

Objecteering

Version 6.0 SP2: Release Notes

Table of contents

Table of contents	2
Modeler	3
CMS	4
C++ Developer (CxxDeveloper).....	4
Documentation	6
DOORS	6
Fortran Developer	6
Java Developer	6
MDA Modeler	7
RAMComponent.....	8
Requirements (RequirementsManager).....	8
SQL Designer.....	8

Modeler

- The "Save as" and "Copy" commands from UML 1.4 diagrams are now available in read only mode.
- Links and link ends can now be correctly cut and pasted.
- Abstract elements are now displayed in italics.
- Signals can now be created within interfaces.
- Data types and enumerations can now be unmasked in interfaces.
- Use cases can now be created within classes.
- No more error messages when deploying the MDAModeler MDAC.
- General improvement of model components.
- The "File/Open" menu now opens on the "work" directory.
- Interface realizations and template bindings now have different graphical representations.
- Enumerations can now be created in class diagrams.
- It is now possible to unmask a link using drag and drop. Objecteering no more crashes on a Ctrl-D in UML 1.4 diagrams.
- It is now possible to select an association role in the explorer from a class diagram.
- "Unmask/All contents" command under a package now unmask all elements.
- Accents are now properly managed in the description of RAMCs.
- Interfaces are now abstracts by default.
- Objecteering can now be used in command line.
- MDACs parameters values are now kept when upgrading an MDAC.
- The selection of a role in a diagram now refreshes the auxiliary window.
- Tagged values parameters are now displayed in diagrams.
- The field "Type Constraint" of the return parameter edition box has been renamed "Type constraint / String size".
- Correction of an Audit issue regarding sequence diagrams.
- DataFlows names are now displayed in the diagrams.
- The case is no more ignored on Attributes' Initial Value in the Java Properties box.
- UML 1.4 Collaboration diagrams are no more migrated into UML 2.0 diagrams, both UML 1.4 Collaboration diagram and UML 2.0 Composite structure diagram can now be created.
- Display of Help menu entry now matches the IHM standards.
- A class' destructor can now be declared abstract.
- Cycles between DataFlows are no more reported by the Audit.
- Sequence Messages which have Dependency with <<throw>> links are now correctly migrated.
- Correction of a crash occurring when connecting a Requirements project to a non Requirement CMS repository.
- Improvement of the Sequence Message deletion process.
- Correction of a crash occurring when trying to create a traceability link between elements and Requirements in Read Only mode.
- WinMerge bundle was upgraded from 2.4.0.0 to 2.6.4.0.
- Notes can now be created/unmasked directly on a diagram.
- Correction of a crash occurring when creating a Class Association in Class diagrams.
- Correction of a crash occurring when unmasking Template Binding between 2 UseCases in Class diagrams.
- Work Products cannot be edited twice anymore.
- It is now possible to copy/paste stereotypes from the auxiliary properties window.
- Auxiliary Window is now refreshed when deploying an MDAC.

- It is now possible to add Notes, Tagged Values, Stereotypes, and Constraints on Enumeration Literals
- Tagged values are now correctly managed via the MDACs tabs.
- RAMCs are now correctly managed by the Diff Merge facility.
- Correction of a crash occurring when drag and dropping a Template Class into its own dialog box.
- Template Parameters name are now correctly displayed after a drag and drop to an Attribute type.
- Work Products are now correctly propagated in any case.
- CxxModule is no more automatically replaced by the CxxDeveloper MDAC when migrating a 5.3.0 database.
- Correction of an error occurring during a Check-In with a Java code retrieval.
- Improvement of memory management during MDACs packaging.
- Colors are now correctly saved in State Diagrams.
- Diagrams are present again in the generated documentation.
- Required and Provided Interfaces can only be created between Component/Class/Part/Instance and Part/Instance.
- Errors occurring during automatic merge are now displayed in an end of session report.
- Correction of an issue regarding attributes' colors in diagrams.
- Toolbar's help bubbles are now displayed.
- Clearer audit messages are displayed when trying to make Template Parameter a Parent for some forbidden elements.

