Release Notes 2.3.0

g9 2.3.0

April 20, 2016

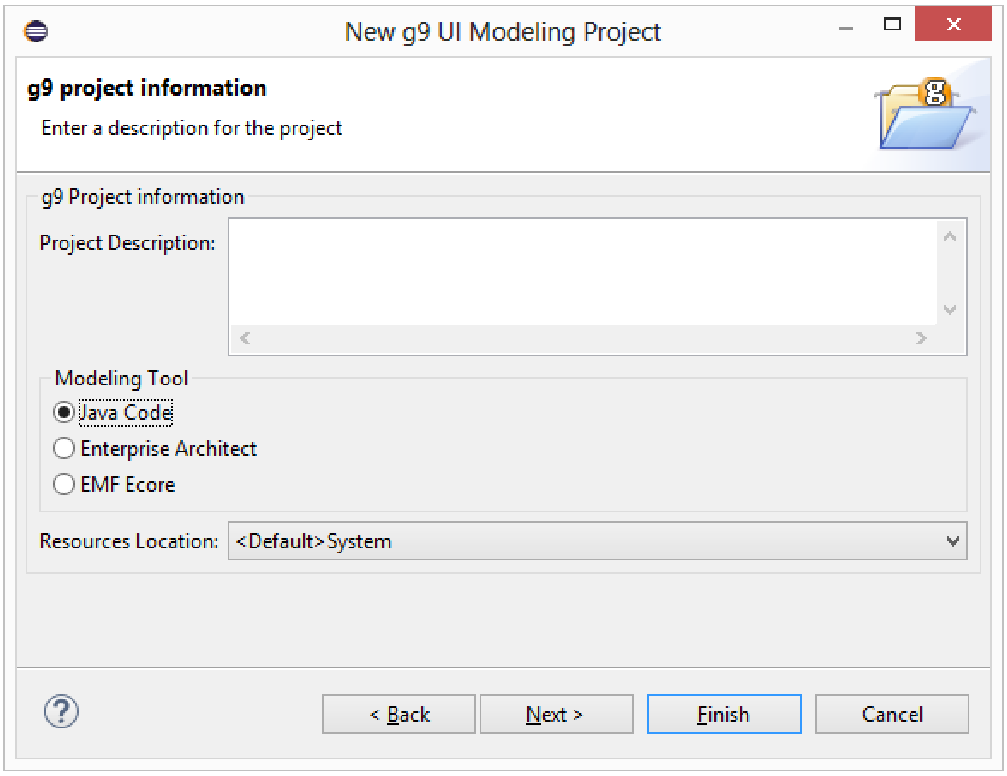
Date: 20-Apr-2016 13:58
URL: https://esito-conf.inmeta.com/display/g9RN/Release+Notes+2.3.0

Table of Contents

1 News and Changes

1.1 Modeling with Java source code as the Model Source

From this version you may create domain models from your written java source. An internal domain model in g9 can be distilled from java source code by specifying *Java Code* as modeling tool when creating a new g9 UI modeling Project. As source for the model project an eclipse java project must be specified. By default all java files contained in the project will be used as source, but files can be excluded. Files added to java project will automatically be incorporated.



New g9 UI Modeling Project

g9 project information

Enter a description for the project

g9 Project information

Project Description:

