



EPworX 12.0

Release Notes

12.0.42203

PDTS GmbH
Moeringgasse 20
A-1150 Vienna
Vienna Commercial Court, FN 73236 a, ATU 15125308
<https://www.frequentis.com>
Support: WFM-support@frequentis.com

All rights reserved. No part of the document may be reproduced, processed, or distributed in any form or by any means, electronic or mechanical, for any purpose, without the written permission of PDTS GmbH.

Company or product names mentioned in this document may be subject to protection under trademark law, brand law, or patent law.

Preamble

The release notes for a major software release of *EPworX* cover only the new features, changes, and improvements that are included at the time of the initial release of that major version.

Any subsequent changes made within the same major release are documented in the changelog.

Therefore, if you want to get a complete overview of all new features, changes, and improvements introduced in a specific revision of a major release, you need to consult both the release notes and the changelog up to and including that revision.

Contents

1	Introduction	5
2	Architecture	6
2.1	System requirements	6
2.1.1	Java version related notes	6
2.2	Libraries	6
2.2.1	Jakarta EE	6
2.2.2	Axis2 to CXF	7
2.2.3	CometD without WebSockets	7
2.3	Licensing	7
2.4	OIDC authentication using an IAM system	8
2.5	Synchronizing users with an IAM system	8
2.6	Upgrade	8
3	User Interface	9
3.1	React-based user interface	9
3.2	Dojo-based user interface	11
3.3	Which client is used and when	11
4	Administration	13
5	API	14
5.1	Logos	14
5.2	Icons	14
5.3	Java API	14
5.4	Configuration properties	15
5.4.1	Removed properties	15
5.4.2	Deprecated properties	15
5.4.3	Properties with new default values	16
6	Removed components and functions	17
	Appendices	18
A	Migration from Axis2 to CXF	19
A.1	Overview and Background	19
A.2	Prerequisites	19
A.3	Step-by-Step Migration	20

CONTENTS

A.3.1	Step 1: Generate the Service Interface from Your WSDL	20
A.3.2	Step 2: Implement the Service Interface	20
A.3.3	Step 3: Migrate Handlers to Interceptors	21
A.3.4	Step 4: Migrate Modules to Features	22
A.3.5	Step 5: Annotate the Service with Interceptors and Features	22
A.3.6	Step 6: Deploy the Service	23
A.4	Troubleshooting	24
B	Library updates	25
B.1	Added libraries	25
B.2	Removed libraries	27
B.3	Libraries with changed versions	29
C	Java interfaces, classes, and methods	33
C.1	New interfaces, classes, and methods	33
C.2	Changed interfaces, classes, and methods	34
C.3	Deprecated interfaces, classes, and methods	34
C.4	Removed interfaces and classes	34
C.5	Removed methods	34
D	Removed database tables and columns	52
D.1	Removed database tables	52
D.2	Removed table columns	52

1. Introduction

First things first: as of release 12.0 the product formerly known as @enterprise has been renamed to *EPworX*. The version numbering continues from the previous major version, so the first release of *EPworX* is version 12.0. *EPworX* stands for "Enterprise Process worX" and in essence describes a software solution for

- modeling,
- analysis,
- automation and
- optimization

of your business processes.

In 2023, version 11 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 12.0 we made all these changes and moved on to a state-of-the-art technological base and deliver a modern user interface.

Klagenfurt, June 2026

2. Architecture

2.1 System requirements

EPworX 12.0 needs at least Java version 25.

The current versions of the following browsers are supported: Chrome, Edge, Firefox, and Safari.

2.1.1 Java version related notes

- Starting with Java 18, UTF-8 is the default charset for standard input and output. For more information, please refer to the Java section in the [Installation and Configuration Guide](#) of *EPworX*.

2.2 Libraries

Almost all libraries have been replaced with current versions.

A complete list of all added, updated, or removed libraries can be found in [Library updates](#).

2.2.1 Jakarta EE

Up to `@enterprise` version 11.0, the basic underlying platform specifications for *EPworX* were “Java EE” 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 of the community (the Eclipse Foundation). The new platform branding name is “Jakarta EE”.

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 Java EE.

The community had to come up with a new namespace and agreed to use `jakarta.*` for Jakarta EE. (for details: please see <https://jakarta.ee>).

With *EPworX* version 12.0 we switched to the `jakarta` namespace. As a consequence, all `javax.*` imports in the code base have been replaced with `jakarta.*` imports. Additionally all custom code using `javax.*` imports has to be changed to `jakarta.*` imports as well.

2.2.2 Axis2 to CXF

As there was hardly any development on Apache Axis2, and it also brought too many outdated dependencies (e.g. no support for jakarta.* namespace), we decided to replace Apache Axis2 with Apache CXF. A detailed description of how to migrate your web services from Axis2 to CXF can be found in [Migration from Axis2 to CXF](#).

2.2.3 CometD without WebSockets

If CometD is used without WebSockets, either because the WebSocket transport is disabled in the CometD servlet initialization parameters or because the client does not support WebSockets, then the CometD requests must be excluded from the CompressionFilter, because the CompressionFilter cannot handle asynchronous requests. This can be achieved by adding `excludePathPatterns=/wf/cometd/.*` to the initialization parameters of the CompressionFilter within the Web components section of the configuration. If the CometD requests are not excluded in this scenario, then the client will not receive any notifications.

If you require compression for CometD requests in this situation and you are running *EPworX* on a different application server than Jetty, you may enable an alternative compression mechanism if offered by your application server.

Please note: If an alternative compression mechanism is used, then the CompressionFilter must be disabled within the Web components section of the configuration.

2.3 Licensing

In *EPworX* 12.0 the licensing mechanism has been changed.

Key characteristics of the new license model are:

- licenses have a purpose, e.g. test, production, development ... ¹
- licenses have an issue date and an expiry date
- licenses relate to a specific customer and installation
- licenses are cryptographically signed and can be verified by the software
- licenses are granted for a certain amount of named users (or unlimited)
- power users with administrative capabilities are licensed separately
- licenses can restrict monthly process volume (number of finished processes, tasks, batch jobs, web service calls)

An old style license will *not* work in *EPworX* 12.0 you need a new one.

Please contact epworx@pds.at to work out the details for a new license matching your purposes.

¹This has only an informational purpose and does not restrict the usage of the software.

2.4 OIDC authentication using an IAM system

EPworX now provides the possibility to use an external Identity and Access Management (IAM) system for user logins. Although only Microsoft Entra ID and Keycloak are officially supported, it should work for any IAM supporting the OpenID Connect (OIDC) standard.

For more details see section [OIDC authentication using an IAM system](#) in the [installation manual](#).

2.5 Synchronizing users with an IAM system

When using an IAM system for managing the users and their roles, it is desirable to synchronize the users and their roles from the IAM system to *EPworX* in an automated way. Unfortunately, there is no standard way to do this, as each IAM system has its own API and data model, especially for the roles and role assignments. Therefore, we cannot provide a generic solution for this problem.

But what we do provide is the possibility to implement your own user synchronization via the `com.groiss.iam.sync.UserSynchronizer` interface. This is especially useful as we call that interface if a user logs in via the IAM system and the user is not yet known in *EPworX*. In that case, the provided interface implementation gets the opportunity to fetch the required data from the IAM system and create the user and his role assignments in *EPworX* based on that information.

For more details see section [Synchronizing users with an IAM system](#) in the [installation manual](#).

2.6 Upgrade

When upgrading from version 11, the needed upgrade steps depend on the actual version you are upgrading from:

- revision 42029 or higher: the upgrade can be performed automatically, therefore please refer to the [Automatic Upgrade](#) section of the [installation manual](#).
- lower than revision 42029: a manual upgrade must be performed, which is describe in section [Manual Upgrade](#) of the [installation manual](#).

When upgrading from version 10, please follow the instructions in the [Performing an Upgrade of EPworX prior to 11.0](#) section of the [installation manual](#).

The support for a direct upgrade path from *EPworX* versions older than @ep10.0 has been removed.

3. User Interface

EPworX 12.0 comes with two end user interface implementations:

- the already known Dojo-based user interface, which is still supported and can be used as before, and
- a completely new user interface based on React.

Both clients come with a new look, but the React-based client is more modern and provides a better user experience. From the functionality perspective, the Dojo client is still more complete than the React client, but the React client will be extended during the lifetime of *EPworX* 12.0 and already provides a new dashboard that is not available in the Dojo client. Also it contains almost all functionality that is currently needed by most existing projects or when starting a new one.

It is not our main goal that at the end the React client will provide all the functionality of the Dojo client, but rather provides the functionality that is currently used by existing customers or is considered as being useful and helpful.

3.1 React-based user interface

The Dojo toolkit served us well during the last 10 and more years, but it is not very actively developed anymore, and it is hardly possible to find developers with experience in Dojo - for us as a vendor as well as for our customers and partners. Therefore, we decided to implement a new user interface based on a state-of-the-art, well-known and widely spread framework - and we opted for React.

The result is not a React-based clone of the Dojo client, it is a modern, responsive and user-friendly interface that preserves the proven features and concepts, but also introduces new ones and improves those being a bit outdated.

The first striking difference is the home page, which is now a dashboard providing the user an overview of his work.

But also the rest of the client has been reworked and modernized.

The main changes are:

