

## **webMethods EntireX**

**Software AG IDL Extractor for PL/I**

Version 10.7

October 2020

This document applies to webMethods EntireX Version 10.7 and all subsequent releases.

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

Copyright © 1997-2020 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>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

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: EXX-EEXPLIEXTRACTOR-107-20220422**

## Table of Contents

1 About this Documentation .....	1
Document Conventions .....	2
Online Information and Support .....	2
Data Protection .....	3
2 Using the Software AG IDL Extractor for PL/I .....	5
Extracting Software AG IDL File from Local PL/I Source File .....	6
Extract Software AG IDL File from a Remote PL/I RPC Environment .....	10
Extraction Result .....	16
Preferences .....	16
3 Using the IDL Extractor for PL/I in Command-line Mode .....	19
Command-line Options .....	20
Example .....	21
4 PL/I to IDL Mapping .....	23
IDL Extractor for PL/I Input .....	24
IDL Extractor for PL/I Output .....	24
Mapping PL/I Data Types to Software AG IDL .....	24
Functions .....	26
Structures .....	26
Arrays .....	27
Aligned .....	27
PL/I to IDL Restrictions .....	27
5 RPC Environment Manager for IDL Extractor for PL/I .....	29
6 RPC Environment Monitor .....	33



# 1 About this Documentation

---

- Document Conventions ..... 2
- Online Information and Support ..... 2
- Data Protection ..... 3

## Document Conventions

---

Convention	Description
<b>Bold</b>	Identifies elements on a screen.
Monospace font	Identifies service names and locations in the format <code>folder.subfolder.service</code> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies:  Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies:  Text you must type in. Messages displayed by the system. Program code.
{ }	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 (...).

## Online Information and Support

---

### Product Documentation

You can find the product documentation on our documentation website at <https://documentation.softwareag.com>.

In addition, you can also access the cloud product documentation via <https://www.software-ag.cloud>. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

### Product Training

You can find helpful product training material on our Learning Portal at <https://knowledge.softwareag.com>.

## Tech Community

You can collaborate with Software AG experts on our Tech Community website at <https://tech-community.softwareag.com>. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at <https://github.com/softwareag> and <https://hub.docker.com/publishers/softwareag> and discover additional Software AG resources.

## Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at <https://empower.softwareag.com>. Many services on this portal require that you have an account. If you do not yet have one, you can request it at <https://empower.softwareag.com/register>. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

## Data Protection

---

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 Using the Software AG IDL Extractor for PL/I

---

▪ Extracting Software AG IDL File from Local PL/I Source File .....	6
▪ Extract Software AG IDL File from a Remote PL/I RPC Environment .....	10
▪ Extraction Result .....	16
▪ Preferences .....	16

The EntireX PL/I Wrapper provides access to RPC-based components from PL/I applications. It enables you to develop both client and server applications. This chapter describes how to use the IDL Extractor for PL/I.

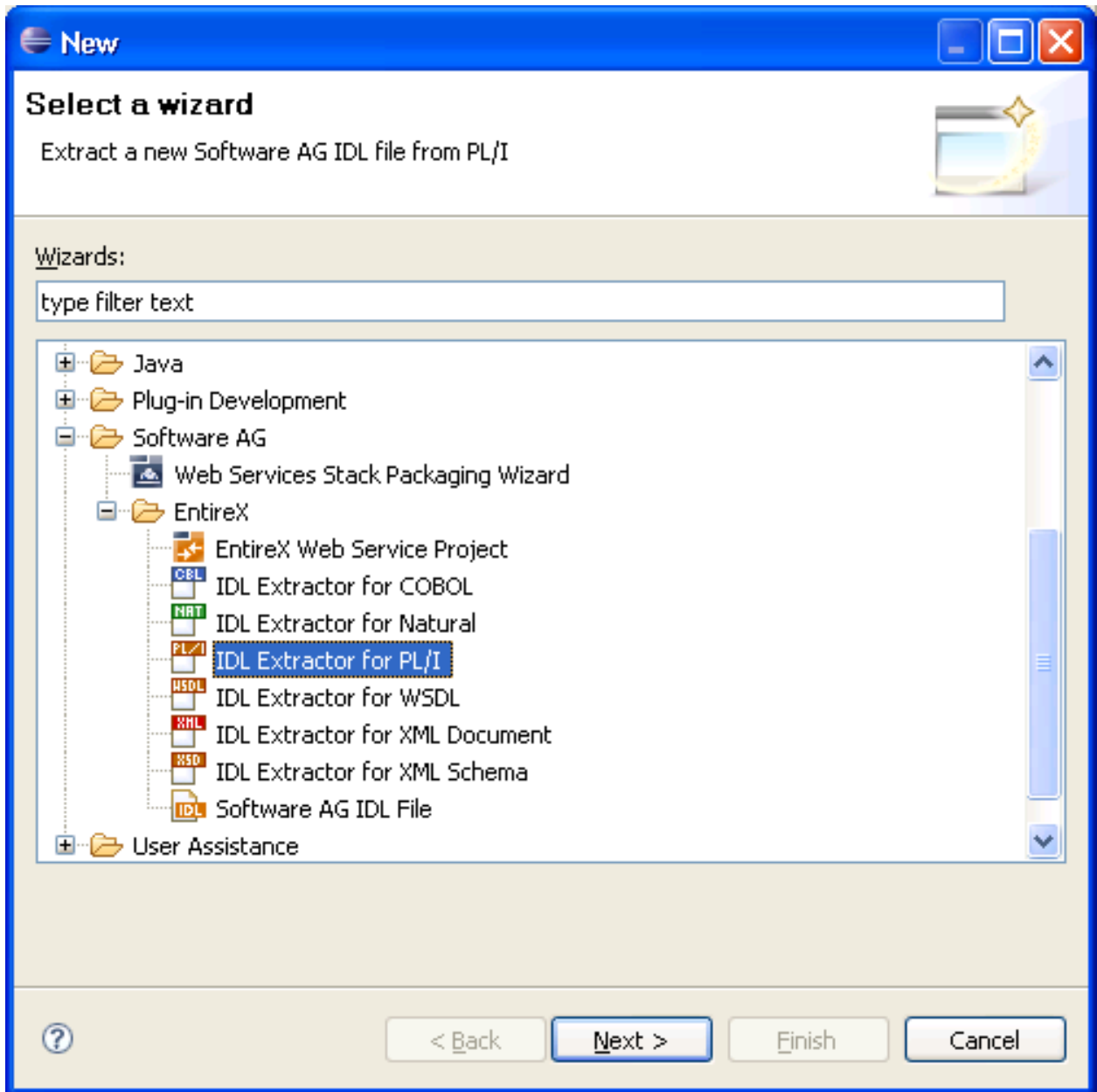
## Extracting Software AG IDL File from Local PL/I Source File

