7.2.20 Web Platform Release Notes
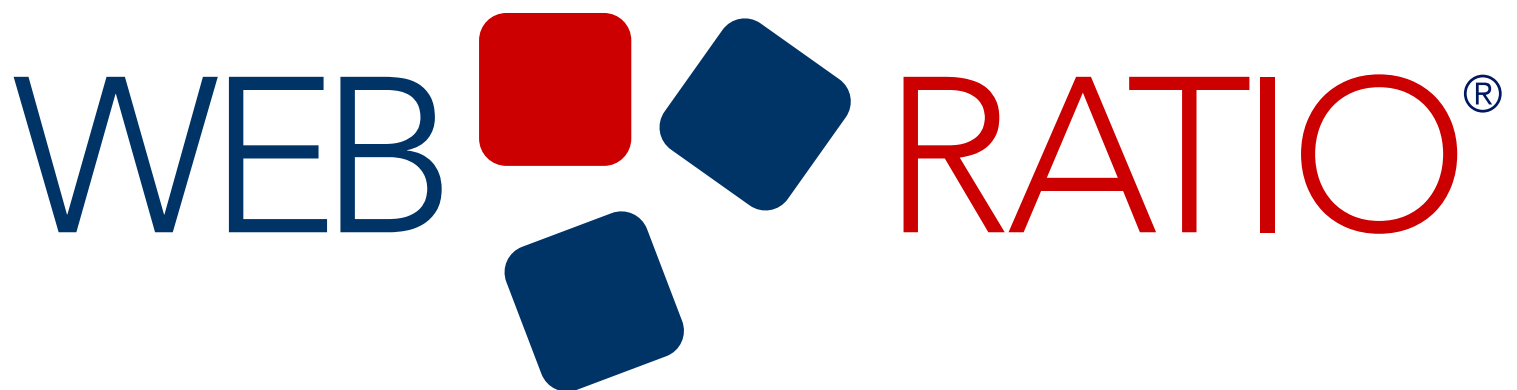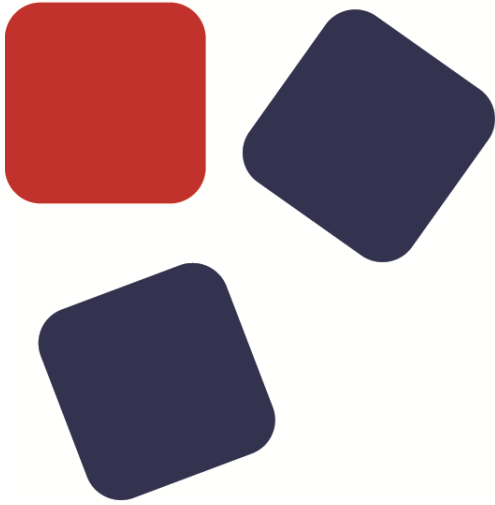
WEB RATIO®

RELEASE NOTES

Copyright © 2022 WebRatio s.r.l – All rights reserved.

TABLE OF CONTENTS

IMPORTANT NOTICE

WebRatio Platform 7.2.20 is a Security Release. We recommend upgrading to 7.2.20 to ensure the security of your web applications and data. This release includes some moderately critical fixes. The main one is the **upgrade of log4j** library to log4j2 due to the several vulnerabilities that came out in this December. Several other non-security issues have also been fixed.

Applications generated with WebRatio Platform 7.2.20 now require the Java Runtime Environment (JRE) 1.8 or higher.

WADL GENERATION

New in 7.2.6

WebRatio introduced the WADL generation for all modeled REST Web Services just in the same way for the WSDL generation for SOAP Web Services. WADL means Web Application Description Language (WADL). The official website http://www.w3.org/Submission/wadl/ describes the WADL as follows:

"An increasing number of Web-based enterprises (Google, Yahoo, Amazon, Flickr to name but a few) are developing HTTP-based applications that provide programatic access to their internal data. Typically these applications are described using textual documentation that is sometimes supplemented with more formal specifications such as XML schema for XML-based data formats. WADL is designed to provide a machine process-able description of such HTTP-based Web applications."

The URL at which you can find the generated WADL definition is composed in this way:

http://<host>:<port>/<application-name>/wadl.do

You can see in the following image an example of the generated WADL.

REST WEB SERVICE INVOCATION USING WADL

WebRatio introduced the possibility to model a REST Web Service invocation using its WADL definition. If the web service provider you are using for you web project expose WADL for the REST Web Service you can use this definition to easily model the invocation to a REST web service method. The strategy used in WebRatio Platform is similar to the one applied to the SOAP Web Services invocation through REST. You can add a Web Service Provider referencing the WADL definition and set the "Resource Type" to WADL.
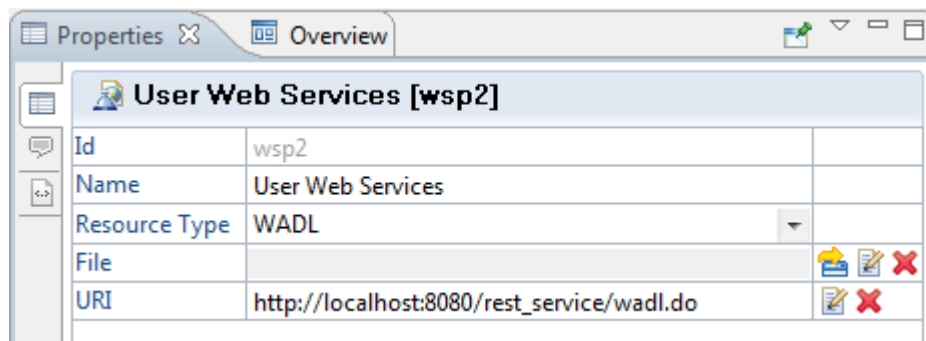
You can also add an XSD Resource referencing the WADL Definition, so that it's possible to synchronize the Domain Model with this resource and create the related set of entities and relationships.
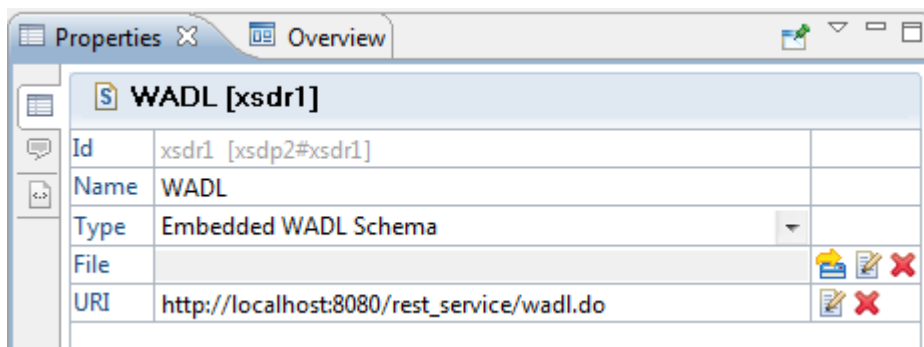
When you model the invocation of a REST Web Service Method using the Restful Request Response Operation you can easily choose between the methods listed in the WADL Definition. The operation configures itself correctly so that it's easier to model the call.