Modeling Tool

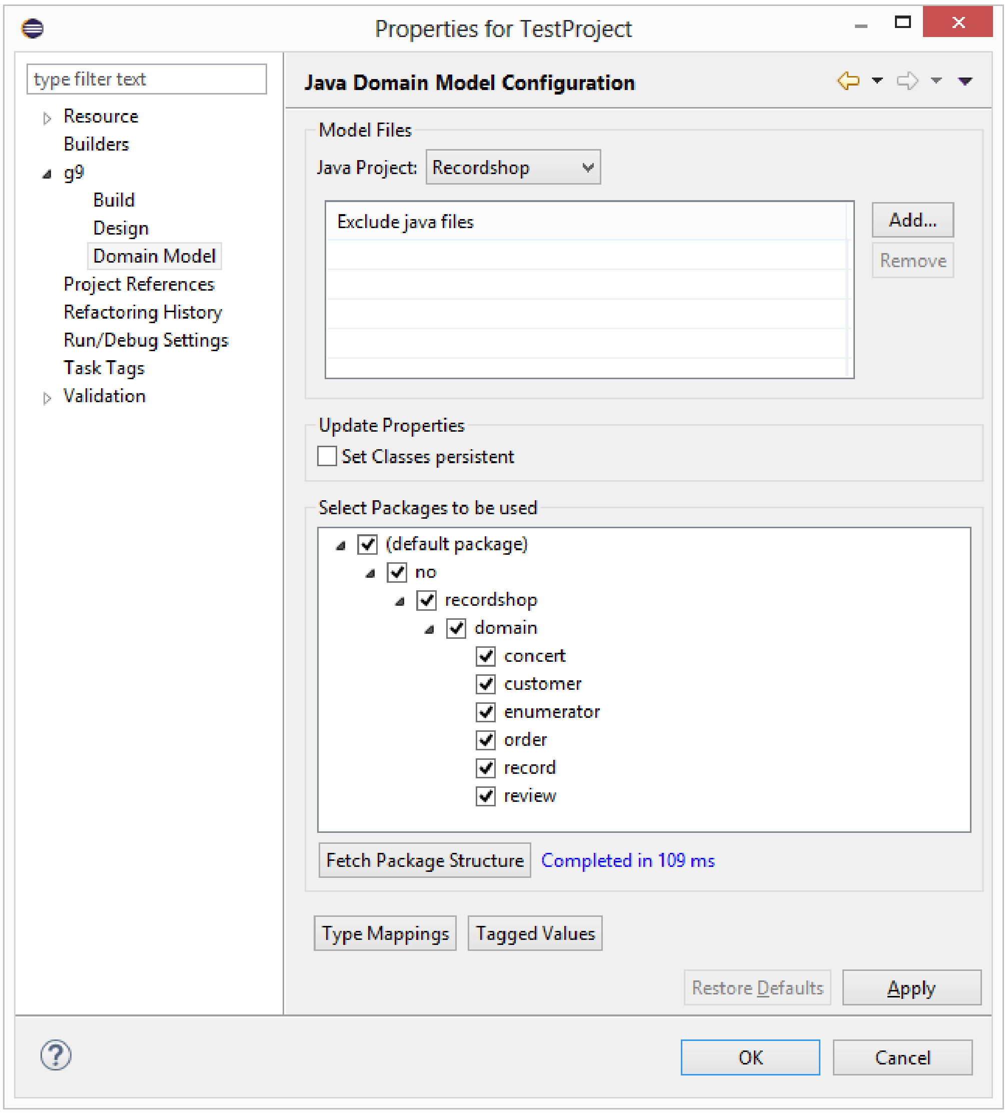
● Java Code

○ Enterprise Architect

○ EMF Ecore

Resources Location: <Default> System

As with the other model tool types, the package structure must be fetched. All packages used as packages in the java files will be included in the package structure and packages may be included or excluded as the model base.