---

- [Start the Wizard](#)
- [Select a Source](#)
- [Select the File Container](#)

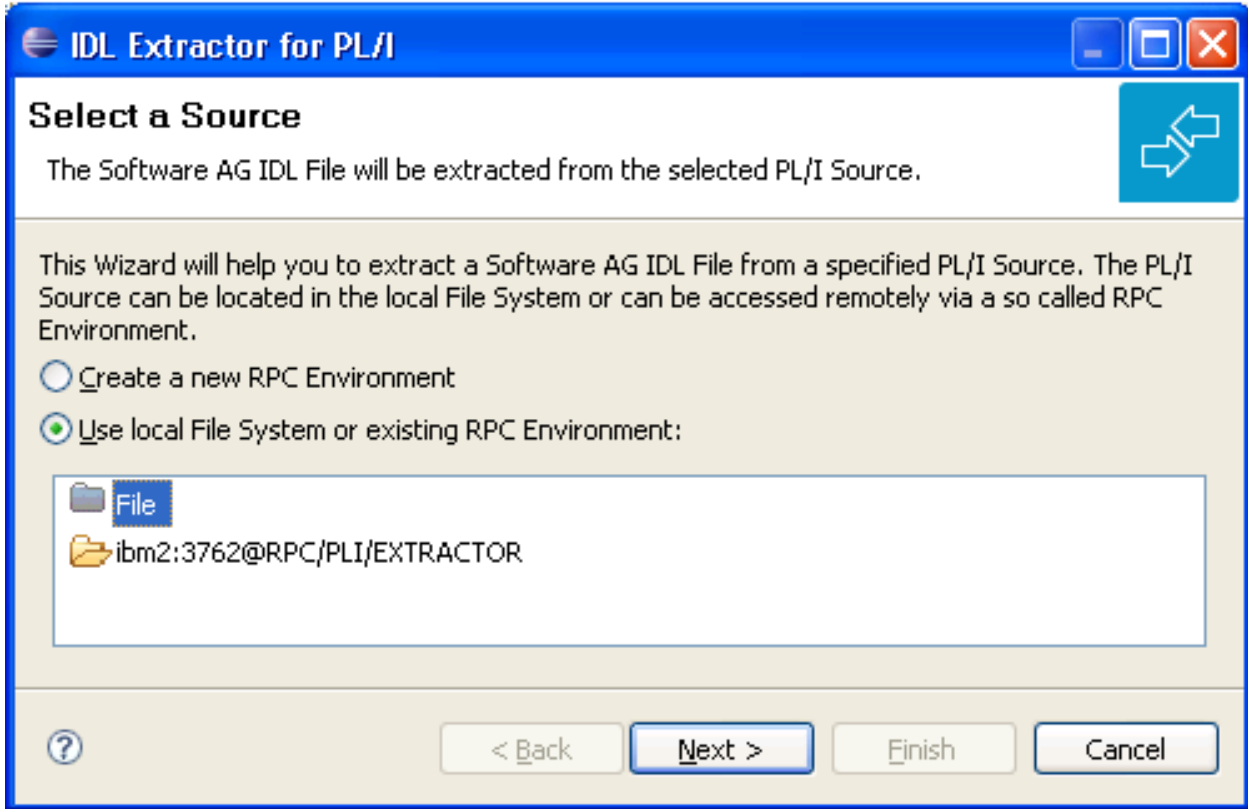
### Start the Wizard

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.

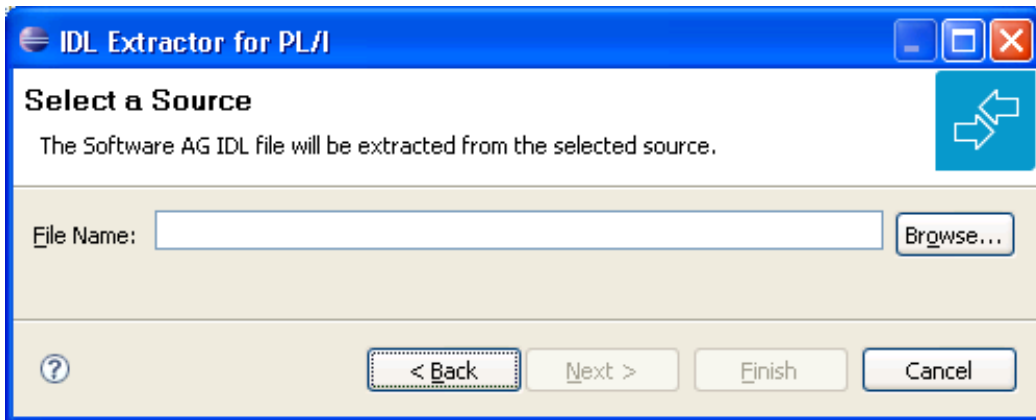


### Select a Source

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.