NOTE: If you do not have a WADL Definition at disposal you can still model REST Web Service invocation without it.

JSON I/O COMPONENT

 New in 7.2.6

WebRatio introduced the JSON I/O Component, which gives the ability to export or import an entity of the Domain Model according to the referenced WADL. If the entity is not mapped on a WADL it's possible to configure the JSON Document structure directly on the operation. The document generated by the operation is in JSON format.

SOLICIT OPERATION IMPROVEMENTS

New in 7.2.6

The Solicit Service Operation has been improved so that now it is possible to choose for the REST Web Service the JSON Content Type. In the Properties View, now you can choose between "XML" or "JSON" when you set in the "Invocation Style" property REST.
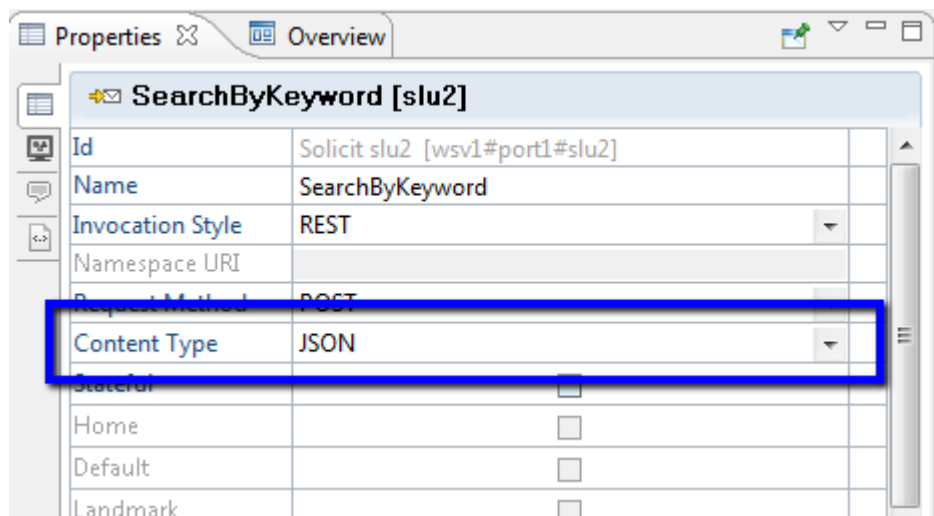
MAIL OPERATION IMPROVEMENTS

New in 7.2.5

The Mail Operation has been improved so that now it is possible to provide dynamically the information of the SMTP Server to be used. In this way the Web Application can accomplish some business logic in order to decide which SMTP Server use and then provide it to the Mail Operation.

Moreover it is now possible to send certified emails (PEC) through a secure SMTP Server. You just have to configure the SMTP Server properties and check the SSL flag.

