

**ARTIST**  
**FP7-317859**



***Advanced software-based seRvice provisioning and  
migraTion of legacy Software***

---

**ARTIST Tool Suite for Enterprise Architect**  
Installation & User Guide

---

|                                     |                 |
|-------------------------------------|-----------------|
| <b>Editor(s):</b>                   | Konrad Wieland  |
| <b>Responsible Partner:</b>         | SparxSystems CE |
| <b>Software-Version:</b>            | v0.1.109        |
| <b>Date:</b>                        | 25/09/2014      |
| <b>Distribution level (CO, PU):</b> | Public          |

|                        |            |
|------------------------|------------|
| <b>Project Number:</b> | FP7-317859 |
| <b>Project Title:</b>  | ARTIST     |

|                               |   |
|-------------------------------|---|
| <b>Abstract:</b>              | <p>The Artist Tool Suite for Enterprise Architect contains different Plug-Ins for Enterprise Architect developed within the Artist project. This documents describes how to install and use these plug-ins.</p>   |
| <b>Licensing information:</b> | <p>This tool suite is licensed under MIT-License.</p> <p>Copyright (c) 2014 SparxSystem Software GmbH</p> <p>Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:</p> <p>The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.</p> <p>THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.</p> |

---

---

# Table of Contents

---

---

Table of Contents ..... 3

1 Installation..... 4

    1.1 Prerequisites ..... 4

    1.2 Installation Steps..... 4

    1.3 Uninstall ..... 5

2 User Guide..... 5

    2.1 Model Preparation ..... 5

        2.1.1 Support for referenced parents ..... 5

        2.1.2 Type Checker and Resolver ..... 5

    2.2 EA-Eclipse Bridge..... 6

        2.2.1 Generic profile for exporting .Net specifics ..... 6

    2.3 Dependency Generator ..... 7

    2.4 Code Generation incl. Model Validation ..... 7



# 1 Installation

The Artist Tool Suite for Enterprise Architect (EA) can be installed using the MSI-file. This file can be downloaded from the public Artist Git Hub repository.

The installation wizard helps you to install the Artist Tool Suite for EA correctly.

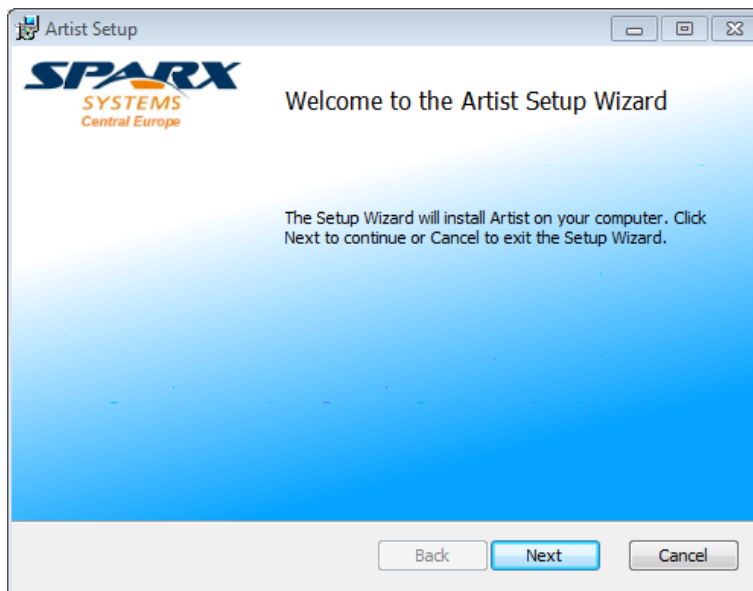
## 1.1 Prerequisites

The following tools are prerequisites for installing the tool suite:

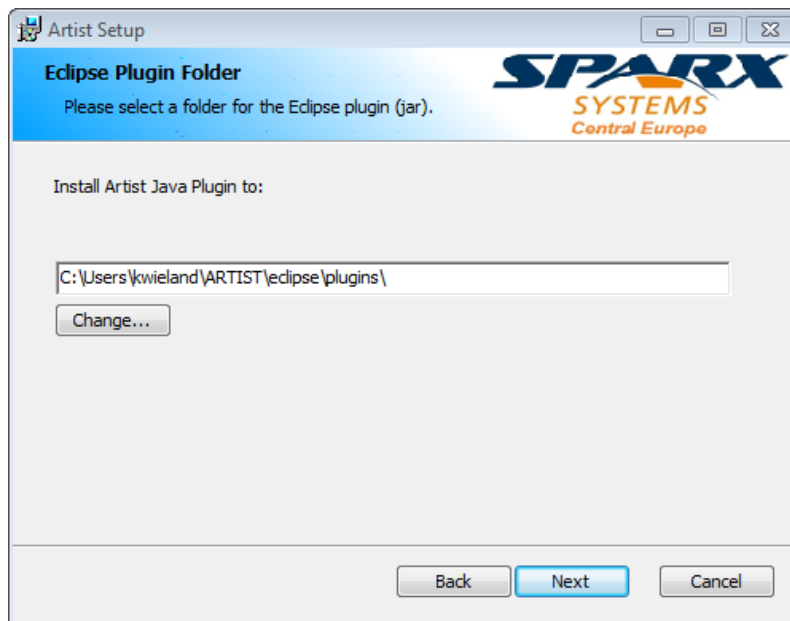
- Windows 7 or higher
- Enterprise Architect 9 or higher
- Eclipse Juno or Kepler (newer versions are not tested)

## 1.2 Installation Steps

Double-click the ArtistSetup.msi file for starting the installation wizard and click on “Next”:



In the next step, please select the “plugins” folder of your eclipse bundle. In this folder a jar file is copied. Please note: The Eclipse plugin (jar) is also copied the installation folder of the tool suite under C:\Program Files (x86)\SparxSystems\Artist\ClientSources.



Click on “Next” and then “Finish” to complete the installation.

You can now find under C:\Program Files (x86)\SparxSystems\Artist the installed tool suite.

## 1.3 Uninstall

If you want to uninstall the tool suite, please open the “Program and Features” overview of your windows machine.

|                         |                            |            |         |              |
|-------------------------|----------------------------|------------|---------|--------------|
| AMUSE Pro Edition       | LieberLieber Software G... | 23.09.2014 | 26,9 MB | 2.2.11       |
| AnkhSVN 2.5.12471.17    | AnkhSVN Team               | 01.08.2014 | 8,87 MB | 2.5.12471.17 |
| Artist                  | SparxSystems CE            | 25.09.2014 | 13,3 MB | 0.1.109      |
| assignPDFverify 1.0.9.0 | A-Trust                    | 20.01.2014 |         | 1.0.9.0      |
| calibre 64bit           | Kovid Goyal                | 17.07.2013 | 164 MB  | 0.9.39       |

Select “Artist” and then click on “Uninstall” for uninstalling the tool suite and deleting all data from your computer.

## 2 User Guide

### 2.1 Model Preparation

#### 2.1.1 Support for referenced parents

If your classes in your EA model have references to external classes which are not part of the model, you can use this plug-in to automatically generate placeholder classes for them. Right-click on the namespace root package (indicated with a small red icon) in the Project Browser and select “Extension -> Artist -> Generate Parent Classes”. These classes now can be exported together with your model.