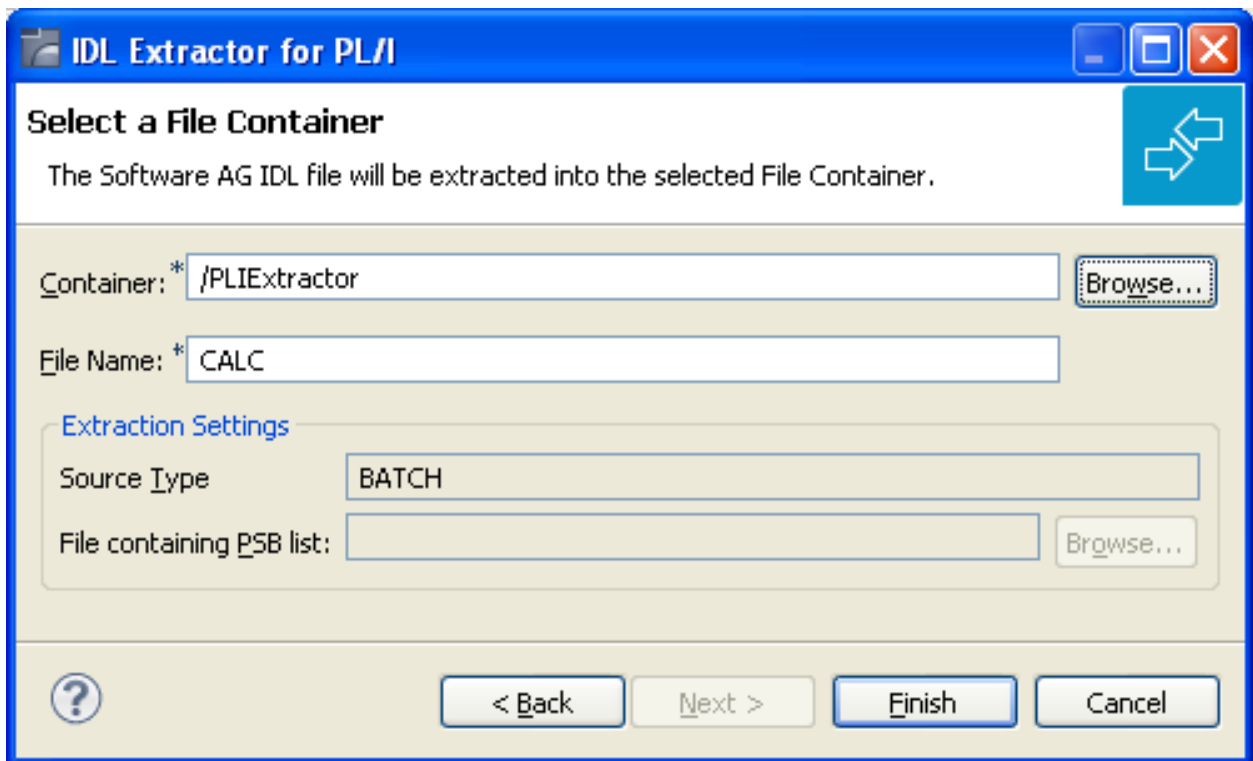


## Select the File Container

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

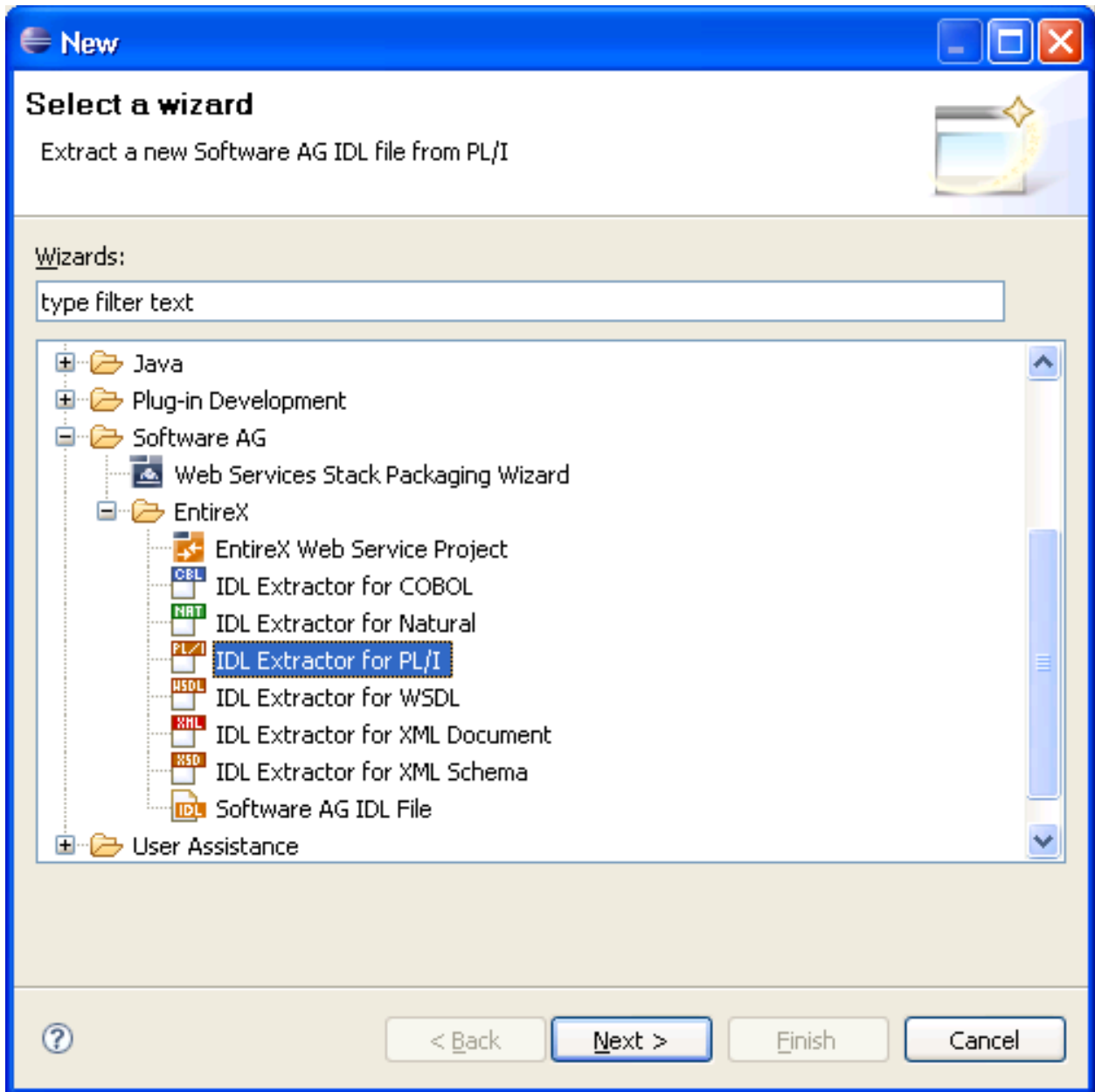
---

This section covers the following topics:

- [Start the Wizard](#)
