Release Notes 2.4.0

g9 2.4.0

June 27, 2016

Date: 27-Jun-2016 13:15
URL: https://esito-conf.inmeta.com/display/g9RN/Release+Notes+2.4.0

Table of Contents

1 News and Changes

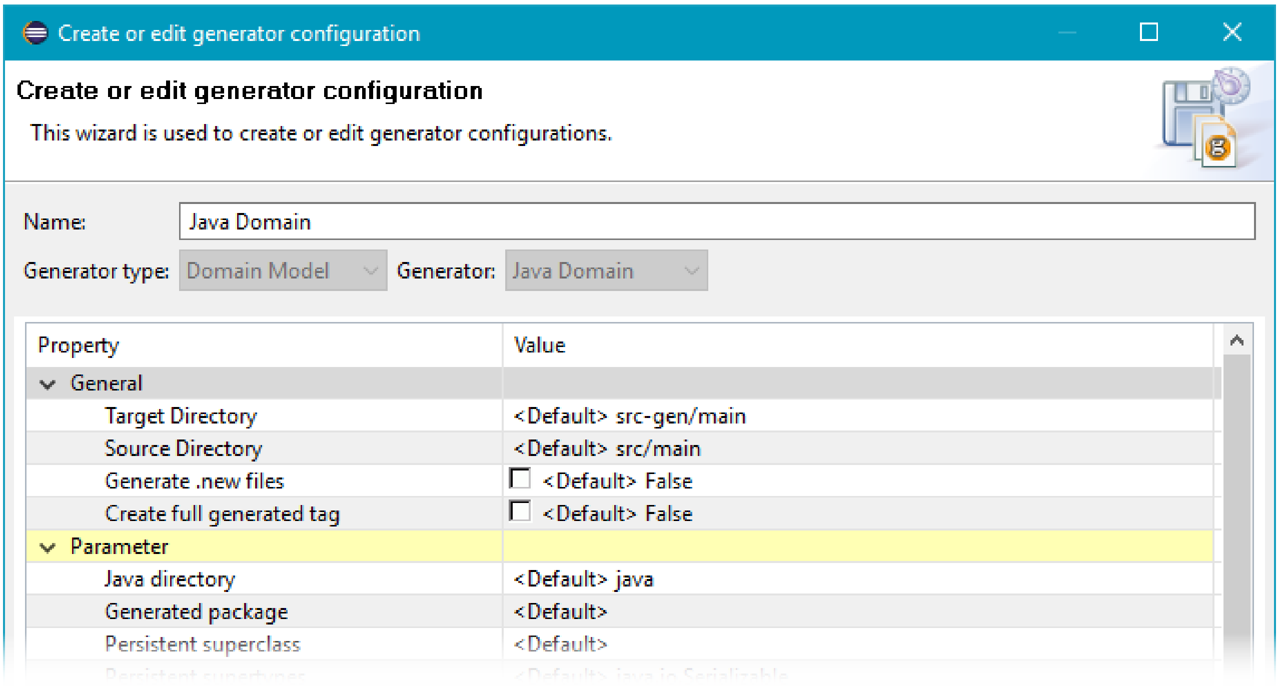
1.1 g9 plugins require Java 8 and Eclipse Mars

The g9 2.4 eclipse plugins require Java version 8, and Eclipse Mars or newer. The g9 runtime, however, can be run with Java 7.

1.2 Separation of generated and manually maintained code

The generators now supports splitting generated and manually maintained code into two separate source hierarchies. This new behavior is triggered by setting the "Source directory" generator parameter to a different value than the "Target directory" parameter. With the new behavior, if a file is present in the "Source directory" it will not be generated to the "Target directory". A ".new" file will instead be generated to the "Target directory" if the "Generate .new files" parameter is true.

An example of the default settings for a generator configuration is shown below. Generated files will be in the "src-gen/main" hierarchy, while manually maintained code is in the "src/main" hierarchy.



Create or edit generator configuration

Create or edit generator configuration

This wizard is used to create or edit generator configurations.

