Java Add-In will guarantee to maximize the productivity and quality because it helps a tool to adapt Java code generation and reverse engineering easily and quickly after it transforms UML model to Java source and Java source to UML model.

Index of Java Add-In Help

- 1. Java Add-In Overview
- 2. Java Profile
- 3. EJB Profile
- 4. J2EE Framework
- 5. Option Configurations
- 6. Java Reverse Engineering
- 7. Java Code Generation
- 8. <u>FAQ</u>

This chapter discusses a general overview of Java Add-In: functions and configuration procedures.

- Java Add-In Functions
- <u>Java Add-In Configurations</u>

Java Add-In provides the following functions.

Java Profile

Java profile is provided to allow visual modeling of Java concepts such as package, class, interface, imports, modifier and so on in StarUML(tm). Include Java profile when you start your StarUML(tm) project, in order to apply the features of the Java language in your software modeling.

EJB Profile

In addition to Java profile, Java Add-In provides EJB profile for the enterprise development environment.

Java Framework (J2EE)

JJava Add-In provides Sun Microsystems' J2EE (Java 2 Enterprise Edition) in the Model Framework format.

Java Reverse Engineering

Java Add-In provides the reverse engineering function that generates StarUML(tm) models by analyzing source codes written in Java.

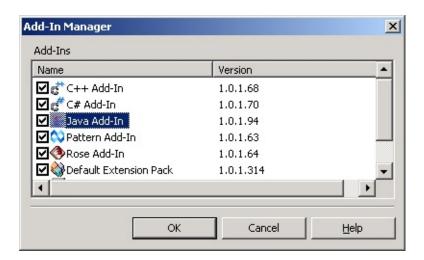
Java Code Generation

Java Add-In provides the forward engineering function that generates Java source codes by analyzing StarUML(tm) models.

Once Java Add-In is installed, it is enabled for use in StarUML(tm) by default. The installed Add-Ins can be enabled or disabled through Add-In Manager in StarUML(tm). If an Add-In is disabled, no main menu and popup menu items related to it are displayed, and no StarUML(tm) events are relayed to it.

Procedure for Enabling Java Add-In:

- 1. Select the **[Tools] -> [Add-In Manager...]** menu in StarUML(tm).
- 2. At the Add-In Manager dialog box, check the "Java Add-In" checkbox in the Add-In list.



3. Click the **[OK]** button to close the dialog box.

Procedure for Disabling Java Add-In:

- 1. Select the **[Tools] -> [Add-In Manager...]** menu in StarUML(tm).
- 2. At the Add-In Manager dialog box, uncheck the "Java Add-In" checkbox in the Add-In list.
- 3. Click the **[OK]** button to close the dialog box.

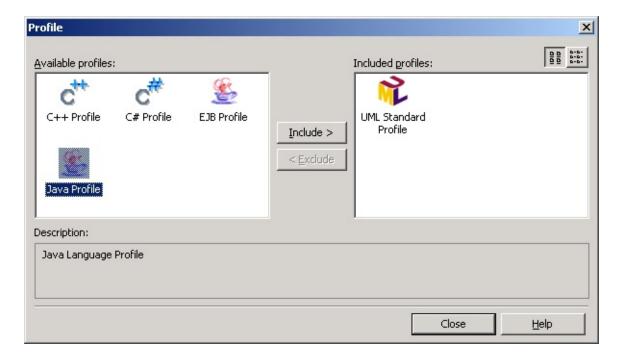
This chapter describes Java profile: configuration procedures and definitions.

- Including Java Profile
- Excluding Java Profile
- <u>Java Profile Definition List</u>

Java profile must be included in the project in order to utilize the Stereotypes, TagDefinitions, and DataTypes defined in Java profile.

Procedure for Including Java Profile:

- 1. Select the **[Model] -> [Profiles...]** menu.
- 2. At the Profile Manager window, select "Java Profile" from the "Available profiles" list on the left.



- 3. Click the **[Include]** button or hit Alt-I to move "Java Profile" to the "Included profiles" list.
- 4. Click the [Close] button to close the Profile Manager window.
- 5. Java profile is included in the current project.

Note

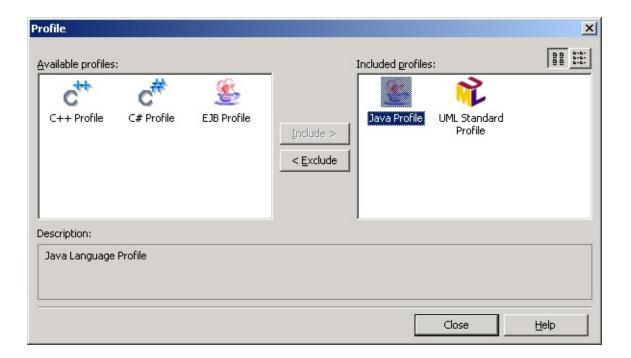
Opening a project with Java profile on another StarUML(tm) system that

 does not have Java profile installed may result in loss of model element extension information (Stereotype, Tagged Value, etc).