- [Select an RPC Environment](#)
- [Create RPC Environment \(Optional\)](#)
- [Select Data Set \(Optional\)](#)
- [Select Source and Extract](#)

### Start the Wizard

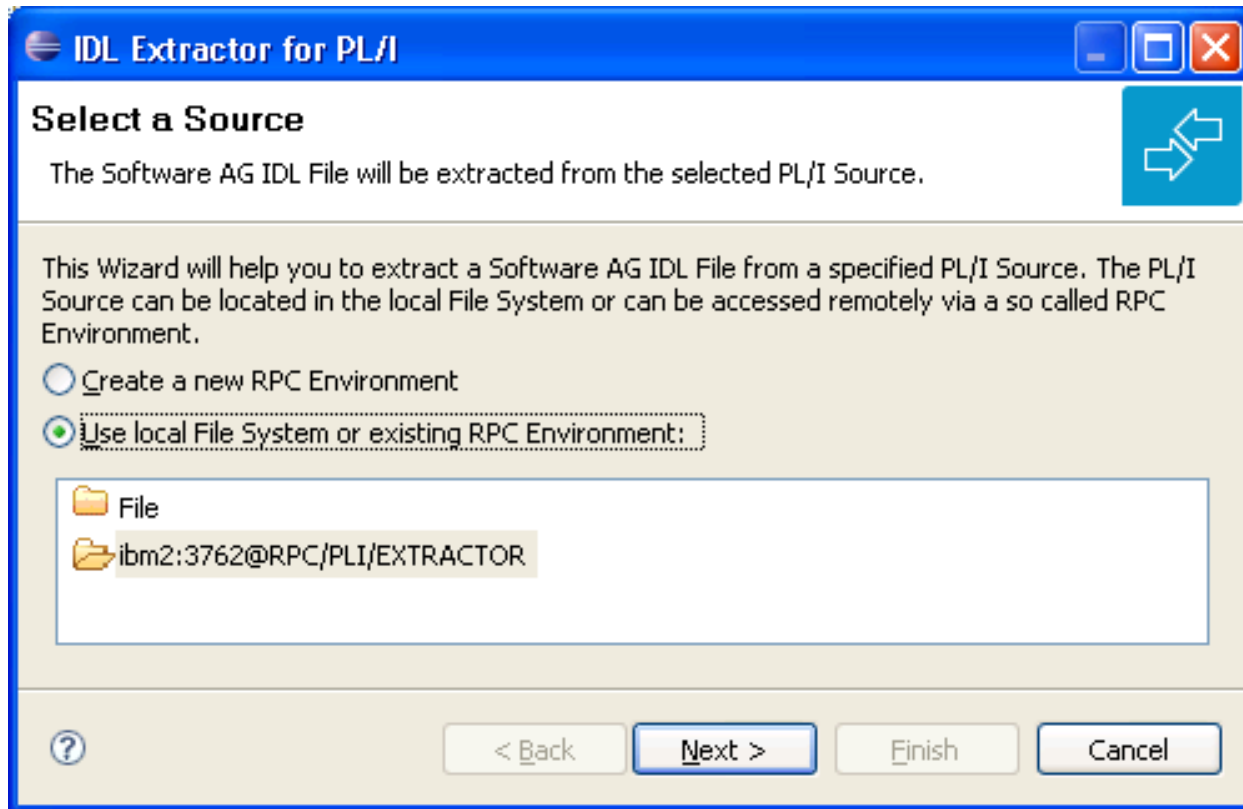
Start the **IDL Extractor for PL/I Wizard**.