- a new color scheme, which is more modern and fresher

3. User Interface

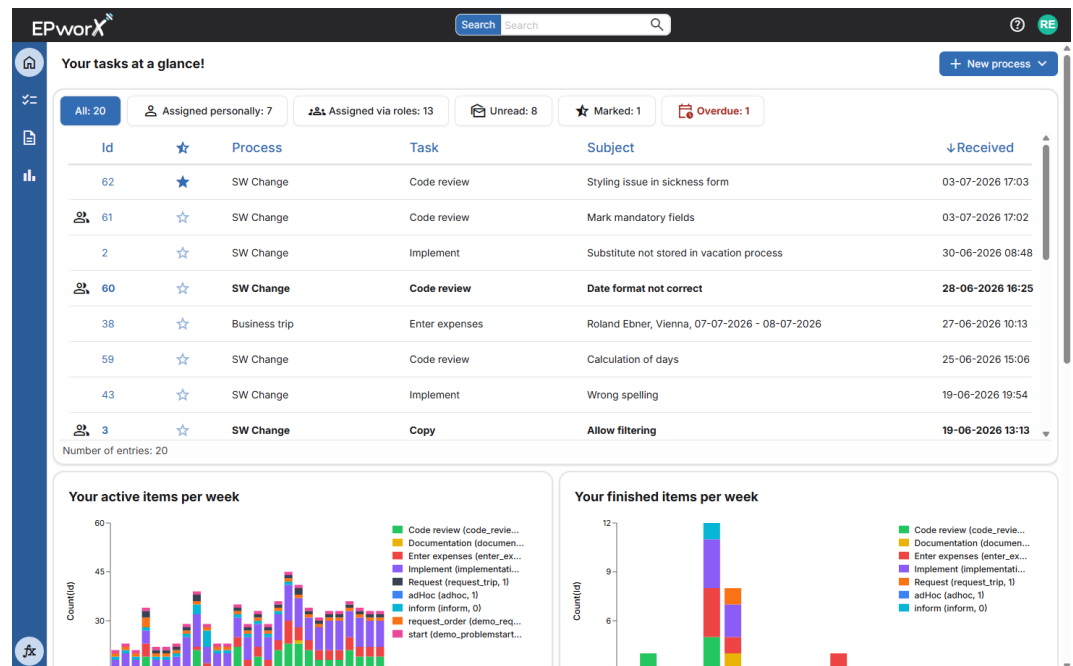


Figure 3.1: React-based user interface - home page with dashboard

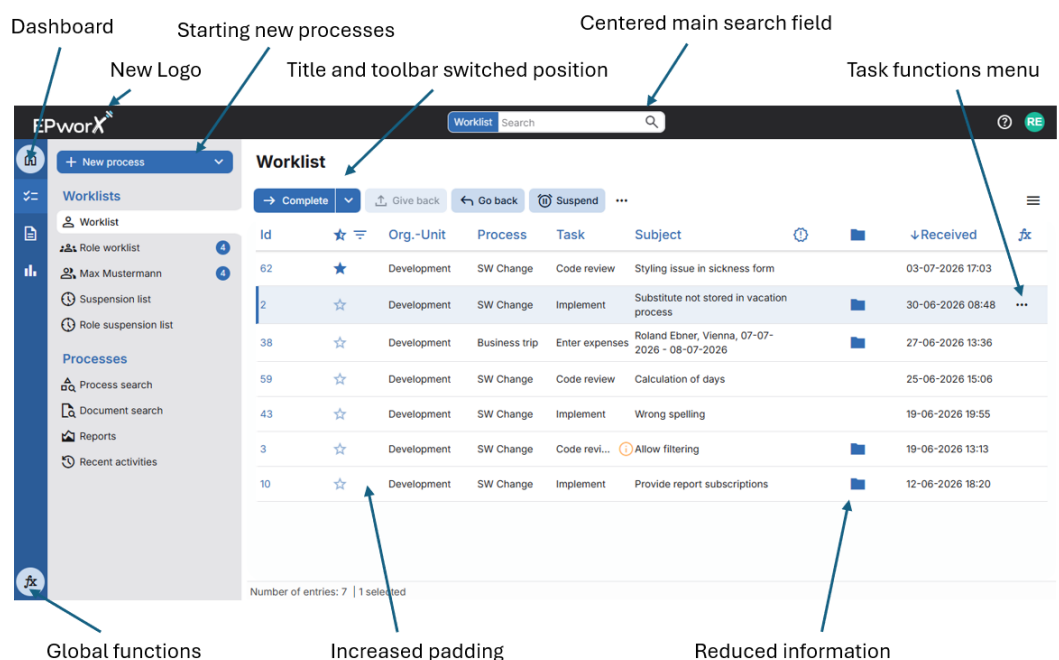


Figure 3.2: React-based user interface - worklist

- a new logo, representing the new product name
- a home page with a dashboard providing the user an overview of his work

3. User Interface

- starting a new process is now the top level element in the navigation bar
- title and toolbar in the main area switched positions, so that the toolbar is now always at the top of the table
- the main search field is now in the center of the top bar
- column 'Documents' reduced to show only if documents are available or not
- the button for the task function menu is only available if a row is selected or on hover to reduce visual clutter
- global function menu is placed at the bottom of the side bar, so that it is always available and not only if the 'Tasks' tab is selected
- increased padding and spacing in the tables to improve readability

3.2 Dojo-based user interface

From the functionality perspective, the client has not been changed, but the appearance has been improved in several ways. The main goal was to make the user interface look fresher and more modern - and to make it a better match with the React- based user interface.

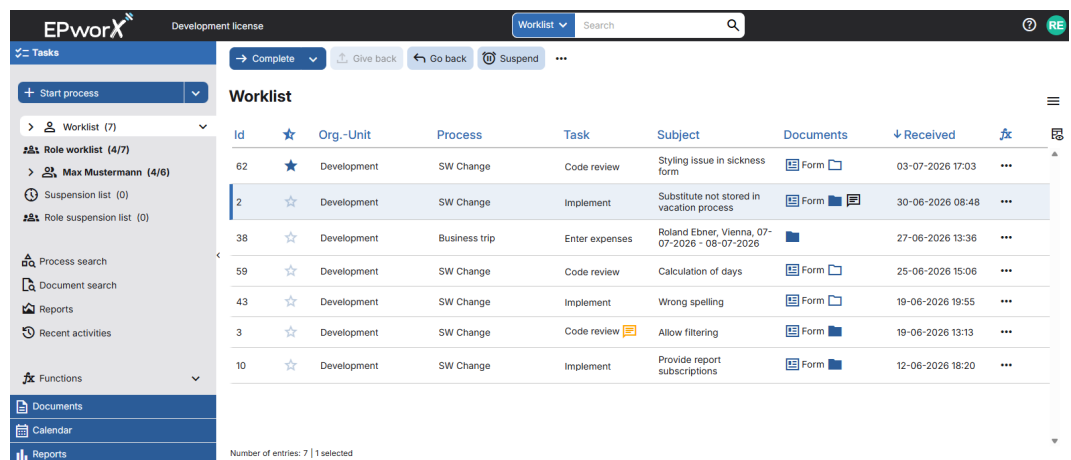


Figure 3.3: Dojo-based user interface

But note: this interface is to be considered as deprecated. It is not actively developed anymore and will be removed in the next version of *EPworX*. Therefore, we recommend switching to the React-based user interface as soon as possible.

3.3 Which client is used and when

This depends on the GUI Configuration assigned with the highest precedence to the user logging in. If the GUI Configuration is tagged as being applicable for Dojo, the

3. User Interface

Dojo-based client will be used. If it is tagged with React, the React-based client will be used. If no GUI Configuration is assigned to the user, the React-based client will be used with its default configuration.

Besides this mechanism, the user can also enforce the client to be used by adding the following path after the context root the URL used to navigate to *EPworX* :

- `/servlet.method/com.groiss.smartclient.Main.showMainPage` for the Dojo-based client
- `/ui` for the React-based client

This might be useful if the user needs a certain functionality that is only available in one of the clients. But therefore appropriate GUI Configurations must be available to the user for both of the clients.

Also note that the user must not be logged in to *EPworX* when using these modified URLs as it is not supported to switch between the clients during a session. Therefore it is also not possible to use both clients simultaneously in the same browser application. If the user wants to use both clients at the same time, he needs to use two different browsers, e.g. Chrome for the React client and Edge for the Dojo client.

4. Administration

- User has a new checkbox 'Administrative account' in the user detail dialog. If this checkbox is checked, the user can access the administration UI. Note that this checkbox can only be checked for as many users as administrative accounts are licensed in the license key.
- The following changes have been made in context of GUI Configurations:
 - a GUI Configuration now needs to be tagged as being applicable for Dojo, React or Mobile. All existing configurations have been tagged as Dojo by the upgrade mechanism.
 - a new property `componentKey` has been added to the various elements in a GUI Configuration. This property is used to identify the component to be used for this element in React.
 - for action elements a new property `actionKey` has been added. This property is used to identify the action to be marked as the primary action in React.
- For functions, a new property `componentKey` has been added. This property is used to identify the component to be used for this element in React.
- Web services: caused by the change of the used library, some changes were also made in the web service administration parts, e.g. uploading AAR files, deleting, or activating/deactivating web services are not supported anymore via the UI.

5. API

5.1 Logos

The names of the *EPworX* logo files have been renamed:

- from *enterprise.svg* to *epworx.svg*,
- from *enterprise-inverse.svg* to *epworx-inverse.svg*,
- from *enterprise-diamond.svg* to *epworx-small.svg*, as well as
- from *enterprise-inverse-diamond.svg* to *epworx-inverse-small.svg*.

If you shadowed those logo files you need to rename them in your *lang/default/images* folder accordingly.

Additionally, new logo files have been introduced which are used in the login dialog: *epworx-large.svg* and *epworx-inverse-large.svg*

5.2 Icons

The icon library changed from icomoon to Google Material Symbols (<https://fonts.google.com/icons>). The icons are used in the sharp style, which is the default style of the Material Symbols library.

Migration notes for customizations in Dojo:

- LESS variables for Material Symbols icons use the prefix `@ms-` instead of `@icon-` (e.g. `@ms-check` instead of `@icon-checkmark`).
- All available variables are defined in `fonts/material_variables.less`.
- icomoon icons and `@icon` variables remain available for now for backwards compatibility.
- icomoon is still used internally for mime-type icons in the DMS (e.g. PDF, Word, Excel).

5.3 Java API

As always, when providing a new major version, we clean up our public Java API with great care. You can find all new, changed and removed interfaces, classes and methods in appendix [Java interfaces, classes, and methods](#).

5.4 Configuration properties

5.4.1 Removed properties

The following properties have been removed as the the functionality using them has been removed or replaced with an alternative.

Configuration Property	Alternative
avw.distri.blocksize	No alternative
avw.distri.classes	No alternative
avw.distri.classes.archive	No alternative
avw.distri.classes.historic	No alternative
avw.distri.replication.disabled	No alternative
avw.distri.semaphore.concurrent	No alternative
avw.distri.semaphore.waiting	No alternative
avw.distri.synch_offset	No alternative
avw.dyn_class_load.enabled	No alternative
avw.license	ep.licensing.license. Old-style licenses are no longer recognized. A new license must be requested.
avw.permanent_secure_rmi	No alternative
avw.plain_rmi	No alternative
avw.rmi.enablefull	No alternative
avw.rmi.plain_exportport	No alternative
avw.rmi.port	No alternative
avw.rmi.ssl_exportport	No alternative
avw.secure_login_rmi	No alternative
avw_distri_activated	No alternative
calendar.agenda.days	No alternative
ep.notification.usetemplate	Recreate the old notification via a message template (see System Administration guide – Message templates) and add it to the needed process (see System Administration guide – Processes). When false, the old template (NotificationMask.html) was used. Got replaced by message templates.
ep.use.htmlforms	No alternative
replication.sync.tune_permissions	No alternative
rmi.dgc.log	No alternative
ssl_requireclientcertificate_over_rmi	No alternative

Table 5.1: Removed configuration parameters

5.4.2 Deprecated properties

The following properties have been marked as deprecated and may be removed in the next major version.

5. API

Configuration Property	Alternative
ep.process.details.tab	None. Cannot be maintained via configuration UI anymore. Not supported in the React-based client.
ep.process.details.toolbar	None. Cannot be maintained via configuration UI anymore. Default value is now true. Not supported in the React-based client.
ep.allow.html.description	None. Could not already be maintained via configuration UI. Will be removed in next major version.
ep.exception.allowhtmlmessage	Set on needed ApplicationException before throwing it via setAllowHTMLMessage(boolean). Could not already be maintained via configuration UI. Will be removed in next major version.

Table 5.2: Deprecated configuration properties

5.4.3 Properties with new default values

The default values of the following properties have been changed. This will only affect new installations, existing installations will keep their actually stored values.

Configuration Property	Old value	New value
avw.start.with.form	false	true
dms.use.texteditor	false	true
ep.process.details.toolbar	false	true

Table 5.3: Configuration properties with changed default values

6. Removed components and functions

Several outdated and/or unused components have been removed:

- As the Derby database is no longer maintained by its developers, the Derby database configuration has been removed as an option in the fill-in help for the database configuration. Existing installations will still work as Derby is still supported in *EPworX* 12.0 but it is recommended to migrate to a different database system as the support for Derby will be removed in a future version of *EPworX*.

Here is the official statement from the Apache Derby project:

"On 2025-10-10, the Derby developers voted to retire the project into a read-only state. Derby development and bug-fixing have ended. No further releases will be published. The Derby website and wiki are frozen in place..."