Java Domain Model Configuration

Model Files

Java Project: Recordshop

Exclude java files

Update Properties

□ Set Classes persistent

Select Packages to be used

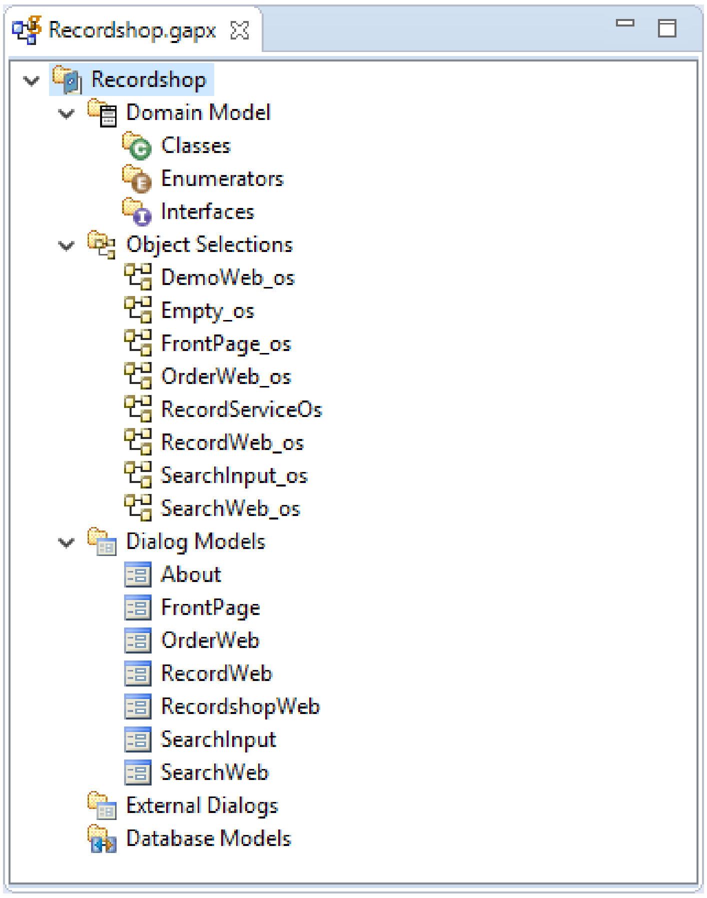
```
☑ (default package)
  ☑ no
    ☑ recordshop
      ☑ domain
        ☑ concert
        ☑ customer
        ☑ enumerator
        ☑ order
        ☑ record
        ☑ review
```

Completed in 109 ms

See Modeling with Java code as Model Source.

1.2 Application models

The Application model in g9 now supports all buildable objects. This adds Class, Enumerator, Interface and Database Model to the model.



Recordshop.gapx

- Recordshop
  - Domain Model
    - Classes
    - Enumerators
    - Interfaces
  - Object Selections
    - DemoWeb_os
    - Empty_os
    - FrontPage_os
    - OrderWeb_os
    - RecordServiceOs
    - RecordWeb_os
    - SearchInput_os
    - SearchWeb_os
  - Dialog Models
    - About
    - FrontPage
    - OrderWeb
    - RecordWeb
    - RecordshopWeb
    - SearchInput
    - SearchWeb
  - External Dialogs
  - Database Models

See the documentation for the Application model.

Application Build

A new build rule is that all application objects will build with the build configuration specified for the application root. This is additional to any build configuration already specified for the objects outside of the application context. This makes the build system more flexible and easier to configure for full and automatic build use.

All included objects in the application will build with the same *Build Configuration* as its application's. This application build comes in addition to any other build configuration the objects have. You may create applications for each build purposes such as separating UI, Reports and services. The *Build Application* menu is defined on Applications only and it will build Application level artifacts for all members of the application.

Some generated artifacts can only be built if the model is part of an application:

- Dialog Model Report (*Applications* and *Dialog Models*)
- Object Selection Report (*Applications* and *Object Selections*)
- REST client (*Interfaces*)
- REST server (*Interfaces*)
- SOAP/Castor (*Interfaces*)
- WS Client (*Interfaces*)
- WSDL (*Interfaces*)

See the documentation for Performing a build operation.

1.3 Build menu changes

1.3.1 Build Application

On the application node, "Build Application" in effect replaces the former "Build Top + Build Detail" menu items. The "Build" menu item now only builds the Application model itself, not the included model elements.

1.3.2 Top/Detail

is removed from the build menus.

1.4 Generator API changes

Generators will have to be modified to take one object at a time. The IGenerator implementation is now called for each resource rather than once for all resources. This will in most cases simplify the code since there is no longer a need to iterate a resources collection.

See the documentation for the new APIs.

- IGenerator interface
  - doGenerate(EObject eo)
  - doValidate(EObject eo)

Furthermore, generators can now accept multiple object types. Most Dialog Model generators now also adds Application to the input list. The application type also obsoletes the *topLevel* generator attribute in the *plugin.xml* file for generator projects.