### 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 RPC Server documentation for Batch | IMS | BS2000.

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.



**New RPC Environment for IDL Extractor for PL/I**  
Define a new RPC Environment.

**Broker Parameters**  
 Broker ID: myhost:1971  
 Server Address: \*RPC/PLISRV1/CALLNAT   
 Timeout (Seconds): 60

**EntireX Authentication**  
 User ID:   
 Password:

**RPC Server Authentication**  
 RPC User ID:   
 RPC Password:

**Extractor Settings**  
 Enter names, or use filter for range of values (wildcards \* and ? on any position, < and > as final character only).  
 Data Set Name: ETS\*  
 File Name:

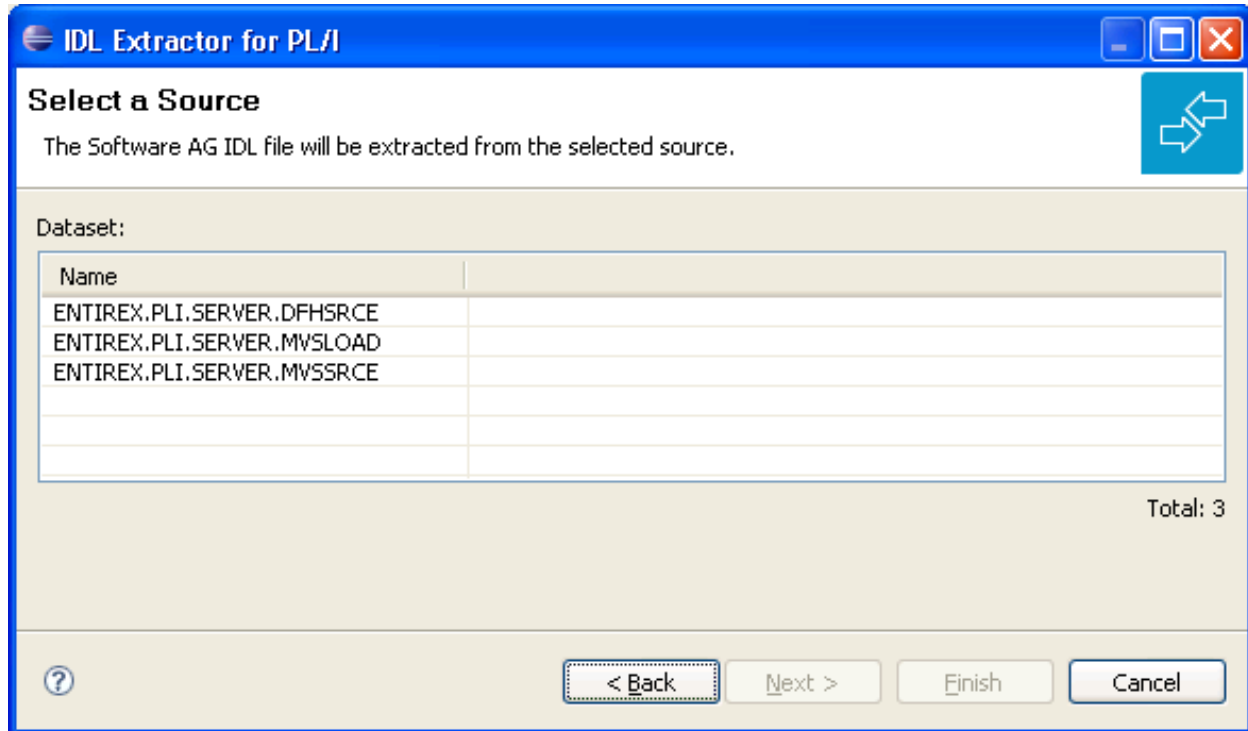
**Wrapper Settings**  
 Save locally  
 Save remotely  
 Target Library Name:

**Environment Name**  
 Default (myhost:1971@RPC/PLISRV1/CALLNAT)  
 Other: myhost:1971@RPC/PLISRV1/CALLNAT

The RPC environments are managed in the Preferences. See [RPC Environment Manager for IDL Extractor for PL/I](#).