Java profile can be excluded from the current project. Once C# profile is excluded, Stereotypes, TagDefinitions and DataTypes defined in the profile cannot be used in the project.

Procedure for Excluding Java Profile:

- 1. Select the **[Model] -> [Profiles...]** menu.
- 2. At the Profile Manager window, select "Java Profile" from the "Included profiles" list on the right..



- 3. Click the **[Exclude]** button or hit Alt-E to remove "Java Profile" from the "Included profiles" list.
- 4. Click the [Close] button to close the Profile Manager window.
- 5. Java profile is excluded from the current project.

 Re-including Java profile after excluding it does not restore the previously edited tagged values of the model elements. 				

Stereotype

Java profile contains definitions for the following stereotypes.

Sterotype	Target Element	Description
< <javasourcefile>></javasourcefile>	Component	Source file with Java code
< <javaclassfile>></javaclassfile>	Component	File with compiled Java source code
< <javaachivefile>></javaachivefile>		JAR file contains the class for Java applet, image files and sound files as one compressed file to allow quick downloading for web browsers.

TagDefinition

Java profile contains definitions for the following tag definitions.

TagDefinition	Type	Target Element	Description	
JavaStrictfp	Boolean	Class, Interface, Operation	Executes all real number calculations by strictly conforming to the IEEE 754 standard.	
JavaStatic	Boolean	Class	Used for loading and unloading the target class to and from the base memory when running JavaVM.	
JavaVolatile	Boolean	Attribute, AssociationEnd	Variables with this keyword are not processed for variable optimization by the compiler.	
JavaTransient	Boolean	Attribute, AssociationEnd	The 'transient' flag allows the interpreter to process class instances as persistent objects.	
JavaCollection	String	Attribute, AssociationEnd	Defines the Java Collection type name.	
JavaDimensions	Integer	Attribute, Parameter,	Indicates the array dimension of	

		AssociationEnd	the declared object.
JavaNative	Boolean	Operation	Method that supports JNI (Java Native Interface).
JavaThrows	String		Used for relaying exceptions of the method above to other methods.
JavaFinal	Boolean	Parameter	Indicates constant type arguments.

DataTypes

Java profile contains definitions for the following basic Java data types.



- float
- boolean
- char
- double
- short
- long
- byte
- void
- Boolean

- Byte
- Character
- Double
- Float
- Integer
- Long
- Object
- Short
- String
- Void

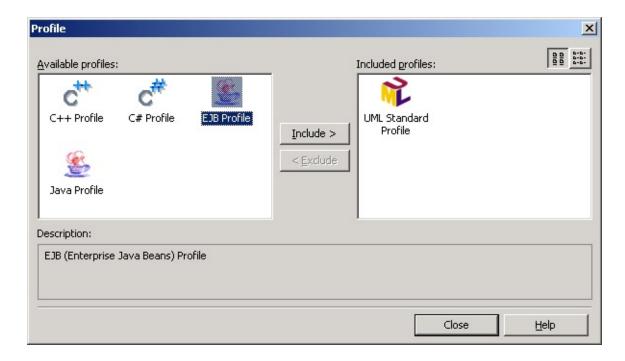
This chapter describes EJB (Enterprise JavaBeans) profile: configuration procedures and definitions.

- Including EJB Profile
- Excluding EJB Profile
- EJB Profile Definition List

EJB profile must be included in the project in order to utilize the Stereotypes, TagDefinitions, and DataTypes defined in EJB profile.

Procedure for Including EJB Profile:

- 1. Select the **[Model] -> [Profiles...]** menu.
- 2. At the Profile Manager window, select "EJB Profile" from the "Available profiles" list on the left.



- 3. Click the **[Include]** button or hit Alt-I to move "EJB Profile" to the "Included profiles" list.
- 4. Click the **[Close]** button to close the Profile Manager window.
- 5. EJB profile is included in the current project.

Note

Opening a project with EJB profile on another StarUML(tm) system that

• does not have EJB profile installed may result in loss of model element extension information (Stereotype, Tagged Value, etc).				

EJB profile can be excluded from the current project. Once EJB profile is excluded, Stereotypes, TagDefinitions and DataTypes defined in the profile cannot be used in the project.

Procedure for Excluding EJB Profile:

- 1. Select the **[Model] -> [Profiles...]** menu.
- 2. At the Profile Manager window, select "EJB Profile" from the "Included profiles" list on the right.