1.5 Support for Value Objects in the domain model

1.5.1 Domain Models

The domain model classes have a new boolean property, *Value Object*. If set to true, the class is treated as a value object. The setting of this property is done when doing an *Update from Domain Model*.

Enterprise Architect: In Enterprise Architect a value object class is modeled by creating a *Data Type* element rather than a *Class* element. When doing an *Add Element* on a package the *New Element* dialog contains *Data Type* as a possible type.

Java source domain model: When Java code is used as the domain model in g9, inner classes are treated as value object classes. Additionally, classes annotated with the JPA *@Embeddable* annotation or with the g9 class annotation with the *isValueObject* field set to true, are also treated as value objects.

Ecore/Xcore: A class is considered a value object if the model contains an *EDataType* wrapping the class. However, *EDataType* elements annotated as a g9 Usertype are not considered as such wrappers.

⚠ The g9 annotation for Usertype has changed. In Ecore/Xcore all EDataTypes were imported as Usertypes into g9. When exporting a g9 domain model to Ecore, g9 needs to create EDataType instances not based on Usertype instances in the domain model. Such EDataType elements were given the g9 annotation *isNotUserType*. This annotation is now obsoleted, and all EDataType instances created based on an UserType is now given the annotation *isUsertype* when the model is exported. On import from Ecore/Xcore only EDataType instances marked with this annotation will be imported as a Usertype.

In the Ecore/Xcore model any isNotUserType annotation should be removed, and all EDataType instances that need to be Usertype should be marked with the isUsertype annotation. Exporting a g9 model to Ecore will automatically create the correct annotations.

1.5.2 Database Models

In the Database Model, value objects do not give rise to tables. Persistent attributes in a value object are included as columns in a table if the table is based on a class with an attribute referencing the value object. In the same way, if the value object contains persistent associations, foreign key columns will be incorporated in the table.

A table containing columns for a value object will also contain a group embracing all columns for that value object, including foreign key columns. Value objects may contain other value objects, and this is also reflected in the Database Model through groups containing groups as members.

1.5.3 Object Selections

To support value objects in the Object Selection, the structure is enlarged. Value object classes as such cannot participate as roles in the Object Selection. But if a class has an attribute referencing a value object class, the attribute itself is included and the attributes from the value object is shown as child nodes. This inclusion of child attributes is recursive.

When adding a new role to a role via associations, not only associations from the main class is available, but also associations defined in value objects included in the class. This behavior is also recursive.

1.5.4 Dialog Models

In the Dialog Model, attributes in the value object classes are seen through the object selection attributes. Data Items are connected to Object Selection attributes as before. The only difference is the name of the Data Item. It used to be shown as *RoleName.AttributeName*. Now the attribute part can be a sequence of ". "-separated names representing the actual path from the class to the attribute, i.e. *Customer.address.street*. Here *Customer* is the role containing an attribute *address*, which references a value object containing the *street* attribute.

As earlier, when renaming a Data Item the attribute part of the attribute cannot be renamed.

1.5.5 Generator support

The following generators support value objects: *Java Domain, JGrape*, *Hibernate* and all database schema generators (Derby, MySQL, Oracle, SQLserver, Sybase).

The Java Domain generator will create immutable Java classes for value objects, with a constructor containing all attributes. A private empty constructor is also created to support ORM frameworks, which will set all attributes to a type dependent zero value. There are no setters for the attributes in a value object class, and the *equals* and *hashCode* methods are based on all attributes in the class.

The JGrape generator is updated to support immutable value object classes.

For value object columns in the Database Model, the Hibernate generator will create *component* elements in the mapping. *Component* elements may be nested if a value object class contains value object attributes.

The database schema generators automatically supports value objects, as value object attributes are normal columns in the Database Model.

1.6 User defined type mappings for the Hibernate generator

The Hibernate generator now supports type mappings. Current Hibernate generator configurations will be upgraded to include the default g9 Type Mappings on a g9 model project upgrade.