Name: Java Domain

Generator type: Domain Model Generator: Java Domain

Property	Value
General	
Target Directory	<Default> src-gen/main
Source Directory	<Default> src/main
Generate .new files	□ <Default> False
Create full generated tag	□ <Default> False
Parameter	
Java directory	<Default> java
Generated package	<Default>
Persistent superclass	<Default>
Persistent supertypes	<Default> java.io.Serializable

A file can be moved from the generated source code hierachy to the manually maintained hierarchy. This will prevent the generator from writing the file the next time the generator is run, and the file contents can be maintained by the developer.

The new behavior is the default for new generator configurations, and the default value for the "Generated package" parameter is now blank. If "Target directory" and "Source directory" have the same values, there will be no change in behavior.

Existing generator configurations will keep the old behavior until the "Source directory" parameter is changed by the user.

1.3 Import models from databases

A g9 modeling project needs a domain model, and currently the supported domain model sources are Ecore /Xcore, Enterprise Architect, and Java code. In this version you can import a database schema to an Ecore /Xcore model, or Java code. The imported database schema can then be used as a domain model in a g9 modeling project.

A database schema will be imported to an existing project in the Eclipse workspace. If the project has an Xtext nature, an Xcore model will be created. If the project has a Java nature, Java code will be created. For other projects, an Ecore model will be created.

The imported domain model will contain classes, groups and associations. Database tables results in domain model classes, and the columns of a table will become attributes in the class. Foreign keys in the database schema are treated as associations. Groups are created from indexes and primary keys in the database schema containing more than one column.

The *Import Database Model* wizard is available from the g9 category in the *Import* dialog or from the *Import* menu. The wizard contains

- Setup of database connection
- A list of tables where name may be edited
- A list of attributes where name may be edited
- A list of type mappings where model type may be edited

See the Import Database Model page for a detailed description.

1.4 Java Domain generator

1.4.1 g9 annotations

The Java Domain generator now can generate g9 annotations to the domain classes. Set the "Generate g9 annotations" parameter to "True" to activate this feature.

1.4.2 JPA annotations

The Java Domain generator now can generate JPA annotations to the domain classes. Set the "Generate JPA annotations" parameter to "True" to activate this feature.

1.4.3 Groups, Converters and Usertypes

The Java Domain generator now supports generating Groups, Converters and Usertypes. These model elements are only generated if they are referenced from a Class, an Interface, or an Enumerator that is also generated. Set the "Generate groups", "Generate converters" or "Generate Usertypes" parameter to "True" to activate this feature.

1.5 Xcore generator

The Xcore generator generates an Xcore representation of the g9 domain model. One Xcore file is written for each package that contains a *Class*, an *Enumerator* or an *Interface* to be generated. For referenced *Usertypes, Converters* and Groups, their package files are also written. The generator adds g9 annotations from the g9 domain model. See Xcorex definition.

1.6 New DB properties in the domain model

Table Name, Column Name, Insert Prohibited and Update Prohibited

The property Table Name is part of the general DB properties available on classes while the three other properties are part of the DB properties available on user types and attributes in the domain model. All four properties are available in all model tools supported.

The Insert prohibited and Updated Prohibited was already available in the database models, but they may now get their default value from settings in the domain model.

The default setting for the properties themselves are empty/undefined for Table Name and Column Name and False for the two prohibiting properties.

1.6.1 Enterprise Architect

A new version of the g9 profile is available containing three new tagged values under the PropertyDB stereotype for attributes:

- g9 DB:Column Name
- g9 DB:Insert Prohibited
- g9 DB:Update Prohibited

The tagged value **g9 DB:Table Name** can be used on classes, but it is not connected to any class stereotype.