SIMPLE CALCULATED ATTRIBUTES

New in 7.2.5

WebRatio introduced the possibility to add derived attributes having a simple expression for their derivation without the need to create a view in the persistent database. The derivation wizard now shows a new option, which is the "Simple Calculated Attribute" option. For all attributes specified using a concatenation of other attributes of the same entity or for simple calculations, you can now use this option.
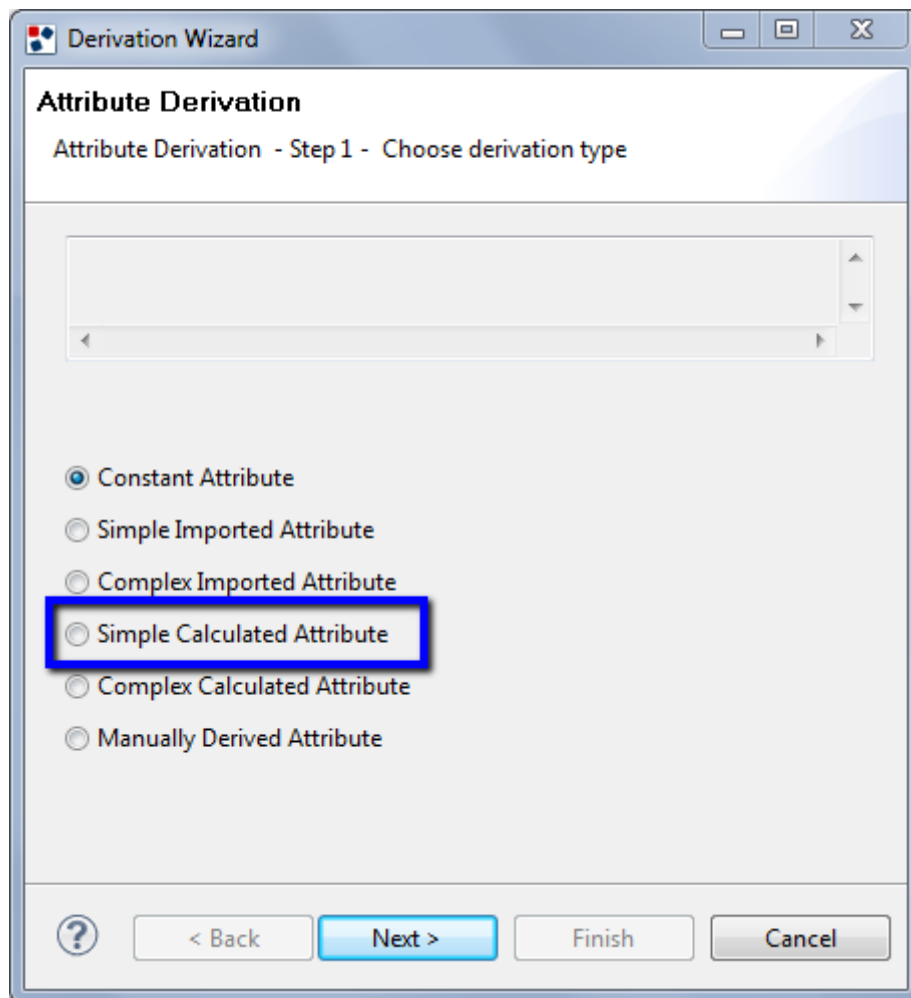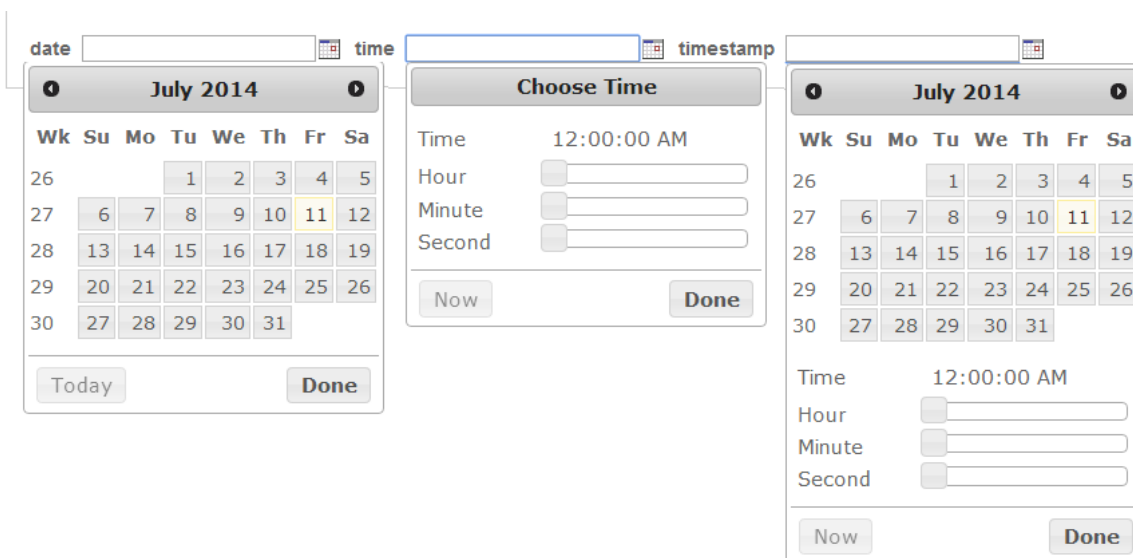
DYNARCH CALENDAR REMOVAL

New in 7.2.5

WebRatio changed the JavaScript widget used by default to render date, time and timestamp fields. In particular, the Dynarch Calendar library has been removed in favor of the new up-to-date jQuery UI Datepicker (http://jqueryui.com/datepicker/) and its Timepicker extension (http://trentrichardson.com/examples/timepicker/).In the following image, you can see the appearance of the new widgets. To learn how to migrate your custom style projects in order to be compliant with this change, please refer to the "Migrate from WebRatio 7.2.4 to WebRatio 7.2.5" section of the **WebRatio 7.2 Migration Guide** available at learn.webratio.com.

LOOP OPERATION "BREAK" FLOW

New in 7.2.5

The Loop Operation has been improved so that it's possible to specify a dedicated flow to be executed when the business logic needs to break the loop before its end. For this purpose, the Loop Operation now can have an additional OK Flow with "break" code. Refer to the following image for an example.
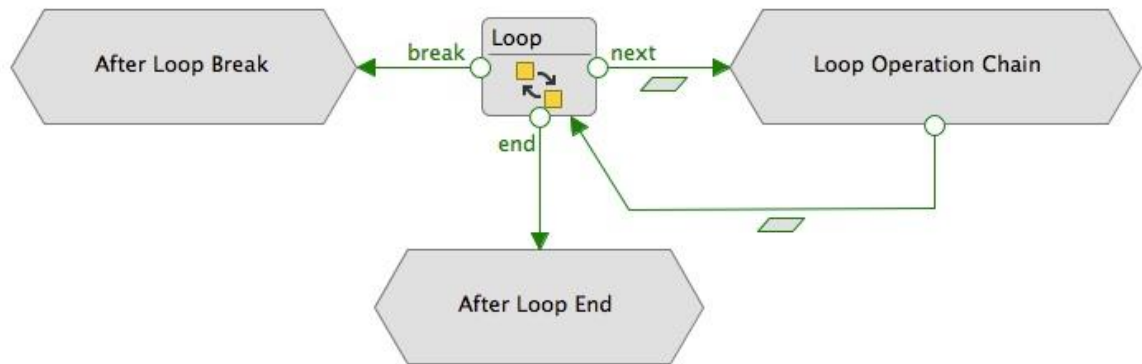
LOCAL STORAGE OF CLIENT RUNTIME LOG

New in 7.2.5

The Client Runtime Log contains all the log messages related to the execution of the AJAX framework requests and JavaScript methods. It's now possible to store this log on the local storage of the client web browser. In this way you have a persistent log at your disposal and the Web Application user can send the log to the administrator in case of problems with the Web Application. This is the suggested configuration for a deployed Web Application. To learn how to configure the Client Runtime Log in this way and to know which configuration options you have at your disposal, read the "**Configuring Client Runtime Logging**" article at this link. You can see in the following image what the user sees when an error occurs.

New Cloud Account

New in 7.2.4

WebRatio enriched the options available for deploying your Web Applications in the cloud. In the available cloud accounts list, you can now find the new WebRatio Account. Buy a WebRatio Application Cloud Plan on our website, get your security credentials, and deploy your Web Application in the cloud. Check out our WebRatio Application Cloud Plans offer and choose your preferred size. Then open WebRatio Platform and deploy your Web Application in the cloud.

