@ENTERPRISE

@enterprise 11.0

Release Notes

March 2025

FREQUENTIS AG

FREQUENTIS AG

Innovationsstraße 1
1100 Wien
Austria

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

Document Version 11.0.39523

Contents

1 Introduction

In 2018, version 10 of *@enterprise* has been released and since then a lot of changes and enhancements have been delivered. However, some changes are not feasible within the same major version of the software, namely:

- changes in the user interface,

- in the API,

- changing the directory structure,

- replacing the included libraries,

- and the required Java version.

In version 11 we made all these changes and moved on to a state-of-the-art technological base and deliver a modern user interface.

Klagenfurt, April 2023

2 Architecture

2.1 System requirements

@*enterprise* 11 needs at least Java version 17.
The current versions of the following browsers are supported: Chrome, Edge, Firefox, and Safari.

2.2 Libraries

Almost all libraries have been replaced with the current version, see the file `Copyright` or contents of the lib directory for details.

2.2.1 Axis2 and Rampart Support

We also updated the Axis2 library to the newest available version. However, it should be noted that there is currently no Rampart support available for this version. This may affect users who rely on Rampart for security purposes. The corresponding ticket for this issue is: RAMPART-449. Please refer to this ticket for further information. We're going to re-integrate Rampart as soon as a compatible version is available.

2.3 JavaEE vs. JakartaEE

Up to and including @*enterprise* version 10.0, the basic underlying platform specifications for @*enterprise* were "JavaEE" which was owned and driven by Oracle.
Oracle decided to lay the future responsibility for the development and evolution of the specifications into the hands community (the Elipse Foundation). The new platform branding name is "JakartaEE". While this step for a community driven effort can be viewed in a generally positive light, there remains a main complication. Oracle chose to disallow the use of the `javax.*` namespace of JavaEE.
The community had to come up with a new namespace and agreed to use `jakarta.*` for JakartaEE. (for details: please see https://jakarta.ee).
Since this is a potential breaking change for all components and applications targeting the platform, the need for a migration strategy from javax to jakarta arises.
Efforts for such a migration are under way in the community and industry, but are not yet uniformly available. In the light of the situation, we decided:

- up to @*enterprise* 10, nothing will change, there will not be any support for JakartaEE in the future.

- for @*enterprise* 11, we still use the old namespace (the JavaEE8 platform), committing us to the following versions of Servlet Containers: Jetty 10.X, Tomcat 9.X, Weblogic 14.1

At the moment, we do not have any plans to provide an JakartaEE variant of @*enterprise* 11, but are committed to target this platform for future releases of @*enterprise*.
If there is the pressing need to target the JakartaEE platform / Jakarta namespace, please do contact us.

2.4 Directory structure

The directory structure of @*enterprise* and application files has been changed. We distinguish three areas:
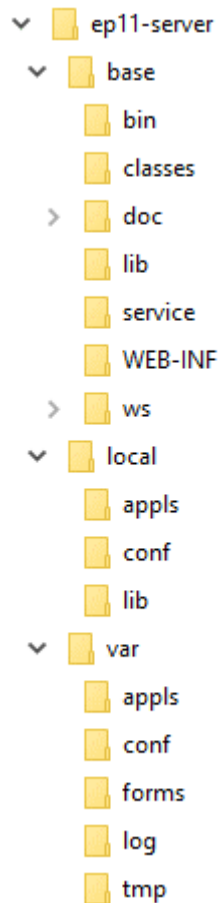
Figure 1: **Directory structure**

- BASE: program elements, that are not changed by the customer or in the local installation

- LOCAL: project specific files, that are not changed locally

- VAR: variable files: application configurations, log files, etc.

The startup scripts `epstart.bat` and `epstart.sh` are in the BASE area, they should not be changed, but their behavior can be customized:

- using environment variables: `EP_CONFFILES` or java system property `ep.conffiles` define the path(s) of the configuration file(s), `EP_BASE`, `EP_LOCAL`, and `EP_VAR` the above paths. The default name of the configuration file has changed from `avw.conf` respectively `avwservlet.conf` to `ep.conf`.