1.6.2 Ecore/Xcore

Four new annotations are available under g9:

- tableName
- columnName
- isInsertProhibited
- isUpdateprohibited

1.6.3 Java Model

Three new field values are available in the `@G9Attribute` annotation defined in G9Attribute.java:

- columnName
- isInsertProhibited
- isUpdateProhibited

These properties could also be set using the JPA annotation `@Columns`

- name
- insertable - Inverse of InsertProhibited
- updatable - Inverse of UpdateProhibited

One new field value is available in the `@G9Class` annotation defined in G9Class.java

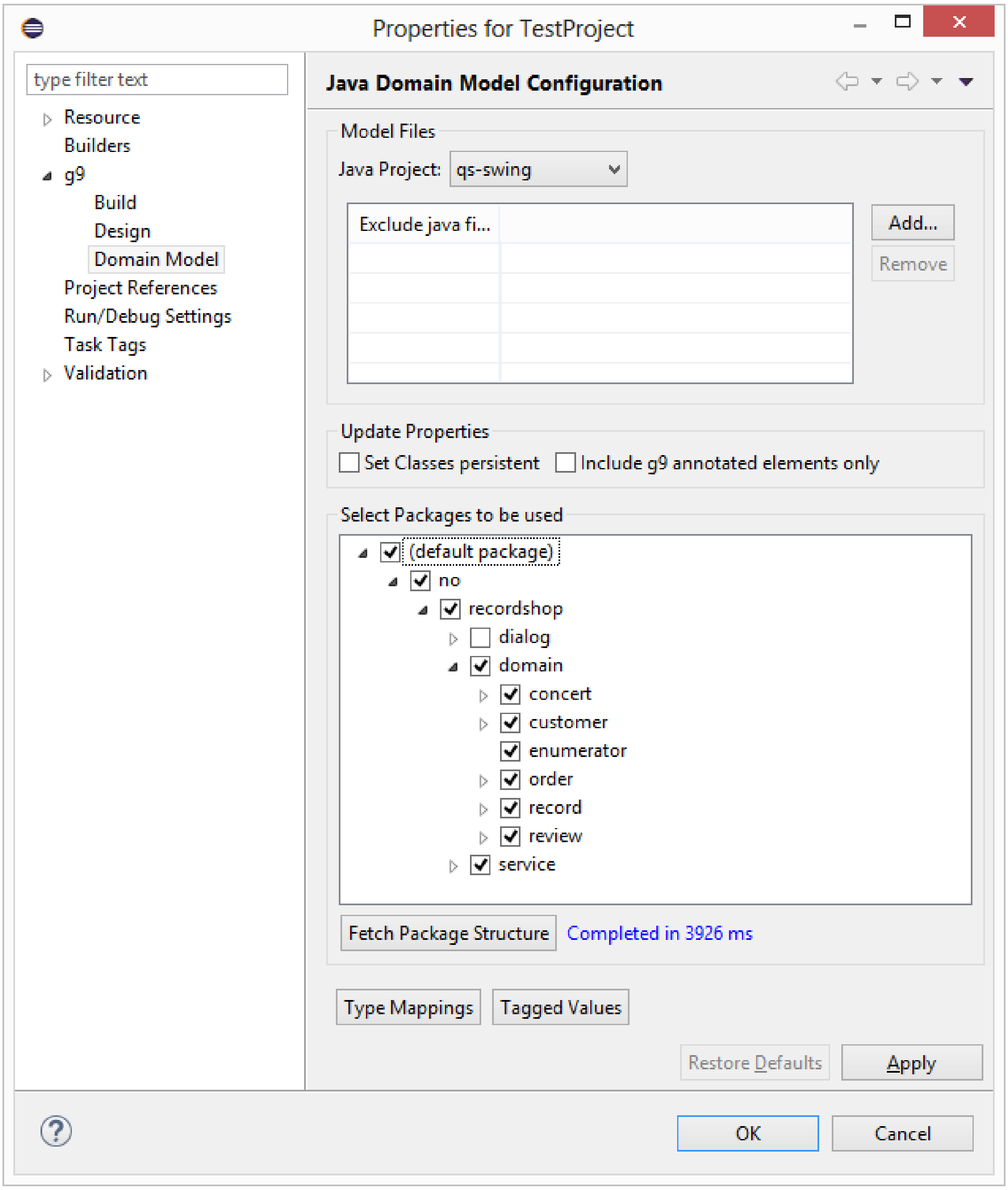
- tableName

This property could also be set using the JPA annotation `@Entity`

- name

1.7 Modeling with Java source code as the Model Source

On the g9 project property page *Domain Model* for *Java Domain Model Configuration* is a new check box added: *Include g9 annotated elements only*.