Embedded Tomcat Update

New in 7.2.4

WebRatio updated the version of the Tomcat embedded in WebRatio Platform. Now you will find Tomcat 7.0.53, which solves several issues with the previous version. You can find detailed information about Tomcat 7.0.53 at this link: http://tomcat.apache.org/tomcat-7.0-doc/index.html

Apache Tomcat 7

Version 7.0.53, Mar 25 2014

BPM Style Removal

New in 7.2.3

WebRatio removed the BPM style from the set of predefined styles. This style was used as the default for BPM-based projects. Starting from the current WebRatio version, the WebRatio style is the default for all project types. So, BPM-based projects will now use the WebRatio style. The WebRatio style has been updated in order to be able to correctly handle the layout of the BPM components. You can see an example of the new layout for BPM project in the following image. To learn how to migrate existing projects, please refer to the **Migrate from WebRatio 7.2.2 to WebRatio 7.2.3** section of the **WebRatio 7.2 Migration Guide**.

WEB ◆ RATIO®

🏠 › Home

Login

Select one of the demo users to enter the application.

Username: [   ]   **Password:** [   ]   [Login]

Demo Users

	Username	Group	
›	John Doe	Employee	select
›	Scott Tiger	Employee	select
›	Ronald Still	Supervisor	select
›	Ray Herbert	Supervisor	select
›	Douglas Martin	Treasurer	select
›	Adam Hendricks	Treasurer	select
›	Admin	Administrator	select

RESTFUL REQUEST RESPONSE OPERATION

New in 7.2.3

WebRatio added a new operation to the Service Operations set that allows you to easily invoke a **Restful Web Service** based on JSON Documents exchange. What you need to know is the Web Service endpoint, and the structure request and response documents. You can configure the Restful Request Response Operation by adding request and response parameters to build the desired request and correctly handle the web service result.

Configured parameters are used to bind the input/output coupling parameter to a specific JSON Document node.
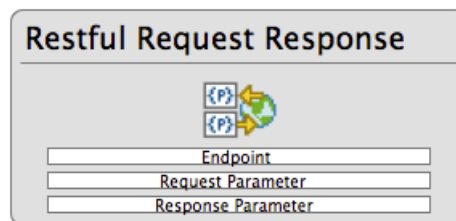
PUBLISH RESTFUL WEB SERVICE WITH JSON RESPONSE

New in 7.2.3

WebRatio improved the facilities used to model a Web Service by providing the ability to receive a JSON request, and to send a JSON response back to the requestor.

The Solicit Operation now discovers, at runtime, the kind of request (JSON or XML) by inspecting the request's content-type header.

The Response Operation now has an additional property that allows you to decide whether you want to send back an XML or a JSON response. You can also set the request method and response status code according to the implemented function.

EXPIRE STYLE PAGE

New in 7.2.2

WebRatio added the ability to handle, in a user-friendly way, what happens when a user triggers an action on the Web Application after the session is expired and the Web Application is not able to fulfill the request. This situation happens when the user leaves the browser idle for a long time or when the application server restarts for some reason. In this case, you may want to show the user a page explaining that the session is expired and that the user needs to restart the navigation (for example, starting from the home page). You can see a sample of the page in the following screenshot.
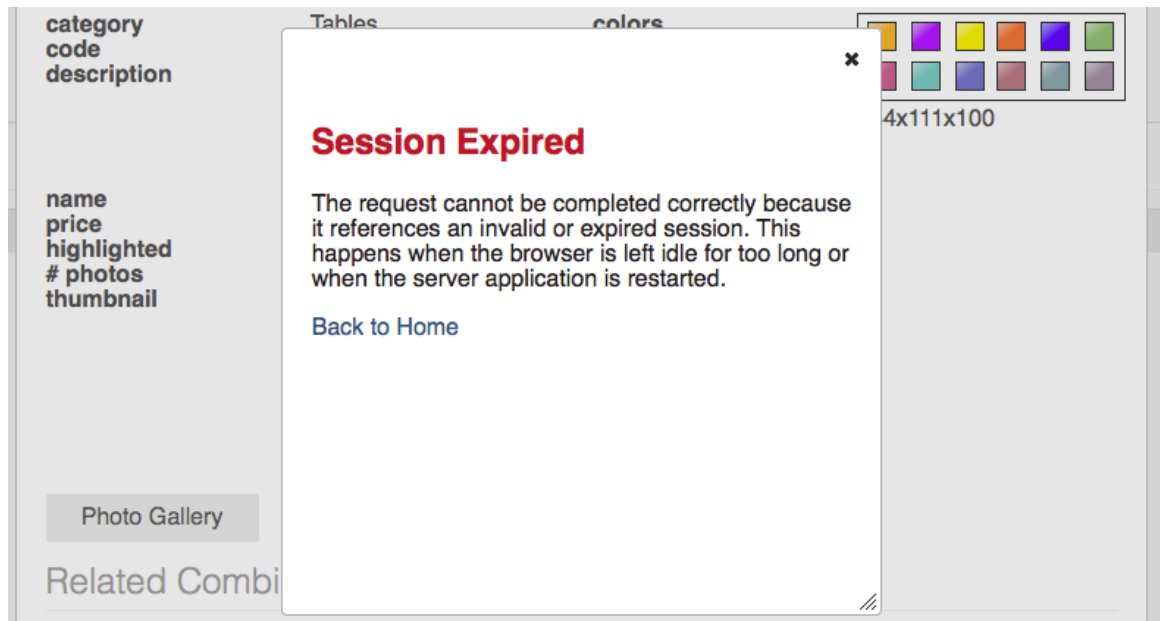
This page can be customized using the **Expire Style Page** property in the Layout Tab of the Properties View. By default you have a page similar to the one shown screenshot, but if you prefer you can create your own Expire Style Page in your Style Project and then set the corresponding property.
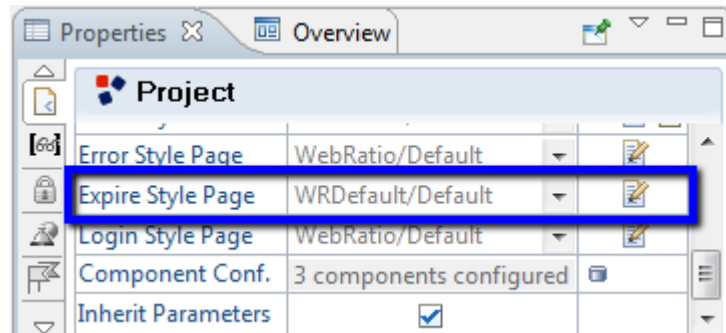
WRDefault Style Changes
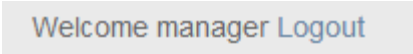
The WRDefault style has been changed in order to make easier the process of changing the style to another one and to be used easier. These changes include:

1. The "Localized.unit.template" of the Message Component has been removed.

   This template was applied on a Message Component that has a localized message. If you changed the message to a non-localized one, you needed to select the "Normal.unit.template" as the Component Layout in order to make things work. The "Normail.unit.template" now has an additional layout parameter "Message Localized" to handle the localized message.

   Refer to the "Migrate from WebRatio 7.2.0 to WebRatio 7.2.1" section **WebRatio 7.2 Migration Guide** to learn how to change your web project in order to remove all references to the obsolete "Localized.unit.template".

2. A new layout template for the Details Component has been added. This template is named "User.unit.template" and can be used to show all the information related to the user currently logged in your Web Application. You can see an example of the template result in the following image.

   Welcome manager Logout

   The "Welcome" text can be changed to a custom one using the "Prefix Text" Layout Parameter. Refer to the *Web Modeling User Guide -> Reference -> WR Default* section of the **Product Documentation** for further information.

3. The rich text editor plugin used for showing field of type text/html has been updated to the latest release. Now the WRDefault style includes the CKEditor 4.3. This new version of the plugin allows to handle correctly the case in which the field is set as not modifiable.

   Refer to the "Migrate from WebRatio 7.2.0 to WebRatio 7.2.1" section **WebRatio 7.2 Migration Guide** to learn how to migrate your custom field templates.

WEBRATIO STYLE CHANGES

The WebRatio style is the style applied by default to all Web Projects. This style helps you model and test quickly your Web Application without focus on the graphic. The WebRatio style has been changed in order to be compliant with the new WebRatio visual identity. A Web Project generated with this version of the WebRatio style looks like the following image.
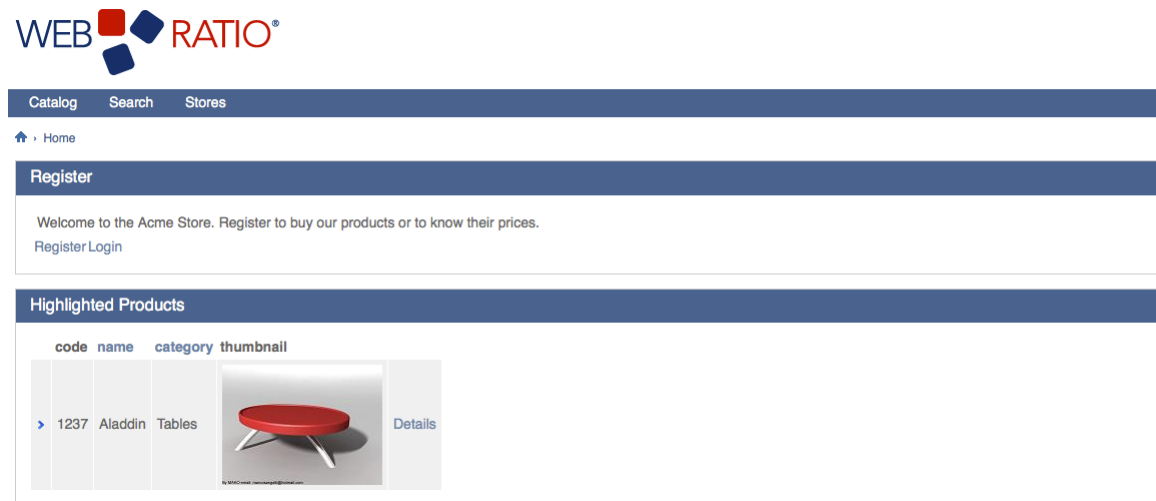
Moreover, in order to be able to change easily the style applied on a Web Project to another style downloaded from the WebRatio Store, the list of the custom locations available for this style has been changed. The available Custom Locations are: "User", "Page Links", "Top", "Left" and "Right".

Refer to the "Migrate from WebRatio 7.2.0 to WebRatio 7.2.1" section **WebRatio 7.2 Migration Guide** to learn how to change your Web Project in order to use the new Custom Locations.

BPM STYLE CHANGES

The BPM style which is automatically set on Web Project connected to a BPM Project has been changed in order to be compliant with the new JavaScript framework adopted by WebRatio 7.2. The framework used is JQuery, so the BPM style has been improved removing the usage of the old Prototype framework.

CACHE OF CSS AND JAVASCRIPT RESOURCES

The cache is able to understand if a static resource has been changed or not and to load the resource only when there are changes.

This approach allows to reduce the time necessary in order to load a page of the Web Application on the browser, since the static resources are downloaded once and then cached.

Moreover, when you deploy a new version of the Web Application, it's not necessary anymore to force the users deleting the browser cache in order to see the changes done on the application graphic. The cache system is able to recognize that the resources are different from the previous ones and then the new version are downloaded and used to render the page to the user.

NEW SUPPORTED DATABASE TYPES

WebRatio 7.2 now supports Microsoft SQL Server 2008 and Microsoft SQL Server 2012. You can choose one of these database types when creating a new database connection in your Web Project.

COOKIE STORAGE PROVIDER

WebRatio 7.2 lets you define on a Web Project a new Service/Data Provider which is named **Cookie Storage Provider**. This provider lets you configure some cookie properties such as the name, the domain, the timeout and an optional encryption key. See the following image for an example.