- with Java argument files `LOCAL/localjavaargs` and `VAR/varjavaargs` for defining heap size, default character set, and other Java environment properties.

Locations of configuration files of @enterprise are neither passed via the java command nor via context parameters in `web.xml` anymore. Configuration files packaged in the `WEB-INF/classes` directory of the `ep110.war` file are ignored.

See the installation manual for further details.

2.5 File names

Several files names changed:

- We renamed the start scripts `ep.bat` and `ep.sh` to `epstart.bat` and `epstart.sh`.

- We renamed the script for Windows service installation to `services\epservice.bat`.

- *@enterprise* artifacts now contain the full version information in their names, e.g. `ep-setup-11.0.23456.jar`, `ep-api-11.0.23456.jar`. The name pattern for the war file did not change, because the name is used as the default for the context root in the application servers.

- Prepended "ep-" to names of some deliverables:

```
bin/ep-adminshell-<rev>.jar
lib/ep-bootstrap.jar
lib/ep-docu-de-<rev>.jar
lib/ep-docu-en-<rev>.jar
lib/ep-icomoon-repackaged-*.jar
lib/ep-dojo-repackaged-*.jar
lib/ep-dojo-repackaged-*-debug.jar
lib/ep-wfxml2svcs-*.jar
```

@enterprise documentation files `ep-docu-en*.jar` and `ep-docu-de*.jar` are part of initial setup now.

2.6 Windows Service

@enterprise 11 uses a new windows service framework, which starts the `epstart.bat` file. Modifying such services is much more flexible than in the previous implementation.

2.7 Web.xml

The configuration file for web components `WEB-INF/web.xml` is a minimal one. The web-components of *@enterprise* are registered dynamically at startup. Existing `web.xml` files from @ep10.0 will not work. The components can be configured via a new section "Web Components" in the Configuration.

2.8 Upgrade

The files `upgrade.bat/upgrade.sh` have been removed, call `epstart -upgrade` instead. We don't supply patch files anymore, the upgrade uses the `ep-setup-11.0.*.jar` and replaces all files in the BASE directory. The automatic upgrade support for *@enterprise* versions less than @ep8.0 has been removed.

For more details, see the installation manual - especially section Performing an Upgrade of @enterprise prior to 11.0.

2.9 Docker support

Improved support for running *@enterprise* in a containerized environment (e.g. Docker). For more details see section Containerized *@enterprise* (Docker) in the installation manual.

2.10 REST API

@enterprise provides now a RESTful API for workflow related operations. For more details please refer to the Application Development Manual.

2.11 Security

- The return type of servlet methods is now checked: methods with signature
  `m(HttpServletRequest, HttpServletResponse)` must return `void`, method with signature
  `m(HttpServletRequest)` must return `com.groiss.gui.Page`.

- All servlet methods have been checked for script-injection.

- Every access to an object in the database via a servlet method is now protected by an permission check. In previous versions it was possible to access the list of users or organizations without any right check. The role *all* has now the necessary additional permissions. If you users don't have this role, additional permissions may be necessary.

- We added the `@Admin` annotation to all servlet methods, which are used only in the Administration.

- XML parsing of resources loaded from unsafe sources (not local file system) is not allowed to access external resources.

- The parameter "Check Referer header" in group Security is now true by default. The new annotation `com.groiss.servlet.EntryPoint` can be used to specify servlet methods which are exempted from the Referer check.

  All internal methods are appropriately annotated. The new default value for the corresponding property `ep.check.http.referer.exempt` in section *Security* of the configuration is empty.

- The session cookie is now always set to `httpOnly` and set to `secure` for HTTPS communication.