Java Project: qs-swing

Exclude java fi...

□ Set Classes persistent □ Include g9 annotated elements only

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

Completed in 3926 ms

1.7.1 Filter for hashCode(), equals() and toString() methods

The following methods are ignored when java code is used as model:

- All constructors.
- All methods overriding a method from an interface or a class where the source is not part of the source code. This implies that reimplementation of methods such as equals, toString etc. are ignored.

- Methods matching utility methods generated by the g9 java generator:
  - `Map<String,Object> getPrimaryKey()`
  - `boolean equalKeys(Object other)`

In addition, the g9 generator produces `static String PK.` This variable is also ignored.

This behavior can be overridden individually by adding the `@G9Method` annotation to an ignored method or `@G9Attribute` to an ignored field.

1.7.2 Support for JPA annotation @Column

The JPA annotation `@Column` is supported and the following fields are read:

- name: Will override the default setting of column names in a database model
- length: Same as g9's DB property Byte (Length)
- precision: Same as g9's DB property Precision
- scale: Same as g9's DB property Scale
- nullable: Same as inverse of g9's PropertyDB NotNull
- unique: Same as g9's PropertyDB Unique

1.7.3 Support for JPA annotation field mappedBy in the association annotations

The JPA annotations `@ManyToMany`, `@OneToMany` and `@OneToOne` contains a field mappedBy referencing the method or field representing the opposite end of a bidirectional association. It is used in the same way as the annotation field `@G9Association.oppositeRole`.

1.7.4 Use @G9Exclude to exclude elements from the domain model

The following elements defined inside a class or an interface can use this annotation and thereby be excluded:

- Methods
- Fields
- Classes
- Interfaces
- Enumerators

Outer classes, interfaces and enumerators are not excluded in this way. They are excluded by adding their . java file to the list of excluded files in the Java Domain Model configuration dialog.

1.7.5 Change in reference to Converter and Group

The type of `@G9Attribute.converter` is changed from String to Class.

The field value should reference the converter class as shown in this example:
`@G9Converter(value=no.recordshop.domain.converter.LineItemConverter.class)`

The type of the `@G9Group` annotation is changed from ElementType.FIELD to ElementType.TYPE. This is a pure tagging annotation without fields. The group class is used in an ordinary class by creating a field with the group class as data type. The fields in the group class define the content of the group and all fields should match by name a field, a group field or an association in the class containing the group.

1.7.6 Import set/get methods as attributes or associations

Property methods will not be imported as methods, but they may be imported as attributes or as associations.

Three types of methods are regarded as property methods and should match the following pattern:

- `boolean isFieldname()`
- `anytype getFieldname()`
- `void setFieldname(anytype parameter)`

In the get method the type should not be void. The first character of the fieldname should always be an uppercase letter.

Methods regarded as property methods will never be imported as methods, but they may be imported as attributes or as associations.

The name will be the same as the field name part of the method name, but with the first letter in lower case.

If a field exists with the same name as the name derived from a property method, only the field is imported and the property methods and their annotations will be ignored, with one exception the fields visability is taken from the get/is method.

If a field does not exists, the properties and annotations are taken from the get/is method. The set method will be ignored and its annotation will be ignored.

If only a set function exists it will be imported as a method and not as an attribute or an association.

In addition to these rules any method returning a non-void type and defined as a column method via JPA annotations, will be imported as an attribute or as an association. In this case the attribute name will be the same as the method name.

1.7.7 Name changes for java model g9 annotations

The following annotation fields are changed as explained:

`@G9Attribute`

- Name of *constraint* is changed to *constraints*
- Name of *fieldlength* is changed to *fieldLength*
- Name of *displayrule* is changed to *displayRule*
- Type of *primaryKeyCreator* is changed from *String* to the new enumerator *PrimaryKeyCreator*. Legal values:
    - Database
    - Application

`@G9EnumValue`

- Name of *displaytext* is changed to *displayText*
- The new int field *ordinalNumber* can be used to set ordinal number for an enumerator value

`@G9Association`

- Name of association *name*. This is a new field compatible with the associationName annotation for Ecore/Xcore

1.8 Modeling with Ecore/Xcore

The Ecore import should convert collections of primary types to collections of wrapper types.