CMS

- The "PVCSDimensionsModule" has been renamed "Dimensions".
- Correction of a bug that could occur when using non hierarchic imports.
- RAMCs are now automatically deployed when needed, using the "RamcList.conf" configuration file.
- The 'Update generation work products before check-in' parameter is now taken into account by CxxDeveloper.
- Multiuser's "*.read", "*.write" and "multiuser.lock" files are now in the \$MU/Locks/users/ directory instead of the \$MU/Locks directory. This greatly improves performances when locking the repository from a distant network.
- Dimensions MDAC now supports Merant Dimensions 8.0 and Serena Dimensions 9.1
- CMSynergy MDAC now supports CMSynergy 6.3 and 6.4 (The 'Locks' directory is not versioned anymore, only 'Shared' mode CM Synergy projects are now supported, Personal/Insulated CM Synergy projects are not supported anymore. Existing repositories have to be recreated by the CM Synergy repository administrator.)

C++ Developer (CxxDeveloper)

- The automatic creation of constructors, destructor and operators is now activated when a class is created in a diagram.
- Enumerations and enumeration literals are now managed by doxygen generation.
- Parameter cardinalities are now taken into account.
- A new parameter, "Use description as doxygen", has been added.
- Abstract methods are now automatically declared virtual.
- Cardinalities of return parameters are now taken into account.

- Setting the "throws" field on an operation to "(") prevents it from launching an exception.
- Inner classes are now forward declared correctly.
- Include directives are now correctly generated when a class declares a copy constructor.
- Includes are now correctly generated for generic interfaces.
- Base constructor code is now generated for constructors with parameters.
- The "argument" field in the association end edition box is now populated.
- All thrown exceptions are now generated.
- Declaration of methods pure virtual throwing exceptions is now correctly generated.
- The {}virtual tagged value is no more taken into account on 'cannot be specialized' methods.
- Correction in the generation of the ".hpp" in the package for a template instance datatype.
- The "Arguments" field in the C++ properties window is now correctly taken into account.
- Documentation can now be edited on enumeration literals.
- Doxygen notes are now generated in the .Hpp file with markers.
- Modified header code is now correctly updated after external edition.
- The manual declaration on operation now work correctly.
- Pure virtual methods are generated with a body only if a 'Cxx.Code' note is present on the operation.
- Class templates with static const attribute are now correctly generated.
- No more J error when throwing an exception to a class tagged {}Nocode.
- The pointer type of a qualifier is now taken into account in the generation of the map key.
- Attributes tagged {}NoCode are no more generated.
- General improvement of the old C++ tags conversion.
- All '?' buttons from the Project Edition window now lead to the relevant online Help.
- CxxDeveloper is now fully coupled with the CMS MDACs.
- General improvement of the Project Management feature.
- Generation is now verbose.
- "Inline" keyword is no more added to the declaration of abstract operations.
- "ElementImport" from Enumerations are now generated.
- Addition of a "Multiple return parameter by collection" parameter, making it possible to generate collections for return parameters.
- Virtual tagged values are now removed if operations are changed from "Abstract" to "Cannot be specialized".
- Attributes, Parameters and Associations are now initialized with the Cxx.value Note when it is present (and even if it is empty). Otherwise, the UML value is used instead.
- Addition of a "Automatic guessing strategy" parameter, allowing the user to choose whether the guessing must be automatic or manual.
- Addition of a "Directory for personal ACT files" parameter, allowing the user to provide its own ACT files.
- Introduction of the concept of Smart Generation. A generation that skips elements which don't need to be regenerated, as opposed to Forced generation, which regenerates the whole project.
- Addition of a "Default behavior for file retrieving" parameter, allowing the user to choose the behavior when, upon generation, the generator finds a file which could be updated. Choices are: 'ask', 'retrieve all files' or 'force the regeneration'.
- It is now possible to add an external file, or a set of external files, that need to be compiled with the application. To add those files, create an artifact and stereotype it as <<Cxx.Files.Extern>>. This artifact represents a set of c++ files in the project, not modeled directly in the model, but which need to be in the project directory. This artifact can set filename in two ways. In the File name attribute of the artifact, when

there's only one file to manage. File name must be related to the cxx project directory, and we advise that a '/' be used as directory separator. When there's more than one file to manage, {}Cxx.Filenames tagged value must be used.

- Old {*} C++ Tagged Value is now correctly converted.
- Correction of a J error occurring when launching the 'Convert old C++ tags' command on a migrated model containing read-only stereotypes.
- Static Const Associations with List as Container type are now correctly generated.
- Correction of a J error occurring when launching a check-in on Requirements projects.
- Associations from Class to Datatype are now correctly generated.
- Correction of J Errors occurring when generating model with empty {}Cxx.Name or {}Cxx.Container Tagged Values.
- Get accessors are now generated as Const.

Documentation

- Profiles, templates and J code are now visible and modifiable, in order to import and redefine them in a new MDA project.

