@enterprise 8.0 Release Notes

Friday 15th December, 2017

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 8.0.22989

1 Introduction

This fresh document lists the changes and new features in release 8.0 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 setup80.jar. An installed JDK of at least version 1.6.0 is required. The JRE must be at least version 1.5.0.

Hint: The JDK 1.6 is required for compiling @enterprise applications.

Alternatively the installation can be initiated from the command line via

java -jar setup80.jar

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

Caveat: The current versions of *MySQL* and database-drivers are buggy. The first analysis shows us that some combinations seems to be stable (e.g. database-version 5.0.15 and database-driver 5.0.3). For this reason *MySQL* is supported experimentally only!

1.2 Upgrade

Further information about the upgrade procedure can be found in the installation- and configuration-guide.

Hint: Please note that the file structure of @enterprise 8.0 has been reorganized (see section 2.2.9).

2 New Features

2.1 Technology

2.1.1 XForms

XForms is a standard defined by the W3C consortium for the definition of web forms. The principle behind them is described as follows: "Traditional HTML Web forms don't separate the purpose from the presentation of a form. XForms, in contrast, are comprised of separate sections that describe what the form does, and how the form looks. This allows for flexible presentation options, including classic XHTML forms, to be attached to an XML form definition." (cited from http://www.w3.org/MarkUp/Forms/).
In @enterprise, XForms can be used as an alternative to HTML forms. The advantages of XForms make this technology an excellent choice for all further web form implementations.

2.1.2 Web Services

The Web services framework AXIS2 has been integrated. Authorization based on WS-Security is integrated within the @enterprise authorization module.
For additional information about Web Service Integration in @enterprise please refer to chapter 'Web Services' in the 'Application Development Guide'.

2.1.3 Worklist Cache

The worklist cache now contains the following additional information:

- process instance objects,

- origin of activity instances: allows the distinction between a users own items and items received via substitution.

2.1.4 ACL Cache

The ACL Cache has been improved: It is no longer necessary to refresh the cache manually when changes in the permission structure occurred. Since version 8.0 the ACL Cache is active by default. Detailed information about configuring the ACL Cache can be found in chapter 'ACLCache' of the 'Installation and Configuration' manual.

2.2 Application Development

2.2.1 Process Definition

- It is possible to use a Java method for defining task agents (see chapter 'Properties of an Activity' of the 'System Administration' manual).

- Process instance ids and process subjects can be defined via regular expressions. More information can be found in chapter 'Process Properties' - point 'Common' - of handbook 'System Administration'.

- A name can be specified for process forms. This name is shown in the user interface, the id is still used in the process definition and the API (see chapter 'Process Properties' of handbook 'System Administration'). Note: when changing such a name within a process you may need to refresh the worklist cache if there exists running instances of that process. Otherwise the names shown in the worklists may remain the old ones.

- At the end-nodes of parallelism (andjoin, orjoin and end-node of a parallel for) a method can be defined which will be called by each parallel path reaching the end (see chapter 'Control Structures', sections 'Parallel execution' and 'Parallel For' of handbook 'System Administration').

- The id of a form can now be changed using the process editor.

- When alternative icons for a task or subprocess could not be loaded, the default icon will be displayed grey and crossed out to signal that kind of problem.

2.2.2 Script-Integration

Groovy scripts can be used alternatively to Java methods in various places: preprocessing, postcondition, task-functions, etc. In the administration interface multi-line input fields are provided where scripts can be defined.

More about the usage of groovy in @enterprise can be found in chapter 'Application Methods Called by the Engine' of the 'Application Development Guide'.

2.2.3 Process Definition Report and Component Editor

A comprehensive report for process definitions is available in the toolbar of process definitions table. It shows an HTML Page containing all aspects of a process definition, including the script, the graphical representation, the used components (tasks, roles, forms, escalations) and form visibilities.

In the process detail view, the new tab "Components" allows to directly edit the components of the process.

2.2.4 Graphical Definition of user interface descriptions

The XML-files which specify a user interface can be maintained via an editor (via 'Administration - *Application* (e.g. default) - GUI Configuration'). The definition of DefaultURLs from previous versions of @enterprise is obsolete.

2.2.5 Definition of toolbar actions

Additional toolbar actions can be defined in a modular way. There is no more need to include such definitions in the admin.xml. A reference to a toolbar actions takes the form of the name of its enclosing file and the name of the function separated by a period (i.e. <myfile>.<myfunction>). An example can be found in the DMS related section in the 'Application Development Guide'.

2.2.6 Process and Task escalations

Additional escalation types have been introduced:

- Unseen Tasks: tasks (IDLE or ACTIVE) not seen by anybody

- Batch Steps and Sync-Steps with status IDLE

2.2.7 HTML Tables

The implementation of tables has been unified. The advanced features familiar (column filter, stored filter, scrolling body) are now available for all tables.