When modeling in Xcore, collections are by default interpreted as either List or Set.

- Integer[] is interpreted as a List<Integer>
- unordered Integer[] is interpreted as a Set<Integer>

Such a declaration using any of the java primitive types will give illegal java code: long[] number was interpreted as a List<long>

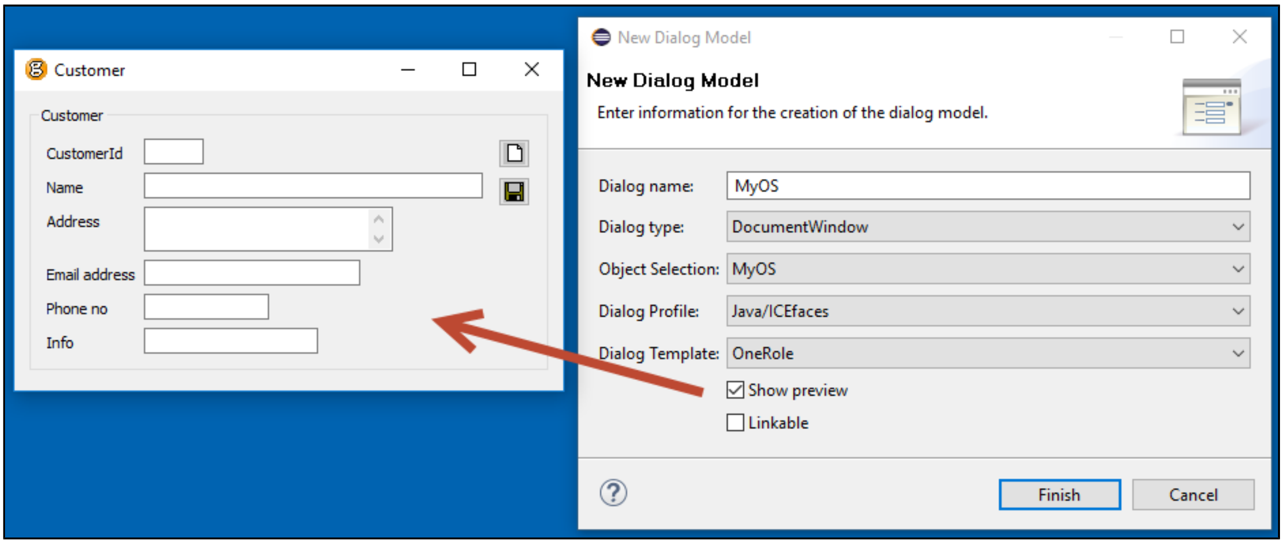
Now java primitives are automatically converted to their wrapper type, giving the result List<Long>.

This conversion is only done when using the [] syntax in Xcore, and only when not using the g9 annotation: `@g9(isArray=”true”)`

1.9 Preview from New dialog wizard

Toggle a "Show preview" checkbutton to show a preview while changing settings in the New DIalog Model wizard.

New Dialog Wizard gets a checkbox for showing preview. This can be very useful for selecting the right dialog template.



New Dialog Model

New Dialog Model

Enter information for the creation of the dialog model.

Dialog name: MyOS

Dialog type: DocumentWindow

Object Selection: MyOS

Dialog Profile: Java/ICEfaces

Dialog Template: OneRole

☑ Show preview

□ Linkable

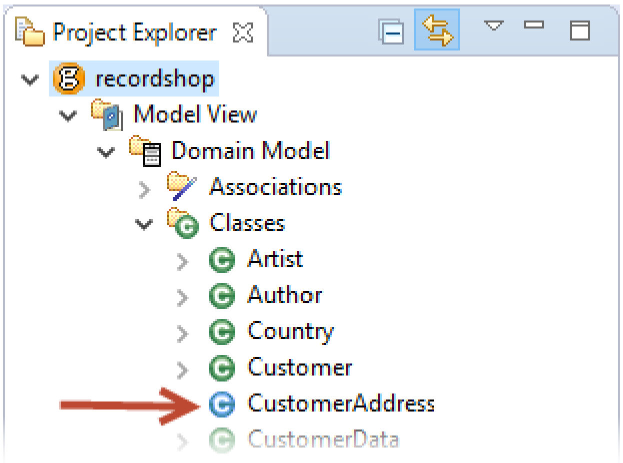
Finish Cancel

The preview will appear as a non-focused window as soon as the checkbox is checked and it will disappear when unchecking or finishing the wizard. It will update as changes are made to the relevant parameters. The Dialog wizard is modal so it is not possible to move or resize the preview while the Dialog wizard is open.