The default type mappings are the same as was previously hard coded into the generator. While the default type mappings only contains model types, both model types and g9 Usertypes can be mapped.

Some combinations of g9 types and database types are still generated without consulting the type mapping. The g9 type *text* with dbtype *char* will always be generated as *gva_text*. The g9 type *date* in combination with dbtype *int* will give *gva_intdate* as result, and a date column with a *legacy type* will be generated as that legacy type. A g9 type *boolean* with dbtype *char* will be generated as *gva_textboolean*. Enumerators will always be generated as *gva_enumerator*, unless the dbtype is *char* or *varchar* which will result in *gva_textenumerator*.

1.7 Runtime

Added new runtime component: g9-annotation-<version>.jar. This component contains the g9 annotations to be used when Java Code is the domain model source for a g9 model project.

1.8 Default text file encoding on model files

The encoding on model files has previously been set to *ISO-8859-1* by g9. New files will now get the Eclipse Workspace text file encoding, which is *Cp1252* by default on Windows. See Preferences > General > Workspace if some other encoding is needed.

1.9 Model Presentation

The g9 Package Presentation menu is renamed to Model Presentation.

2 Generated code and Model Changes

2.1 Changes in model files

Project file (.gproject)

- g9Version="2.3.0.v<release-version>"
- Default Hibernate Mapping is added

Domain Model files (.gdox):

- The file is reorganized because the elements now are sorted

Application Model files (.gapx):

New elements in the file:

- <AppClasses>
- <AppEnumerators>
- <AppInterfaces>
- <AppDatabases>

Database Models (.gdmx):

The internal structure of DBRelation is changed:

- <Reference name="FromColumn" changed to DBRelation attribute FromColumn
- <Reference name="MemberTable" changed to DBRelation attribute MemberTable
- <Reference name="ToColumn" changed to DBRelation attribute ToColumn

2.2 Changes in generated code

Regeneration of code is necessary.

For the ICEfaces generator, only the Enumerators in use by an application are now generated.

3 Dependent versions

g9 runtime libraries uses 3. party open source libraries. Supported versions in this release are (red are upgrades since 2.1.0):

Library	Version		Library	Version
Spring	4.0.9.RELEASE		Hibernate	5.0.1.Final 5.0.7.Final
ICEfaces	4.1.0 4.0.0-GA		JSF	2.2.10
log4j	2.4.1		Jetty	9.2.6.v20141205
Jackson	2.6.3		Spring WS	2.2.2.RELEASE
Castor	1.3.3		JasperReports	6.2.0
Joda Time	2.4		Apache POI	3.9
EJB	3.2.0		Guava	18.0

4 Improvements and corrected Errors

If nothing is written in the *Release Comment* section about an error, the error is corrected and a comment is not necessary.

4.1 g9

SUP-1744 - Copy/paste of methods Under the Methods tab in the dialog designer does not work properly

SUP-2981 - Multiselecting data elements and changing style leads to strange style display

SUP-2992 - Removing Generator configuration does not remove the configuration from Build Configurations

SUP-2993 - In the New project wizard the Back button should disable the Finish button when moving from page 3 to page 2

SUP-3712 - Removed Top/Detail builds for Application and DBMapping

SUP-3835 - No Style should add a Problem
Release Comment: When a skin has no style set for a specific widget type it will not create a warning in the resources model but rather where used, ie "Skin has no style for widget type Button".

SUP-3897 - Labels are misplaced/disappears when undoing changes followed by redoing of the same changes

SUP-4145 - .gproject is being saved twice

SUP-4387 - External model can't be created when ids are wrong
Release Comment: The handling of errors should never result in a failed external model. If the model is inconsistent, there would be problem markers in the model. The build should still go on and build what it can and mark the failing artifacts with an X and make the stack trace available in the error log.

SUP-4429 - ClassCastException when closing dialog model editor

SUP-4491 - Table header inherits Table font, not Header font
Release Comment: Font and foreground color is now "inheriting" the same way as background color. If label's style explicitly sets as "inherited" it will get the value from the table's header style when it is positioned in a header.