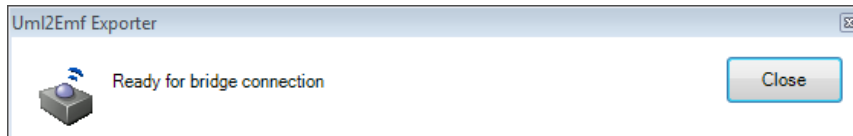
#### 2.1.2 Type Checker and Resolver

When reverse engineering source code with EA, it might happen, especially when importing additional frameworks, that types of operations, parameters, etc. are not referenced. Instead of references string values are used. With this Artist feature you can resolve these issues

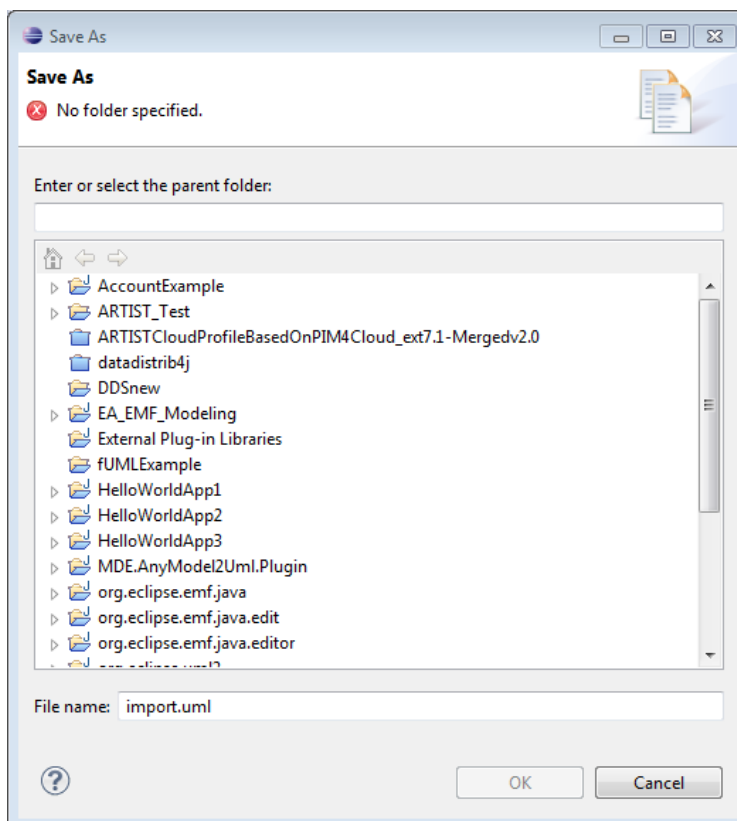
automatically. In addition, the namespace structure is also considered. These features are executed before the parent classes are generated (cf. section 2.1.1) by right-clicking a package in the Package Browser and by selecting “Extension -> Artist -> Generate Parent Classes”.

## 2.2 EA-Eclipse Bridge

To export the data to Eclipse, right-click a package and select “Extensions -> Artist -> UML2EMFExporter”. The Exporter-Service is opened visualized by the following window:



Inside Eclipse you can start to import the model, by clicking on “Artist -> Import Model from Bridge”. You are forced to select a folder in the Eclipse Package Explorer.



After confirming, the .uml file is created in the specified folder. In addition, a UML profile is generated as described above. After successfully creating the file, the bridge connection is closed automatically.