DOORS

- It is now possible to import the DOORS module containing utf-8 characters.
- Improvement in the management of user rights during export operations.
- General improvement in the import operation.
- Newly created DOORS "Objects below" are now correctly imported in Objecteering model.

Fortran Developer

- Models migrated from 5.3.0 are now correctly generated.

Java Developer

- Enumerations and enumeration literals are now managed by the reverse and javadoc generation.
- Classes stereotyped <<interface>> are now generated as interfaces.
- Correction of a bug concerning the reversal of classes directly under the root package.
- Javadoc on enumerations are now correctly retrieved in Model Driven mode.
- Correction of a bug concerning the reversal of enumerations that contains methods and attributes.
- Correction in the generation of methods of an interface that throws exceptions
- The {}JavaBind tagged value is now taken into account when set on an generalization link between two generic classes.
- The {} javaExtern tagged value is now taken into account when applied to a datatype in a class.

- The {}nocode tagged value is now correctly taken into account.
- Work Products are now correctly propagated.
- "Create the methods to be implemented/Redefined" commands now work from RAMCs.
- Objecteering now generates ant files instead of makefiles.
- General improvement of the imports generation.
- It is now possible to set elements' visibility to "undefined" with the Java Properties tab.
- General improvement of the generation of the "@param" and "@return" tagged values.
- Enumerations' visibility is now taken into account.
- Corrections regarding the {}JavaTypeExpr tagged value generation.
- Addition of a {}JavaNocode tagged value preventing model elements from being generated.
- Addition of a "Key Type" field in the Java Properties tab, available for Associations, Attributes and Parameters with multiple cardinality.
- Correction regarding the generation of Classes containing Enumerations which implement Interfaces.
- Interface contained into Classes are now generated.
- General improvement of the reverse feature.
- Addition of an "objingid" annotation for classes, interfaces, enumerations, attributes, associations and operations, allowing the Round Trip to find the modeling element in the Objecteering project when it is reversed.
- 'Generate java code', 'Generate Ant file', 'Build Ant file', 'Create a Jar file' and 'Execute Ant target' buttons have been added to the JarFile's Java tab.
- General improvement of the reverse of the generics.
- Interfaces' template parameters and enumerations are now reversed.
- Enumerations literals are now correctly reversed and generated.
- Annotations are now reversed on constructors.
- JavaAnnotations are now correctly reversed.
- Useless @return annotations are no more generated on setters.
- Model Driven markers are no more generated in Round Trip mode.
- A comprehensive error message is now shown when the reverse tries to create elements under Read-Only elements.
- 'Automatically create diagrams on initial reverse' option now works correctly.
- The 'run an Applet' button now works correctly.

MDA Modeler

- The selection of a directory to export patterns is now allowed.
- Correction of a J error when a navigation node or a template's metaclass is modified.
- Correction of a J error when a template is applied using MDARuntime.
- The update of the framework is now possible.
- Addition of a new DocBook tag to generate an RTF file inclusion field. This tag generates a hyperlink for the HTML and PDF formats.
- Template selection is now possible.
- The generation of XML files (DocBook) is now allowed.
- The resource directory is now created before packaging.
- Addition of help bubbles for icons in the "MDA" property page.
- Addition of a new button to create old documentation templates.

RAMComponent

- The “Remove model component” command now works correctly.
- The “.ramc” extension is now correctly managed in the different file choosers.

Requirements (RequirementsManager)

- Correction of bug that was blocking the modification of Requirements objects in the generated documentation.
- Update of the Requirements toolbar of generated documentation.
- Correction of a presentation problem in the generated documentation.
- Addition of a Uno package (Open Office) making it possible to export Open Office documents towards Objecteering.
- XML file extension is now set by default while choosing the XML file generation path in the generated documentation.

SQL Designer

- Diagrams generated with the physical model are now in UML2 version.
- The diagram coming with the First Steps is now in UML2 version.
- Correction in Oracle predefined types.
- Tooltips are now available for the SQLDesigner tool bar.
- Modified parameters for the ‘Run SQL’ command are now taken into account.
- Attribute’s ‘SQL type ‘ field now allow to choose between all accessible types.
- ‘SQL type constraint’ field can now be used for all types (UML, SQL Server, Oracle...)
- Oracle’s IN OUT parameters of a stored procedure are now correctly generated.
- Default value are no longer automatically set for Sybase and SQLServer’s CHAR and VARCHAR.
- It is now possible to generate SQL code if a 'char' parameter of an operation has its field "Max" set.