- The support for databases DB2 and MySQL (experimental) has been removed as well as we do not consider them relevant anymore.
- Support for interacting with *EPworX* from other Java programs via Remote Method Invocation (RMI) using `com.groiss.wf.Session` has been removed, and all related classes have been removed from the code base as well. The same applies for the connection of such a program to the notification mechanism as it depends on `com.groiss.wf.Session`.

For worklist related functionality, the REST API can be used as an alternative.

- The Online Archive has been removed as it was not used anymore. In this context, also the replication of build time data to a second *EPworX* server instance has been removed as it was only used for the Online Archive.
- Support for WfXML2 has been removed as this is a very old and outdated standard which is not used anymore. Please use the REST API for integration with other systems.
- Form types of type "HTMLForm" have been deprecated many versions ago, but were still supported at a certain extent for backward compatibility. With version 12.0 this support has now been removed.

Appendices

A. Migration from Axis2 to CXF

This chapter walks you through the complete process of migrating an existing Axis2-based web service to Apache CXF.

A.1 Overview and Background

Apache Axis2 and Apache CXF are both SOAP web service frameworks, but they differ significantly in their architecture and philosophy. Axis2 uses a modular, container-like architecture with its own deployment descriptors (`services.xml`, `axis2.xml`) and a concept of modules (`.mar` files) that bundle handlers together. CXF, on the other hand, follows a more lightweight, annotation-driven approach that integrates tightly with JAX-WS standards.

The key conceptual mappings you need to understand for the migration are:

Axis2 Concept	CXF Equivalent	Purpose
<code>services.xml</code>	<code>@WebService</code> annotation	Service configuration and metadata
Handlers	Interceptors	Pre/post-processing of messages
Modules (<code>.mar</code>)	Features	Bundled, reusable capabilities
<code>axis2.xml</code>	Annotations	Endpoint configuration

A.2 Prerequisites

Before you begin the migration, make sure you have the following:

- The WSDL file of your existing Axis2 service: This is the contract that defines your service's operations, messages and data types. If you do not have it as a standalone file, you can usually retrieve it from a running Axis2 service by appending `?wsdl` to the endpoint URL.
- Access to the CXF `wsdl2java` tool: This command-line utility ships with every CXF distribution and is used to generate Java source code from your WSDL.
- Your existing business logic: You will need the implementation code from your current Axis2 service. The migration does not require rewriting your business logic.
- Any existing handler implementations: If your Axis2 service uses handlers, you will need to rewrite them as CXF interceptors.

A.3 Step-by-Step Migration

A.3.1 Step 1: Generate the Service Interface from Your WSDL

The CXF `wsdl2java` tool reads your WSDL and generates a set of Java classes: the service endpoint interface (SEI), request and response types, object factories and related artifacts. Run the following command, replacing the package name and WSDL path with your own values:

```
1 wsdl2java -d src/main/java -p com.custom.package YourService.wsdl
```

Listing A.1: Usage of `wsdl2java`

- The `-d` flag specifies the output directory for the generated source files.
- The `-p` flag sets the Java package for all generated classes.

After running this command, you will find several generated files in the specified directory. The most important one is the service endpoint interface (e.g., `CustomServicePortType`), which defines the Java methods corresponding to the operations declared in your WSDL. **Important:** Do not modify the generated interface. It represents the contract as defined in your WSDL. If you need to change the interface, update the WSDL first and regenerate.

A.3.2 Step 2: Implement the Service Interface

Create a new Java class that implements the generated service endpoint interface. This is where you place your business logic, typically by copying it over from your existing Axis2 service implementation. The critical difference from Axis2 is that you must annotate this class with `@WebService` so that CXF can recognize it as a deployable service endpoint. Here is an example of what the implementation class should look like:

```
1 @WebService(  
2     serviceName = "CustomService",  
3     portName = "CustomServicePort",  
4     targetNamespace = "http://custom.namespace/",  
5     endpointInterface = "com.custom.package.CustomServicePortType"  
6 )  
7 public class CustomServiceImpl implements CustomServicePortType {  
8  
9     @Override  
10    public CustomResponse customOperation(CustomRequest request) {  
11        // Your business logic here  
12    }  
13 }
```

Listing A.2: Implementation class

Each attribute in the `@WebService` annotation serves a specific purpose:

- `serviceName` must match the name attribute of the `<wsdl:service>` element in your WSDL.

- portName must match the name attribute of the <wsdl:port> element in your WSDL.
- targetNamespace must match the targetNamespace of your WSDL.
- endpointInterface is the fully qualified class name of the generated service endpoint interface. This links your implementation to the contract.

A.3.3 Step 3: Migrate Handlers to Interceptors

In Axis2, handlers are components that sit in a processing pipeline and can inspect or modify SOAP messages before or after the service's business logic runs.

CXF uses a similar concept called interceptors. Like Axis2 handlers, interceptors sit in a processing chain and can read or modify messages. However, CXF's interceptor model is more fine-grained. Instead of simple "inbound" and "outbound" chains, CXF organizes interceptors into phases that represent distinct stages of message processing. A full list of phases can be found here: <https://cxf.apache.org/docs/interceptors.html>

Here is an example of a CXF interceptor:

```
1 import org.apache.cxf.interceptor.Fault;
2 import org.apache.cxf.message.Message;
3 import org.apache.cxf.phase.AbstractPhaseInterceptor;
4 import org.apache.cxf.phase.Phase;
5
6 public class CustomInterceptor
7     extends AbstractPhaseInterceptor<Message> {
8
9     public CustomInterceptor() {
10         // Specify the phase in which this interceptor runs.
11         super(Phase.PRE_INVOKE);
12     }
13
14     @Override
15     public void handleMessage(Message message) throws Fault {
16         // This method is called for every incoming request.
17
18         // Example: Log the operation being called
19         String opName = message.getExchange()
20             .getBindingOperationInfo()
21             .getName().getLocalPart();
22         System.out.println("Operation_called:_" + opName);
23
24         // To abort processing and return a fault:
25         // throw new Fault(
26         //     new RuntimeException("Access denied"));
27     }
28 }
```

Listing A.3: CXF Interceptor

If your Axis2 handler also processed outbound (response) messages, you need to create a separate outbound interceptor. In Axis2, a single handler could often handle

both directions. In CXF, inbound and outbound interceptors are distinct classes, each registered in a separate chain.

If you need to control the order of execution among multiple interceptors in the same phase, CXF allows you to declare ordering constraints in the constructor:

```

1 public CustomInterceptor() {
2     super(Phase.PRE_INVOKE);
3     // Ensure this interceptor runs before another one:
4     addBefore(OtherInterceptor.class.getName());
5     // Or ensure it runs after another one:
6     addAfter(AnotherInterceptor.class.getName());
7 }

```

Listing A.4: Phase ordering

A.3.4 Step 4: Migrate Modules to Features

In Axis2, modules are packaged as .mar files and represent reusable bundles of functionality that can be engaged on a service. A module typically contains one or more handlers along with a module descriptor (module.xml). Common examples include the Rampart module for WS-Security and the Sandesha2 module for WS-ReliableMessaging.

The CXF equivalent of an Axis2 module is a Feature. A Feature is a component that bundles together one or more interceptors into a reusable unit.

The following table shows some Axis2 module-to-CXF Feature mappings:

Axis2 Module	CXF Feature / Equivalent	Notes
Rampart (WS-Security)	WSS4JInInterceptor WSS4JOutInterceptor	/ CXF uses WS-Security interceptors directly rather than wrapping them in a module-like construct
Addressing	WSAddressingFeature	Enables WS-Addressing headers on request and response
Sandesha2 (WS-RM)	RMFeature	Enables WS-ReliableMessaging
Custom logging module	LoggingFeature	Logs inbound and outbound SOAP messages
Custom module with handlers	Custom Feature (extend AbstractFeature)	Bundle your interceptors into a reusable Feature class

A.3.5 Step 5: Annotate the Service with Interceptors and Features

Once you have converted your handlers to interceptors, you need to wire them to your service implementation. In CXF, this is done using annotations on the implementation class. This replaces the handler chain configuration you previously had in your Axis2 services.xml.

Here is a complete example showing how to attach interceptors and features:

```

1 @WebService(...)

```

```
2 @InInterceptors(interceptors = {
3     "com.custom.package.CustomInterceptor"
4 })
5 @InFaultInterceptors(interceptors = {
6     "com.custom.package.CustomInFaultInterceptor"
7 })
8 @OutInterceptors(interceptors = {
9     "com.custom.package.CustomOutInterceptor"
10 })
11 @OutFaultInterceptors(interceptors = {
12     "com.custom.package.CustomOutFaultInterceptor"
13 })
14 @Features(features = {
15     "org.apache.cxf.feature.LoggingFeature"
16 })
17 public class CustomServiceImpl
18     implements CustomServicePortType {
19     // ...
20 }
```

Listing A.5: Annotations

Note the following details:

- @InInterceptors registers interceptors for the inbound (request) processing chain.
- @InFaultInterceptors registers interceptors for the inbound (request) fault processing chain.
- @OutInterceptors registers interceptors for the outbound (response) processing chain.
- @OutFaultInterceptors registers interceptors for the outbound (response) fault processing chain.
- @Features activates features. Each feature is referenced by its fully qualified class name. CXF will instantiate the feature and let it register its own interceptors internally.

All referenced interceptor and feature classes must have a public no-argument constructor, since CXF instantiates them by class name.

A.3.6 Step 6: Deploy the Service

Deployment in CXF is simpler than in Axis2. There is no `services.xml`, no `.aar` archive and no dedicated deployment directory. Instead, you simply place the compiled service class (and its dependencies) on the classpath of your installation. CXF will discover and register the service automatically at startup.

Follow these steps to deploy:

1. Place your compiled service class, interceptors, features and the WSDL file into the classpath of your installation.
2. Enable the property `Activate web services` under *Administration* → *Configuration* → *Web components*.
3. Restart the server.
4. Verify that your service was detected by navigating to *Admin-Tasks* → *Communication* → *Local services*. Your service should appear in the list indicating it started successfully.

A.4 Troubleshooting

If your service does not appear in the Local services list after restarting, check the following:

Missing `@WebService` annotation: CXF will only auto-detect classes that carry the `@WebService` annotation. Double-check that the annotation is present and that all its attributes are correct.

Interceptor instantiation errors: If an interceptor or feature class cannot be found or lacks a no-argument constructor, CXF will log an error during startup. Check the server logs for `ClassNotFoundException` or `NoSuchMethodException` entries.

Phase mismatch: If your interceptor runs in the wrong phase, it may not have access to the data it expects.

Property not enabled: Make sure the `Activate web services` property is set to `true`. Without it, the CXF servlet will not be activated and no services will be published.

B. Library updates

This chapter lists all libraries which have been added, updated, or removed in *EPworX* 12.0.

