@enterprise 9.0 Release Notes

Groiss Informatics GmbH

Strutzmannstraße 10/4
9020 Klagenfurt
Austria

Tel: +43 463 504694 - 0
Fax: +43 463 504594 - 10
Email: support@groiss.com

Document Version 9.0.33982

Contents

1 Introduction

This document presents the changes and new features in version 9.0 of **@enterprise**.

2 End user GUI

The end user GUI has been completely rewritten to improve the usability and give it a modern look. The main changes are:

- Drag and drop can be used for moving worklist entries between folders. For example, the actions "take" and "give back" can be done using drag and drop. The document management also supports drag and drop.

- HTML Frames are not used anymore, Iframes are used only for application forms.

- Many GUI components - tree, table, pop-up windows, etc. - are taken from the open source library dojo (http://http://dojotoolkit.org/).

- context sensitive help: Clicking F1 or the help button opens a context sensitive help.

- Client notification: The worklist refreshes automatically if another user sends a work item to the logged in user.

Apart from the different look additional features have been implemented:

2.1 Additional features

2.1.1 Worklist

Additional columns can be configured:

- application: the application of the process definition of the activity instance

- lastAction: the previous action on the activity instance, for example: process start, finish, recall.

2.1.2 Input filter for user folders

A condition can be defined for each user folder. Entries are moved automatically or manually to the appropriate folder depending on the defined expression.

2.1.3 Process tracking

This new function allows to track the process. It sends an e-mail to the observer whenever the process or some specific task of a process instance is finished.

2.1.4 Structured history

The process history can now be viewed as structure. This may be very helpful if the process structure is complex (parallelisms, loops).

2.1.5 Recent activities

This functions shows the processes the user has processed today, yesterday, last week, and last month.

2.1.6 User profile

The user profile contains some information about the user (name, roles, e-mail address, phone number, photo, ...). It is shown on several places in the user interface, for example in the process history.

2.1.7 Combining GUI configurations

In **@enterprise** installations where more than one application is used it may be convenient having a gui configuration combining the features of all applications. The gui configuration `combined` has the following features:

- It contains a placeholder (APP_TREE) after the document section for navigation components from the applications. Applications must put a file *applid*/navigationcomponent.xml into the classpath.

- It uses a worklist adapter that combines the adapters defined in the configuration parameter `ep.worklist.adapters`. The interface methods (getList, modifyColumns, etc.) are called for each of the given classes.

2.2 Document management

2.2.1 Sign PDF documents

The DMS offers this function to sign PDF files using certificates. This method can be used for example for signing invoices compliant the German and Austrian regulations.

2.2.2 Open Office support

@enterprise offers a template engine for generating MS Word and PDF files based on Open Office (Libre Office). Templates can be created in ODT file format.

2.2.3 Up- and download of zip files

The function zip-upload (key: admin.upload) uploads a zip file and extracts the contents into the current DMS folder. The function zip-download (key: admin.download) creates and downloads a zip file containing the selected documents or folders. These functions are not in the standard configuration.

3 Administration

3.1 Reorganization of administration

Several object classes now have an application context. This means their administration is done in the application folder:

- value lists

- Timer

- Mailboxes

- Stored queries

- Web service servers and clients

An application export now contains the instances of these classes.

3.2 Application reports

For applications, processes and forms reports in HTML and PDF format can be created. The reports are intended to be used as technical documentation of an **@enterprise** application.

3.3 Copy function

Copy objects like users, roles, processes, etc.

3.4 User

The user form contains the following additional fields:

- Salutation

- Name suffix

- Gender (male/female)

Furthermore a picture can be added to the user form. This picture is displayed for example in user profile.

3.5 Forms

The form wizard has been adapted and allows the creation of indices on columns and the definition of dependencies (referential integrity).
HTML forms are deprecated and no longer an option when creating new forms, XHTML and XFORMS are the remaining options (With setting the configuration parameter ep.use.htmlforms to true, you can still create HTML forms).

3.6 Process editor

Subprocesses are displayed in a own process editor window now.

Caveat: The combination of *Internet Explorer 11* and SSL-connection does not work correctly with process editor!

