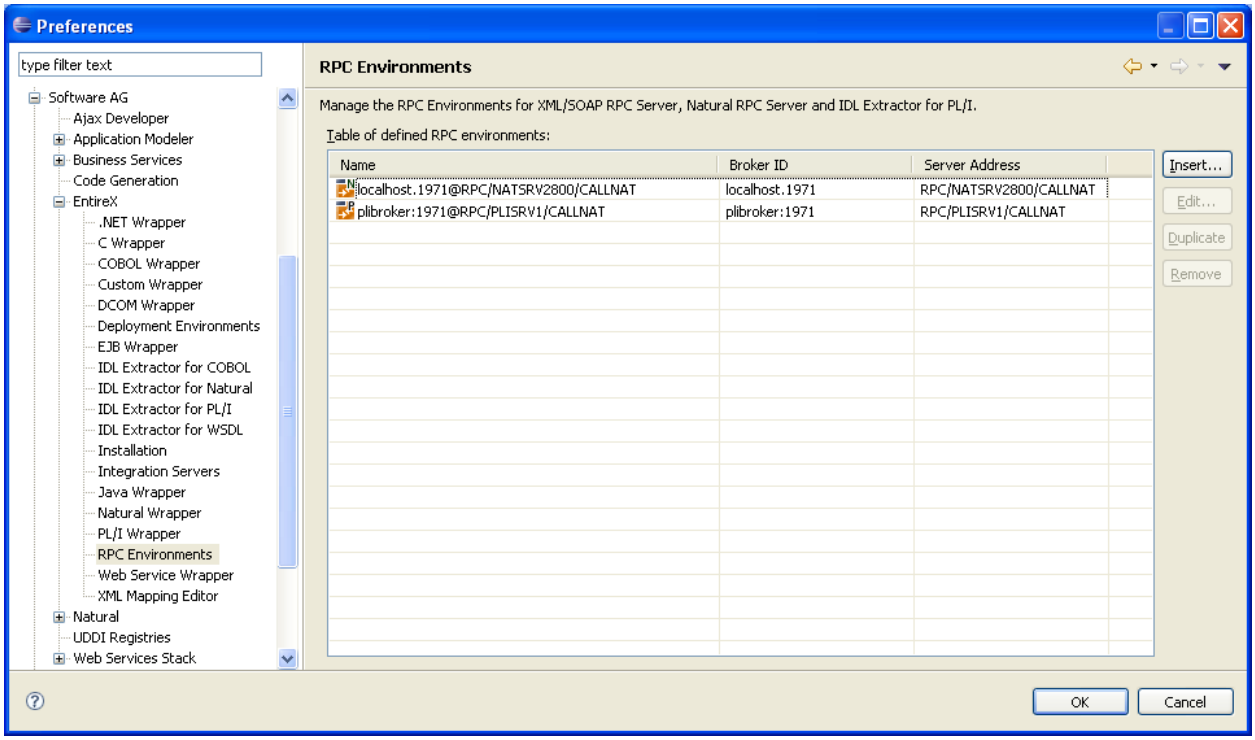
4 RPC Environment Manager

The RPC environment is managed on the RPC environment preference page. The RPC environments can be created, edited and removed. There are several types of RPC environment: Natural, PL/I and XML/SOAP. The RPC environment type will be used to prepare the selection lists of the following wizards:

- Natural RPC Server
- IDL Extractor for PL/I
- XML/SOAP RPC Server

Use the *RPC Environment Monitor* to check the availablity of each RPC environment.