B.1 Added libraries

Library	Version
com.aayushatharva.brotli4j:brotli4j	1.23.0
com.aayushatharva.brotli4j:service	1.23.0
com.fasterxml.jackson.jakarta.rs:jackson-jakarta-rs-base	2.22.0
com.fasterxml.jackson.jakarta.rs:jackson-jakarta-rs-json-provider	2.22.0
com.fasterxml.jackson.module:jackson-module-jakarta-xmlbind-annotations	2.22.0
com.google.code.findbugs:annotations	3.0.1
com.google.crypto.tink:tink	1.21.0
com.google.j2objc:j2objc-annotations	3.1
com.google.protobuf:protobuf-java	4.33.0
com.groiss:ep-dojo-repackaged	1.17.3
com.groiss:ep-icomoon-repackaged	1.0.1
com.sun.xml.messaging.saaj:saaj-impl	3.0.5
io.dropwizard.metrics:metrics-json	4.2.26
io.swagger.core.v3:swagger-annotations-jakarta	2.2.52
io.swagger.core.v3:swagger-core-jakarta	2.2.52
io.swagger.core.v3:swagger-integration-jakarta	2.2.52
io.swagger.core.v3:swagger-jaxrs2-jakarta	2.2.52
io.swagger.core.v3:swagger-models-jakarta	2.2.52
jakarta.el:jakarta.el-api	6.0.0
jakarta.enterprise:jakarta.enterprise.cdi-api	4.1.0
jakarta.enterprise:jakarta.enterprise.lang-model	4.1.0
jakarta.inject:jakarta.inject-api	2.0.1
jakarta.interceptor:jakarta.interceptor-api	2.2.0
jakarta.mail:jakarta.mail-api	2.1.5
jakarta.servlet:jakarta.servlet-api	6.1.0
jakarta.websocket:jakarta.websocket-api	2.2.0
jakarta.websocket:jakarta.websocket-client-api	2.2.0
javax.cache:cache-api	1.1.0
net.jcip:jcip-annotations	1.0

B. Library updates

Library	Version
net.shibboleth:shib-networking	9.1.6
net.shibboleth:shib-security	9.1.6
net.shibboleth:shib-support	9.1.6
net.shibboleth:shib-velocity	9.1.6
org.apache.commons:commons-fileupload2-core	2.0.0-M5
org.apache.commons:commons-fileupload2-jakarta-servlet6	2.0.0-M5
org.apache.commons:commons-text	1.15.0
org.apache.cxf:cxf-core	4.2.2
org.apache.cxf:cxf-rt-bindings-soap	4.2.2
org.apache.cxf:cxf-rt-bindings-xml	4.2.2
org.apache.cxf:cxf-rt-databinding-jaxb	4.2.2
org.apache.cxf:cxf-rt-features-logging	4.2.2
org.apache.cxf:cxf-rt-frontend-jaxws	4.2.2
org.apache.cxf:cxf-rt-frontend-simple	4.2.2
org.apache.cxf:cxf-rt-security	4.2.2
org.apache.cxf:cxf-rt-security-saml	4.2.2
org.apache.cxf:cxf-rt-transport-http	4.2.2
org.apache.cxf:cxf-rt-ws-addr	4.2.2
org.apache.cxf:cxf-rt-ws-policy	4.2.2
org.apache.cxf:cxf-rt-ws-security	4.2.2
org.apache.cxf:cxf-rt-wsdl	4.2.2
org.apache.httpcomponents.client5:httpclient5	5.5
org.apache.httpcomponents.client5:httpclient5-cache	5.5
org.apache.httpcomponents.core5:httpcore5	5.3.4
org.apache.httpcomponents.core5:httpcore5-h2	5.3.4
org.apache.wss4j:wss4j-ws-security-policy-stax	4.0.1
org.apache.xmlgraphics:batik-anim	1.19
org.apache.xmlgraphics:batik-awt-util	1.19
org.apache.xmlgraphics:batik-bridge	1.19
org.apache.xmlgraphics:batik-codec	1.19
org.apache.xmlgraphics:batik-constants	1.19
org.apache.xmlgraphics:batik-css	1.19
org.apache.xmlgraphics:batik-dom	1.19
org.apache.xmlgraphics:batik-ext	1.19
org.apache.xmlgraphics:batik-gvt	1.19
org.apache.xmlgraphics:batik-i18n	1.19
org.apache.xmlgraphics:batik-parser	1.19
org.apache.xmlgraphics:batik-script	1.19
org.apache.xmlgraphics:batik-shared-resources	1.19
org.apache.xmlgraphics:batik-svg-dom	1.19
org.apache.xmlgraphics:batik-svggen	1.19
org.apache.xmlgraphics:batik-transcoder	1.19
org.apache.xmlgraphics:batik-util	1.19
org.apache.xmlgraphics:batik-xml	1.19
org.apache.xmlgraphics:xmlgraphics-commons	2.11
org.cometd.java:cometd-java-server-http-jakarta	9.0.0
org.cometd.java:cometd-java-server-websocket-jakarta	9.0.0
org.eclipse.angus:angus-activation	2.0.3
org.eclipse.angus:angus-mail	2.0.5
org.eclipse.jetty.compression:jetty-compression-common	12.1.10
org.eclipse.jetty.compression:jetty-compression-gzip	12.1.10

B. Library updates

Library	Version
org.eclipse.jetty.ee11.websocket:jetty-ee11-websocket-jakarta-client	12.1.10
org.eclipse.jetty.ee11.websocket:jetty-ee11-websocket-jakarta-common	12.1.10
org.eclipse.jetty.ee11.websocket:jetty-ee11-websocket-jakarta-server	12.1.10
org.eclipse.jetty.ee11.websocket:jetty-ee11-websocket-servlet	12.1.10
org.eclipse.jetty.ee11:jetty-ee11-annotations	12.1.10
org.eclipse.jetty.ee11:jetty-ee11-plus	12.1.10
org.eclipse.jetty.ee11:jetty-ee11-servlet	12.1.10
org.eclipse.jetty.ee11:jetty-ee11-webapp	12.1.10
org.eclipse.jetty.ee:jetty-ee-webapp	12.1.10
org.eclipse.jetty:jetty-annotations	12.1.10
org.ehcache:ehcache	3.12.0
org.jasypt:jasypt	1.9.3
org.jSpecify:jSpecify	1.0.0
org.jvnet.staxex:stax-ex	2.1.0
org.opensaml:opensaml-core-api	5.1.6
org.opensaml:opensaml-core-impl	5.1.6
org.pac4j:jakartaee-pac4j	8.0.3
org.pac4j:pac4j-core	6.5.2
org.pac4j:pac4j-jakartaee	6.4.2
org.pac4j:pac4j-oidc	6.5.2
org.springframework:spring-core	6.2.18
xml-apis:xml-apis-ext	1.3.04

B.2 Removed libraries

Library	Version in 11.0
antlr:antlr	2.7.7
com.fasterxml.jackson.jaxrs:jackson-jaxrs-base	2.16.2
com.fasterxml.jackson.jaxrs:jackson-jaxrs-json-provider	2.16.2
com.fasterxml.jackson.module:jackson-module-jaxb-annotations	2.16.2
com.github.stephenc.jcip:jcip-annotations	1.0-1
com.google.errorprone:javac-shaded	9+181-r4173-1
com.google.googlejavaformat:google-java-format	1.7
com.squareup.moshi:moshi	1.15.1
com.squareup.moshi:moshi-adapters	1.15.1
com.squareup.okio:okio	3.7.0
com.squareup.okio:okio-jvm	3.7.0
com.sun.activation:jakarta.activation	1.2.2
com.sun.istack:istack-commons-tools	3.0.12
com.sun.mail:jakarta.mail	1.6.7
com.sun.xml.bind.external:relaxng-datatype	2.3.6
com.sun.xml.bind.external:rngom	2.3.6

B. Library updates

Library	Version in 11.0
com.sun.xml.bind:jaxb-impl	2.3.8
com.sun.xml.bind:jaxb-jxc	2.3.3
com.sun.xml.dtd-parser:dtd-parser	1.4.5
com.sun.xml.fastinfoset:FastInfoset	2.1.0
com.sun.xml.ws:jaxws-tools	2.3.3
commons-fileupload:commons-fileupload	1.6.0
io.swagger.core.v3:swagger-annotations	2.2.21
io.swagger.core.v3:swagger-core	2.2.21
io.swagger.core.v3:swagger-integration	2.2.21
io.swagger.core.v3:swagger-jaxrs2	2.2.21
io.swagger.core.v3:swagger-models	2.2.21
jakarta.jws:jakarta.jws-api	2.1.0
javax.xml.bind:jaxb-api	2.3.1
net.shibboleth.utilities:java-support	8.4.2
org.apache.axis2:axis2-adb	1.8.2
org.apache.axis2:axis2-adb-codegen	1.8.2
org.apache.axis2:axis2-ant-plugin	1.8.2
org.apache.axis2:axis2-codegen	1.8.2
org.apache.axis2:axis2-corba	1.8.2
org.apache.axis2:axis2-fastinfoset	1.8.2
org.apache.axis2:axis2-java2wsdl	1.8.2
org.apache.axis2:axis2-jaxbri-codegen	1.8.2
org.apache.axis2:axis2-jaxws	1.8.2
org.apache.axis2:axis2-json	1.8.2
org.apache.axis2:axis2-kernel	1.8.2
org.apache.axis2:axis2-metadata	1.8.2
org.apache.axis2:axis2-mtompolicy	1.8.2
org.apache.axis2:axis2-saaj	1.8.2
org.apache.axis2:axis2-soapmonitor-servlet	1.8.2
org.apache.axis2:axis2-transport-base	1.8.2
org.apache.axis2:axis2-transport-http	1.8.2
org.apache.axis2:axis2-transport-jms	1.8.2
org.apache.axis2:axis2-transport-local	1.8.2
org.apache.axis2:axis2-transport-mail	1.8.2
org.apache.axis2:axis2-transport-tcp	1.8.2
org.apache.axis2:axis2-transport-udp	1.8.2
org.apache.axis2:axis2-xmlbeans	1.8.2
org.apache.axis2:axis2-xmlbeans-codegen	1.8.2
org.apache.axis2:mex	1.8.2
org.apache.groovy:groovy-servlet	4.0.28
org.apache.groovy:groovy-templates	4.0.28
org.apache.james:apache-mime4j-core	0.8.11
org.apache.rampart:rampart-core	1.8.0
org.apache.rampart:rampart-policy	1.8.0
org.apache.rampart:rampart-trust	1.8.0
org.apache.woden:woden-core	1.0M10
org.apache.ws.commons.axiom:axiom-api	1.4.0
org.apache.ws.commons.axiom:axiom-dom	1.4.0
org.apache.ws.commons.axiom:axiom-impl	1.4.0
org.apache.wss4j:wss4j	3.0.3
org.codehaus.jettison:jettison	1.5.4