1.10 Value Objects

The Swing generator now supports Value Object.

In the Domain Model is Value objects shown as blue icons as a opposed to the green Class icon.



2 Generated code and Model Changes

2.1 Changes in model files

Project file (.gproject)

New property for Source Directory.

Domain Model files (.gdox):

New tagged values AttributeInsertProhibited and AttributeUpdateProhibited.

Resources (.grdx):

Added to Skins: StandardOneColumn.
OneRole Dialog Template uses the StandardOneColumn skin.

2.2 Changes in generated code

Regeneration of code is necessary.

3 Dependent versions

g9 runtime libraries uses 3. party open source libraries. Supported versions in this release are:

Library	Version		Library	Version
Spring	4.0.9.RELEASE		Hibernate	5.0.1.Final 5.0.7.Final
ICEfaces	4.1.1 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

Special version of g9 runtime library without support for Spring WS (available on demand):

Library	Version		Library	Version
Spring	4.2.6.RELEASE		Hibernate	5.0.1.Final 5.0.7.Final
ICEfaces	4.1.1 4.0.0-GA		JSF	2.2.10
log4j	2.4.1		Jetty	9.2.6.v20141205
Jackson	2.6.3		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-4946 - Not able to delete dialogs/OS from application model

SUP-4966 - Cancel/reopen of Domain Model project property page loses column width in display of file list
Release Comment: Table type widgets in various wizards are fixed.

SUP-5031 - Background color in property editor switches for every other attribute

SUP-5032 - Build on Domain Model node in Application Model should be removed

SUP-5041 - Context menu for Database Models in an application model don't contain Add all... and New...

SUP-5048 - Application model "forgets" dialog/OS names after rename

SUP-5049 - Classes with same name fails in application models

SUP-5050 - Rename of OS does not update Application models
Release Comment: Rename of a model will automatically update any application models referencing the model.

SUP-5051 - Cached OOModel should be removed when domain model is changed on file system
Release Comment: Database models were synchronized with the old version of the domain model.

SUP-5052 - Rename of tables in database model is not refreshed

SUP-5054 - Rename of models does not check the new name against already existing names
Release Comment: This was the case for dialog models, object selections, application models and database models.

SUP-5055 - Rename not allowed for database models
Release Comment: The action is now allowed

SUP-5057 - The default name for a generator configuration should be the generator name

SUP-5062 - Avoid exporting the default property values as g9 Ecore annotations
Release Comment: The input and output justification annotations for attributes are now only exported if the value differ from the default value.

SUP-5065 - Preview for New Dialog Model wizard
Release Comment: Toggle a "Show preview" check-button to show a preview while changing settings in the New Dialog Model wizard.

SUP-5066 - Support for src-gen and src for all generators
Release Comment: The generators now supports splitting generated and manually maintained code into two separate source hierarchies. This new behavior is triggered by setting the "Source directory" generator parameter to a different value than the "Target directory" parameter. With the new behavior, if a file is present in the "Source directory" it will not be generated to the "Target directory". A ".new" file will instead be generated to the "Target directory" if the "Generate .new files" parameter is true. The new behavior is the default for new generator configurations, and the default value for the "Generated package" parameter is now blank. Existing generator configurations will keep the old behavior until the "Source directory" parameter is changed by the user.

SUP-5068 - New DB properties in domain model: Table Name, Column Name, Insert Prohibited and Update Prohibited
Release Comment: See the News and Changes section.

SUP-5079 - Support for generating Xcore domain models
Release Comment: The Xcore generator generates Xcore models with g9 annotations from a g9 domain model.

