

webMethods EntireX

Merging and Refactoring Software AG IDL

Version 10.3

October 2018

This document applies to webMethods EntireX Version 10.3 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 © 1997-2018 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>.

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-EEXXMERGEREFACOR-103-20191129

Table of Contents

1 About this Documentation	1
Document Conventions	2
Online Information and Support	2
Data Protection	3
2 Merging and Refactoring Software AG IDL	5
Refactoring a Single IDL File	6
Merging and Refactoring Multiple IDL Files	8
Notes on Merging	10
Command-line Mode	11

1 About this Documentation

▪ Document Conventions	2
▪ Online Information and Support	2
▪ Data Protection	3

Document Conventions

Convention	Description
Bold	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

Software AG Documentation Website

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

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at <https://empower.softwareag.com/>.

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](#).

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at https://empower.softwareag.com/public_directory.asp and give us a call.

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

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 Merging and Refactoring Software AG IDL

▪ Refactoring a Single IDL File	6
▪ Merging and Refactoring Multiple IDL Files	8
▪ Notes on Merging	10
▪ Command-line Mode	11

IDL refactoring is a process that checks all programs and structures in a single library if they contain identical groups. All identical groups are extracted in a single structure in the same library, and replaced with a structure reference. If a structure exists that is identical to the structure to be created, all references will point to the existing structure and a new one will not be created. Two groups are identical if each group has the same number and order of parameters, and each parameter in one group has the same name and the same type as the corresponding parameter in the other group. IDL refactoring can be performed on single or multiple IDL files.

Refactoring a Single IDL File

In the case of a single IDL file, all programs and structures in every single library are checked if they contain identical groups. You will be asked for the name of the new IDL file.

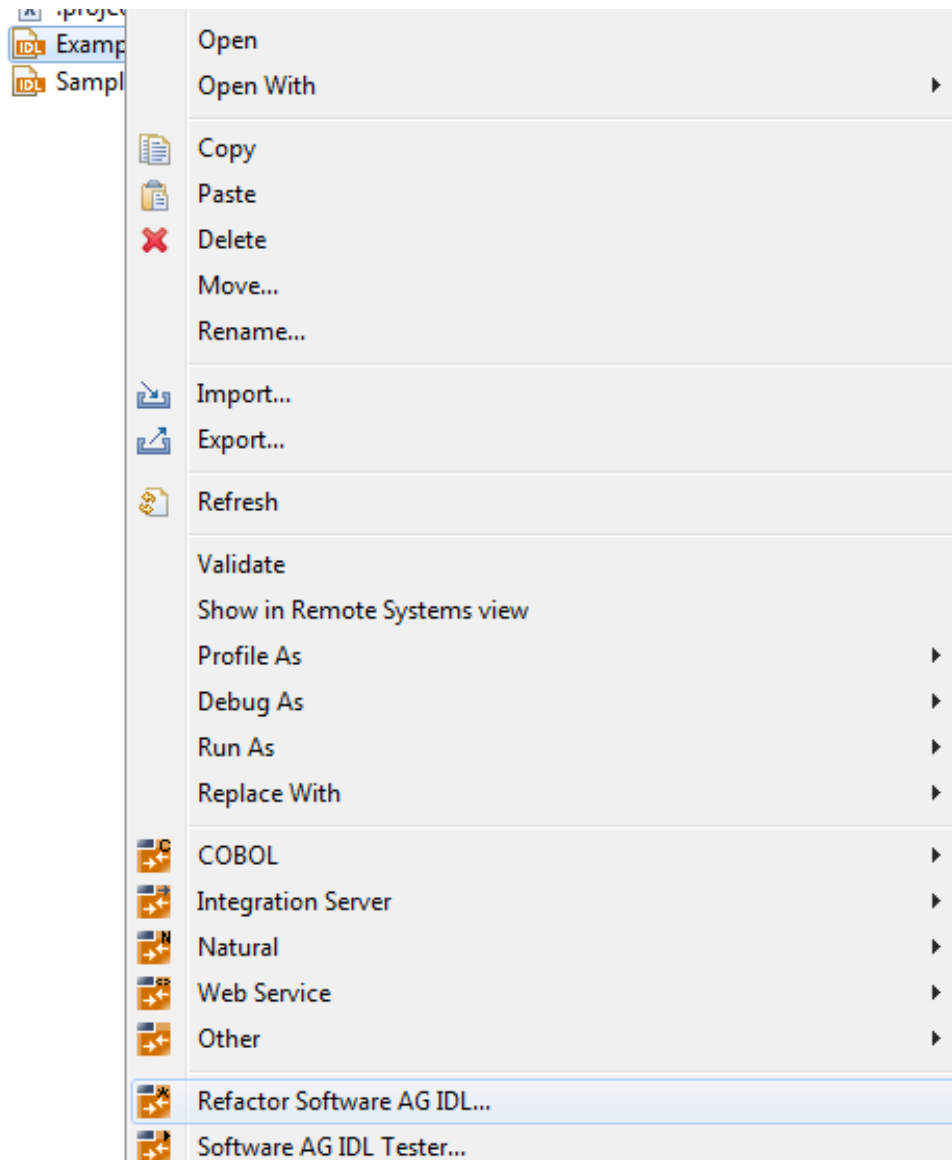
To illustrate refactoring, create an example IDL file *Example.idl*:

```
library 'EXAMPLE' is
  program 'SUM' is
    define data parameter
      1 Operands
        2 Operand1      (I4)  In
        2 Operand2      (I4)  In
      1 Function_Result (I4)  Out
    end-define

  program 'SUBTRACTION' is
    define data parameter
      1 Ops
        2 Operand1      (I4)  In
        2 Operand2      (I4)  In
      1 Function_Result (I4)  Out
    end-define

  program 'MULTIPLICATION' is
    define data parameter
      1 Operands
        2 Operand      (I4)  In
        2 Multiplier   (I4)  In
      1 Function_Result (I4)  Out
    end-define
```