- 3. Click the **[Exclude]** button or hit Alt-E to remove "EJB Profile" from the "Included profiles" list.
- 4. Click the [Close] button to close the Profile Manager window..
- 5. EJB profile is excluded from the current project.

Note

Re-including EJB profile after excluding it does not restore the previously

• edited tagged values of the model elements.

Stereotype

EJB profile contains definitions for the following stereotypes.

Sterotype	Target Element	Description
< <ejbcreatemethod>></ejbcreatemethod>	Operation	The specified operation indicates EJB Create Method.
< <ejbfindermethod>></ejbfindermethod>	Operation	The specified operation indicates EJB Finder Method.
< <ejbremotemethod>></ejbremotemethod>	Operation	The specified operation indicates EJB Remote Method.
< <ejbremoteinterface>></ejbremoteinterface>	Class, Interface	The specified class indicates EJB Remote Interface.
< <ejbhomeinterface>></ejbhomeinterface>	Class, Interface	An abstract stereotype. The specified class indicates EJB Home Interface.
< <ejbsessionhomeinterface>></ejbsessionhomeinterface>	Class, Interface	An implementation of the abstract stereotype Home Interface. The specified class indicates EJB Session Home.
< <ejbentityhomeinterface>></ejbentityhomeinterface>	Class, Interface	An implementation of the abstract stereotype Home Interface. The specified class indicates EJB Entity Home.
< <ejbprimarykey>></ejbprimarykey>	Dependency	Indicates use of EJB Primary Key Class for EJB Entity Home.
< <ejbcmpfield>></ejbcmpfield>	Attribute	The specified attribute indicates the containermanaged field for the CMP-based EJB Entity Bean.
< <ejbprimarykeyfield>></ejbprimarykeyfield>	Attribute	An implementation of the stereotype EJBCmpField. The specified attribute indicates the

		primary key field for the CMP-based EJB Entity Bean.
< <ejbrealizehome>></ejbrealizehome>	Dependency	Indicates abstraction of EJB Home Interface for EJB Implementation Class.
< <ejbrealizeremote>></ejbrealizeremote>	Dependency	Indicates abstraction of EJB Remote Interface for EJB Implementation Class.
< <ejbimplementation>></ejbimplementation>	Class	The specified Class indicates EJB Implementation Class.
< <ejbenterprisebean>></ejbenterprisebean>	Subsystem	An abstract stereotype. The specified subsystem indicates EJB Enterprise Bean.
< <ejbsessionbean>></ejbsessionbean>	Subsystem	An implementation of the abstract stereotype EJBEnterpriseBean. The specified subsystem indicates EJB Session Bean.
< <ejbentitybean>></ejbentitybean>	Subsystem	An implementation of the abstract stereotype EJBEnterpriseBean. The specified subsystem indicates EJB Entity Bean.
< <ejbreference>></ejbreference>	Association	Association with the specified direction indicates the referenced EJB Enterprise Bean.
< <ejbaccess>></ejbaccess>	Association	The specified association indicates the security regulation relation between the subsystems indicated by the actor and the stereotype EJBEnterpriseBean.
< <ejb-jar>></ejb-jar>	Package	The specified package indicates EJB-JAR.
< <ejbdescriptor>></ejbdescriptor>	Component	The specified component indicates EJB Deployment

		Descriptor.
< <ejbclientjar>></ejbclientjar>	Dependency	Indicates whether ejb-client-jar is using EJB-JAR.

TagDefinition

EJB profile contains definitions for the following tag definitions.

TagDefinition	Type	Target Element	Description
EJBSessionType	Enumeration	Class	
EJBRoleNames	String	Operation	A comma-separated string. Defines the security regulation for the specified operation.
EJBTrasAttribute	Enumeration	Operation	Defines the transaction management policy as Not Supported, Supports, Required, RequiresNew, Mandatory or Never.
EJBEnvEntries	String	SubSystem	Defines environment listings similar to attributes to be read when EJB Enterprise Bean is running. Each list is comma-separated and a single list is in the format of <name, type,="" value="">.</name,>
EJBNameInJAR	String	SubSystem	The name of EJB Enterprise Bean. Generally, the EJB Remote Interface name is used.
EJBReferences	String	SubSystem	Used for defining reference to other Enterprise Beans. Each list is comma-separated