### 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**.

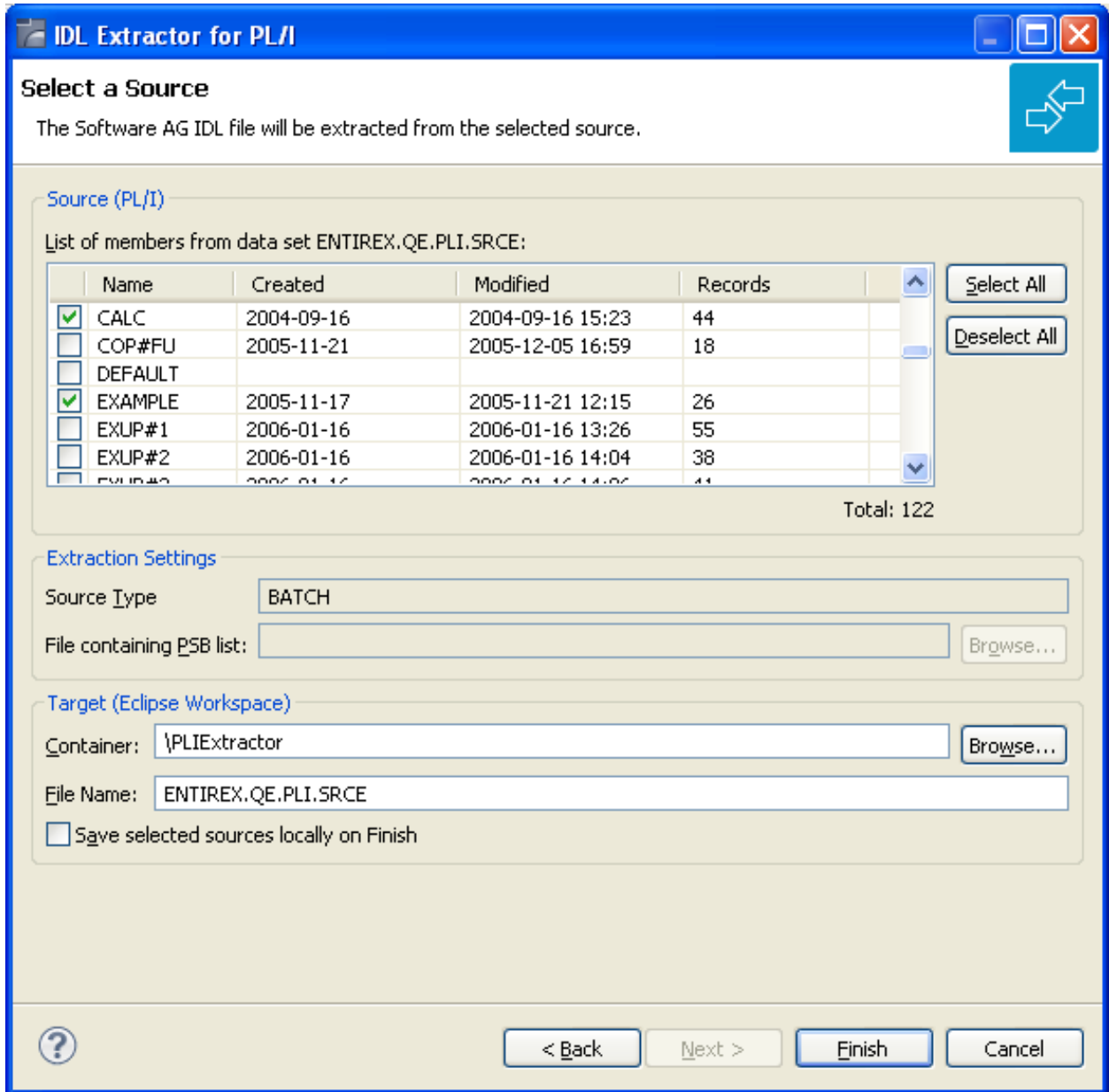


## Select Source and Extract

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 RPC Server for IMS 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 RPC Server for IMS. 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.

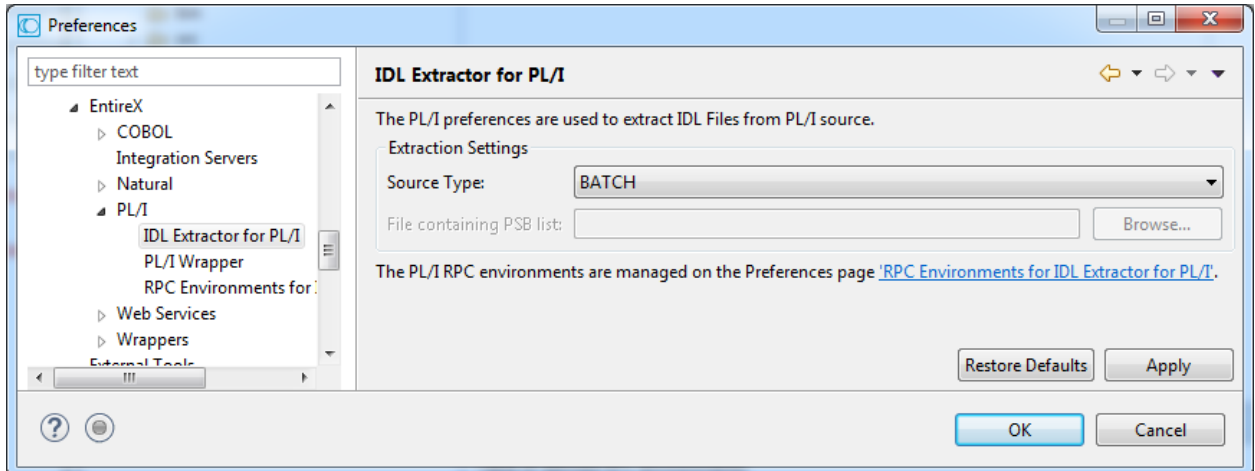
## Preferences

---

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.





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

---

- Command-line Options ..... 20
- Example ..... 21

## Command-line Options

See *Using EntireX in the Designer 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 EntireX design-time starter as described under *Using EntireX in the Designer 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.



