

@enterprise 6.4

Release Notes

Groiss Informatics GmbH
Klagenfurt, January 2006

1 Introduction

This release notes describe the changes and new features in release 6.4 of @enterprise.

1.1 Installation and Configuration

The system is delivered as self-extracting jar file. The installation can be started with a double-click on the file `setup64.jar`. An installed JDK of at least version 1.4.2 is required.

When you have only a command line, start the installation with:

```
java -jar setup64.jar
```

Follow the steps of the setup procedure. Further information about the installation and configuration can be found in the installation guide.

1.2 Upgrade

To upgrade from a prior version perform the following steps:

1. Make a backup!
2. Extract the files to a new directory.
3. Copy your existing configuration file and forms directory to the corresponding directories of the new version.
4. Start the server and login as `sysadm`. You should now be redirected to the upgrade page. Perform the necessary database upgrades.

If upgrading from a version previous to 4.0, please contact our support.

2 New Features

2.1 HTML Client

The HTML client uses a new default XML file (`standard.xml` located in the `classes` directory). For the case that you prefer to work with the old layout (like the one used in @enterprise 6.3 and previous versions) we left a file named `standard_old.xml` in the same directory. Rename this file to `standard.xml` and you will see the old HTML client layout. Anyway, we recommend to use the new layout because the old XML file will not be updated any more, thus it is not guaranteed that all functionality of @enterprise 6.4 is accessible there.

The following features are new:

- The overall design of the user client changed.
- Calendar: When using the new standard xml a calendar tab is included. It allows the definition of appointments and resources.
- You can customize the columns of the worklist. Click on the icon right to the rightmost column header to get the menu for selecting columns.
- Short search: In the search tab you can search for processes using a single input field: The system searches the string in the following data: process forms, process subject, process id.
- User dashboard: The default page contains a dashboard and there are three new elements in the dashboard menu:
 - Worklist Overview: an overview of the worklist entries of the current users.
 - News folder: shows the contents of the "News" folder (the folder must be a subfolder of the Common folder).
 - Appointments: Calendar appointments from today and tomorrow.

2.2 Log Viewer

The log viewer is a separate program for viewing @enterprise log files. It can be downloaded from our web server.

2.3 Forms

- The icon defined in the form type detail mask is used in the worklist and document management system.
- More than one attribute can be specified for form name and sort order.
- The width and height of a form can be defined, this information is used for subform windows.
- The subform table can be customized using a *tablehandler*. The class must implement the interface `com.groiss.dms.FormTableHandler`. It is "registered" in the `tablefield` tag as attribute `"tablehandler"`.
- multiple templates per form type can be used: all documents of the selected type in the template folder can be selected as a template.

2.3.1 XHTML forms

When loading a form you can specify the path to an XHTML form instead of loading an HTML page. The behaviour of XHTML forms is different in the following respects:

- The XHTML form must be in correct XML syntax, but we make no validation against the DTD.
- The form is read from the classpath and parsed whenever it is shown. No "Replace HTML" is necessary when only the HTML code is changed. When database and class changes are wanted (for example an additional field) then the "Replace HTML" action is still necessary.
- The behaviour of the form can be adapted using an event handler implementing the interface `com.groiss.dms.XHTMLFormEventHandler`.

2.4 Style

The new client contains a link to the Style-configuration where each user can define the appearance of the client. See tab "Extras", link "Style-Configuration". The defaults are read from a property file `style.prop` in the `avw.jar` file.

Note that we have dropped some style classes we don't use anymore. If you use them in applications, add them to the `avwlocal.css` style sheet.

2.5 RMI Extension

We provide the possibility to use fixed ports for RMI. You can specify a port for

- Plain RMI,
- RMI with secure login,
- RMI via SSL

services.

The server will use those 3 ports (and the configured RMI registry port), so it is possible to open just those specific ports in the networking infrastructure to facilitate RMI communication via firewalls.

Nota bene: the notification mechanism does not use RMI at all. Therefore client notification will not work even with this extension.

3 API

Some obsolete classes, which we are not using anymore - and are not in the API - have been moved to a separate jar file, `obsolete.jar`. Include the jar file in your classpath, if you are using such classes.

3.1 HttpRequest, HttpResponse

The self-made `HttpRequest` and `HttpResponse` classes have been removed from the `@enterprise` code. They are still part of the distribution and work as usual but we removed all methods using these classes from non-obsolete `@enterprise` classes. There may be one case where application code can be affected: When you overwrite the `HTMLForm.showForm` or `update` methods. Use the `FormEventHandler` for this purpose.