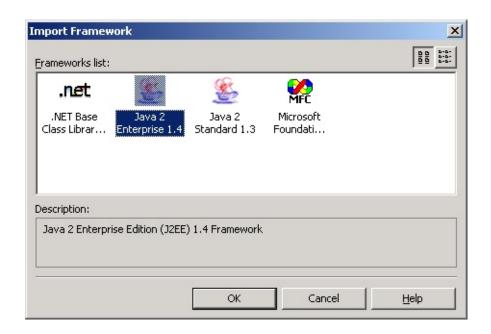
			and a single list is in the format of <name, home,="" remote="" type,="">.</name,>
EJBResources	String	SubSystem	Used for defining the external resources for EJB Enterprise Bean. Each list is commaseparated and a single list is in the format of <name, auth="" type,="">.</name,>
EJBSecurityRoles	String	SubSystem	Defines security regulations for all EJB Enterprise Bean operations. Each list is comma-separated and a single list is in the format of <name, link="">.</name,>
EJBTransType	Enumeration	SubSystem	Indicates whether EJB Session Bean transactions are managed by EJB Session Bean or container. The value is either Bean or Container.
EJBPersistenceType	Enumeration	SubSystem	Indicates whether EJB Entity Bean persistence is managed by EJB Entity Bean or container. The value is either Bean or Container.
EJBReentrant	Boolean	SubSystem	Indicates whether EJB Entity Bean can reenter.

The J2EE (Java 2 Platform, Enterprise Edition) Framework is the J2EE development platform model included in Java 2 Platform, Enterprise Edition. In StarUML(tm), a Model Framework refers to a software model that expresses class libraries or application frameworks such as MFC, VCL, and JFC. Using a framework with the project greatly increases the convenience of the user for modeling software that depends on a certain class library or application framework.

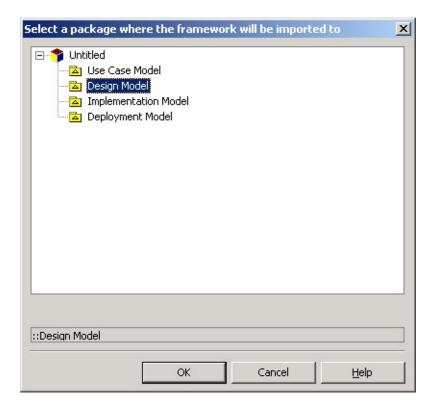
• <u>Using the J2EE Framework</u>

Procedure for Importing the J2EE Framework:

- 1. Select the **[File] -> [Import] -> [Framework...]** menu.
- 2. At the **[Import Framework]** dialog, select the J2EE Framework from the list and click the **[OK]** button.



The Select Element dialog box will appear, to determine under which element the J2EE framework will be located. Select an element (package, model, subsystem or project) to contain the framework and then click the **[OK]** button.



4. The framework is included under the selected element.

✓ Note

Importing a framework does not store the framework elements in the

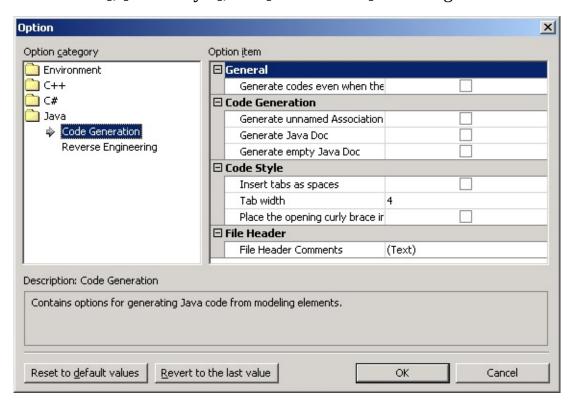
- project. Since framework units are referenced by the project, the framework unit files must be present when opening the project.
- To delete imported frameworks, remove the respective framework units.

This chapter discusses procedures for configuring the Java Add-In environment and describes the option items in detail.

- Code Generation Option Configuration
- Reverse Engineering Option Configuration

Code Generation Option Configuration Top Previous Next

Code Generation Option ([Tools] -> [Options]) is the group of option items for code generation by Java Add-In. This category includes the [General], [Code Generation], [Code Style], and [File Header] sub-categories.



General

Option Item Default	Description
Generate codes even when there is no profile	Specifies whether to generate codes even when Java profile is not loaded in the project (not recommended).

Code Generation

Option Item	Default	Description
Generate 1 file		
each for Class, Struct, Interface, and	l ruo	Specifies whether to generate 1 file each for 1 member (Class, Struct, Interface, and Enum).

Enum		
Generate unnamed AssociationEnd	False	Generates unnamed AssociationEnd as a reference type field for Java member. In this case, the field identifier is indicated as "UnspecifiedType" with a number at the end.
Generate Java Doc	HAICE	Specifies whether to generate modeling element documentation as Java Doc.
Generate empty Java Doc	False	If checked, Java Doc is generated even when the modeling element documentation does not have any values (empty).