SUP-4508 - External elements should not be included in build resources

SUP-4628 - Resources/Profiles are not read correctly
Release Comment: A project configuration error will set a project in an invalid state. It will create an Error in problems view and an exclamation mark on the project node when the project is opened. It will also put an exclamation mark on the missing resources / profiles file. Such a mode will also disable build and validation functionality.

SUP-4788 - Undo/redo of name setting does not move objects sorted by name

SUP-4797 - Add Toolbar/Menubar from palette does not work

SUP-4848 - Group/ungroup children on domain classes is not shown in menu

SUP-4851 - Menu item Go To Domain element is missing for methods in object selections
Release Comment: Domain model Methods are no longer shown in Object Selections roles since these have no function there.

SUP-4853 - Generator API for validating and compiling one object at a time
Release Comment: Custom generator API has changed to generate for one object at a time. This makes building for larger projects smoother whereas progress bar is more accurate and cancel works better.

SUP-4854 - Adding an event to a dialog window collapses all containers

SUP-4942 - Packages should be listed first when using Composite or Package Hierarchical view

SUP-4956 - Datatypes in Ecore/Xcore should by default not end up as Usertypes in g9
Release Comment: In Ecore/Xcore all EDataType were imported as Usertype into g9. When exporting a g9 domain model to Ecore, g9 needs to create EDataType instances not based on Usertype instances in the domain model. Such EDataType object was given the g9 annotation *isNotUserType*. This annotation is obsoleted and all EDataType instances created based on an UserType is now given the annotation *isUsertype* when the model is exported. On import from Ecore/Xcore only EDataType instances marked with this annotation will be imported as a Usertype.

In the Ecore/Xcore model any isNotUserType annotation should be removed and all EDataType instances that need to be Usertype should be marked with the isUsertype annotation. Exporting a g9 model to Ecore will automatically create the correct annotations.

SUP-4957 - Generalization between interfaces gets Inherited As property
Release Comment: The property was set in the domain model file when updating from an Ecore or an Xcore model. The property was not shown in the property view.

SUP-4958 - The categories are not sorted in the domain model
Release Comment: When opening domain model in an editor the categories will sort alphabetically. It will also arrange XML nodes accordingly so there will be a big difference when using text compare.

SUP-4971 - Editing the association property of an object selection role fails
Release Comment: When selecting the property, the displayed value changed to an internal identity value and the correct value was not selected in the list of the combo. If multiple values were available, changing the value did not have any effect.

SUP-4978 - Renamed g9 "Package Presentation" to "Model Presentation" to avoid duplicating menus

SUP-4979 - Fixes in export and import from Xcore when using wrappers
Release Comment: Wrappers defined in Xcore become data types in Ecore. Such data types used to be imported as user types into g9. Now such data types are by default not imported as user types. Only when annotated g9 with field isUsertype="true" will they become user types. However, attributes, methods and method parameters defined by such wrappers will get the same model type in g9 independent of the presence of the annotation.

SUP-4981 - The default Maximum name length in database profiles is wrong
Release Comment: The values in the system delivered database profile are set as follows:

- Derby: 128
- MySQL: 64
- Oracle: 30
- SQLServer: 128
- Sybase: 30

In a user defined database profile these values are also the default values for the *Maximum name length* property.

SUP-4989 - External dialog models are not available to the dialog model generators
Release Comment: External dialog models which are modeled in g9 are now available to the generators.

SUP-4990 - External dialogs should have a folder in application model

SUP-4991 - External dialog models cannot be used as targets for Open actions

SUP-4992 - Charset & UTF-8 support for model files
Release Comment: Character encoding for new model files will follow eclipse project/workspace setting. Note that although Eclipse windows default is "Cp1252" it will map to "windows-1252" in the XML header.

SUP-4994 - StackOverflowError from database model
Release Comment: The error correction triggers changes in database model files giving differences in column and table settings in relation. The changes do not have any consequences neither for the models nor for the code generated from them. Database models will be upgraded.