3 User Interface

Only gentle changes have been done in the end user interface. The main goal was to make the user interface easier to understand for novices: reducing the number of buttons and toolbars.
Hand in hand with this effort we made the interface more tidy: fewer lines, borders, colors and more padding between the elements.
Third, the appearance became "softer" by reducing usage of bold text, reducing contrast and using round edges.
The main changes are:

- The main toolbar shows only the first (most important) four elements, the others are accessible via the extension button "...".
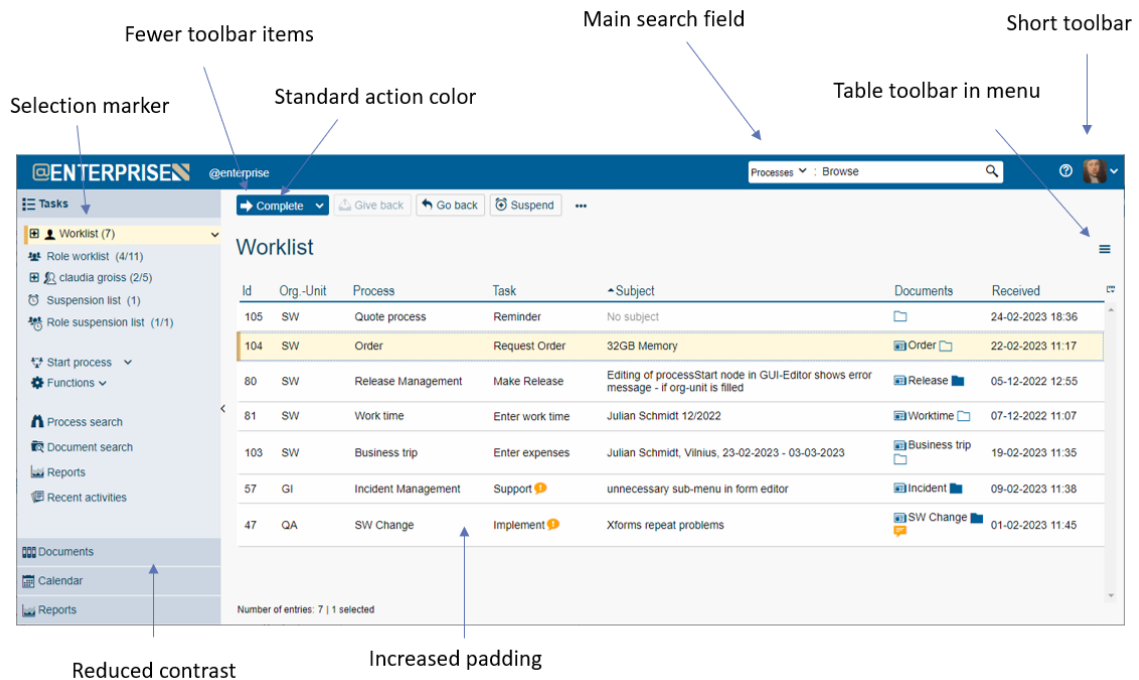
Figure 2: **User interface**

- The extra toolbar for tables (worklist, document list) has been removed and replaced by a burger menu containing these functions.

- In the header, a new central search field allows searching in the current table, searching process instances, and documents.

- Logout, Select GUI configuration, and Administration buttons are now in the user dropdown.

- the collapsed (narrow) navigation is default for the standard configuration (`standard.xml`).

- different appearance of tab headers: no borders, no icons, counter for documents, nodes and mails are shown in tab header.

- In the form toolbar the button "Save and Complete" has been removed. You can change the default behavior using the configuration parameter `avw.form.toolbar`.

- New page "About" introduced in user profile containing version infos and server name.

- Password Policy is part of the dialog "Change password" now.

- Date-columns in reporting are not right aligned anymore, to be consistent with .worklist

- User avatars are shown instead of or additional to user names on several places in the user interface.

- The login page now contains a - customizeable - background image.

4 Process Modeling

- Additional keywords for the context object in "sync", "registerForEvent" and "raiseEvent" process nodes can be used.