Code Style

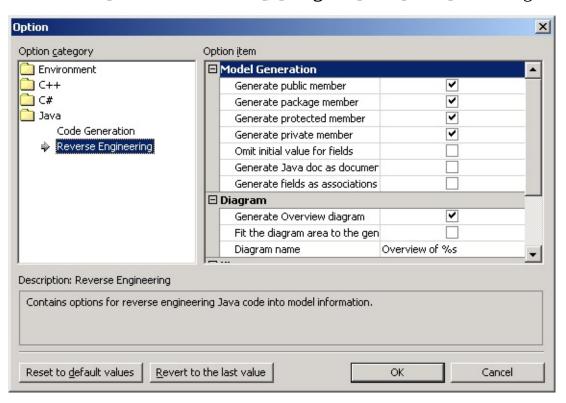
Option Item	Default	Description
Insert tabs as spaces	False	Uses space instead of tab for indentation.
Tab width	4	Specifies the number of spaces to be used when inserting tabs as spaces.
Place the opening curly brace in the new line	False	Places the opening curly brace "{" in the new line.

File Header

Option Item	Default	Description
		Adds the comments in the beginning of the source file.
		(Default)
		//
		//

Reverse Engineering Option Configuration

Reverse Engineering Option Configuration (**[Tools] -> [Options...]**) is the group of reverse engineering option items for Java Add-In. This category includes the **[Model Generation]**, **[Diagram]** and **[View]** sub-categories.



Model Generation

Option Item	Default	Description
Generate public member	True	Specifies whether to generate class and interface members with public visibility.
Generate package member	True	Specifies whether to generate class and interface members with package visibility.
Generate protected member	True	Specifies whether to generate class and interface members with protected visibility.
Generate		Specifies whether to generate class and interface

private member	True	members with private visibility.
Omit initial value for fields	False	Does not include the field initial value in the attribute model information.
Generate Java doc as documentation	False	Reads Java Doc in the source code and registers it as documentation information. Java Doc of each method is analyzed for tag information and entered as documentation information for each parameter of the operation model.
Generate fields as associations	False	Analyzes the field information in the source code to establish association relationships with the respective field type models. If unchecked, field information is generated as attributes for the respective class model

Diagram

Option	D. C. 1.	
Item	Default	Description
Generate Overview diagram		Specifies whether to generate the Overview diagram for the generated model. If unchecked, the following diagram and view options are ignored.
Fit the diagram area to the generated view area		Enlarges the Overview diagram to fit the generated view area.
Diagram name	Overview of %s	Specifies the Overview diagram name. The package name can be included in the diagram name by using %s (e.g. Overview of %s).

View

Option Item	Default	Description
Suppress the Attribute area of Class	False	Suppresses the Attribute compartment of the Class View when generating the Overview diagram.
Suppress the		Suppresses the Operation compartment of the Class

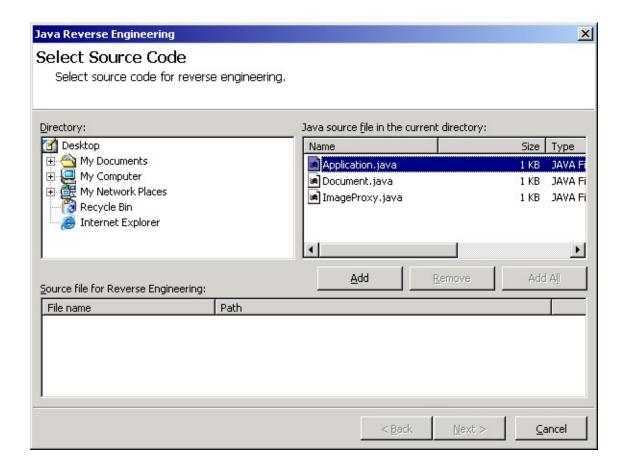
Operation area of Class	False	View when generating the Overview diagram.
Hide operation signature	False	Hides the operation signature when generating the Overview diagram.
Generate Generalization and Realization views only for relations	False	Generates generalization and realization views only for relations when generating the Overview diagram. When used appropriately with other view options, this option is very useful for drawing the inheritance relations of overall classes and interfaces within the package.

Java reverse engineering analyzes Java source files and converts them into UML models. It is useful for source inspection, system structure analysis, and re-designing.