3.2 com.groiss.cal

New class `com.groiss.cal.impl.GermanHolidays`. The concrete holidays can be defined in the configuration: Define the class as `Calendar` class, then restart the server and reload the configuration menu. A new link "Feiertage" will appear.

The method `getDateFormatter` returns thread-safe date formatters for different formats. Constants are defined for the different formats. `getDateFormatter` and `getCalendar` are two methods allowing to use calendar implementations different from `GregorianCalendar`. Use the configuration parameter `calendar.class` and `date.format.class` to specify their class names.

3.3 com.groiss.component

We removed `ApplicationAdapter` and added the class `ServiceAdapter`.

3.4 com.groiss.dms

The method `getName(DMSForm)` has been added to the interface `FormEventHandler` to allow defining the name of a form. The `FormEventAdapter` class is a new adapter class to the `FormEventHandler` interface. An additional pair of interface and adapter class is `XHTMLFormEventHandler` and `XHTMLFormEventAdapter`. They are used for XHTML-forms. The helper class `FormContext` defines the environment when an event handler method is called.

The `FormTableHandler` allows implementers to customize the form table. You can specify the table handler class in the `tablefield` element of an HTML-form as attribute `tablehandler`.

Method `DMS.listSubforms(DMSForm, int)` will now sort the result as configured in the administration. New Methods in DMS: `countSubforms` returns the number of subforms, `DMS.createWebLink` creates a `WebLink`, and `getTemplates` returns the templates for a form type.

3.5 com.groiss.ds

A new class `ListComparator` can be used to sort lists (for example table rows).

3.6 com.groiss.gui

New class `TabbedDialog` for showing dialog windows containing tabs.

3.7 com.groiss.log

The method `logError(Throwable error, String errHeader, int level)` allows to log an error on a given log level. It has been added to the interface `ILogger` and the implementation `Logger`.

3.8 com.groiss.org

`OrgData` has three new methods: `OrgData.setUserProperty` and `getUserProperty` for accessing the user properties and `listUsersWithRole` for finding all users having a role.

3.9 com.groiss.store

StoreUtil.setFromDB An enhanced approach to setting the members of a persistent object from the contents of a ResultSet has been implemented. The new implementation drastically decreases the overhead of this operation. This is mainly achieved by caching of database metadata in the persistence layer of @enterprise itself instead of repeated determination of the needed metadata at each access to a ResultSet.

A prerequisite of the new approach is the consequent and constant ordering of the attributes in the select list resp. the ResultSet. Therefore, usage of the asterisk ("*") as a shorthand way to denote all columns of a table or query is strongly discouraged. So, do not use "select * ..." or "select t.* ..." in your queries.

This affects only queries where the select list is not constructed by @enterprise itself. There is one single call where this takes place, that is the list2() method in the SQLObject and the Store classes. All other queries should continue to work unaffected by the enhanced implementation. When calling list2(), use one of the variants of StoreUtil.dbAttrs() for the consistent order of the select list.

Note that setFromDB() has been deprecated in class StoreUtil and is now final in class SQLObject. The new call is StoreUtil.setFromRS(). The old semantics and implementation of setFromDB() is now realized by a setFromRSByName() call.

The behaviour of setFromDB() has been slightly changed to ease the transition for peculiar queries (e.g. queries with column aliases; union constructs, order by clauses with positional column notation, ...). If the configuration property database.setfromdb.compatibility is set to "1", setFromDB() and list2() call use the old implementation (call setFromRSByName()); otherwise, setFromDB() uses the new implementation by calling setFromRS().

Lock The class Lock provides a database lock.

Store The new method getNoCache reads and returns an object from the database bypassing the thread cache. It can be used to read or restore the original object - without the uncommitted modifications of the current thread.

Additional variants of the methods getTable and getValue have been added allowing to set return value types.

3.10 com.groiss.util

MailSender: Attachments can be files or MimeBodyParts.

3.11 com.groiss.wf

The new class SystemAction provides some methods useful for preprocessing and postconditions. Method WfEngine.goBackToLastInteractive is new.

3.12 com.groiss.wf.batch

BatchJobs will be compensated when going back. The interface BatchAdapter has the new method doCompensate(BatchJob) for perform an action.

4 Support

If you have problems with this version contact us under the email support@groiss.com.