  - `parent`: the immediate parent of the current activity instance (can be a process instance or a parfor node)
  - `processInstance`: the innermost (sub-)process instance reachable from the current activity instance
  - `mainProcess`: the top level process instance for the current activity instance

  The traditional keyword `process` is deprecated. It has the same (unchanged) semantics as the new "parent" keyword, but the naming gave the wrong impression that it always would be a process instance.

  `IEventHandler.onRegister()` is called without any parameter. The current process instance can be obtained via `WfEngine.getInstance().getContext().getProcessInstance()`.

- XForms: additional binding attribute `as-text` introduced. Similar to readonly, but value is shown as text, not in a readonly input/select/textarea.

- XForms data binding: The following data types are allowed for the different controls:

  1. `range`: numeric
  2. `input, secret`: all except `xsd:base64Binary, xsd:hexBinary`
  3. `textarea`: `string, longString`

- XForms: `appearance` attribute for `select`:

  - `full`: checkboxes,
  - `compact` and `minimal`: select list with size: min(number-of-values, n), where n is 4 for minimal and 10 for compact.

  `appearance` attribute for `select1`:

  - `full`: radio buttons
  - `compact`: select list, size: min(number-of-values, 10)
  - `minimal`: dropdown list

  The size attribute can also be specified explicitly.

- New property `allowModifications` in GUI-configuration XML for types `query` and `table`. If true, insert/update/delete using `JSONLoader` is allowed.

- Avoid loading `import.xml` definitions from file system, only classpath-resources are supported.

- CSS class `scFlat` has been removed from @*enterprise*.

- `.less` is now the default extension for theme files, `.css` is still possible.

5 Administration

- In the administration there is a new tree entry "All applications" for viewing the elements (processes, forms, timers, etc.) of all applications.

- `ep.forms.generate.jar` parameter (group Other parameters) has three values: "generate" the `appl_forms.jar`, "ignore error" and "don't generate".

- Role "Owner" has permission "Share".

- @ep standard message templates changed from HTML to XHTML and unified in styling.

- The ids of message templates must now be unique.

- For object classes and form types at most one permission list may now be assigned to in tab "Default permissions". Assigning multiple permission lists discriminated by organizational units is not provided anymore as it was not interpreted by the system.

- Changed default JDBC URL for SQLServer to account for newer versions of the driver which (try to) enable encryption by default.

- Application version: after an upgrade the application version stored in the database omits the part after the underscore. In the detail mask we show the implementation version if it differs from the value in the database.

- For managing keywords, object specific permission "Create/View/Edit/Delete objects" are needed. Permission 'Edit Configuration' is still needed to access the keyword management at all.

- Application is no longer a mandatory field for reports. The right "execute" for an application is needed for adding reports to an application and editing reports belonging to an application.

- Bundled HTTP URLs related to cluster operations in class `com.groiss.cluster.ClusterSupport` according to the following mapping:

  `com.dec.avw.html.HTMLNodes.redirect` → `redirect`
  `com.dec.avw.html.HTMLNodes.unbind` → `unbindAndRedirect`
  `com.dec.avw.html.HTMLNodes.redirectRMIClient` → `getRedirectAddresses`
  `com.dec.avw.html.HTMLNodes.getLoginNodes` → `getAllRedirectAddresses`

  `com.groiss.cluster.ClusterInfo.getNodeStatusHAProxy` → `getNodeState`

  `com.groiss.httpd.jetty10.EPHazelcastInfo.getState` → `getHazelcastState`
  `com.groiss.httpd.jetty10.EPHazelcastInfo.getState` → `getHazelcastSessionInfo`

  `com.dec.avw.html.HTMLNodes.clientInfo` → `getClientInfo`

  Please note that the old URLs will not work, change references to them accordingly in your installations.