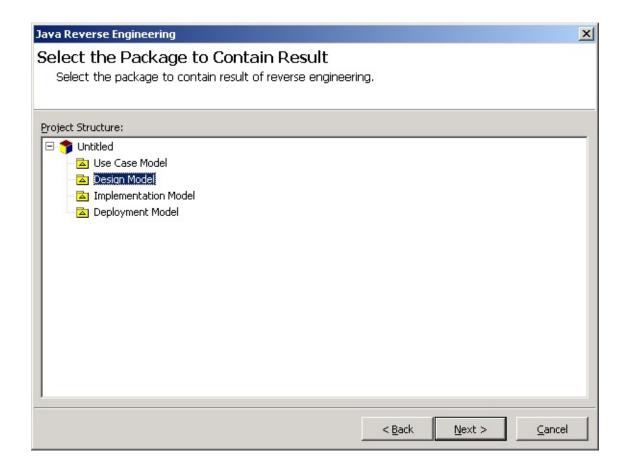
- Java Reverse Engineering
- Reverse Engineering Option Configuration

Procedure for Reverse Engineering:

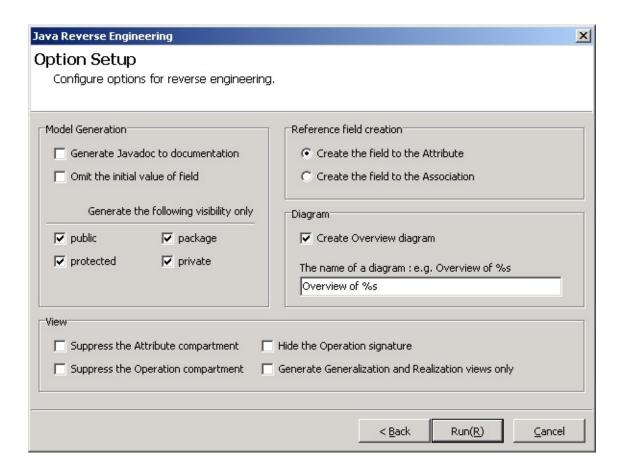
- 1. In StarUML(tm), select the **[Tools] -> [C#] -> [Reverse Engineer...]**
 - At the [Select Source Code] page in the [Java Reverse Engineering]
- 2. dialog box, select a source and click **[Add]**. Click **[Next]** once you have completed adding the target sources for reverse engineering.



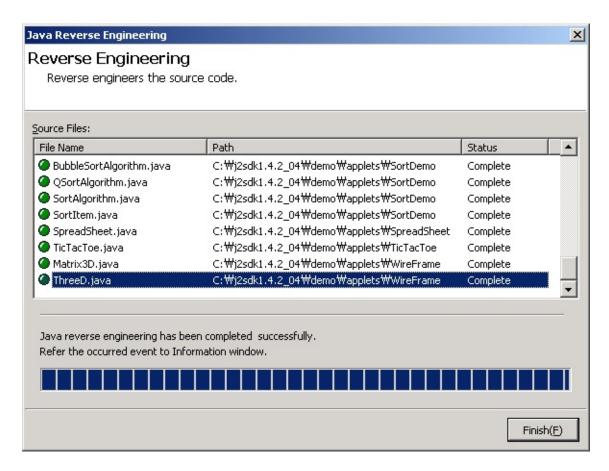
3. At the **[Select the Package to Contain Result]** page, select a package to contain the output results from the package tree and click **[Next]**.



4. At the **[Option Setup]** page, select the reverse engineering options and click **[Run]**. Reverse engineering will start now.



5. The **[Reverse Engineering]** page will show the reverse engineering progress status and return reverse engineering failure or success results.



☑Note

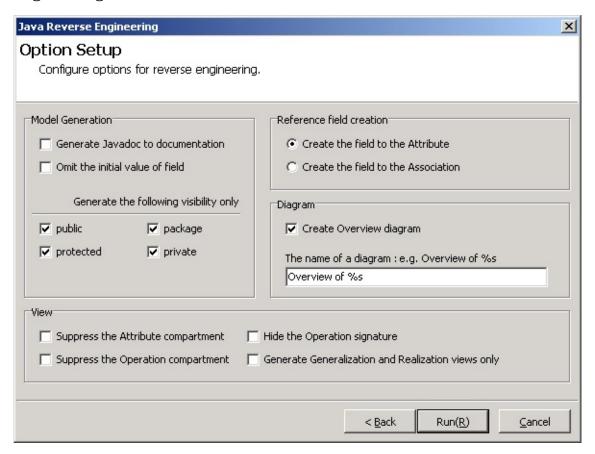
If Java reverse engineering is executed without including Java profile, the

• following dialog box will appear asking whether you want to include Java profile. Select "Yes(Y)" to continue the reverse engineering process.



Reverse Engineering Option Setup Screen

This is the screen for configuring the options required for Java reverse engineering.



Model Generation

Model Generation includes various options for model generation.