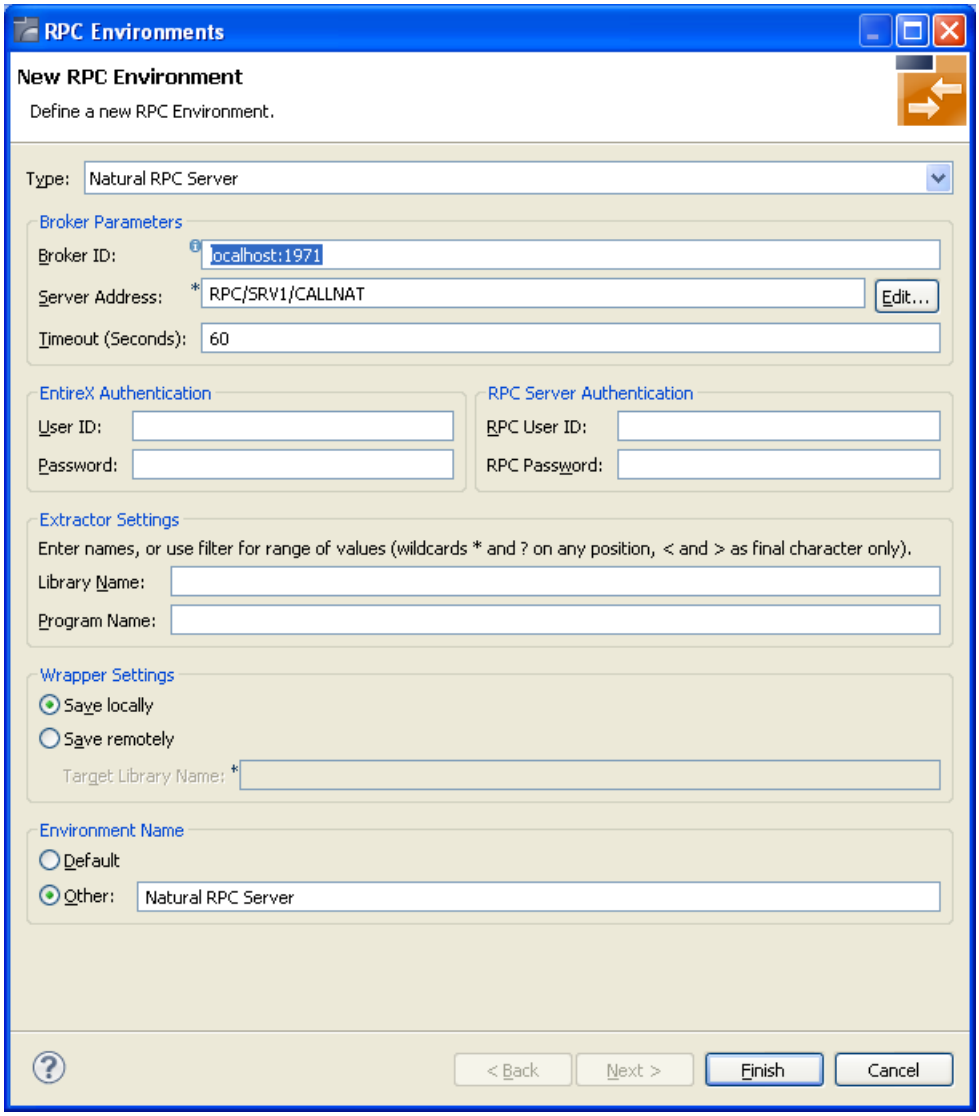
Using these wizards, you can add new RPC environments of the respective type. To manage these RPC environments, open the **Preferences** page.



To edit an existing RPC environment, select the table row and press **Edit...**. If multiple entries are selected, the first entry is used.

To remove an RPC environment, select the table row and press **Remove**. You can select multiple environments.

To create a new RPC environment, choose **Insert...**.



Choose the **Type** and enter the required fields: **Broker ID**, **Server Address** and a unique **Environment Name**, which will have the default format *brokerID@serverAddress*. The given **Timeout** value must be in the range from 1 to 9999 seconds (default: 60).

EntireX Authentication describes the settings for the broker, and **RPC Server Authentication** describes the settings for the RPC server.

The **Extraction Settings** are used for the IDL Extractors (Natural and PL/I) only. Use them to specify the name of the **Dataset/Library** and the **Member/Program** name.

The **Wrapper Settings** are used for Natural Wrapper only, and you can specify the operation type and target library name (not available for **Save locally**).

5 RPC Environment Monitor

The RPC Environment Monitor is part of the EntireX Workbench. It is an Eclipse view that provides a quick overview of the availability of the defined RPC environments in your workspace.

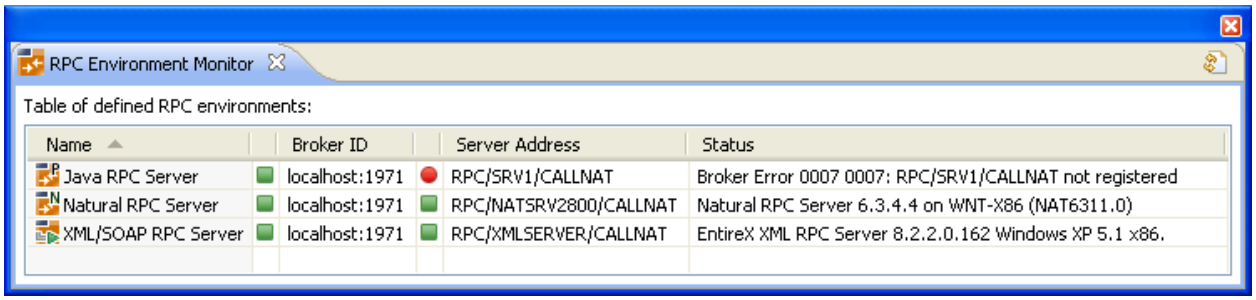
To open the RPC Environment Monitor from the EntireX perspective

- Choose **Window > Show View > RPC Environment Monitor**.

To open the RPC Environment Monitor from a non-EntireX perspective

- Choose **Window > Show View > Other > Software AG > RPC Environment Monitor**.

The RPC environments are managed on the **Preference** page. See *RPC Environment Manager*.



The status check starts when the view is opened. To force an additional check, choose **Refresh** from the **Views** toolbar. The status check can be cancelled in the dialog that appears or within the Eclipse progress view. When the check is complete or if it cancelled, the following symbols indicate the status of the corresponding item. The table will be reloaded every time a status check is started to make sure all stored RPC environments are available.

Symbol	Status
	Running.
	Not running.
	Unknown (at the beginning of the check or if the check was cancelled).

Note: Additional status information (including error messages) is displayed when refreshing the view (by a ping command to all specified RPC servers).