SUP-5080 - Unix line endings on Windows gives wrong line-endings in generated source

SUP-5085 - Resources TreeView images have wrong ref
Release Comment: System default and sample projects: default.grdx. Removed image refs and function types.

SUP-5088 - The Ecore export should use a lower cased first letter on group references
Release Comment: Group references from a class are now generated with a lower cased first letter.

SUP-5095 - Property view is not updated after a rename of an editor element

SUP-5099 - Double click in Problems view does not bring you to the element in the domain model
Release Comment: The error only occurs when the domain model is based on java classes. The correction implies changes of the internal id for all domain model elements. It is recommended to do a complete update when doing the first "Update from Domain Model" with this g9 version.

SUP-5101 - Images for Value objects
Release Comment: Value objects are blue as a opposed to the green Class objects. They differ substantially and needs to be easily distinguished.

SUP-5102 - Search and Go To functionality support for Value objects

SUP-5104 - The domain model is validated after the first Type Mapping change
Release Comment: Changes in project properties should not trigger complete validation. After a Domain Model Update some remaining validation requests were processed at a later stage.

SUP-5109 - Default value for column/group property Unique in database models should be True when member cardinality is 1
Release Comment: This is the case both with cardinality 1 and 1..1 which are compatible.

SUP-5113 - Specific charset get handled wrong when saving models
Release Comment: When files had different internal charset settings in XML header than the workspace, it would mess up on save.

SUP-5117 - Dialog Template has Skin refs <Not Set>, should be Standard
Release Comment: If the Dialog Template in system has Skin ref <Not set>, it is changed to Standard. Private Resource files are not changed.

SUP-5119 - Add OneColumn Skin
Release Comment: OneRole dialog template is generated with one column data

SUP-5121 - Schema generation does not create Unique constraint for columns and groups
Release Comment: This corrections affects the schema generated for Sybase, SQLServer, MySQL and Derby. Only columns and groups defined both as unique and as index were given a unique constraint. Now any column and group defined as unique will be given this constraint.

SUP-5125 - NPE in SOAP/Castor when the return type of a method is an Usertype
Release Comment: When method return types are an Usertype, the generated response type in the Castor mapping file is now handled in the same manner as if the return type is a domain model class. The same applies to method parameter types.

SUP-5130 - Dialogs modeled in EA should have special annotations in Application model
Release Comment: Enterprise Architect modeled dialogs are distinguished by an yellow square annotation in top left corner. These dialogs cannot be deleted or modified within the EA modeled application.

SUP-5132 - Copy of model does not update internal id
Release Comment: When copying model files on filesystem and thereafter editing them in model editor would lead to NPE and other errors.

SUP-5137 - Delete of groups of dialog widgets is very slow

SUP-5139 - Restore button on Build Configurations does not work for aggregated build configurations

4.2 Swing

SUP-5058 - Support for Value Objects in the Swing generator

4.3 Java runtime

SUP-5090 - TAD-912, SUP-4643 - "Remove unloaded domain objects from the result after find actions" is not implemented in the EJB version of runtime
Release Comment: In the g9 version 1.12 when implementing this functionality, only the local version was changed. The needed changes were not implemented in the EJB3 version of the g9 runtime.

4.4 Java Domain

SUP-5073 - Support for generating JPA annotations in the Java Domain generator
Release Comment: The Java Domain generator can generate JPA annotations to the domain classes. Set the "Generate JPA annotations" parameter to "True" to activate this feature.

SUP-5075 - Generated Java classes don't have skeleton methods for implemented model interfaces
Release Comment: A skeleton method is now generated in the class for each method from the modeled interfaces of the class.

SUP-5076 - Support for generating g9 annotations in the Java Domain generator
Release Comment: The Java Domain generator now can generate g9 annotations to the domain classes. Set the "Generate g9 annotations" parameter to "True" to activate this feature.

SUP-5089 - The constructor of Value object classes without local attributes doesn't compile

SUP-5092 - Support for generating Groups, Converters and Usertypes in the Java Domain generator
Release Comment: The Java Domain generator now supports generating Groups, Converters and Usertypes. These model elements are only generated if they are referenced from a Class, an Interface, or an Enumerator that is also generated. Set the "Generate groups", "Generate converters" or "Generate Usertypes" parameter to "True" to activate this feature.