B. Library updates

Library	Version in 11.0
org.cometd.java:cometd-java-server-websocket-javax	6.0.15
org.eclipse.jetty.ee8.websocket:jetty-ee8-websocket-javax-client	12.0.19
org.eclipse.jetty.ee8.websocket:jetty-ee8-websocket-javax-common	12.0.19
org.eclipse.jetty.ee8.websocket:jetty-ee8-websocket-javax-server	12.0.19
org.eclipse.jetty.ee8.websocket:jetty-ee8-websocket-servlet	12.0.19
org.eclipse.jetty.ee8:jetty-ee8-annotations	12.0.19
org.eclipse.jetty.ee8:jetty-ee8-nested	12.0.19
org.eclipse.jetty.ee8:jetty-ee8-plus	12.0.19
org.eclipse.jetty.ee8:jetty-ee8-security	12.0.19
org.eclipse.jetty.ee8:jetty-ee8-servlet	12.0.19
org.eclipse.jetty.ee8:jetty-ee8-webapp	12.0.19
org.eclipse.jetty.toolchain:jetty-javax-websocket-api	1.1.2
org.eclipse.jetty.toolchain:jetty-servlet-api	4.0.6
org.eclipse.jetty.websocket:jetty-websocket-jetty-api	12.0.19
org.eclipse.jetty.websocket:jetty-websocket-jetty-client	12.0.19
org.eclipse.jetty.websocket:jetty-websocket-jetty-common	12.0.19
org.eclipse.jetty.websocket:jetty-websocket-jetty-server	12.0.19
org.eclipse.jetty:jetty-ee	12.0.19
org.glassfish.hk2.external:jakarta.inject	2.6.1
org.glassfish.jaxb:codemodel	2.3.6
org.glassfish.jaxb:jaxb-xjc	2.3.6
org.glassfish.jaxb:xsom	2.3.6
org.jetbrains.kotlin:kotlin-stdlib	1.9.21
org.jetbrains.kotlin:kotlin-stdlib-jdk7	1.8.21
org.jetbrains.kotlin:kotlin-stdlib-jdk8	1.8.21
org.jetbrains:annotations	13.0
org.opensaml:opensaml-core	4.3.2
org.owasp.encoder:encoder	1.2.3
xalan:xalan	2.7.3

B.3 Libraries with changed versions

Library	Version in 11.0	Version in 12.0
ch.qos.logback:logback-classic	1.3.14	1.5.33
ch.qos.logback:logback-core	1.3.14	1.5.33
com.fasterxml.jackson.core:jackson-annotations	2.16.2	2.22
com.fasterxml.jackson.core:jackson-core	2.16.2	2.22.0
com.fasterxml.jackson.core:jackson-databind	2.16.2	2.22.0

B. Library updates

Library	Version in 11.0	Version in 12.0
com.fasterxml.jackson.dataformat: jackson-dataformat-yaml	2.16.2	2.22.0
com.fasterxml.jackson.datatype: jackson-datatype-jsr310	2.16.2	2.22.0
com.fasterxml.woodstox:woodstox-core	6.5.1	7.2.0
com.github.librepdf:openpdf	2.0.2	3.0.5
com.github.ziplet:ziplet	2.4.1	2.5.0-atlassian-4
com.google.code.gson:gson	2.10	2.13.2
com.google.errorprone:error_prone_ annotations	2.26.1	2.47.0
com.google.guava:failureaccess	1.0.2	1.0.3
com.google.guava:guava	33.1.0-jre	33.6.0-jre
com.hazelcast:hazelcast	5.4.0	5.7.0
com.ibm.icu:icu4j	74.2	78.3
com.nimbusds:nimbus-jose-jwt	9.37.3	10.9
com.nimbusds:oauth2-oidc-sdk	11.10.1	11.37.2
com.sun.istack:istack-commons-runtime	4.0.1	4.1.2
commons-beanutils:commons-beanutils	1.9.4	1.11.0
commons-codec:commons-codec	1.16.1	1.19.0
commons-io:commons-io	2.19.0	2.22.0
commons-validator:commons-validator	1.7	1.10.1
de.inetsoftware:jlessc	1.12	1.16
io.dropwizard.metrics:metrics-core	4.2.25	4.2.26
io.github.classgraph:classgraph	4.8.165	4.8.184
io.github.java-diff-utils:java-diff- utils	4.12	4.17
jakarta.activation:jakarta.activation- api	2.0.1	2.1.4
jakarta.annotation:jakarta.annotation- api	1.3.5	3.0.0
jakarta.ejb:jakarta.ejb-api	3.2.6	4.0.1
jakarta.jms:jakarta.jms-api	2.0.3	3.1.0
jakarta.transaction:jakarta.transaction- api	1.3.3	2.0.1
jakarta.validation:jakarta.validation- api	2.0.2	3.0.2
jakarta.ws.rs:jakarta.ws.rs-api	2.1.6	3.1.0
jakarta.xml.bind:jakarta.xml.bind-api	3.0.1	4.0.5
jakarta.xml.soap:jakarta.xml.soap-api	1.4.2	3.0.2
jakarta.xml.ws:jakarta.xml.ws-api	2.3.3	4.0.3
jaxen:jaxen	2.0.0	2.0.4
joda-time:joda-time	2.12.7	2.14.2
net.bull.javamelody:javamelody-core	1.99.3	2.8.0
net.minidev:accessors-smart	2.5.0	2.6.0
net.minidev:json-smart	2.5.0	2.6.0
org.apache.commons:commons-collections4	4.4	4.5.0
org.apache.commons:commons-compress	1.26.1	1.28.0
org.apache.commons:commons-lang3	3.14.0	3.20.0
org.apache.groovy:groovy	4.0.28	5.0.6
org.apache.groovy:groovy-json	4.0.28	5.0.6
org.apache.groovy:groovy-xml	4.0.28	5.0.6

B. Library updates

Library	Version in 11.0	Version in 12.0
org.apache.logging.log4j:log4j-api	2.23.1	2.26.0
org.apache.logging.log4j:log4j-to-slf4j	2.23.1	2.26.0
org.apache.neethi:neethi	3.2.0	3.2.2
org.apache.pdfbox:fontbox	3.0.2	3.0.7
org.apache.pdfbox:pdfbox	3.0.2	3.0.7
org.apache.pdfbox:pdfbox-io	3.0.2	3.0.7
org.apache.santuario:xmlsec	3.0.4	4.0.4
org.apache.velocity:velocity-engine-core	2.3	2.4.1
org.apache.ws.xmlschema:xmlschema-core	2.3.0	2.3.2
org.apache.wss4j:wss4j-bindings	3.0.3	4.0.1
org.apache.wss4j:wss4j-policy	3.0.3	4.0.1
org.apache.wss4j:wss4j-ws-security-common	3.0.3	4.0.1
org.apache.wss4j:wss4j-ws-security-dom	3.0.3	4.0.1
org.apache.wss4j:wss4j-ws-security-stax	3.0.3	4.0.1
org.bouncycastle:bcpkix-jdk18on	1.78	1.84
org.bouncycastle:bcprov-jdk18on	1.79	1.84
org.bouncycastle:bcutil-jdk18on	1.78	1.84
org.checkerframework:checker-qual	3.42.0	3.37.0
org.codehaus.woodstox:stax2-api	4.2.1	4.3.0
org.cometd.java:cometd-java-api-client	6.0.15	9.0.0
org.cometd.java:cometd-java-api-common	6.0.15	9.0.0
org.cometd.java:cometd-java-api-server	6.0.15	9.0.0
org.cometd.java:cometd-java-common	6.0.15	9.0.0
org.cometd.java:cometd-java-server-common	6.0.15	9.0.0
org.cometd.java:cometd-java-server-websocket-common	6.0.15	9.0.0
org.cryptacular:cryptacular	1.2.6	1.2.7
org.eclipse.jetty.websocket:jetty-websocket-core-client	12.0.19	12.1.10
org.eclipse.jetty.websocket:jetty-websocket-core-common	12.0.19	12.1.10
org.eclipse.jetty.websocket:jetty-websocket-core-server	12.0.19	12.1.10
org.eclipse.jetty:jetty-client	12.0.19	12.1.10
org.eclipse.jetty:jetty-http	12.0.19	12.1.10
org.eclipse.jetty:jetty-io	12.0.19	12.1.10
org.eclipse.jetty:jetty-jmx	12.0.19	12.1.10
org.eclipse.jetty:jetty-jndi	12.0.19	12.1.10
org.eclipse.jetty:jetty-plus	12.0.19	12.1.10
org.eclipse.jetty:jetty-security	12.0.19	12.1.10
org.eclipse.jetty:jetty-server	12.0.19	12.1.10
org.eclipse.jetty:jetty-session	12.0.19	12.1.10
org.eclipse.jetty:jetty-util	12.0.19	12.1.10
org.eclipse.jetty:jetty-util-ajax	12.0.19	12.1.10
org.eclipse.jetty:jetty-xml	12.0.19	12.1.10
org.glassfish.hk2.external:aopalliance-repackaged	2.6.1	3.0.6
org.glassfish.hk2:hk2-api	2.6.1	3.0.6

B. Library updates

Library	Version in 11.0	Version in 12.0
org.glassfish.hk2:hk2-locator	2.6.1	3.0.6
org.glassfish.hk2:hk2-utils	2.6.1	3.0.6
org.glassfish.jaxb:jaxb-core	3.0.2	4.0.9
org.glassfish.jaxb:jaxb-runtime	3.0.2	4.0.9
org.glassfish.jaxb:txw2	3.0.2	4.0.9
org.glassfish.jersey.containers: jersey-container-servlet-core	2.42	3.1.11
org.glassfish.jersey.core:jersey-client	2.42	3.1.11
org.glassfish.jersey.core:jersey-common	2.42	3.1.11
org.glassfish.jersey.core:jersey-server	2.42	3.1.11
org.glassfish.jersey.inject:jersey-hk2	2.42	3.1.11
org.glassfish.jersey.media:jersey- media-multipart	2.42	3.1.11
org.jfree:jfreechart	1.5.4	1.5.6
org.jgroups:jgroups	5.2.23.Final	5.5.5.Final
org.jodconverter:jodconverter-core	4.4.7	4.4.11
org.jodconverter:jodconverter-local	4.4.7	4.4.11
org.jpmmml:pmml-model	1.6.4	1.7.7
org.json:json	20240303	20260522
org.mnode.ical4j:ical4j	3.2.17	4.2.5
org.odftoolkit:odfdom-java	0.12.0	0.13.0
org.opensaml:opensaml-messaging-api	4.3.2	5.1.6
org.opensaml:opensaml-profile-api	4.3.2	5.1.6
org.opensaml:opensaml-saml-api	4.3.2	5.1.6
org.opensaml:opensaml-saml-impl	4.3.2	5.1.6
org.opensaml:opensaml-security-api	4.3.2	5.1.6
org.opensaml:opensaml-security-impl	4.3.2	5.1.6
org.opensaml:opensaml-soap-api	4.3.2	5.1.6
org.opensaml:opensaml-soap-impl	4.3.2	5.1.6
org.opensaml:opensaml-storage-api	4.3.2	5.1.6
org.opensaml:opensaml-xacml-api	4.3.2	5.1.6
org.opensaml:opensaml-xacml-impl	4.3.2	5.1.6
org.opensaml:opensaml-xacml-saml-api	4.3.0	5.1.6
org.opensaml:opensaml-xacml-saml-impl	4.3.0	5.1.6
org.opensaml:opensaml-xmlsec-api	4.3.2	5.1.6
org.opensaml:opensaml-xmlsec-impl	4.3.2	5.1.6
org.slf4j:jcl-over-slf4j	2.0.16	2.0.18
org.slf4j:jul-to-slf4j	2.0.16	2.0.18
org.slf4j:log4j-over-slf4j	2.0.16	2.0.18
org.slf4j:slf4j-api	2.0.16	2.0.18
org.xhtmlrenderer:flyingsaucer-core	9.7.1	10.2.2
org.xhtmlrenderer:flyingsaucer-pdf	9.7.1	10.2.2
org.yaml:snakeyaml	2.2	2.6

C. Java interfaces, classes, and methods

This chapter lists all Java interfaces, classes and methods which have been added, changed, removed, or deprecated in *EPworX* 12.0 and provides alternatives where applicable.

C.1 New interfaces, classes, and methods

The following interfaces, classes and methods have been added to the public API, see tables [C.1](#) and [C.2](#) for details.