In the Web Project you can decide which information must be stored inside a cookie by associating context parameters to a certain Cookie Storage Provider. You can refer to a provider setting the related property in the Properties View of the Context Parameter. The following image shows you an example.
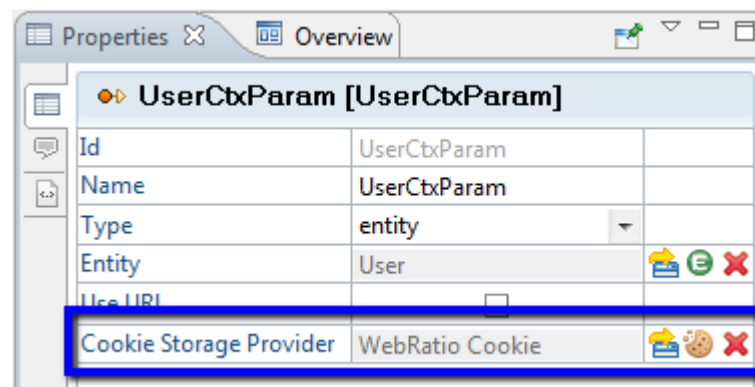
SUMMARY PAGE FOR PUBLISHED REST WEB SERVICES

WebRatio 7.2 now supports the generation of a specific section of the Web Application which contains a summary of all the Web Services Methods exposed using REST. This page contains the list of methods with an example of the URL to use for calling the method itself.

Moreover it's possible to include a description for each method and for each parameter required by the web service method. It's also possible to see an example of the provided response. You can see an example of the summary page in the following image.

ManageEmployee

The "ManageEmployee" method allows to create or update an employee. In order to create a new employee you should pass to the web service all the information about the employees. In order to update an existing employee you must provide also the employee primary key. The web service gives you back all the employees information. This method does not allow you to delete employees. Use the method "DeleteEmployee" for this purpose.

Example URL

```
GET   http://localhost:8080/PublishingWebServices/RESTWebServices/Employees/ManageEmployee.do?firstname={VALUE}&lastname={VALUE}&birthdate=
```

Input Parameters

Name	Type	Notes
firstname	xsd:string	The employee first name. It's mandatory.
lastname	xsd:string	The employee last name. It's mandatory.
birthdate	xsd:date	The employee birthdate. It's mandatory. The format of the date must be yyyy-MM-dd.
email	xsd:string	The employee email. It's optional. Provide a valid email.
sex	xsd:string	The employee sex. Available options are: male or female. It's optional.
employeeId	xsd:int	The employee primary key. It's mandatory if you want to update the information about an exising employee.

Output Parameters

Name	Type	Notes
employees	{urn:common.model}employeesType	The created or updated employee information.