5.1 Renamed parameters

- `avw.dms.standard_tablemodel` → `ep.dms.tablehandler`

- `javaclient.itemlimit.Referencefolder` → `ep.referencefolder.maxelements`

- `javaclient.no_notify.folder_content_change` → `ep.referencefolder.nocontentnotification`

- `httpd.jetty.cookie.secure` → `http.cookie.forceSecure` if false: the secure attribute of the session cookie will be set when created over a secure channel (HTTPS); if true: the secure attribute of the session cookie will be set regardless of HTTP or HTTPS (e.g. behind an SSL-terminating proxy). This works in embedded Jetty as well as in ServletContainers. A migration from the old "httpd.jetty.cookie.secure" property is provided with the following mapping (old → new):
  false → false
  true → false
  force → true

Migration is done automatically, i.e. if present the configuration file will contain the old and the new parameter with the value held by the old one.

5.2 Application directory for the default application

In *@enterprise* 11.0 we set the application directory for the default application to `appls/default`. This directory will be used as the root directory for storing resources created in context of the default application:

- additional language resource files

- additional GUI configurations

- templates of form types

It is still highly recommended not to create any resources in the default application except additional language resource files as this is the only way to customize the labels and messages of the basic *@enterprise* system. Additional forms or GUI configuration should always be defined in context of a custom application.

When upgrading from an older version to *@enterprise* 11.0 the application directory will only be set if no such directory is already stored in the application object representing the default application.

If the default directory can be set, we also create that directory automatically on the file system. Already existing custom resources will not be moved to the new location but remain in their current directories. The server will still be able to find them and they can still be changed using the Administration UI. New resources will be stored in the new location - with one exception: if there exists already custom language resource bundles in the old location newly created files will be stored there as well.

To avoid confusion about from where a resource is loaded we recommend to move existing resources to the new application directory, e.g.:

- `classes/com/dec/avw/resource/String_en_US.properties` to
  `local/appls/default/classes/com/dec/avw/resoure/String_en_US.properties`

- `classes/mygui.xml` to
  `local/appls/default/classes/mygui.xml`

- `forms/default/forms/myform.xhtml` to
  `local/appls/default/classes/default/forms/myform.xhtml`

The corresponding database objects (i.e. GUI configuration and form types) need not to be changed.

6 API

6.1 General

- Private constructors have been added to the following classes:

  ```
  com.groiss.event.EventManager
  com.groiss.office.DocumentManager
  com.groiss.cal.CalRegistry
  com.groiss.component.ApplicationContext
  com.groiss.component.BeanManager
  com.groiss.component.DefaultResource
  com.groiss.gui.table.ExtraLine
  com.groiss.gui.table.TableAdapterUtil
  com.groiss.util.Assert
  com.groiss.util.Base64
  com.groiss.wf.html.PublicHTMLUtils
  com.groiss.org.AuthUtil
  com.groiss.reporting.QueryEngine
  ```

- Added `getApplication()` methods to the interfaces `com.groiss.timer.TimerEntry`, `com.groiss.mail.MailBox` and `com.groiss.messaging.MessageTemplate`.

- Added factory methods to the @ep standard interfaces (`Store`, `DMS`, `OrgData`, `WfEngine`, `Admin`) to replace `ServiceLocator.getX()`. E.g. use `Store.getInstance()` instead of `ServiceLocator.getStore()`.

- Introduced `com.groiss.wf.ApplicationAdapter.of(Application)` to get the registrered instance. Also streamlined the inheritance-hierarchy of `ApplicationAdapter/DefaultApplicationAdapter`.

- The method `com.groiss.org.OrgData.getReferencedObjects` has been moved to the `com.groiss.server.Admin` interface.

- Using non API classes: The visibility of some fields in non-API classes have been reduced. If such fields are used directly in your application code, replace them with API methods.

- Persistent attributes of Java classes representing system defined forms do now have Java modifier `private`. All intermediate form related classes now use modifier 'protected' for their persistent attributes.