3.7 Server monitor

JavaMelody (https://code.google.com/p/javamelody/) is used for showing the run-time characteristics of the server. Several measurements are undertaken, the results are displayed graphically.

3.8 Mail handling

Two enhancements:

- Support of diverse authorization types for sending and receiving emails.

- Mails can be sent directly or by using the integrated mail queue. A timer handles the mail queue. This makes sending mails more robust against network and mail server problems.

3.9 Mailbox

Enhancement of action handling for process starts: A process and a start organizational unit can be defined. Moreover, the text of an optional receipt mail can be defined.

3.10 @enterprise and application upgrade

This new component allows to download and upgrade patches for **@enterprise** or applications via GUI from an *application repository*. Moreover, a timer can be configured that checks the availability of new versions of **@enterprise** or any of the installed application.
Groiss Informatics will maintain a site containing the latest version and patches of **@enterprise** and of some applications we provide (CRM, HR-Processes, ITSM). It is possible for a customer to maintain his own application repository containing patches of his own applications. Such a repository can facilitate the distribution of patches to several **@enterprise** installations.

3.11 Configuration

@enterprise reads the system configuration from a file. Several additional features have been implemented in this version:

- Storage of configuration parameters in multiple files is possible. You can even specify a directory containing configuration files. This feature can be used for example for separating cluster-wide and node-specific parameters in a cluster configuration.

- It is possible to have read-only configuration files.

- The configuration parameters of an application can be organized in several configuration groups for keeping the masks usable. The groups can be configured in the application administration, tab configuration.

- When editing the configuration you can view the default value, set a parameter to the default value and see the configuration file that contains the parameter.

- When a configuration file is changed from the GUI, a backup file is created. You can configure the number of backup files that are kept.

- You can define a callback for configuration changes by adding a PropertiesChangeListener to the configuration object. Example:

```
Configuration.get().addPropertiesChangeListener(
  new PropertiesChangeListener(){
    @Override
```

```
      public void propertiesChange(
        Map<String, Pair<Object, Object>> events) {
        ..
      }
    }, "aclcache.active", "aclcache.objectrights.maxelems");
```

The method addPropertiesChangeListener adds a handler for the two parameters to the **@enterprise** configuration.

Note, that the reconfigure method is no longer part of the service interface.

4 Workflow engine

4.1 New constructs

The engine and process editor supports the following new features:

- wait: A wait step can be used to halt the process execution for a time duration or until a point of time.

- goto end: This step allows to jump to the end of the process.

- Condition for parfors: A method (with boolean return value) or an expression can be defined, that defines whether a parfor branch is created for a form.

4.2 Definition of process start

- In the past processes could only be started from the GUI, if the first step was an interactive task. Now, all task nodes on a path from the begin node without another task or goto node in between are considered as start nodes. These start nodes may have different properties, for example agent definition, attached functions. On process start the user can select one start step if his role assignments match to more than one step.

- A process can be started by a timer (*ProcessStartTimer*), by incoming mail (depending on mailbox definition) or be a web service.

4.3 Escalation handling

The timeout task attached to the process has been removed. The definition of escalations has been moved from the task definition to the process definition. Escalations can be defined for every step that has a "duration": task, sync, batch step, web-service receive, and the process itself.

4.4 Batch processing

The exception behavior of batch job methods `doStart()` and `afterCompletion` has been changed. If an exception occurs, a rollback is done now instead of commit.

5 Reporting

5.0.1 Wizard

The user interface for creating reports has been reimplemented. The wizard-like structure makes it easier to define the different aspects of a report.

5.0.2 ReportTimer

The timer executes stored queries in defined periods. Depending on the defined output type,

- a chart as JPEG file

- a Excel file

- a CSV file

- a PDF file

- a XML file

can be created which can be sent per email to defined receivers or stored in the DMS or on the file system.

6 API and architecture

For API changes see the section *Upgrading an application* below.

6.1 User interface APIs

The new user interface brings some new APIs for writing GUI components in Javascript, based on the dojo framework.