Sample Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ManageEmployeeResponse>
  <employees>
    <employeeId>859770437</employeeId>
    <firstname>xyzzy</firstname>
    <lastname>xyzzy</lastname>
    <email>xyzzy</email>
    <birthdate>2013-12-18</birthdate>
    <photo>ampXeVRBajZBU2xmdWJRbktrSzlzVFY0S1BTN294</photo>
    <sex>xyzzy</sex>
  </employees>
</ManageEmployeeResponse>
```

The generation of this page must be explicitly enabled. There's a property you can use for this purpose which is located in the Project Properties View. Refer to the following image.
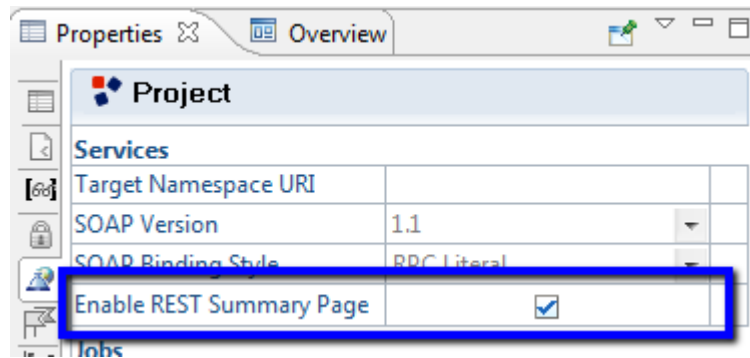
In order to include a description in the page, you have to add comments on the model elements composing the web service method. Comments can be written in the Properties View, selecting the "Comment" tab.

INTRODUCING IFML NOTATION

WebRatio 7.2 introduces the first steps for IFML compliance. IFML is the new OMG standard language for modeling the application user interface. IFML, the result of the collaboration between WebRatio and the OMG, is now available as a beta version.

The Web Project Editor has been changed in order to be compliant with the IFML notation in terms of symbols and colors.

If you are familiar with WebML, please read the document **Differences between the WebML and IFML notation** - stored in the Learning Center - to find out the differences between WebML and IFML and to become familiar with the new notation.

HELP REORGANIZATION

The main menu Help option has been reorganized to show just a few options. The central place for documentation about how to effectively use WebRatio is the Learning Center.

Help		
📖	Learning Center	
⑦	WebRatio Product Documentation	
🔍	Search	
	Dynamic Help	
	Key Assist...	Ctrl+Shift+L
	Tips and Tricks...	
	Cheat Sheets...	
🛒	WebRatio Store...	
🔑	License Information	
🔑	Reset License Activation	
	Check for Updates	
	Install New Software...	
	About WebRatio Enterprise	
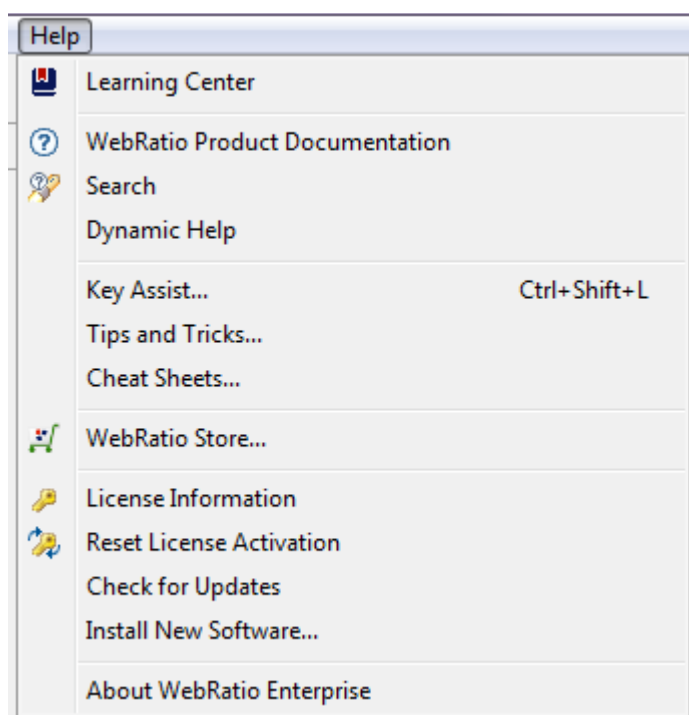
LEARNING CENTER

Discover the learning center where you can access information that increases every day about WebRatio. Watch the brand-new tutorial to get started, or download the up-to-date sample projects from the WebRatio Store. Become a WebRatio Expert by taking advantage of the WebRatio Knowledge Base, which has been enriched with new video tutorials (coming soon) and articles.
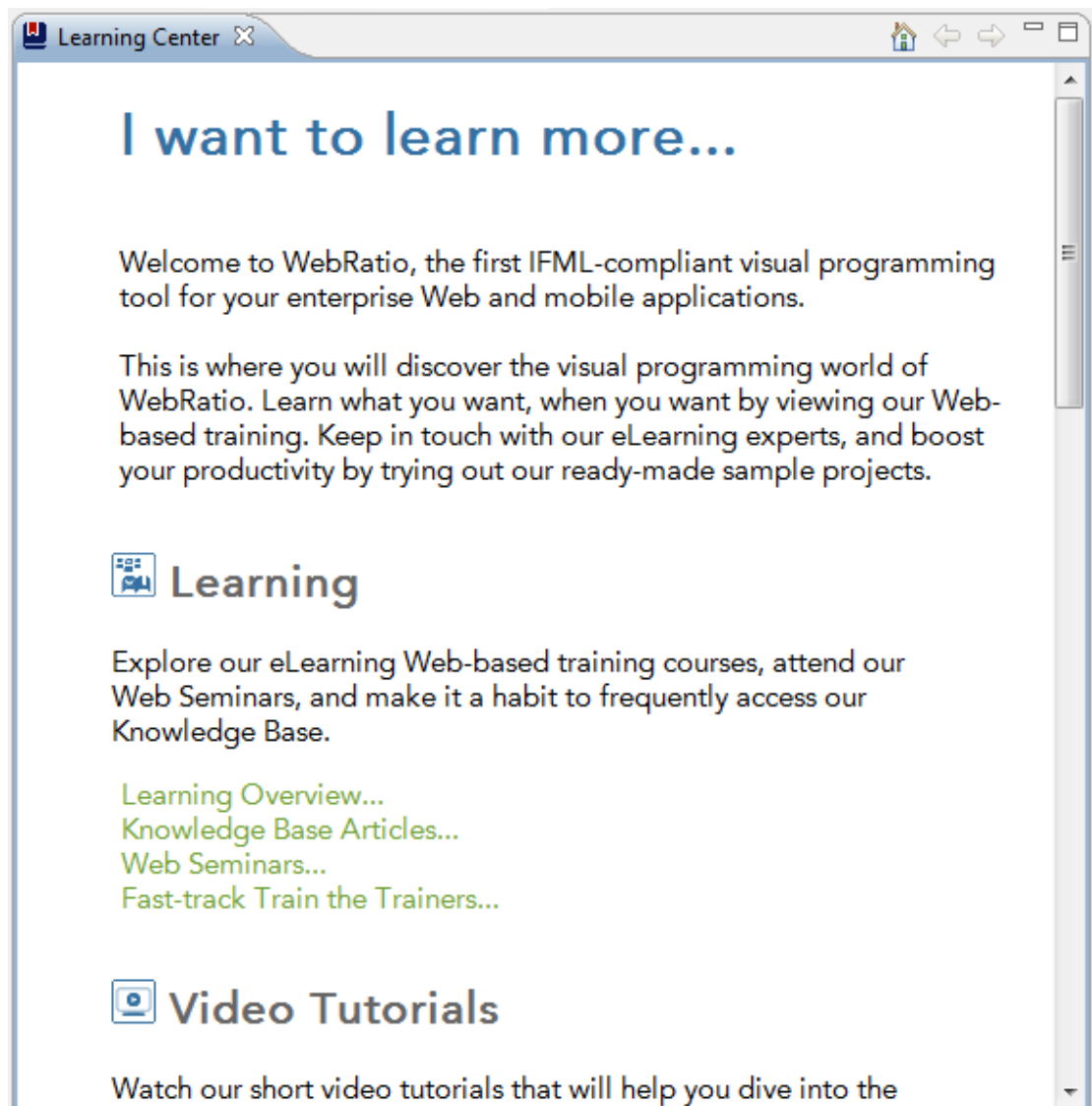
STATIC RESOURCES INCLUSION REVISION

The ways in which users define and use resources, such as CSS and JavaScript files, have been completely revised in order to be more efficient and flexible. Now all resources must be declared in the Style Configuration file (Layout.xml). The "Resources" section of the Layout.xml now contains an editor that allows you to add CSS and JavaScript resources while specifying dependencies between resources. In this way, you just indicate in the template which resource you want to use and the tool will automatically solve all the necessary dependencies.

Refer to the **WebRatio 7.2 Migration Guide** to learn how to change your style projects to be compliant with this new mechanism.

LAYOUT TAGS REFACTORING

The layout tags used to write down custom templates have been revised in order to reduce their number and to remove duplicates and similar tags inherited from previous versions of WebRatio. The refactoring produced a set of deprecated tags that still work but that will be removed from the tool with the next release of WebRatio.

Refer to the **Product Documentation** to see the full list of deprecated tags and go through the **WebRatio 7.2 Migration Guide** to learn how to change your custom templates to be up-to-date with the current implementation.

AJAX RESPONSE PAGE OPTIMIZATION

The JSP page generated when a page is considered AJAX by the tool has been optimized. The main benefits of this change are the decreased size of the JSP page in terms of bytes, an increase of the generation performances, and an optimization of the page execution at runtime. Basically, with this feature you will save time while modeling and the user will have a more effective user interaction.

WEB PROJECT INCLUSION IN ENTERPRISE PROJECT

The Enterprise Project can be now enriched to encompass an entire Web Project. The aim is to allow the reuse of an existing auto-consistent Web Project inside the Enterprise one. The tab showing the Enterprise Project structure now contains a new section, which is the Web Projects section. By right-clicking on the section you can include your own Web Project. You can add as many projects as you want. The projects are quite independent and can be included in the deployment phase.
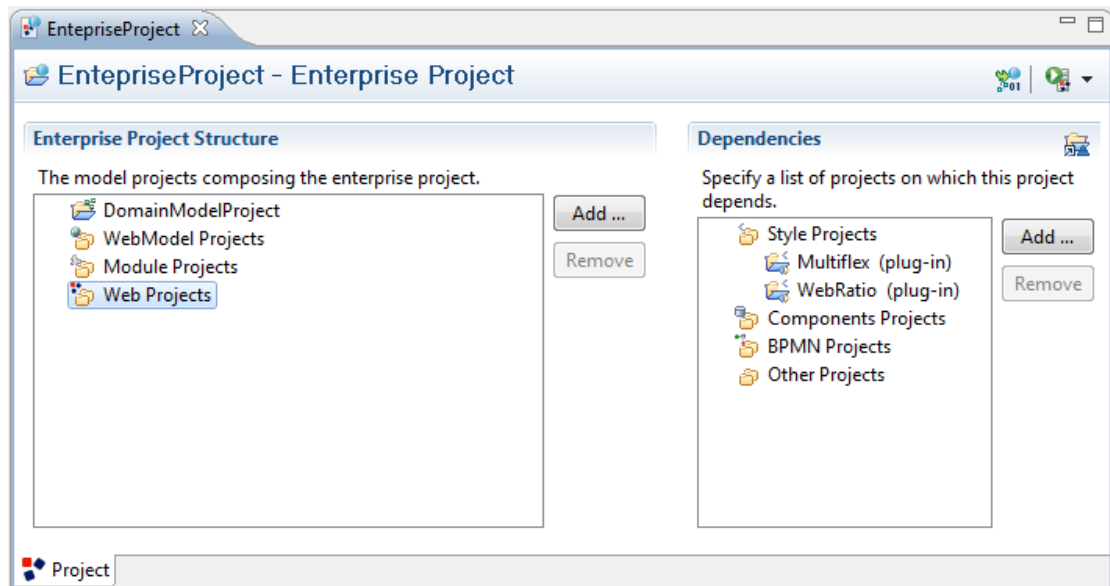
MOVING FROM 7.2.18

This section aims to help you to migrate your project from version 7.2.18 to version 7.2.19.

Log configuration

You must read this section if you have the `RTXLogConfig.properties` and/or the `StrutsLogConfig.properties` in the WebContent/WEB-INF/conf folder of your web project or the `AppLogConfig.properties` in the WebContent/WEB-INF/conf folder of your enterprise project

The upgrade to log4j2 requires to verify and change, if necessary, the log configuration contained in your projects. You must check the content of the `RTXLogConfig.properties`, `StrutsLogConfig.properties` and `AppLogConfig.properties` files so that they match the new required structure. You can migrate your own custom configuration starting from the default one reported below. If the examples are not enough to migrate your configuration please refer to the official log4j2 documentation at these links:

- https://logging.apache.org/log4j/2.x/manual/configuration.html

- https://logging.apache.org/log4j/2.x/manual/appenders.html

RTXLogConfig.properties