6.2 Utilities and data structures

- Class `com.groiss.util.ThreadContext`:

    - New method `switchAttribute()`, which can be used in a try-with-resources block. Upon leaving such a block, the attribute is automatically set back to the old value.
    - `getThreadResponse()` to access the `HttpServletResponse` object in servlet-context. Previously the object could be obtained using `ThreadContext.getAttribute("response")` - this attribute is no longer available.
    - introduce generic signature to `getAttribute(Object key)`

- Introduced `com.groiss.util.SilentCloseable` interface, that can be used in try-with-resource statements and avoids the need to catch any Exception that may be thrown when closing the resource. Currently the interface is returned by `ThreadContext.switchXX()` and `PersistentAspect.add()/.remove()`.

    Action required: The return-type of `PersistentAspect.add/remove` got changed to this new interface. You may have to adapt the type of local variable you store the response into. `AutoCloseablePersistentAspect` got removed, `SilentCloseable` is a drop-in-replacement.

- New method `com.groiss.util.ThreadUtil.sleepSilently()` which catches and ignores `InterruptedException`.

- `com.groiss.servlet.Public` is deprecated, annotate your package, class or method with `@com.groiss.servlet.Access(Access.mode.Public)` instead.

- New method `com.groiss.util.Assert.nonNull(var, varName)` checks if a variable is not null.

- Class `com.groiss.util.Settings` contains two additional methods: `getLocalDir` and `getVarDir`.

- `com.groiss.util.FileUtil` provides new methods `delete(File)`, `getFileFromLocalPath(String)`, `getFileFromVarPath(String)`.

- `com.groiss.util.StringUtil.getCharset()` is deprecated. Use `StandardCharsets.UTF_8` or `StandardCharsets.UTF_8.getName()` instead.

- `com.groiss.component.ServiceAdapter` is deprecated, use `Service` interface.

- `com.groiss.component.Configuration.getBoolean` now returns a `boolean` instead of a `java.lang.Boolean`.

- `com.groiss.util.ApplicationException`: added additional constructors, removed the method `getCallingClass()` without replacement.

- New constructor added to `com.groiss.http.Parameter` to enable masking the value of parameters when they are logged (e.g. when used for sending a password).

- Logging for `com.groiss.http.Client changed`: Used client configuration and response content will only be logged at log level TRACE. Post parameters and payload of the request will only be logged completely in case of TRACE but will be logged truncated at level DEBUG.

- Unused method `com.groiss.smartclient.JsonData.supportsPaging()` has been removed.

- Deprecated mail handling methods in `com.groiss.mail.HTMLMail`, `com.groiss.util.MailSender` and `com.groiss.smartclient.mail.MailActions` removed, use `com.groiss.messaging.MessageTemplate`.

- Annotation `com.groiss.util.SuppressFBWarnings` has been removed without replacement.

- Removed `com.groiss.org.SSLAuth` class. `com.groiss.demo.ClientCertDemoAuth` does not redirect to non-SSL.

- The method `Map<String, String> com.groiss.ws.WebserviceOperation.invoke(Map<String, ? extends Object>)` directly returns the mapped return-values instead of the `OMElements`.

6.3 Persistence

- Additional types supported for fields of Persistents: char, Character and StringBuilder.

- Changed the deprecated method `com.groiss.store.StoreUtil.getTableDefinition()` to `getColumnDefinitions()`. The old method did not work reliably. The new one provides the column information reliably and allows to get information also on transient fields.

- Removed `com.groiss.store.StoreUtil.toNativeScript`, `com.dec.gi.sql.DBUtil.toScript` and `DBUtil.toNativeScript`. Use `StoreUtil.toNative()` instead. Replacement patterns are always written in uppercase and enclosed within %.

- Removed ancient store functionalities:

  - archive schema: Methods `getSchema()` and `setSchema()` removed from `com.groiss.store.Store` interface
  - different stores: Method `setStore()` removed from classes `Persistent`, `XHTMLForm`, `ObjectTable`. Method `getStore()` in `Persistent` always returns the system store.
  - internal alternative database connection (class `DBConnBean`) removed.