6.2 New libraries

We use the following versions of libraries:

- Servlet 3.1

- Java 7

- JEE 6

- JDOM 2.0

- DOJO 1.9

- SLF4J

- Jetty 9

- Axis 1.6.2

6.3 LDAP

@enterprise offers the implementation `com.groiss.ldap.BasicUserDirectorySyncer` for Active Directory synchronization with **@enterprise** users.

6.4 Logging

@enterprise uses the library SLF4J for logging. It can show messages on five levels (ERROR, WARN, INFO, DEBUG, TRACE). The advantage of this library is, that you can define several loggers for your classes an modules. In the configuration, the log level for each logger can be defined separately.

6.5 Right system

The right system has been reimplemented. A consistent API has been defined for handling the special behaviour of some classes in an efficient way.

7 Security

Several actions have been taken to avoid possible vulnerabilities:

7.1 SQL Injection

This vulnerability happens when user input is embedded in SQL statements. In **@enterprise** all statements are prepared statements now to avoid SQL injection.

7.2 Cross-site scripting

Cross-site scripting (XSS) enables attackers to inject client-side scripts into Web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same origin policy.

@enterprise encodes user data and parameters to avoid cross-site-scripting.

7.3 Admin session

An own session for accessing the administration is used. This allows a better protection of administration functions: For example, if an administrator is logged in for doing normal work, he can not accidentally make operations where administration rights are used.
For this reason all **@enterprise** operations are classified:

- Public: executable without authorization,

- User: for authorized users only,

- Admin: for administration users only - in a dedicated administration session.

The classification is done using the Annotation com.groiss.servlet.Access with one of the above three values. It can be attached to a method, class or package.

It is possible to define an own IP interface and port for admin sessions. Together with some router or firewall, access to administration can be restricted to some sub-net.

8 Upgrading an application

@enterprise 9.0 is not fully compatible to previous versions of **@enterprise** for two reasons: We now use the most current libraries, for example of JDOM, Java enterprise edition, Axis, Jetty. That alone led to some incompatibilities.

Second, the new user interface caused several changes to existing APIs. For example, the links to **@enterprise** pages like the worklist or process history will no longer work. However, the old GUI remains part of the distribution: If you want to use **@enterprise** 9.0 without the new GUI, this may be an option for minimizing the porting effort. The gui configuration id for this compatibility mode is `standard_old`.

In the following, we describe the changes relevant for upgrading an application:

8.1 New libraries

Replace the libraries linked to your project with the libs from the **@enterprise** distribution. Most likely, you will get some compile errors now. If you have used the JDOM library, most of the errors caused by the new version can be fixed by replacing the package name from org.jdom to org.jdom2. If you get other errors and warning from third-party libraries, please contact the documentation of them.

8.2 Logging framework

The logging methods in the Settings class are deprecated. It is recommended to use the logging framework SLF4J directly. Add the following line into your code:

```
private static final Logger logger = LoggerFactory.getLogger(
        thisclassname.class);
```

The defined logger object can now be used for logging (methods error, warn, info, etc). You can define one such object per file or one per application, depending on how fine-grained you plan to set the log levels.

Directly streaming on the log file is still possible with `Settings.getLogWriter`.

8.3 SQLObject, LoggedObject, LockedObject and DistributedObject

Some versions ago the interface `Persistent`, the abstract class `PersistentObject` and the service `Store` have been introduced to replace SQLObject and its descendents. Now we set their methods (list, get, insert, update, delete,...) deprecated. The classes LockedObject and DistributedObject have been removed.

8.4 Persistent, FormHandler and PersistentEventHandler

The methods with signature onX() (X in insert, update, delete) in these interfaces and classes have been renamed to beforeX(). Following methods have been added:

- afterInsert(): This method is called after the object has been inserted into the database.

- afterUpdate(): This method is called after the object has been updated.

- afterDelete(): This method is called after the object has been deleted.

In the adapter classes for `FormHandler` and `PersistentEventHandler` the old method are deprecated.

8.5 WfEngine, DMS

The user handling in these interfaces has been unified: The user is always taken from `ThreadContext` (ThreadContext.getThreadPrincipal).
The method `setUser` in the interface `WfEngine` has been removed. Several methods in the DMS interface with the user as argument are deprecated.
The methods `getAcl` and `setACL` in DMSObject have been renamed to `getPermissionList` and `setPermissionList`.