By selecting this file in the Designer, the **Refactor Software AG IDL...** command in the context menu is enabled.



Executing the command and entering the target IDL file will result in a file *Example.refactored.idl*:

```

Library 'EXAMPLE' is
  struct 'Operands' is
    define data parameter
      1 Operand1    (I4)
      1 Operand2    (I4)
    end-define

  program 'SUM' is
    define data parameter
      1 Operands      ('Operands')  In Out
      1 Function_Result (I4)         Out
    end-define
  
```

```
program 'SUBTRACTION' is
  define data parameter
    1 Ops          ('Operands')  In Out
    1 Function_Result (I4)      Out
  end-define

program 'MULTIPLICATION' is
  define data parameter
    1 Operands    In Out
      2 Operand      (I4)
      2 Multiplier  (I4)
    1 Function_Result (I4) Out
  end-define
```

As can be seen from above, the common group with parameters

```
Operand1    (I4)
Operand2    (I4)
```

is extracted as a single structure and the former groups are transformed to structure references. However, the group `Operands` from the `MULTIPLICATION` program is not replaced, because its members have different names, although parameters are equal in quantity and type.

Now, let us assume the *example.idl* file already contains a structure such as:

```
struct 'strct' is
  define data parameter
    1 Operand1    (I4)
    1 Operand2    (I4)
  end-define
```

Although its name differs, the structure's signature (number of parameters, parameter names and types) is the same as the one to be created. In this case, a new structure will not be created, but all references will point to the existing one - `strct`.

Merging and Refactoring Multiple IDL Files

If there is more than one IDL file, all files are first merged and then refactoring is run on the assembled file. You will be asked for the new IDL file's name.

To illustrate IDL merging, create two IDL files *Example1.idl* and *Example2.idl* with the following content:

■ Example1.idl

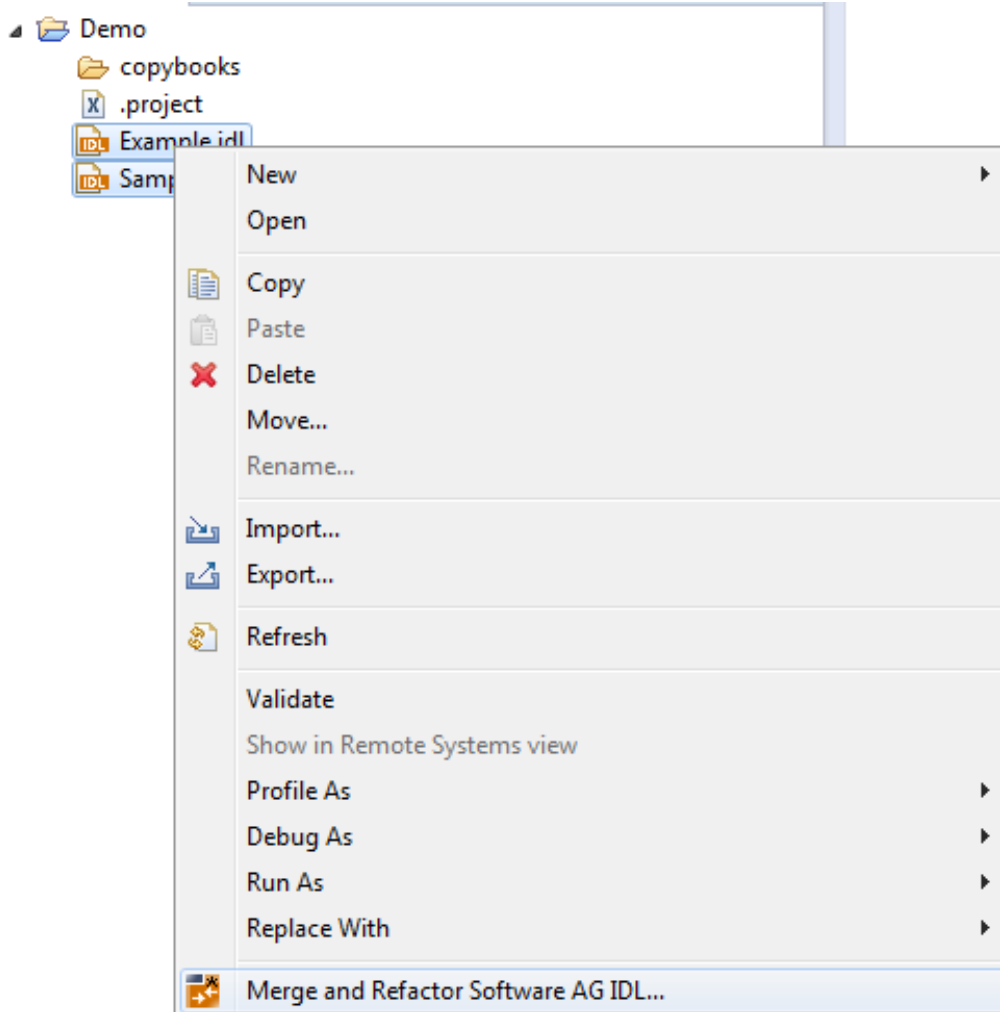
```
library 'EXAMPLE' is
  program 'SUM' is
    define data parameter
      1 Operands
        2 Operand1      (I4) In
        2 Operand2      (I4) In
      1 Function_Result (I4) Out
    end-define

    program 'MULTIPLICATION' is
      define data parameter
        1 Operands
          2 Number       (I4) In
          2 Multiplier   (I4) In
        1 Function_Result (I4) Out
      end-define
```

■ Example2.idl

```
library 'EXAMPLE' is
  program 'SUBTRACTION' is
    define data parameter
      1 Ops
        2 Operand1      (I4) In
        2 Operand2      (I4) In
      1 Function_Result (I4) Out
    end-define
```

Selecting these two files in the Designer will enable the **Merge and Refactor Software AG IDL...** command in the context menu.



Executing this command and entering the target IDL name will result in exactly the same file content as *Example.refactored.idl*. As the two source IDL files have libraries with one and the same name, an attempt to merge their programs is made. If successful, the result is a single `EXAMPLE` library containing all programs from the both libraries. This resulting library is then refactored as described above.

Notes on Merging

- Two libraries with different names will be simply copied into the target IDL file.
- For libraries with same names, an attempt to merge their programs and structures will be made. For a merge operation to be successful, the name of each program and structure must be unique. The only exception are pairs of programs or structures that are exactly the same. For example, if *Example2.idl* above contains the `MULTIPLICATION` program from *Example1.idl*, merging will be successfully completed.

Command-line Mode

See *Using EntireX in the Designer Command-line Mode* for the general command-line syntax. There are no specific command-line options for IDL Refactoring. If a single IDL file is passed, it will be refactored. If more than one IDL file is passed, they will be merged into one IDL file and it will be refactored. Example:

```
<workbench> -idl:refactor C:/Demo/example1.idl C:/Demo/example2.idl
```

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