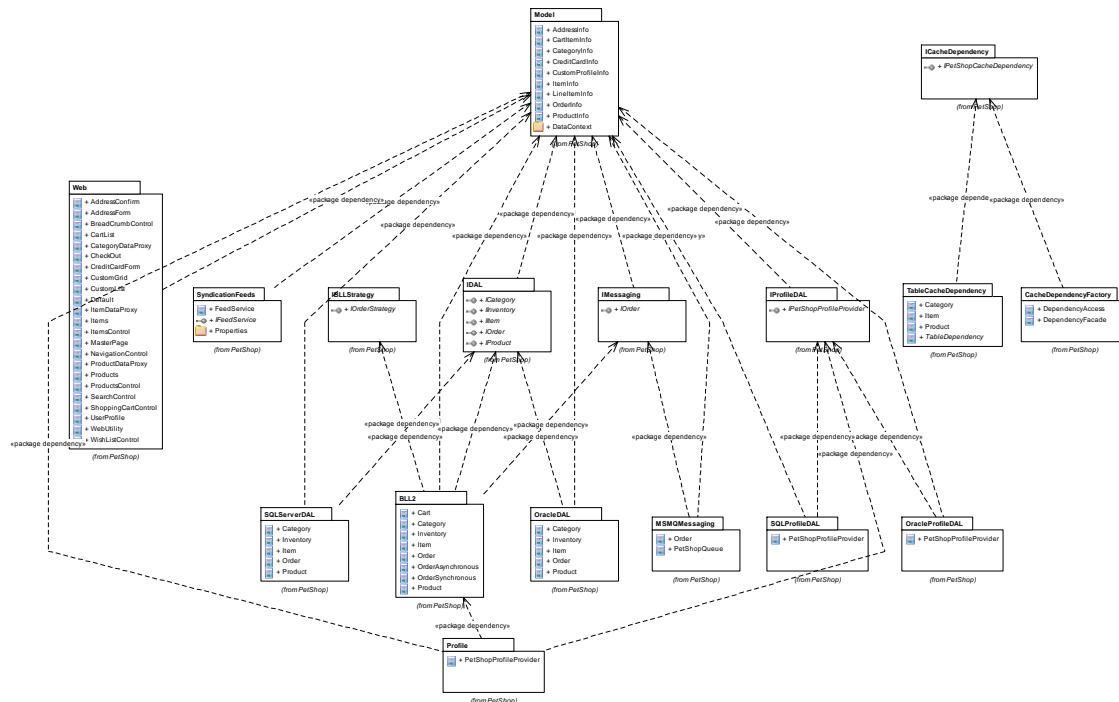
Please note: Currently only UML Class Diagrams are supported.

### 2.2.1 Generic profile for exporting .Net specifics

To cover all C# specifics inside a UML Class model, Enterprise Architect is storing this information as so-called Tagged Values to the dedicated Elements. These elements can be classes, interfaces, enumeration, operations or attributes. To export this information, a generic C# UML profile is automatically created inside Eclipse and its stereotypes are applied to the dedicated elements.

## 2.3 Dependency Generator

With the Artist Tool Suite you are able to visualize the dependencies your model packages to better understand your already imported legacy application. Right-click a package in the project browser and select “Extensions -> Artist -> Resolve Package Dependencies”. A diagram is automatically generated visualizing the dependencies between your packages. The following diagram is generated for the PetShop Example used in the Artist project.