# 4 PL/I to IDL Mapping

---

- IDL Extractor for PL/I Input ..... 24
- IDL Extractor for PL/I Output ..... 24
- Mapping PL/I Data Types to Software AG IDL ..... 24
- Functions ..... 26
- Structures ..... 26
- Arrays ..... 27
- Aligned ..... 27
- PL/I to IDL Restrictions ..... 27

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 ( <i>n</i> )	<i>An</i>	Alphanumeric	1
CHARACTER (*)	AV	Alphanumeric variable length	
CHARACTER ( <i>n</i> ) VARYING	AV <i>n</i>	Alphanumeric variable length with maximum length	1
GRAPHIC ( <i>n</i> )	K( <i>n</i> *2)	Kanji fixed length	2
GRAPHIC (*)	KV	Kanji variable length	
GRAPHIC ( <i>n</i> ) VARYING	KV( <i>n</i> *2)	Kanji variable length with maximum length	2
BIT ( <i>n</i> )	B( <i>n</i> /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( <i>m</i> )9[V( <i>n</i> )9]'	N <i>m</i> [. <i>n</i> ]	Unpacked decimal	4
PIC '( <i>m</i> )9[V( <i>n</i> )9]'	NU <i>m</i> [. <i>n</i> ]	Unpacked decimal unsigned	4
FIXED DECIMAL ( <i>m</i> , <i>n</i> )	P( <i>m</i> - <i>n</i> )[. <i>n</i> ]	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  $n \leq 7$ ,  $n \leq 29$  and  $m+n \leq 29$ .
5. *m, n*, are numbers, where  $n \leq 7$  and  $m \leq 29$ .

## 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;
```

This 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);
```