Added interface/class
com.groiss.iam.sync.UserSynchronizer
com.groiss.mail.MailHandler
com.groiss.ws.SOAPWebServiceModuleProvider

Table C.1: New interfaces and classes

Added method
com.groiss.gui.UserAgent.isReactClient()
com.groiss.gui.table.ColumnDescription.getComponentKey()
com.groiss.gui.table.ColumnDescription.setComponentKey(String)
com.groiss.org.OrgData.checkAdmin()
com.groiss.org.OrgData.isAdmin()
com.groiss.org.User.LICENSE_ADMIN
com.groiss.org.User.LICENSE_REGULAR
com.groiss.org.User.checkAdmin()
com.groiss.org.User.getLicenseType()
com.groiss.org.User.isAdmin()
com.groiss.org.User.setLicenseType(int)

Table C.2: New methods

C.2 Changed interfaces, classes, and methods

The following interfaces, classes and methods have been changed in the public API, see tables [C.3](#), [C.4](#) and [C.5](#) for details.

C.3 Deprecated interfaces, classes, and methods

The following interfaces, classes and methods have been deprecated in the public API, see table [C.6](#) for details.

C.4 Removed interfaces and classes

The following interfaces and classes have been removed from the public API, see table [C.7](#) for details:

C.5 Removed methods

The following methods have been removed from the public API, see tables [C.8](#) to [C.19](#) for details.

Changed	Change
Page HTMLToolbar.show(String actions, String target, boolean verticalToolbar, boolean showIcons, HttpSession sess, Map<Object, Object> params)	Page HTMLToolbar.show(String actions, String target, boolean verticalToolbar, boolean showIcons, Map<Object, Object> params)
Additional Info: HttpSession-Parameter not needed anymore.	
Page HTMLToolbar.show(String actions, String target, boolean verticalToolbar, ToolbarShape shape, HttpSession sess, Map<Object, Object> params)	Page HTMLToolbar.show(String actions, String target, boolean verticalToolbar, ToolbarShape shape, Map<Object, Object> params)
Additional Info: HttpSession-Parameter not needed anymore.	
com.groiss.notification.NotificationSuite. publishImmediate(NotificationEvent) throws Exception	com.groiss.notification.NotificationSuite. publishImmediate(NotificationEvent)
Additional Info: Checked Exception not thrown anymore.	
com.groiss.notification.NotificationSuite. publishImmediate(NotificationItem, short) throws Exception	com.groiss.notification.NotificationSuite. publishImmediate(NotificationItem, short)
Additional Info: Checked Exception not thrown anymore.	
com.groiss.notification.NotificationSuite. publishImmediate(NotificationItem, short, Agent) throws Exception	com.groiss.notification.NotificationSuite. publishImmediate(NotificationItem, short, Agent)
Additional Info: Checked Exception not thrown anymore.	
com.groiss.notification.NotificationSuite. notifyUser(User, String, Map<String, ?>) throws Exception	com.groiss.notification.NotificationSuite. notifyUser(User, String, Map<String, ?>)
Additional Info: Checked Exception not thrown anymore.	
com.groiss.notification.NotificationSuite. notifySession(HttpSession, String, Map<String, ?>) throws Exception	com.groiss.notification.NotificationSuite. notifySession(HttpSession, String, Map<String, ?>)
Additional Info: Checked Exception not thrown anymore.	

Table C.3: List of methods with changed signature

Changed	Change
<code>com.groiss.cluster.ClusterSupport.getRedirectAddresses</code> (<code>HttpServletRequest</code> , <code>HttpServletResponse</code>)	The RMI address will no longer be part of the response.
Additional Info: The order of the addresses in the response has change. The HTTP address will be in the first line instead of the second and the HTTPS address will the second line instead of the third.	
<code>com.groiss.cluster.ClusterSupport.getAllRedirectAddresses</code> (<code>HttpServletRequest</code> , <code>HttpServletResponse</code>)	The RMI address will no longer be part of the response.
Additional Info: The order of the addresses in the response has change. The HTTP address will be in the first line instead of the second and the HTTPS address will the second line instead of the third.	
<code>void com.groiss.cluster.ClusterSupport. getNodeState(HttpServletRequest, HttpServletResponse)</code>	Does not return <code>SC_NOT_FOUND</code> if the node is load balanced, but logins are disabled without evicting sessions (blue)

Table C.4: List of methods with changed behavior

Changed	Change
com.groiss.notification.BasicNotificationItem	The class is now abstract.
Additional Info: No instances of this class can be created anymore. A new class extending com.groiss.notification.BasicNotificationItem has to be created. All classes extending com.groiss.notification.BasicNotificationItem have to implement the getTopic method.	

Table C.5: List of interfaces and classes with changed signature

Deprecated	Alternative
<code>com.groiss.gui.table.TableAdapterUtil</code>	<code>com.groiss.gui.table.ColumnDescription</code>
Additional Info: Avoid the need to transform it into old <code>ColumnDescription</code>	
<code>com.groiss.org.OrgData.ADMIN</code>	Use <code>textttcom.groiss.org.OrgData.isAdmin()</code> to check if a user is an admin.
Additional Info: Is not used anymore due to changes in the licensing model.	
<code>com.groiss.org.OrgData.NAMED_USER</code>	No alternative
Additional Info: Is not used anymore due to changes in the licensing model.	
<code>com.groiss.wf.ApplicationAdapter.modifyDetailPanels (KeyedList<String, NavigationTreeNode>, StringBuilder, ProcessInstance, ActivityInstance)</code>	<code>com.groiss.wf.ApplicationAdapter.modifyDetailPanels (KeyedList<String, NavigationTreeNode>, ProcessInstance, ActivityInstance)</code>
Additional Info: Add a new node having an attribute "region=top" to the passed nodes if you want render a modified header component in the detail view. Such a node can also be defined in your GUI XML and used in this method, e.g. <code>nodes.put("myCustomHeader", GUIConfiguration.getNode("myGuiConfig.myCustomHeader"))</code> . In this case, you can also use the id of that node and alternatively add it to a specific process definition via tab 'General' in the details view of a process definition in the administrative UI (see administrative handbook for further details).	
<code>com.groiss.wf.DefaultApplicationAdapter.getNotificationMail (User, ActivityInstance, String)</code>	Recreate the notification via a message template
Additional Info: Create a message template for the notification (see System Administration Guide – Message templates) and assign it to the affected process (see System Administration Guide – Processes).	
<code>com.groiss.wf.ServiceLocator</code>	Use <code>WfEngine.getInstance()</code> , <code>OrgData.getInstance()</code> , <code>Store.getInstance()</code> , <code>Admin.getInstance()</code> , <code>DMS.getInstance()</code>
<code>com.groiss.wf.html.HTMLUtils.getWorklist</code>	No alternative
Additional Info: Considered to be not useful anymore nowadays.	
<code>com.groiss.wf.html.Worklist.getTitle</code>	No alternative
Additional Info: Was not used anymore since version 10, and was never supported in the SPA worklist UI.	

Table C.6: Deprecated interfaces, classes and methods

C. Java interfaces, classes, and methods

Removed Class/Interface	Alternative
com.groiss.cal.CalInfoAdapter	Implement CalInfo instead
com.groiss.component.ServiceAdapter	Implement interface com.groiss.component.Service directly
com.groiss.dms.DMSTableAdapter	Implement DMSTableHandler instead
com.groiss.dms.FormEventAdapter	Implement FormEventHandler instead
com.groiss.dms.FormTableAdapter	Implement FormTableHandler instead
com.groiss.dms.RemoteDMS	No alternative
com.groiss.dms.XHTMLFolderFormEventAdapter	Implement XHTMLFolderFormEventHandler instead
com.groiss.dms.XHTMLFormEventAdapter	Implement XHTMLFormEventHandler instead
com.groiss.fileimport.ImportHandler	Implement FileImportHandler instead
com.groiss.gui.JavaScriptLoader	No alternative
com.groiss.mail.MailHandler2	com.groiss.mail.MailHandler
com.groiss.org.RemoteOrgData	No alternative
com.groiss.servlet.Public	Annotate your class with @com.groiss.servlet.Access(Access.mode.Public) instead. Also allows you to annotate packages and methods.
com.groiss.store.PersistentEventAdapter	Implement PersistentEventHandler instead
com.groiss.store.RemoteStore	No alternative
com.groiss.util.RMIThreadContext	No alternative
com.groiss.wf.RMIService	No alternative
com.groiss.wf.RemoteWfEngine	No alternative
com.groiss.wf.Session	No alternative
com.groiss.wf.SessionFactory	No alternative
com.groiss.wf.html.WorklistAdapter	Implement WorkList directly instead
com.groiss.wfxml.Partner	No alternative
com.groiss.wfxml.WfXML2	No alternative
com.groiss.wfxml.WfXMLDocForm	No alternative
com.groiss.wfxml.WfXMLFolder	No alternative
com.groiss.wfxml.WfXMLForm	No alternative
com.groiss.wfxml.WfXMLNote	No alternative
com.groiss.wfxml.WfXMLObject	No alternative
com.groiss.wfxml.WfXMLWebLink	No alternative
com.groiss.ws.ClasspathResolver	No alternative
com.groiss.ws.client. ConfigurationContextFactory	No alternative
com.groiss.ws.server.MessageHandler	No alternative

Table C.7: Removed interfaces and classes

Removed	Alternative
List<Right> ApplicationAdapter.getFinishRights()	No alternative
Right ApplicationAdapter.getFinishRight(ActivityInstance ai)	No alternative
public final IPermissionMapping PermissionMapping.getParentMapping()	Use PermissionMapping.parentRewrite(PermissionQuery)
Additional Info: This method is inherently broken, in case of a heterogeneous list of persistent objects that need to be checked. In that case, the parent mapping for the first object in the list will be returned. Before: DNF result = getParentMapping().rewrite(q); After: DNF result = parentRewrite(q);	
public Map<String, String> Response.getHeaders()	Use Response.getAllHeaders() or Response.getHeaderValues(String) instead.
Additional Info: It cannot handle the case of multiple headers with the same name (case-sensitive) – this method will only return the last value for such headers.	
public Object Application.getAppClass()	Use ApplicationAdapter.of(String) or ApplicationAdapter.of(Application)
Additional Info: ApplicationAdapter.of(String) and ApplicationAdapter.of(Application) do not unwrap the application adapter. If you want to cast the result to your implementation class, use com.groiss.component.LoggingWrapper.unwrap(0) as well. MyApplicationAdapter myAdapter = (MyApplicationAdapter) LoggingWrapper.unwrap(ApplicationAdapter.of(application));	
com.groiss.wf.DefaultApplicationAdapter.getNotificationMail (User, ActivityInstance)	No alternative – use message templates instead
Additional Info: Got called when the configuration parameter ep.notification.usertemplate was set to false.	
com.groiss.wf.DefaultApplicationAdapter.getNotificationMail (User, ActivityInstance, String)	Use message templates to create your notification.
com.groiss.org.LogEntry.getAgent()	Use LogEntry.getCreatedBy() instead
com.groiss.org.LogEntry.getWhen()	Use LogEntry.getCreatedAt() instead
com.groiss.org.LogEntry.getLoggedObject()	Use LogEntry.getVersion() instead

Table C.8: List of removed methods (1)

Removed	Alternative
com.groiss.smartclient.ClientUtil.userInput(String, Supplier<DialogConfiguration>)	Use com.groiss.smartclient.ClientUtil.userInput(String, Consumer<DialogConfiguration>) instead
Additional Info: Substitute () -> new DialogConfiguration() with dlg -> dlg in the second parameter. Before: ClientUtil.userInput("promptTest", () -> new DialogConfiguration()).setMessage("whats your name").setType(Type.PROMPT)); After: ClientUtil.userInput("promptTest", dlg -> dlg.setMessage("whats your name").setType(Type.PROMPT));	
com.groiss.smartclient.ObjectSelectProvider.isCrudEnabled()	Use com.groiss.smartclient.ObjectSelectProvider.allowModifications() instead
com.groiss.store.Persistent.getStore()	Use com.groiss.store.Store.getInstance() instead
com.groiss.util.ApplicationException.isWarning()	Check for level "WARN" returned by getLevel()
com.groiss.util.ApplicationException.setWarning()	Set level through constructor
com.groiss.util.TopLevelException.isWarning()	Check for level "WARN" returned by getLevel()
com.groiss.util.TopLevelException.setWarning()	Set level through constructor of implementing class, since this is an interface.
com.groiss.util.ApplicationException.getException()	Use getCause() instead
com.groiss.util.TopLevelException.getException()	Use getCause() instead
com.groiss.org.OrgData.addRule(PermissionMapping)	Use OrgData.addRule(Class<? extends PermissionMapping>) instead.
Additional Info: Same in OrgDataImpl.	
com.groiss.org.OrgData.get(Class<? extends P>, long)	Use Store.get(Class<? extends P>, long) instead.
Additional Info: Same in OrgDataImpl.	
com.groiss.org.OrgData.get(String, long)	Use Store.get(String, long) instead.
Additional Info: Same in OrgDataImpl.	