2.2.8 HTML GUI

- @enterprise supports the browser Apple Safari and Chrome.

- The name of the toolbar-frame has been changed from *toolbar* to *toolbarframe*. This step was necessary for Apple Safari support.

  Hint: Please note that the process editor cannot be used with Safari 4 and higher!

2.2.9 File and Directory Structure

The structure of @enterprise files has been reorganized:

- The files *avw.jar*, *masks.jar*, *serverarea.jar* and *reporting.jar* are no longer available. The content of these files is combined in the new file **ep.jar**.

- The folder *sql* has been moved into ep.jar.

- The user interface XML files (e.g. standard.xml, admin.xml, etc.) have been moved into ep.jar.

- Files of folder *conf* have been moved to ep.jar, except for:

    - avw.conf
    - avwservlet.conf
    - IconMimetype.txt
    - mime.types
    - schema.ldap

- The javascript files have been reorganized. They can be found in 'serverarea/alllangs/scripts/source' of ep.jar. The obsolete files (e.g. avw.js, form.js, etc.) are in *obsolete.jar* now.

2.3 Administration

2.3.1 File Import

The file import component allows to load data from text files into form tables. No programming is needed since the structure of the import source and the target objects can be specified at runtime. Detailed information about this new feature can be found in chapter 'File Import' of the 'System Administration' manual.

2.3.2 AJAX Components and Date Picker

Object selection has been made more comfortable using AJAX driven components. A new date picker has been integrated which allows fault-tolerant input of date values.
For more information please refer to chapter 'AJAX components integrated in @enterprise' in the 'Application Development Guide'.

2.3.3 Logging

The log file rotation can be configured. A new log file is created either when the server is restarted or at midnight. The number of retained log files can also be configured. Current and previous access- and error-logs can be displayed via the administration interface.

2.3.4 Event Log

The event log mechanism can be used to store events that you want to make persistent. Startup and shutdown events are written to this log. For more information about storing events consult the @enterprise API documentation.
The event log can be accessed via 'Admin Tasks - Server - Events'.

2.3.5 Configuration Changes without Restart

Most configuration changes now don't need a system restart to become effective. In the configuration masks the parameters that still need a restart are marked with an asterisk (*).

2.3.6 @enterprise as service under Windows

A new framework is used (Java Service Wrapper from Tanukisoftware). It has the following features:

- Reliable restart,

- Configuration without the need to manipulate the registry, via the ./services/wrapper.conf file

- Thread-dumps for services are possible (sendsignal.exe in ./services)

More about the used framework and migration from the previously used mechanism can be found in chapter 'Installing a Service' in the 'Installation and Configuration' manual.

2.3.7 Data sources

Datasources can be used instead of direct JDBC-Connections. Only the datasource id is specified in the @enterprise configuration, the datasource itself is defined in the execution environment (application server,...).
Detailed information about datasources in @enterprise can be found in chapter '@enterprise and Datasources' of the 'Installation and Configuration' manual.

2.3.8 Miscellaneous

- User ids may contain special characters. In particular, Email addresses can be used as user ids.

- Forceful log out of users can be accomplished by killing of user-sessions (works cluster-wide).

- Bootstrap mechanism is used, which builds the classpath automatically (see chapter 'Bootstrap in stand-alone server (Jetty)' in the 'Installation and Configuration' manual).

2.4 User Interface

2.4.1 Multiple and Shared Dashboards

Dashboards now have a name, so several dashboards can be maintained and used - Dashboard sharing is supported - an administrator can define dashboards for a group of users.

2.4.2 Shared Worklist Filters

Worklist filters can be shared with other agents. The @enterprise permission system is used.

2.4.3 Usage of read-only fields instead of disabled

The "readonly" HTML property is used for textfield- and textarea-formfields in mode "readonly". Previously the HTML property "disabled" has been used.

2.4.4 Change Agent from Role Worklist

The agent of a task can be changed without the need to take the task in before.

2.4.5 Mobile Device (PDA) support

A special version of the user interface for mobile devices has been implemented. It is tailored for devices with small screens like PDAs or smart-phones and does not use frames, popup windows or excessive JavaScript.
Detailed information about the PDA GUI can be found in chapter 'The Mobile Client' of 'User Manual'.

2.5 DMS

- Column picker and enhanced sorting/filtering capabilities have been added to document tables.

- In addition to the columns the available actions and a Black-/Whitelist can be configured for document folders in the following three ways:

  1. a default for all folder types in the GUI Configuration XML
  2. for each folder type via tab 'Folder Settings' in its detail view
  3. if a user has the permission to edit a folder, he can change its columns via button 'Settings' in the folder property dialog). To change the actions and the Black-/Whitelist he additionally needs the 'administration' right.