Item	Description
Generate Java doc to model documentation	Specifies whether to generate Java Document as StarUML(tm) model documentation.
Omit the	Specifies whether to omit the initial values for Java fields.

public	Specifies whether to generate elements with public access modifiers.
package	Specifies whether to generate elements with package access modifiers.
protected	Specifies whether to generate elements with protected access modifiers.
private	Specifies whether to generate elements with private access modifiers.

Reference field creation

Reference Field Generation specifies generation methods for reference fields when generating models.

Item	Description
Create the field to the Attribute	Specifies whether to generate Java fields as StarUML(tm) model attributes.
Create the field to Associations	Specifies whether to generate Java fields as StarUML(tm) model associations.

Diagram

Diagram specifies diagram generation and the default generation names.

Item	Description
Create	Specifies whether to generate Overview diagram when generating model.
Overview	
diagram	
The name of	Specifies names for Overview diagram generation. The string %s
a diagram	is automatically replaced by the package name.

View

View specified view-related options after model generation.

Item	Description
Suppress the	
attribute	Suppresses the attribute compartment of class models.
compartment	

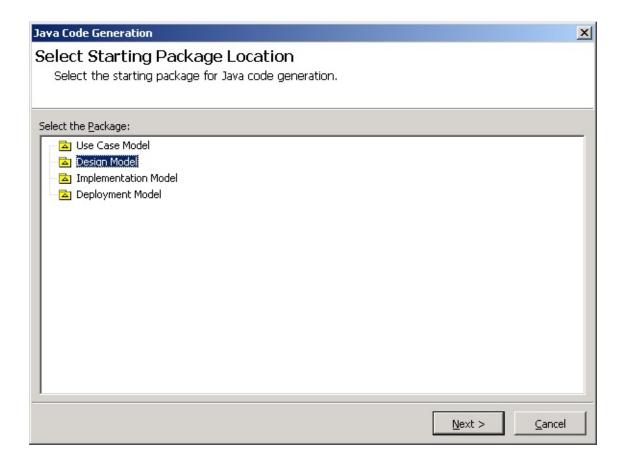
Suppress the operation compartment	Suppresses the operation compartment of class models.
Operation	Specifies whether to display all signatures for operation elements.
חבו	Specifies whether to generate generalization and realization views only for the models generated.

Java forward engineering generates Java source files from StarUML(tm) models.

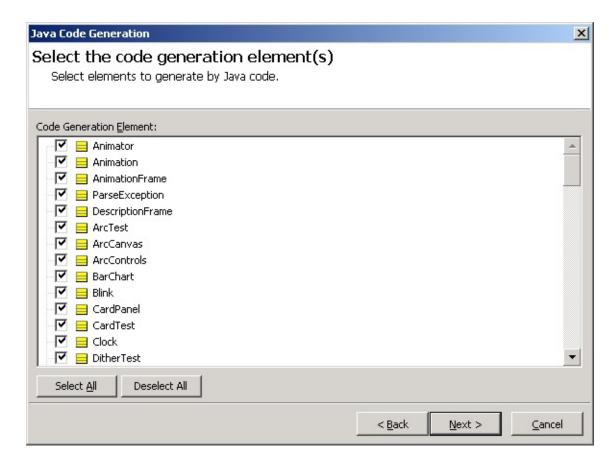
- Java Code Generation
- Code Generation Option Configurations

Procedure for Code Generation:

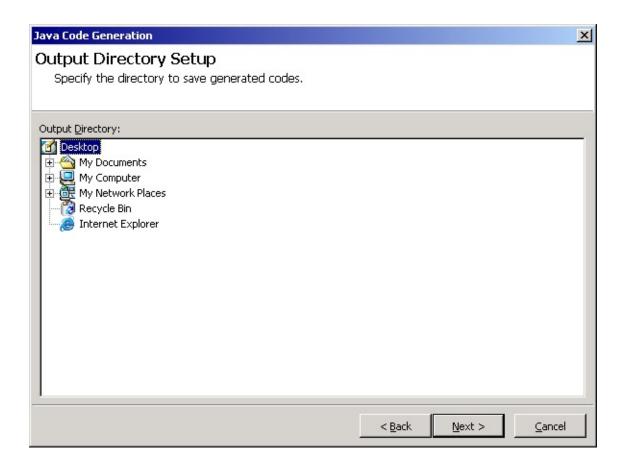
- 1. In StarUML(tm), select the **[Tools] -> [Java] -> [Generate Code...]**
- 2. At the **[Select Starting Package Location]** page in the **[Java Code Generation]** dialog box, select a package and click **[Next]**.