Table C.9: List of removed methods (2)

Removed	Alternative
com.groiss.org.OrgData.getApplication(long)	Use Store.get(Class<? extends P>, long) with Application.class as first parameter instead.
Additional Info: Same in OrgDataImpl.	
com.groiss.org.OrgData.getOrgUnit(long)	Use Store.get(Class<? extends P>, long) with OrgUnit.class as first parameter instead.
Additional Info: Same in OrgDataImpl.	
com.groiss.org.OrgData.getRight(long)	Use Store.get(Class<? extends P>, long) with Right.class as first parameter instead.
Additional Info: Same in OrgDataImpl.	
com.groiss.org.OrgData.getRole(long)	Use Store.get(Class<? extends P>, long) with Role.class as first parameter instead.
Additional Info: Same in OrgDataImpl.	
com.groiss.org.OrgData.getUser(long)	Use Store.get(Class<? extends P>, long) with User.class as first parameter instead.
Additional Info: Same in OrgDataImpl.	
com.groiss.org.OrgData.list(Class<? extends P>, String, String)	Use Store.list(Class<? extends P>, String, String) instead.
Additional Info: Same in OrgDataImpl.	
com.groiss.org.OrgData.list(Class<? extends P>, String, String, Object...)	Use Store.list(Class<? extends P>, String, String, Object...) instead.
Additional Info: Same in OrgDataImpl.	
com.groiss.org.OrgData.list(String, String, Object...)	Use Store.list(String, String, Object...) instead.
Additional Info: Same in OrgDataImpl.	

Table C.10: List of removed methods (3)

Removed	Alternative
com.groiss.org.OrgData.listWithRightCheck(User, Class<? extends P>, String, Object[], Right, Application, boolean)	Use OrgData.listWithRightCheck(User, Class<? extends P>, Right, Application, boolean, String, Object...) instead.
Additional Info: Same in OrgDataImpl.	
com.groiss.org.OrgData.listWithRightCheck(User, String, String, Object[], Right, Application, boolean)	Use OrgData.listWithRightCheck(User, String, Right, Application, boolean, String, Object...) instead.
Additional Info: Same in OrgDataImpl.	
com.groiss.util.Settings.log(String, int)	Use own Logger instead.
Additional Info: Get the logger: static final Logger logger = LoggerFactory.getLogger(Class); Log: logger.level(msg) Where level is one of: error=0, info=1, debug=2, trace=3, warn=new. These are the old level codes mapped to the new levels.	
com.groiss.util.Settings.logError(Throwable)	Use own Logger instead.
Additional Info: Get the logger: static final Logger logger = LoggerFactory.getLogger(Class); Log: logger.error(msg, throwable);	
com.groiss.util.Settings.logError(Throwable, String)	Use own Logger instead.
Additional Info: Get the logger: static final Logger logger = LoggerFactory.getLogger(Class); Log: logger.error(msg, throwable);	
com.groiss.util.Settings.logError(Throwable, String, int)	Use own Logger instead.
Additional Info: Get the logger: static final Logger logger = LoggerFactory.getLogger(Class); Log: logger.level(msg, throwable); Where level is one of: error=0, info=1, debug=2, trace=3, warn=new.	
com.groiss.util.Settings.getLevel()	Use own Logger instead.
Additional Info: Instead of getting an integer back and checking that value, use logger.isLevelEnabled() and check the boolean.	

Table C.11: List of removed methods (4)

Removed	Alternative
com.groiss.util.Settings.setBaseDir(String)	No alternative
com.groiss.util.StringUtil.getCharset()	Use java.nio.charset.StandardCharsets.UTF_8.name() instead.
com.groiss.dms.DMS.listContents(DMSFolder, FormType, String, String, Object[], boolean)	Use DMS.listContents(DMSFolder, FormType, boolean, String, Object...) instead.
com.groiss.dms.DMS.disableRightCheck()	Use PersistentAspect.add(PersistentAspect...) with PersistentAspect.ALL_PERMISSIONS as a parameter instead.
Additional Info: If used with try-with-resources, you do not need to concern yourself with removing the aspect: try (var _ = PersistentAspect.add(PersistentAspect.ALL_PERMISSIONS)) { /* DMS calls */ }	
com.groiss.dms.DMS.enableRightCheck()	Use PersistentAspect.remove(PersistentAspect...) with PersistentAspect.ALL_PERMISSIONS as a parameter instead.
Additional Info: If the above described try-with-resources approach was used, there is no need to call this method manually.	
com.groiss.dms.DMS.checkEdit(User, DMSObject)	Use OrgData.checkRight(Right, Persistent) instead.
com.groiss.dms.DMS.checkView(User, DMSObject)	Use OrgData.checkRight(Right, Persistent) instead.
com.groiss.dms.DMS.checkDelete(User, DMSObject)	Use OrgData.checkRight(Right, Persistent) instead.
com.groiss.dms.DMS.mayEdit(User, DMSObject)	Use OrgData.mayUpdate(User, Persistent) instead.
com.groiss.dms.DMS.mayView(User, DMSObject)	Use OrgData.mayView(User, Persistent) instead.
com.groiss.dms.DMS.mayDelete(User, DMSObject)	Use OrgData.mayDelete(User, Persistent) instead.
com.groiss.dms.DMSDocForm.checkin(User, byte[])	Use DMS.setContent(DMSDocForm, byte[]) instead.
com.groiss.dms.DMSDocForm.checkin(User, InputStream)	Use DMS.setContent(DMSDocForm, InputStream) instead.
com.groiss.dms.DMSDocForm.checkin(User, long, byte[])	Use DMS.setContent(DMSDocForm, InputStream, long, byte[]) instead.
com.groiss.dms.DMSDocForm.getStatus()	Use DMSDocForm.isLocked() instead.

Table C.12: List of removed methods (5)

Removed	Alternative
<code>com.groiss.dms.DMSDocForm.lock()</code>	Use <code>DMS.lock(DMSDocForm)</code> instead.
<code>com.groiss.dms.DMSDocForm.unlock()</code>	Use <code>DMS.unlock(DMSDocForm)</code> instead.
<code>com.groiss.component.BeanManager.bindThreadToTransaction(UserTransaction)</code>	No alternative
<code>com.groiss.component.BeanManager.setExplicitTransactionMode(boolean)</code>	No alternative
<code>com.groiss.store.StoreUtil.getDefaultStore()</code>	Use <code>Store.getInstance()</code> instead.
<code>com.groiss.util.FileUtil.deleteDir(String)</code>	Use <code>FileUtil.deleteDir(File)</code> instead.
Additional Info: Change the call to <code>FileUtil.deleteDir(FileUtil.getDefaultStore().getFileFromBasePath(filename))</code> , or <code>FileUtil.deleteDir(FileUtil.getDefaultStore().getFileFromVarPath(filename))</code> , depending on where the file is located.	
<code>com.groiss.util.FileUtil.getContent(String)</code>	Use <code>FileUtil.getContent(File)</code> instead.
Additional Info: Change the call to <code>FileUtil.getContent(FileUtil.getDefaultStore().getFileFromBasePath(filename))</code> , or <code>FileUtil.getContent(FileUtil.getDefaultStore().getFileFromVarPath(filename))</code> , depending on where the file is located.	
<code>com.groiss.util.FileUtil.getFileFromPath(String)</code>	Use one of the search-root-specific methods <code>FileUtil.getFileFromVarPath(String)</code> , <code>FileUtil.getFileFromLocalPath(String)</code> , or <code>FileUtil.getFileFromBasePath(String)</code> .
<code>com.groiss.util.FileUtil.getNoCommentContent(String)</code>	Use <code>FileUtil.getNoCommentContent(File)</code> instead.
Additional Info: Change the call to <code>FileUtil.getNoCommentContent(FileUtil.getDefaultStore().getFileFromBasePath(filename))</code> , or <code>FileUtil.getNoCommentContent(FileUtil.getDefaultStore().getFileFromVarPath(filename))</code> , depending on where the file is located.	

Table C.13: List of removed methods (6)

Removed	Alternative
com.groiss.store.PersistentObject.onInsert()	Call/Ovenwrite com.groiss.store.PersistentObject.beforeInsert() instead
com.groiss.store.PersistentObject.onUpdate()	Call/Ovenwrite com.groiss.store.PersistentObject.beforeUpdate() instead
com.groiss.store.PersistentObject.onDelete()	Call/Ovenwrite com.groiss.store.PersistentObject.beforeDelete() instead
com.groiss.store.PersistentObject.onRead()	Call/Ovenwrite com.groiss.store.PersistentObject.afterRead() instead
com.groiss.wf.WfEngine.getStartableProcesses(Application)	Use getStartableProcDefs(Application) instead.
com.groiss.wf.WfEngine.gotoTask(ActivityInstance, String, Agent, OrgUnit, String)	Use gotoStep(ActivityInstance, String, Agent, OrgUnit, String) instead.
com.groiss.wf.WfEngine.gotoTask(ProcessInstance, String, Agent, String)	Use gotoStep(ActivityInstance, String, Agent, OrgUnit, String) instead.
com.groiss.wf.WfEngine.makeBranch(ProcessInstance, Task, Agent, OrgUnit, List<String>)	Use makeBranch(ProcessInstance, Task, Agent, OrgUnit) instead.
com.groiss.gui.AbstractComponent.setEncoding(boolean)	No alternative
Additional Info: Method does nothing.	
com.groiss.gui.AbstractComponent.setExpandEmptyElements(boolean)	No alternative
Additional Info: Method does nothing.	
com.groiss.gui.AbstractComponent.setIndenting(boolean)	No alternative
Additional Info: Method does nothing.	
com.groiss.gui.AbstractComponent.setNewLines(boolean)	No alternative
Additional Info: Method does nothing.	