  The most specialized configuration will be taken at runtime.

- DMS function 'new' now shows only those tabs in which the current user is allowed to create objects represented by that tab. E.g. if the user is not allowed to create web links the corresponding tab will not be shown to him.

- Regular expressions can be used to define the names of forms via 'Name Attributes' field of their formtypes. A detailed description of the used regular expressions can be found in section 'Tab: General' of chapter 'Forms' of 'System Administration' manual.

- If a WebDAV capable browser (IE7, ...) is available, documents can be opened in read/write-mode. The necessary steps are described in the 'Installation and Configuration' manual.

- A document download power function is provided. A set of documents can be selected and downloaded packed as a zip archive.

2.6 Reporting

The search and reporting component has been enhanced in several ways:

- direct creation of PDF reports,

- extended library of predefined reports,

- drill down functionality for linking reports,

- allow user-defined toolbar functions on the search result,

- aggregation capabilities to compute sums or averages of column values of multiple lines.

2.7 API

- The biggest change in the API is the use of Java Generics, as introduced in Java 5. Several methods that returned lists of objects now precisely define the return type by specifying the type of the objects contained in the list, for example:

    List<ProcessDefinition> listProcessDefinitions(Application appl)

    The Store.get and list methods have also been redefined so that no casts are needed when assigning the results to the target. Methods that returned Vector now return Lists.

- The handling of the MultipartRequest has been changed. It is no longer necessary to create the MultipartRequest before accessing a parameter (using getParameter). The method *getParameterMap()* returns Map<String,String[]>, according to ServletRequest standard.

2.7.1 Configuration

The Configuration object returns typed values: In an XML specification the types and default values of the configuration parameters are defined. Parameters can be accessed with the methods getProperty(), getBoolean(), getInt(), etc.
The configuration parameters of applications can be defined in the Administration interface (tab 'properties' of the application object).

Hint: Be note that properties of type Boolean must be read with getBoolean() to ensure that checks are done correctly! Checks like "1".equals(Configuration.get().getProperty("bool_prop")) are not correct anymore, because the values *true* or *false* are stored in @enterprise 8.0.

2.7.2 SystemAction

The utility class com.groiss.wf.SystemAction has some additional methods useful in system steps or preprocessing.

2.7.3 Image Locations

Most @enterprise icons were located in the "alllangs" directory. They have been moved to "lang/default/". This new location allows the application developer to define his own images or language dependent images by putting them in "lang/en" or a similar language specific directory. In the image paths of existing applications the "alllangs" component must be removed.

2.7.4 Outputting HTML Text

In previous versions of @enterprise HTML text in table cells has been sent to the browser unescaped. For example the string "text" has been shown as a link. This has the drawback that texts containing the "<" or ">" characters have not been displayed correctly because the browser interprets the text as HTML. In version 8.0 every string is shown as it is, special characters are escaped ensuring that the browser shows them correctly. If you want to put a HTML element into a table cell in one of our table adapter APIs, you have to use either the JDOM Elements for defining the XML or our HTML components from the com.groiss.gui.component package, for example Link.

2.7.5 CalUtil

The implementation of date parsing and formatting now uses the ICU (http://icu-project.org) library instead of the Java built-in classes. This library provides better support for non western languages and calendars.

2.7.6 Exception Handling

@enterprise generally uses RuntimeExceptions. It defines the RuntimeException com.groiss.util.ApplicationException that is used for all kinds of exceptions in the system. Application code normally can not and should not deal with exceptions @enterprise throws. They are handled by the outermost caller, normally the Dispatcher servlet. This servlet presents the exception in the user interface.

2.7.7 RightCheck

Interface RightCheck is more flexible. The may-methods (i.e. mayUpdate()) of previous versions have been replaced by the more generic single method hasRight() which is passed the concerned right as a parameter.
To help the implementors of this new interface the new utility class DefaultRightCheck has been introduced.
Additional information can be found in the 'Application Development Guide'.

2.7.8 DMS Table Modification

Due to the changes in the table handling of our HTML tables the DMSObjectTable class had to be changed in a way which is incompatible with former subclasses (i.e. they will not compile anymore). In former @enterprise versions this class was part of our API and subclasses could be made for changing some properties of the default implementation.
Since @enterprise 7.0 the interface DMSTableHandler has been introduced to allow such modifications without the need to create subclasses. In @enterprise 8.0 class DMSObjectTable has been removed from the API and therefore DMSTableHandler is now the only supported way for modifications.
Interface DMSTableHandler has also changed mainly because of typing the parameters and return types of its methods.
For more information see the API documentation and the corresponding section in the 'Application Development Guide'.

3 Support

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