SUP-4999 - NPE when opening System:default.gmpx and the defined DB profile is missing
Release Comment: Now also shows an exclamation mark when file is missing.

SUP-5005 - Editor tree does not expand correctly initially and suddenly collapses at times

SUP-5007 - "Restore Defaults" for Generator and Domain Type Mappings
Release Comment: Domain model Type Mappings in Project Properties and Type Mappings for Generator configurations now have a "Restore Default" button that will give the default set.

SUP-5009 - Next button should not be enabled on page 2 in New g9 Resources Project

SUP-5018 - Project upgrade for hibernate type mappings
Release Comment: See User defined type mappings for the Hibernate generator.

SUP-5026 - Application Model refreshes a lot

SUP-5042 - Build-, Refresh- and Preview toolbar buttons becomes disabled when unselecting/deleting in a dialog

4.2 ICEfaces

SUP-4986 - Enum converters are generated for all enums in the domain model
Release Comment: The ICEfaces generator now only generates enum converters for the enums used by the dialog models in an application.

SUP-4987 - The application window width is not generated to the application CSS file
Release Comment: The rendered width of the application window is now generated to the CSS file if the Width property of the application window is set.

4.3 Java runtime

SUP-5034 - HibernateSessionFactory hides Spring Exception
Release Comment: The getHibernateProperties and getHibernateResources methods now catch only NoSuchBeanDefinitionExcption, other exceptions are propagated to the caller.

4.4 Hibernate

SUP-5006 - SMG-244 - Support for user defined type mappings for the Hibernate generator
Release Comment: See User defined type mappings for the Hibernate generator**.**

SUP-5018 - Project upgrade for hibernate type mappings
Release Comment: Hibernate generator now supports Type Mappings. Current Hibernate configurations will be upgraded to include the default Type Mappings.

4.5 REST Client

SUP-4960 - RestClient generates illegal model type code for methods
Release Comment: The error occurred when a return type for a method was a generic Set or List type not based on a Usertype.

4.6 Jasper

SUP-4961 - The Jasper Reports generator only builds pom.xml and build.xml when the build is started on the project or Model View level
Release Comment: The files `build.xml` and `pom.xml` are now generated also for builds on a single Dialog or Application Model.

SUP-4962 - Changes to the generator parameters for the Jasper Reports generator
Release Comment: The generator parameter for the `pom.xml` file now correctly uses pom.xml instead of pom.template in its label. Generating `build.xml` now defaults to `false`.

SUP-4964 - Upgrade Jasperreport version to 6.2.0
Release Comment: The generated example pom for Jasper now has a Jasperreports 6.2.0 dependency.

4.7 Known Issues

4.7.1 Differences between Designer and Preview

Both Dialog Designer and Preview use Windows native widgets for display, through SWT.

The Designer adjusts the widget sizes to Windows 7 Theme, which makes the List container look wrong if one is running Windows Classic, 2000 style. Additionally, all containers get a minimum 1 pixel margin to show the selection markers.

The Preview adjusts its widget sizes to Java JFC/Swing, which makes it easy to see the components final sizes and alignments. This, however makes the List container look wrong, since it is more narrow and shorter resulting in horizontal scrollbars and fewer rows.

4.7.2 Known error regarding resource projects and upgrades

The resource model needs an upgrade. If the resource model is located in a g9 resource project missing the .gproject file, two different error situations may occur:

1. g9 will ask you to run upgrade on the project each time eclipse Is restarted and the project node is expanded.
2. g9 will never ask you to upgrade the project.

In both cases do as follows: Open the project property and select g9; then press the Save button to create a new .gproject file. When the project has a .gproject file, run g9->Upgrade Project to upgrade the resource model.