Table C.14: List of removed methods (7)

Removed	Alternative
com.groiss.component.Configuration.demansch(String)	No alternative
com.groiss.component.Configuration.getProperty(String, String)	Define default values in the XML description of the configuration.
com.groiss.component.Configuration.mansch(String)	No alternative
com.groiss.component.Configuration.setSystemProps()	No alternative
com.groiss.ds.MultiMap.getEntries(Object)	Use MultiMap.get(Object) instead.
com.groiss.ds.MultiMap.keys()	Use MultiMap.keySet() instead.
com.groiss.gui.component.SelectList.deleteOption(String, String)	Use SelectList.deleteOption(String) instead.
com.groiss.gui.component.TableContainer.getColumnAttributes(int)	No alternative
Additional Info: This method always returned null.	
com.groiss.ds.TwoLevelMap.firstKeySet()	Use TwoLevelMap.keySet() instead.
com.groiss.gui.HTMLPage.setMask(String)	Use HTMLPage.setPage(String) instead.
com.groiss.gui.table.Table.showToolBarIcons()	Use Table.getToolBarShape() instead.
Additional Info: Interface method. Use implementation.	
com.groiss.gui.table.TableAdapter.showToolBarIcons()	Use TableAdapter.getToolBarShape() instead.
com.groiss.reporting.data.impl.TimeInterval.setValue(Object)	No alternative
com.groiss.store.BulkQuery.execute(String, int)	Use BulkQuery.executeSet(String, int) instead.
com.groiss.store.BulkQuery.execute(String, int, Object...)	Use BulkQuery.executeSet(String, int, Object...) instead.
com.groiss.store.Store.getTable(String, Object...)	Use Store.getTable(String, Object[], int[]) with explicit type information instead.
com.groiss.store.Store.getValue(String, int)	Use Store.getValue(int, String, Object...) instead.
Additional Info: This is one of the methods that should be watched out for when migrating.	

Table C.15: List of removed methods (8)

Removed	Alternative
<code>com.groiss.store.Store.getValue(String, Object[], int)</code>	Use <code>Store.getValue(int, String, Object...)</code> instead.
Additional Info: This is one of the methods that should be watched out for when migrating.	
<code>com.groiss.store.Store.list(Class<? extends P>, String, String, Object[], int)</code>	Use <code>Store.limitedList(Class, int, String, String, Object...)</code> instead.
Additional Info: This is one of the methods that should be watched out for when migrating.	
<code>com.groiss.store.Store.list(Class<? extends P>, String, String, Object[], int, int)</code>	Use <code>Store.limitedList(Class, int, String, String, Object...) instead.</code>
Additional Info: This is one of the methods that should be watched out for when migrating.	
<code>com.groiss.wf.html.HTMLUtils.agentSelectList(HttpServletRequest)</code>	Use <code>ep/widget/smartclient/form/ObjectInputField widget instead.</code>
<code>com.groiss.wf.html.HTMLUtils.getDocumentLink(ActivityInstance, String)</code>	Use <code>HTMLUtils.getDocumentLink(ActivityInstance) instead.</code>
Additional Info: Parameter <code>comingFrom</code> is not supported anymore.	
<code>com.groiss.wf.html.HTMLUtils.roleList(HttpServletRequest)</code>	No alternative. It was only designed to be used by <code>HTMLUtils.selectUserRole(HttpServletRequest)</code> , which has also been removed.
<code>com.groiss.wf.html.HTMLUtils.selectTable(HttpServletRequest)</code>	Use <code>com.groiss.wf.html.HTMLUtils.selectList(HttpServletRequest)</code>
<code>com.groiss.wf.html.HTMLUtils.selectUserRole(HttpServletRequest)</code>	Use <code>ep/widget/smartclient/form/ObjectInputField widget instead.</code>
<code>com.groiss.wf.html.HTMLUtils.showEmptyForm(DMSForm, String, HttpServletRequest)</code>	No alternative.
<code>com.groiss.wf.html.HTMLUtils.showForm(HttpServletRequest, ActivityInstance, String, int, String)</code>	No alternative.
<code>com.groiss.wf.html.HTMLUtils.showWorklist(HttpServletRequest, HttpServletResponse, Application, int, String, String)</code>	No alternative.

Table C.16: List of removed methods (9)

Removed	Alternative
com.groiss.wf.html.HTMLUtils.toJSON(Persistent)	Use StoreUtil.toJsonAsReference(Persistent) or StoreUtil.toJson(Persistent) instead.
com.groiss.wf.html.HTMLUtils.userList(HttpServletRequest)	No alternative. It was only designed to be used by HTMLUtils.selectUserRole(HttpServletRequest), which has also been removed.
com.groiss.servlet.HttpUtils.setNoCache(HttpServletResponse)	Use ServletUtils.setNoCache(HttpServletResponse) instead.
com.groiss.servlet.ServletUtils.setLocale(HttpServletRequest)	Use ServletUtils.setLocale(HttpServletResponse, Principal) instead.
com.groiss.wf.html.Worklist.getHTMLPage()	No alternative.
com.groiss.wf.batch.BatchManager.exceptionHandlingFinished(String)	No alternative.
com.groiss.wf.batch.BatchManager.exceptionHandlingFinished(String)	com.groiss.wf.batch.impl.BatchSupport.exceptionHandlingFinished(String)
com.dec.avw.core.Util.createFormCondition(Class<?>, String, boolean)	No alternative.
com.dec.avw.core.Util.createFormCondition(String, String, boolean)	No alternative.
com.dec.avw.core.Form.getVersionsForDistribution()	No alternative.
com.dec.avw.core.Form.getVersionsForDistribution(PersistentVersion, List<Object>, List<Object>)	No alternative.
com.dec.avw.core.StepInstance.getHomeServer()	No alternative.
com.dec.avw.core.StepInstance.setHomeServer(Server)	No alternative.
com.dec.avw.core.StepInstance.getDependent()	No alternative.
com.dec.avw.core.StepInstance.getProcessData()	No alternative.
com.dec.avw.core.StepInstance.getProcessData(User)	No alternative.

Table C.17: List of removed methods (10)

Removed	Alternative
com.dec.avw.core.StepInstance.getDocsForDistribution(User)	No alternative.
com.dec.avw.core.StepInstance.getDocsForDistribution(User, DMSFolder)	No alternative.
com.groiss.wf.ApplicationAdapter.getPacker()	No alternative.
com.groiss.dms.DMSTableAdapter.modifyTableHeader(List<Pair<String, Object>>)	Use DMSTableHandler.modifyColumns(List<ColumnDescription>) instead.
com.groiss.dms.FormEventAdapter.onInsert(T extends DMSForm)	Use FormEventHandler.beforeInsert(T extends DMSForm) instead.
com.groiss.dms.FormEventAdapter.onUpdate(T extends DMSForm)	Use FormEventHandler.beforeUpdate(T extends DMSForm) instead.
com.groiss.dms.FormEventAdapter.onDelete(T extends DMSForm)	Use FormEventHandler.beforeDelete(T extends DMSForm) instead.
com.groiss.dms.FormTableAdapter.modifyTableHeader(List<Pair<String, Object>>)	Use FormTableHandler.modifyColumns(List<ColumnDescription>) instead.
com.groiss.dms.XHTMLFormEventAdapter.onInsert(T extends DMSForm)	Use XHTMLFormEventHandler.beforeInsert(T extends DMSForm) instead.
com.groiss.dms.XHTMLFormEventAdapter.onUpdate(T extends DMSForm)	Use XHTMLFormEventHandler.beforeUpdate(T extends DMSForm) instead.
com.groiss.dms.XHTMLFormEventAdapter.onDelete(T extends DMSForm)	Use XHTMLFormEventHandler.beforeDelete(T extends DMSForm) instead.
com.groiss.cal.CallInfoAdapter.listEvents(Persistent, Date, Date)	Use CalInfo.listEvents(List<Persistent>, Date, Date) instead.
com.groiss.Server.ServerInfo.getConcurrentUserSessionCount()	No alternative
Additional Info: The concept of “concurrent sessions” does not exist anymore.	
com.groiss.Server.ServerInfoMXBean.getConcurrentUserSessionCount()	No alternative
Additional Info: The concept of “concurrent sessions” does not exist anymore.	

Table C.18: List of removed methods (11)

Removed	Alternative
com.groiss.wf.html.WorkListAdapter.init(HttpServletRequest, int, Application, User)	Use Worklist.init(HttpServletRequest, WorklistDescription, User) instead.
com.groiss.wf.html.WorkListAdapter.modifyTableHeader(List<Pair<String, Object>>)	Use Worklist.modifyColumns(List<ColumnDescription>) instead.
com.groiss.wf.html.WorkListAdapter.modifyTableLine(ActivityInstance, List<Object>)	Use Worklist.modifyTableLine(ActivityInstance, Map<String, Object>) instead.
com.groiss.gui.JavaScriptLoader.getScripts(HttpServletRequest, HttpServletResponse)	No alternative.
Additional Info: References to this method must also be removed in HTML pages.	
com.groiss.server.Admin.setLoginAllowed(String, boolean, String, int, String)	Use setLoginAllowed(String, boolean, String) instead.
Additional Info: Handling of current sessions and successor node were only relevant for RMI clients.	
com.groiss.calender.pers.CalendarImpl.upgradeWholeDayEvents(PrintWriter)	No alternative.
Additional Info: Was only used in patch files from ep8.0 to ep9.0.	
com.groiss.htmladmin.ConfigurationEditor.upgradePasswords(PrintWriter)	No alternative.
Additional Info: Was only used in patch files from ep8.0 to ep9.0.	
com.groiss.htmladmin.UpgradeGuiConfig.convertFolderProps(PrintWriter)	No alternative.
Additional Info: Was only used in patch files from ep8.0 to ep9.0.	
com.groiss.notification.BasicNotificationItem.getTopic()	Implement the getTopic method for every class extending com.groiss.notification.BasicNotificationItem.
Additional Info: Always returned null. Empty topics can not be created anymore.	

Table C.19: List of removed methods (12.0)

D. Removed database tables and columns

D.1 Removed database tables

Affected Table	Additional Info
avw_archivedproc	Not used anymore after removing online archive
avw_lastupdate	Not used anymore after removing build time data synchronisation
avw_wfxml2accesslog	Not used anymore after removing WfXML2
avw_wfxml2asynclistener	Not used anymore after removing WfXML2
avw_wfxml2intobservers	Not used anymore after removing WfXML2
avw_wfxml2observers	Not used anymore after removing WfXML2
avw_wfxml2pd	Not used anymore after removing WfXML2
avw_wfxml2pendingmsg	Not used anymore after removing WfXML2
avw_wfxml2remoteinstance	Not used anymore after removing WfXML2
avw_wfxml2rpd	Not used anymore after removing WfXML2
avw_wfxmlpartner	Not used anymore after removing WfXML2

Table D.1: Removed database tables

D.2 Removed table columns

D. Removed database tables and columns

Affected Table	Removed Columns	Additional Info
avw_classrightrel avw_formtype	avwusage icon, prototype_class, prototype	Not used anymore. Define icons via CSS; prototypes are replaced by template handling
avw_runningnode	currentsessions, successornodeid	Were only relevant for RMI clients.
avw_server	voyagerport	Port is not necessary anymore after removing RMI
avw_stepinstance avw_userrole	homeserver representant, fromdate, todate	Not used anymore. Role representations are handled differently since many versions

Table D.2: Removed table columns