6.4 Organizational data

- `com.groiss.org.OrgData` interface:

  - list and get methods that simply call the corresponding methods in `Store` have been deprecated.
  - `addRule(Class<? extends PermissionMapping> clazz)` should be used instead of `addRule(PermissionMapping)` since the passed object will in fact not be re-used.
  - relaxed signature in methods `hasRightForList` and `listWithRightCheck` do not require casts to `List<Persistent>` anymore. Please note that the elements of the list nevertheless are expected to implement `Persistent`.
  - `getRoles`: new return type `Map<Role, Set<OrgUnit>>`

- New interface `com.groiss.org.HasOwner` introduced. More details about this interface can be found in the its API documentation.

- Use `com.groiss.accesscontrol.PermissionMapping.parentRewrite(PermissionQuery)` in favor of `com.groiss.accesscontrol.PermissionMapping.getParentMapping()` as a save API that also works on heterogeneous lists.

- Fields in `com.groiss.accesscontrol.PermissionQuery` are deprecated, use the new getter methods. Fields in `DNF` are protected.

6.5 Workflow engine und DMS

- `com.groiss.wf.WfEngine.getWorkList` has been renamed to `WfEngine.getWorklist`.

- `com.groiss.wf.html.CombinedWorklist` is now configured in the GUI-configuration XML, see API documentation.

- Removed `com.groiss.wf.batch.ExceptionBatchAdapter` interface. Its sole method `getErrorCode()` was moved to `BatchAdapter`.

- `com.groiss.dms.DMSFolder.getTableModel()` has been removed as there is no support for a DMS-related Table Model anymore.

- Deprecated method `com.groiss.dms.FormEventHandler.getChangedAt(DMSForm)` has been deleted without replacement.

- Class `com.groiss.dms.XHTMLFormEventHandler`: keys returned via handler method `getKeys()` were ignored if keys were configured at the form type. Precedence has been corrected so that the keys of the handler will be returned in that case.

6.6 User interface

- `com.groiss.wf.html.PublicHTMLUtils` has been deleted, `HTMLUtils.selectList` is public now.

- Method `com.groiss.avw.html.HTMLNodes.redirectJavaClient(HttpServletRequest, HttpServletResponse)` has been renamed to `redirectRMIClient(HttpServletRequest, HttpServletResponse)`

- `com.groiss.smartclient.ObjectSelectProvider.isCrudEnabled()` returns now `false` (was `true`). Adapt your subclasses, if necessary.

- `List<Pair<String,Object>> com.groiss.wf.html.Worklist.listFilters( List<ActivityInstance>)` changed to `List<Pair<String,String>> com.groiss.wf.html.Worklist.listFilters()`.

- `com.groiss.gui.component.TableContainer` no longer implements `TableModelListener`.

- Added generic type definition to:

  ```
  com.groiss.dms.FormTableHandler<T extends DMSForm>
  com.groiss.dms.DocumentEventHandler<T extends DMSDocForm>
  ```

Action required: you may need to specify a type-argument when implementing the interface otherwise you may get compiler errors due to generics erasure

Changed generic type definition for:

```
com.groiss.dms.XHTMLFolderFormEventHandler<T extends DMSForm & DMSFolder>
```

Action required: if you already took advantage of the generic specification, the second type argument needs to be removed. It may be replaced by e.g. `com.groiss.dms.DMSFolderForm`.

Removed (unused) generic type definition from:

```
com.groiss.reporting.export.FileReportingExporter
```

General note: the generic type definition comes in handy, when using the auto-generated form-interfaces for form-types with typed getter/setters for the form fields. in that case, using generics, already provides you the exact implementation and you don't need any casting.

- The already deprecated `com.groiss.xforms.ModelInfo` class has been removed from the API.

- Method `com.groiss.wf.ApplicationAdapter.modifyDetailLinks` has been removed. Was only effective in 8.0 compatibility GUI.

6.7 Reporting

- Removed lot of already deprecated methods from reporting API, clarified certain API signatures.