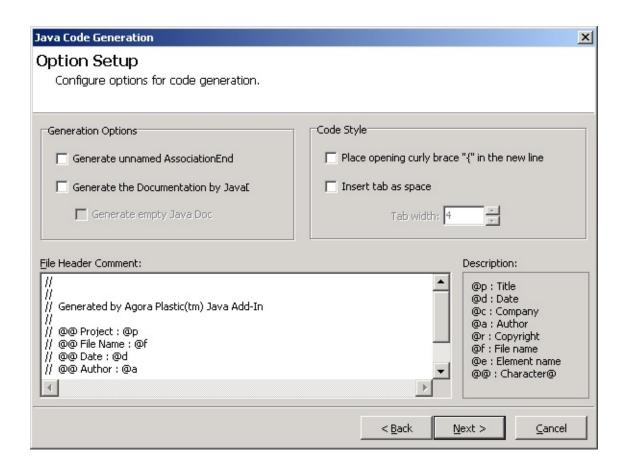
3. At the **[Select the code generation element(s)]** page, select the elements and click **[Next]**.



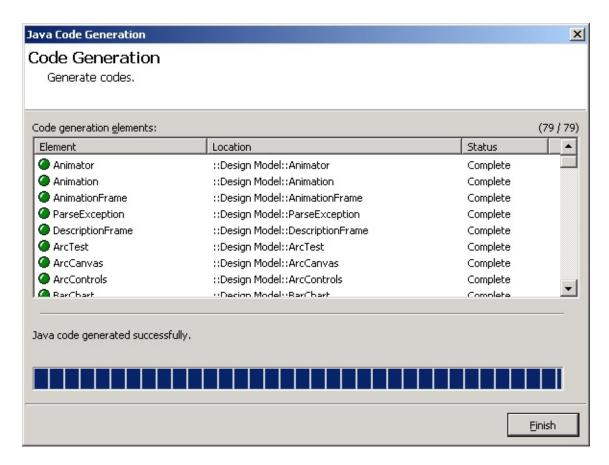
4. At the **[Output Directory Setup]** page, select a directory to save the output sources and click **[Next]**.



5. At the **[Option Setup]** page, select options and click **[Next]**. Reverse engineering will start now.



6. The **[Code Generation]** page will show the code generation progress status and return code generation failure or success results.



✓Note

The following error will occur if Java code generation is executed without

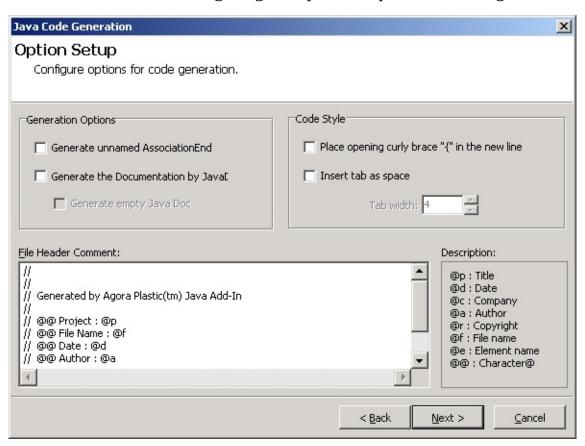
• including Java profile. Please ensure that Java profile is included in the project before executing code generation.



Code Generation Option Configurations

Code Generation Option Setup Screen

This is the screen for configuring the options required for code generation.



Generation Options

Generation options are model-related options for code generation.

Item	Description
Generate unnamed AssociationEnd	Specifies whether to generate AssociationEnd at the end of an unnamed Association.
Generate the Documentation by Java Doc	Specifies whether to generate StarUML(tm) model documentation as Java Doc.
Generate empty	Specifies whether to generate empty StarUML(tm)

Java Doc documentation.	
-------------------------	--

Code Style Options

Editing options are text-related options for code generation.

Item	Description
Place opening curly brace "{" in the new line	Specifies the location of the opening curly brace "{" for code generation.
Insert tab as space	Specifies whether to insert space strings instead of tabs.
Tab width	Specifies the number of spaces for a tab.

File Header Comments

File Header Comments defines the comments for each file head.

Item	Description
	Contains the comments to be inserted in the beginning of the
File header	source file. As described in the "header comments
comments	description" section, the '@' symbol and alphanumeric
	characters can be used to insert specific values here.

FAQ Top Previous

The following are frequently asked questions and answers for using Java Add-In.

- 1. What are the Java Language versions supported by Java Add-In?
- 2. <u>I get a "Java Profile is not loaded" warning window when generating code.</u> What does this mean?

1. Java What are the Java Language versions supported by Java Add-In?

Java Add-In supports Java Language Specification version 1.3, which is the most widely used version.

2. I get a "Java Profile is not loaded" warning window when generating code. What does this mean?

Java Profile must be included in order to use the Java code generation function. <u>Including Java Profile</u>