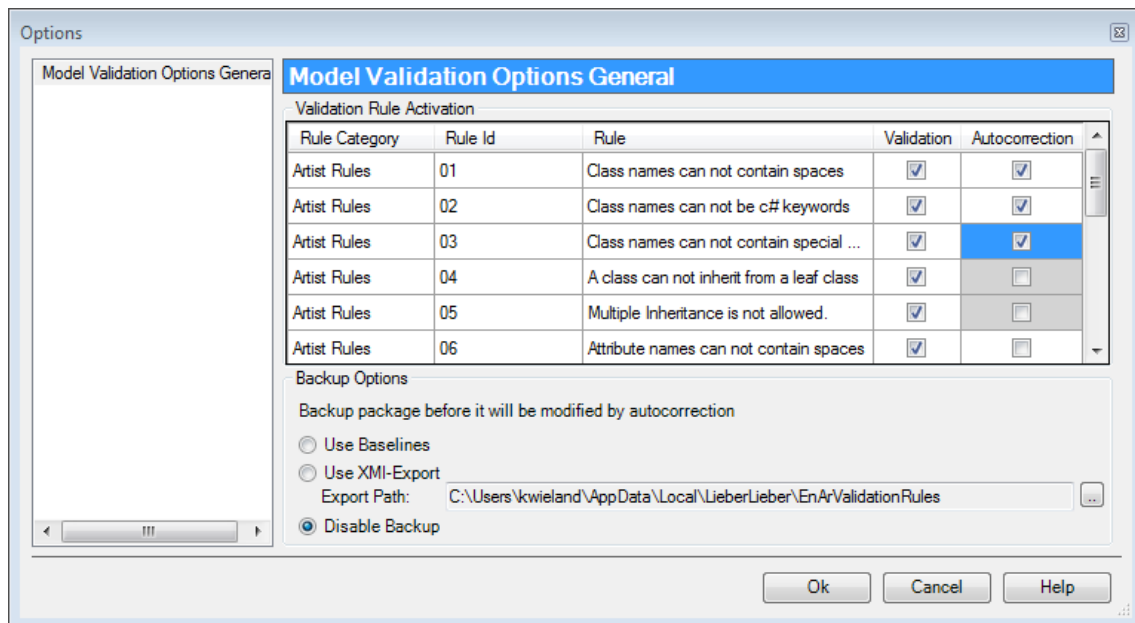


You can delete the generated dependency relationships by right-clicking a package and by selecting “Extensions -> Artist -> Delete Package Dependencies”.

## 2.4 Code Generation incl. Model Validation

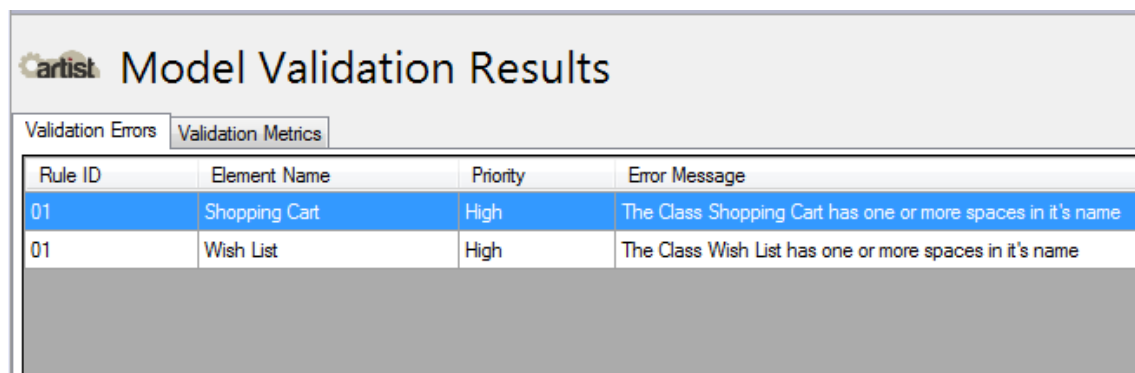
Before generating code out of an EA model, it can be validated to end up with compileable source code.

If the tool suite is loaded successfully you can find a dedicated menu entry under “Extensions” inside Enterprise Architect. For activating or deactivating validation rules you can click on “Extensions -> Artist -> Options”:



In this dialog, you can also activate and deactivate auto-correction for the validation rules. In addition, you can also specify the backup options. Before the tool is modifying the model by auto-correction it exports the model as so called baseline or as XMI file. The validation itself can be started by right-clicking a package in the model and by selecting “Extensions -> Artist -> Model Validation -> Start”.

The detected errors are listed in the “Model Validation Results” window:

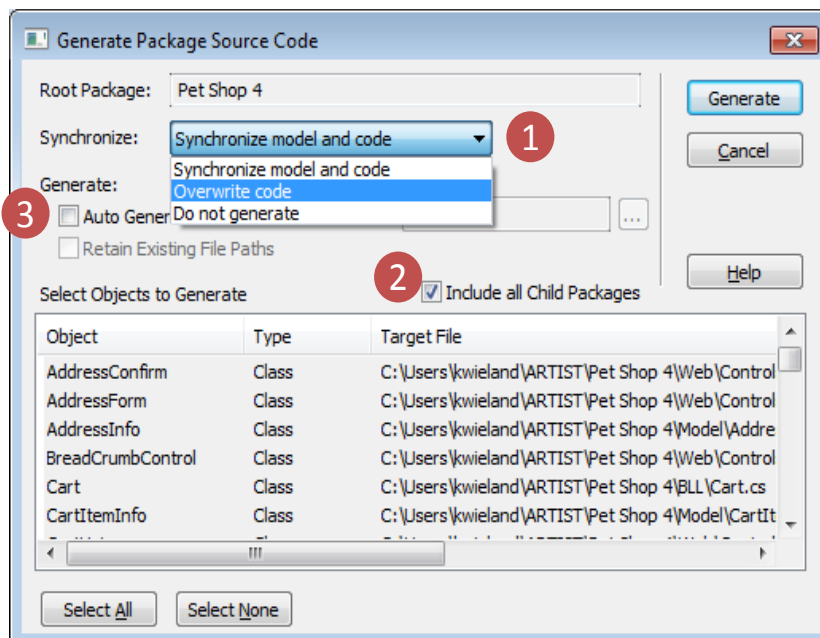


In addition, some basic metrics are provided. Rules can be weighted that is influencing the final result.



| Model Validation Results  |  |                    |                |        |
|---|--|--------------------|----------------|--------|
| Validation Errors   |  | Validation Metrics |                |        |
| Rule ID   | Rule Name  | Checked Elements   | Wrong Elements | Metric |
| 01  | Class names can not contain spaces   | 50                 | 2              | 96%    |
| 02  | Class names can not be c# keywords   | 50                 | 0              | 100%   |
| 03  | Class names can not contain special characters or start with numbers       | 50                 | 0              | 100%   |
| 04  | A class can not inherit from a leaf class                                  | 10                 | 0              | 100%   |
| 05  | Multiple Inheritance is not allowed.                                       | 10                 | 0              | 100%   |
| 06  | Attribute names can not contain spaces                                     | 187                | 0              | 100%   |
| 07  | Attribute names can not be c# keywords                                     | 187                | 0              | 100%   |
| 08  | Attribute names can not contain special characters or start with numbers   | 187                | 0              | 100%   |
| 09  | Attributes must not have the same name as Methods                          | 50                 | 0              | 100%   |
| 10  | Operation names can not be c# keywords                                     | 306                | 0              | 100%   |
| 11  | Operation names can not contain spaces                                     | 306                | 0              | 100%   |
| 12  | Operation names can not contain special characters or start with a number. | 306                | 0              | 100%   |
| 13  | Operations can not be overloaded with the same signature                   | 39                 | 0              | 100%   |
| 14  | Parameter names can not be c# keywords                                     | 443                | 0              | 100%   |
| 15  | Parameters must have names without spaces                                  | 443                | 0              | 100%   |
| 16  | Parameter names can not contain special characters or start with a number  | 443                | 0              | 100%   |
| Model Quality Metric: 99,57%  |  |                    |                |        |
| <div> <div>4</div> <div>Start Page enar Model Validation Results</div> <div></div> </div> |  |                    |                |        |

If you now right-click a package in the project browser and select “Code Engineering -> Generate Source Code” this dialog is opened:



Here, you have basically three different options:

1. If you want to follow a Roundtrip Engineering approach, that is to synchronize the model and code, both are updated accordingly to your changes. You can

- also select “Overwrite code”. With this option you override the before generated code.
2. If your model is structured with the help of packages, the “Include all Child Packages” checkbox should be selected for recursively looping all elements in the model for which code is generated.
  3. Finally, if you want to specify a specific folder in which the code is generated, you have to select the “Auto Generate Files” checkbox. If activated you are forced to select a folder in your file system.

By clicking on “Generate” the source code is generated for each package, class, interface or enumeration including their properties, attributes and operations.

For each package in the model, an equally named folder is generated. You can change the namespace root by right-clicking a package in the project browser and by selecting “Code Engineering -> Set Namespace Root”.

After generating code, you can view the source code inside EA by selecting an element e.g. class, in the project browser and click on “F12”. The code editor including syntax highlighting is opening.

As described above, the source code templates can be adopted following your specific needs by clicking on “Tools -> Source Code Generation Templates...”.