- `com.groiss.reporting.export.DownloadExporter.getDataBytes(T)` got removed, use `com.groiss.reporting.export.DownloadExporter.getDataStream(Query, ReportingTableModel)` instead

6.8 JavaScript

- Removed `dojoConfig.ep_thread_context_user` property, use `ep/config!` module and its `ep.current.user` property.

- Callback parameter for `_Action.getNestedActions()` got removed. The function now needs to return an array or a promise that resolves to an array.

- `msg-util.alert/confirm/yesNoCancel/prompt` don't allow HTML-formatted text to avoid script injection, pass `{allowHTML:true}` to disable encoding

- Removed JavaScript code from `ClientIndex.html` and introduced a `ep/bootstrap-module` handling the initialization. Action required: Adapt any custom `ClientIndex.html` files.

- Deprecated usage of `com.groiss.gui.JavascriptLoader.getScripts`. After loading `dojo.js` the functions from packages `ep.util`, `ep.xforms`, etc. are available.

7 Removed components and functions

Several outdated and/or unused components have been removed:

- Pre 9.0 user interface components

- Old mobile GUI (`standard_old_mobile.xml`). The current implementation of mobile GUI is available as GUI configuration with id `standard_mobile`.

- DB2/ZOS, Pointbase and Firebird are no longer supported.

- ChatService

- Perimeter/central server functionality

- WfXML-1 functionality

- Previous versions of @*enterprise* contained the jar file `obsolete.jarx` with some outdated @*enterprise* Java classes for compatibility. If you have used these classes and need support in upgrading please contact our support.

- The form-type `ODTTemplate` has been removed. If you have instances of this formtype, change their type to standard document before performing the upgrade via dms function "Change type".

- Task function "Mark as process form template" (Id: formtemplate) and its functionality has been removed. You may consider task function "Process templates" (Id: process_templates) as an alternative. For more details about that function please see chapter Standard functions in the System Administration Guide.

- The already deprecated Process-Interface has been removed.

- Support of ancient feature 'virtual roots' removed (configuration parameter `httpd.virtual.roots` and application property `avw.virtual.root`).

- Ancient RMI-based interface `IAVWSession` and all related classes have been removed from the code base. If you are using this interface you may consider to switch to our RMIStore/-DMS/etc classes - but those are deprecated and might be removed in future @*enterprise* versions.

- Interacting with @*enterprise* from other Java programs using `com.groiss.wf.Session` via Remote Method Invocation (RMI) has been deprecated and is planned to be removed in the next major release. The same applies for the connection of such a program to the notification mechanism as it depends on `com.groiss.wf.Session`.

- Parameters no longer used:

  - `AVWHandler.docRoot`: As a consequence the methods `checkOut` and `undoCheckOut` in `com.groiss.dms.DMSDocForm` have been removed as they relied on the removed property, additionally the constant `OUT` has been removed from that class.
  - `ep.allow.clientcondition`: set the condition in the XML configuration
  - `ep.mail.notification.text`: The email is now generated using the message-template with the id "incomingMailNotification".

- services: If you want to add your own services, add them to `ep.services.extensions`, in the user interface on the same location. In a clustered system, it is no longer needed to explicitly include the CoherenceService in the list of started services.

- `httpd.jetty.webxml.location`: In a standalone installation using embedded Jetty, the `WEB-INF/web.xml` file is resolved first against the `local` directory. If it is not found there, it is resolved against the `base` directory. N.B.: deployments within an application server always use the `WEB-INF/web.xml` file of the deployment context.

- `httpd.jetty.cookie.httponly`: session cookies are always created as httpOnly

- `avw.customernumber`

- `avw.reporting.showNoAccess`

- `avw.wlcache.oldspanningsemantics`

- `database.setfromdb.compatibility`

- `ep.allow.formupload`

- `ep.application.config.files`

- `ep.css.legacy.ep9`

- `ep.dms.useperscookie`

- `ep.save.ignore.modes`

- `ep.store.formhandler.behaviour`

- `resource.strings`

- `syspwd`

- `user.search.attr, user.select.attribNames, user.select.attribs`