4.5 EA sync

SUP-5106 - Double escapes of special characters in descriptions
Release Comment: This was a error when modeling with Enterprise Architect. The xml special characters are now treated correctly.

4.6 Xcore sync

SUP-5040 - The Ecore import should convert collections of primary types to collections of wrapper types
Release Comment: See the News and Changes section.

SUP-5072 - The isUnique annotation on attributes and group references in Ecore/Xcore models should be replaced with the isUnique property from ETypedElement
Release Comment: Compatibility: In models already annotated with the isUnique annotation this annotation will override the ecore unique setting.

SUP-5129 - Association annotations are not always read when updating domain model from Ecore/Xcore
Release Comment: The reading is dependent of the end of the association used for the annotations and the sequence the references are read from the model files. This could affect setting of the association name and the properties Persistent, Owner Role and Foreign key. In a 1 to 1 association, a missing Owner Role property could switch the owner/member setting for the association.

SUP-5138 - Methods and parameters don't get model type set correctly when updating from Ecore/Xcore
Release Comment: The error occurs when the type is of an modeled element type Interface or Enumerator. The reference to the element was not set.

4.7 Java sync

SUP-5036 - Java as Model: Need a filter to hashCode/equals/toString-methods
Release Comment: See the News and Changes section.

SUP-5056 - Support for JPA annotation `@Column`
Release Comment: See the News and Changes section.

SUP-5067 - Java as model: set/get methods should be imported as attributes or as associations
Release Comment: See the News and Changes section.

SUP-5071 - Change in reference to Converter and Group in g9's Java Model annotations
Release Comment: See the News and Changes section.

SUP-5074 - Use settings in JPA annotation field mappedBy in the association annotations
Release Comment: The JPA annotations `@ManyToMany`, `@OneToMany` and `@OneToOne` contains a field *mappedBy* referencing to the method or field representing the opposite end of a bidirectional association. It is used in the same way as the annotation field `@G9Association.oppositeRole`.

SUP-5077 - Some of the g9 annotations for java model has names inconsistent with the g9 Ecore annotations
Release Comment: See the News and Changes section.

SUP-5084 - Field and methods with JPA annotation `@Version` should be ignored when updating from a Java model file
Release Comment: Such fields are regarded as column used by optimistic locking and are not seen as part of the model.

SUP-5096 - New annotation `@G9Exclude` used to exclude elements from a java domain model
Release Comment: See the News and Changes section.

SUP-5100 - Update Domain model for a java model project does not set model type for attributes correctly
Release Comment: Fields with a model type from an imported class not being part of the domain model did not get the package path included in the model type, e.g. an attribute with type `java.math.BigDecimal` was only given the model type `BigDecimal`.

SUP-5112 - Final attributes is not set to *const* when updating domain model from java source code

SUP-5115 - Properties for groups is missing when updating domain model from java source code

SUP-5116 - 1 to 1 associations with specified owner role does not always get correct owner class when updating domain model from java source

4.8 Known Issues

4.8.1 Location of the g9 runtime libraries when using the Eclipse Installer (Oomph)

The Oomph based Eclipse Installer uses shared bundle pools for the Eclipse plugins. When using the Eclipse Installer, the location of the g9 runtime files will be in the *plugins* directory in one of the shared bundle pools, and not in the *plugins* directory under the Eclipse installation as reported by g9. Look for the *no.esito.g9.runtime_<version>/jar/mvn* directory in the plugins directory of the bundle pool. The bundle pool can be found in the *Bundle Pool Management* dialog in the Eclipse Installer. Usually, this will be a directory named <home.dir>/.p2/pool.

4.8.2 Known error regarding resource projects and upgrades

This is valid if 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.

4.8.3 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.

4.8.4 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.

4.8.5 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 g9-2.4.0.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 Mars or newer versions of Eclipse.

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 1.3.1, 1.3.2 and 1.4.0
- Xtext version 2.9.1, 2.9.2 and 2.10.0

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: 2.9.1, 2.9.2 and 2.10.0