This example will be mapped to the following 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);

```

This example 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.

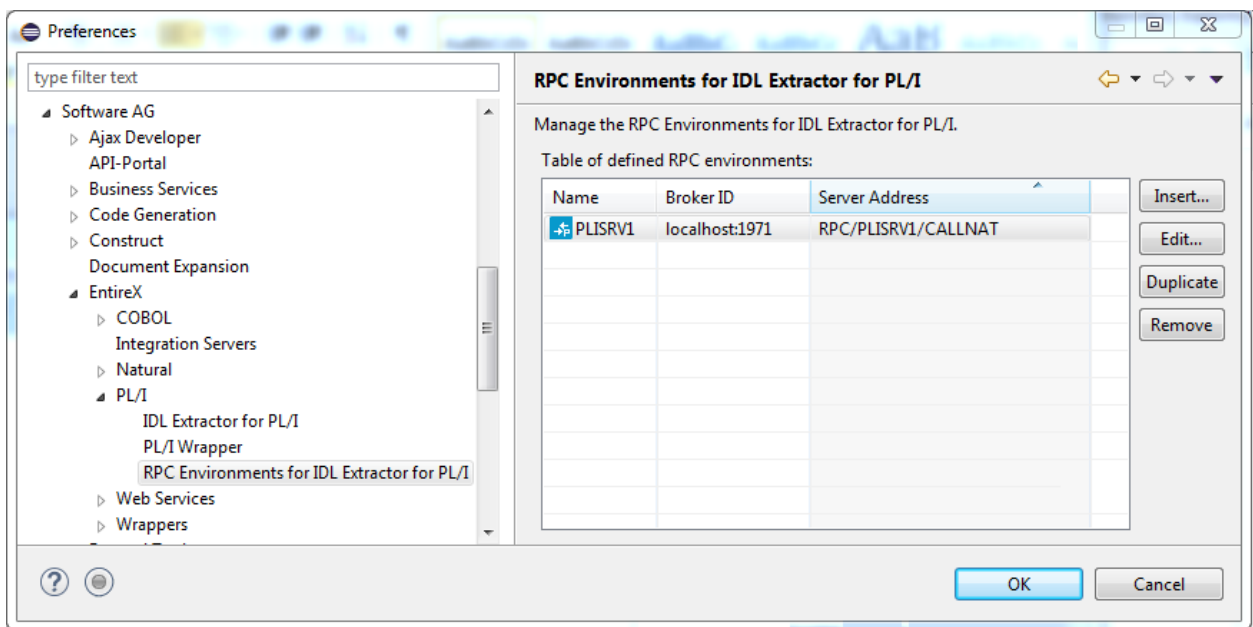


# 5 RPC Environment Manager for IDL Extractor for PL/I

The RPC environment for IDL Extractor for PL/I is managed on the **Preferences** page.

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

Using this wizard, you can add new RPC environments for IDL Extractor for PL/I. To manage these RPC environments, open the **Preferences** page.



## > To edit an existing RPC environment

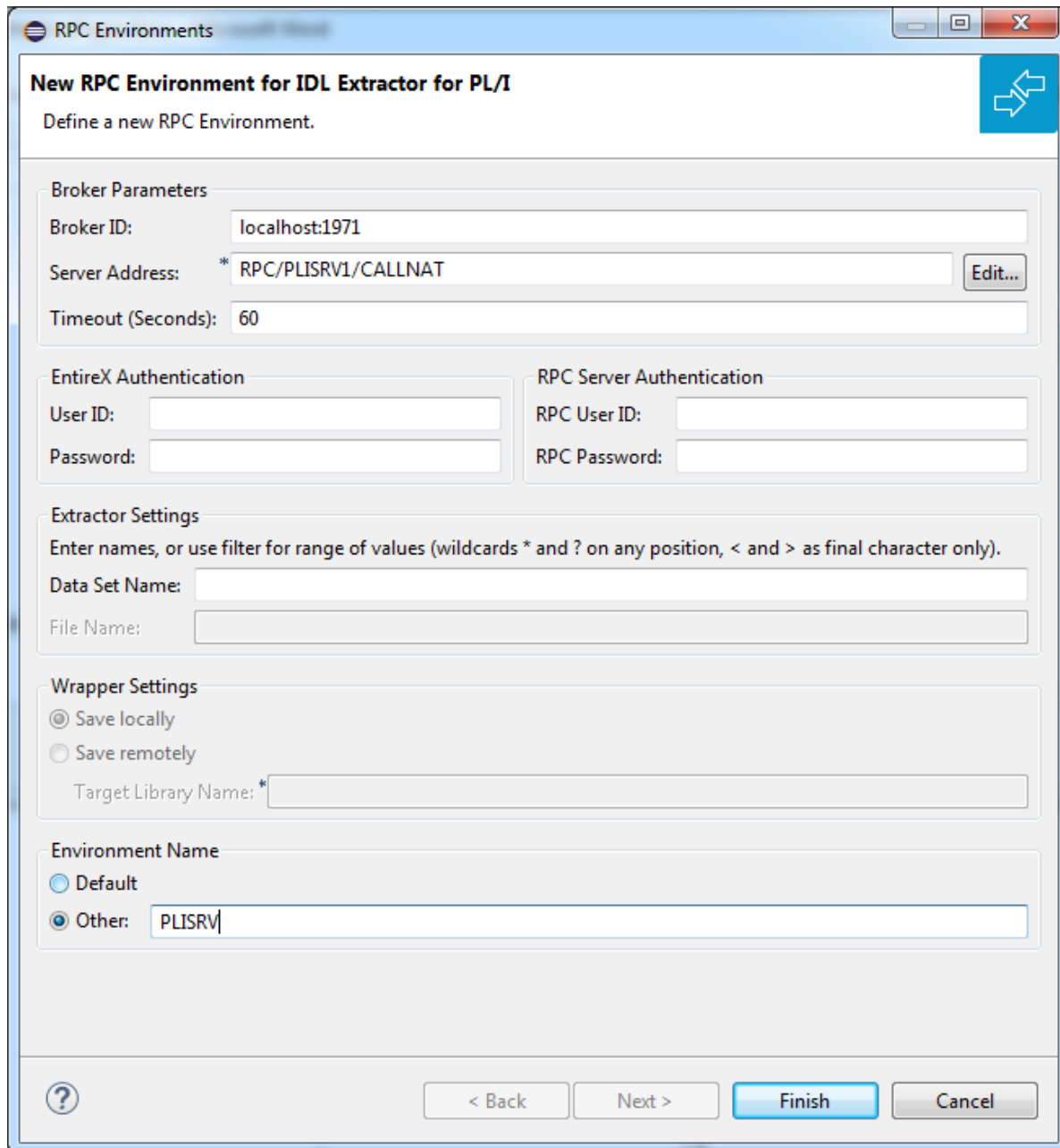
- Select the table row and press **Edit....** If multiple entries are selected, the first entry is used.

> To remove an RPC environment

- Select the table row and press **Remove**. You can select multiple environments.

> To create a new RPC environment

- 1 Choose **Insert...** to call the following screen:



- 2 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.

Use **Extraction Settings** to specify the name of the **Data Set/Library** and the **Member/Program** name.



## 6 RPC Environment Monitor

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

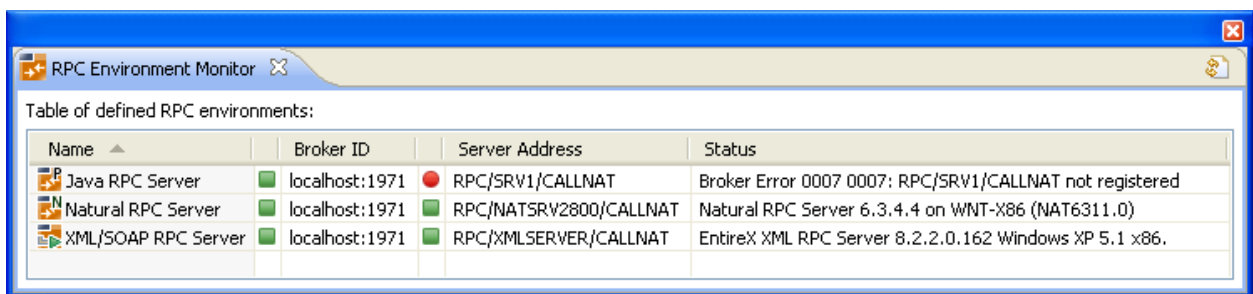
➤ To open the RPC Environment Monitor from the EntireX perspective

- 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 **Preferences** page. See [RPC Environment Manager for IDL Extractor for PL/I](#).



The screenshot shows the 'RPC Environment Monitor' window with a table titled 'Table of defined RPC environments:'. The table has four columns: Name, Broker ID, Server Address, and Status. It lists three RPC servers: Java RPC Server, Natural RPC Server, and XML/SOAP RPC Server. The Java RPC Server is marked with a red circle, indicating an error, while the other two are marked with green squares, indicating they are available.

Name	Broker ID	Server Address	Status
Java RPC Server	localhost:1971	RPC/SRV1/CALLNAT	Broker Error 0007 0007: RPC/SRV1/CALLNAT not registered
Natural RPC Server	localhost:1971	RPC/NATSRV2800/CALLNAT	Natural RPC Server 6.3.4.4 on WNT-X86 (NAT6311.0)
XML/SOAP RPC Server	localhost:1971	RPC/XMLSERVER/CALLNAT	EntireX XML RPC Server 8.2.2.0.162 Windows XP 5.1 x86.

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).