8.6 MessageTemplate and MailSender

The class *MessageTemplate* contains methods for sending mails. The introduction of a mail queue and the request for simpler methods led to a new mechanism. See the description and example in the programming guide. Most of the old send methods are still present in class *MailSender*, but deprecated.

8.7 TableHandler

@enterprise provides the interfaces `DMSTableHandler`, `ObjectTableHandler`, `FormTableHandler` and `Worklist`. They have a similar structure with some differences according to the object types and table use.

	DMSTableHandler	ObjectTable	FormTableHandler	Worklist
init	x	x	x	x
getHTMLPage	x	-	-	x
getTitle	x	-	-	x
getList	x	x	x	x
modifyColumns	x	x	x	x
modifyTableLine	x	x	x	x
modifyActions	x	-	-	-
lineStyle	x	x	x	x

In the `Worklist` interface there are two additional methods: `getAdditionalData` and `listFilter`. The following changes have been made in version 9:

- In `modifyTableLine` the second argument is a `Map`, not a `KeyedList`. Therefore, changing values is done with the method `put` instead of `set`.

- `getList` has a default list as argument.

- The method `init` in Worklist: the parameter req may be null. This happens, if a worklist line is constructed by the notification mechanism.

- The method `getHTMLPage` is only called in old GUI.

- `getTitle`: in new GUI it is not called when the list changes.

14

8.8 ApplicationAdapter

In the application adapter the method `modifyDetailLinks` is only used in old GUI. In the new GUI the new method `modifyDetailPanes` is used. Note, that you con now configure the tabs in the process definition mask.

8.9 XML configuration

The structure of the Gui-configuration file has been changed. See the programming guide for the new structure. You must convert your configurations. If you have a GUI configuration database object, editing the configuration from the administration interface will initiate the conversion. Alternatively, you can start the conversion with the following URL:

```
http://host:port/wf/servlet.method/com.groiss.htmladmin.UpgradeGuiConfig.start
```

Enter the filename (without `.xml`) into the form and start the conversion. There may be errors, because in previous versions the XML files have not been validated against a schema.

8.10 Process Interface

The Process Interface definition has been removed. Starting a process by filling out and submitting a form can be done be defining a process-start node with mode FORM in the XML-configuration.

8.11 User interface

The new user interface has new methods for writing toolbar functions, adapting the details view or insert content into the navigation tree. See the programming guide for examples.
If you don't want to rewrite your functions and navigation links you can use them in the new GUI if you make some minor changes. Normally, the application function makes some changes and redirects then to the worklist or any other page of **@enterprise**. For redirection use the following hints:

- The default window for task-functions is an iframe with id `_compatPane`. Its position is below the toolbar and right from the navigation area. This iframe comes in foreground automatically if a page is loaded into this frame. You can prevent this by setting the window property `_isRedirectWindow` to true (see example below).

- In the new GUI the worklist is cached on the client and the changes are sent selectively. More details how worklists can be refreshed and shown again are described in the **@enterprise** programming guide in section *Worklist data* of chapter *Smartclient*.

- Never make a full refresh of the main page, because a lot of data is cached in the window object and will get lost. However, if you need to redirect to the main page, its url is:

  ```
  http://host.port/wf/servlet.method/
  com.groiss.smartclient.Main.showMainPage?id=theid
  ```

- If you want to show the process details (tabs for form, history, documents, etc.) use the following javascript method: ep.util.showProcessDetails(pi [, selTab])

  pi is the process instance in the following syntax: classname:oid, selTab a string containing the node id of the selected tab, for example admin.proc_history (this is the default).

8.12 Database

Installations using SQLServer as database engine might suffer from heterogeneous types being used for %OIDTYPE% patterns.
The old DB translator (used for SQLServer version < 2000) uses "DECIMAL(20)", the new one (for SQLServer versions >= 2005) uses "BIGINT". In really ancient installations, the type "DECIMAL(28)" might also be in use.

In section *4.5.2 Migration of deprecated MS SQL-Server data types* of Installation- and Configuration Guide two possibilities are described to remedy this potential problem.