4.7.3 New example projects can get cyclic dependency errors

This can happen occasionally due to issues with some versions of Xcore/Xtext. New Example projects can get into a cyclic dependency error situation, shown as an exclamation mark on the project node. In this situation "Clean" or reopening the project does not help, however it is usually resolved by restarting Eclipse

4.7.4 Limitations and known errors for ICEfaces 4.1.0

1. The ACE dataTable component does not show the correct column widths when the header is hidden and Paging mechanism is Scrollbar.
2. Using nested Basic Containers in a Table with paging mechanism set to Scollbar results in misaligned header columns, as ACE does not handle this at runtime. Use paginators instead until this issue is fixed in ICEfaces.
3. The ICEfaces API for JavaScript effects has changed. Currently, JavaScript effect in the g9 runtime does not work with ACE components.
4. The ACE splitPane component's separator is currently not movable. Also, there is an issue with height calculation when scrollbar is enabled.

4.7.5 Slow Eclipse workspaces

If you experience that your Eclipse IDE is becoming slow, it is possible to improve. Here are some improvement suggestions:

1. Plugin cache - "eclipse.exe -clean"
   Reloads all plugin cache (safe - you won't lose anything)
2. Local history
   Remove ".metadata/.plugins/org.eclipse.core.resources/.history/"
3. Markers
   In sub-directories of ".metadata/.plugins/org.eclipse.core.resources/.projects/" remove ".markers" and ".markers.snap" files.
   https://codedrawer.wordpress.com/2011/07/25/slow-or-frozen-eclipse-ide-start-after-a-crash/
4. Settings
   When upgrading major version of Eclipse settings are not cleaned up properly.
   Export your settings, then rename the ".metadata" directory, then import your settings.
   http://blog.pdark.de/2011/09/02/restoring-a-corrupted-workspace-in-eclipse/
5. Create a new Eclipse workspace and import your projects into it.

5 Distribution

The g9-2.3.0.zip distribution consists of the g9 plugin update site.

The g9 plugins contain all generators and the Java run-time jars. A list of g9 plugins is displayed using *Help > About Eclipse*, press the g9 icon and the *Plug-in Details* button.

6 Installation

6.1 g9 requirements

The g9 toolset consists of code that runs on Windows and Linux operating systems. The tool set runs as an Eclipse plug-in and requires Luna or newer versions of Eclipse. g9 has been tested and works well with Luna SR2 and Mars SR1.

6.2 First time installation

Unpack the g9 zip file to a directory of your choice, here called C:\unpack
You may use the directory where the distribution was unpacked as the update site.
Select “Help>Install New Software…”
Press the “Add…” button giving a dialog box with the title “Add Repository”.
Press the “Local…” button and browse to the unpacked g9 directory, giving file name C:/unpack/g9/ in the Location attribute.
Select the given site in the “Work with” combo box.
Expand the Uncategorized node, check g9, press “Next” and do what is demanded.

6.3 After first time installation

When new versions shall be installed, delete the content of the g9 folder and unpack the distribution to the c:\unpack folder. You may use the directory where the distribution was unpacked as the update site. In Eclipse, select “Help>Check for Updates”. Eclipse will check all software for updates and list all new versions. Select the g9 version and install it.

6.4 g9 updatesite

g9 is available from updatesite: **http://updatesite.esito.no/g9**.

6.5 Xcore modeling

To use Xcore for creating the domain models for g9, the Xcore and Xtext plugins must be installed. Some Eclipse packages already have Xcore and Xtext preinstalled. If not, the recommended update site for Xtext is "Xtext All In One - Releases" http://download.eclipse.org/modeling/tmf/xtext/updates/composite/releases/ . The recommended update site for Xcore is http://download.eclipse.org/modeling/emf/emf/updates/2.10.x/.

For Xcore modeling, the following plugin versions have been tested:

- EMF Xcore version 1.2.2, 1.3.1 and 1.3.2
- Xtext version 2.8.4, 2.9.1 and 2.9.2

6.6 Writing generators in Xtend

To be able to create your own generators in the Xtend language, the Xtend and Xtext plugins must be installed. Some Eclipse packages already have Xtend and Xtext preinstalled. If not, the recommended update site is "Xtext All In One - Releases"

For Xtend support, the following plugin versions have been tested:

- Xtend and Xtext versions: 2.8.4, 2.9.1 and 2.9.2