```
status=WARN
packages=com.webratio.rtx.log

# RTX debug appender
appender.RTXLogFiile.name=RTXLogFiile
appender.RTXLogFiile.type=RollingFile
appender.RTXLogFiile.fileName=${logDirectory}/RTX.log
appender.RTXLogFiile.filePattern=${logDirectory}/RTX.log.%i
appender.RTXLogFiile.layout.type=PatternLayout
appender.RTXLogFiile.layout.pattern=%d{dd MMM yyyy HH:mm:ss,SSS}  %-5p
[%t]  (%c:%L) - %m%n
appender.RTXLogFiile.policy.type=SizeBasedTriggeringPolicy
appender.RTXLogFiile.policy.size=1 MB
appender.RTXLogFiile.filter.threshold.type=ThresholdFilter
appender.RTXLogFiile.filter.threshold.level=TRACE

# RTX error appender
appender.RTXErrorLogFile.name=RTXErrorLogFile
appender.RTXErrorLogFile.type=RollingFile
```

```
appender.RTXErrorLogFile.fileName=${logDirectory}/RTXError.log
appender.RTXErrorLogFile.filePattern=${logDirectory}/RTXError.log.%i
appender.RTXErrorLogFile.layout.type=PatternLayout
appender.RTXErrorLogFile.layout.pattern=%d{dd MMM yyyy HH:mm:ss,SSS}
%-5p [%t]  (%c:%L) - %m%n
appender.RTXErrorLogFile.policy.type=SizeBasedTriggeringPolicy
appender.RTXErrorLogFile.policy.size=1 MB
appender.RTXErrorLogFile.filter.threshold.type=ThresholdFilter
appender.RTXErrorLogFile.filter.threshold.level=ERROR

# BPM appender
appender.BPMLogFile.name=BPMLogFile
appender.BPMLogFile.type=RollingFile
appender.BPMLogFile.fileName=${logDirectory}/BPM.log
appender.BPMLogFile.filePattern=${logDirectory}/BPM.log.%i
appender.BPMLogFile.layout.type=PatternLayout
appender.BPMLogFile.layout.pattern=%d{dd MMM yyyy HH:mm:ss,SSS}  %-5p
[%t]  (%c:%L) - %m%n
appender.BPMLogFile.policy.type=SizeBasedTriggeringPolicy
appender.BPMLogFile.policy.size=1 MB
appender.BPMLogFile.filter.threshold.type=ThresholdFilter
appender.BPMLogFile.filter.threshold.level=TRACE

# Use one of the following log levels: ERROR, WARN, INFO, DEBUG, TRACE

# Root logger (always used, unless additity=false)

rootLogger.level=ERROR
rootLogger.appenderRef.0.ref=RTXLogFiile
rootLogger.appenderRef.1.ref=RTXErrorLogFile

# Specific loggers

logger.0.name=org.hibernate.SQL
logger.0.level=ERROR

logger.1.name=org.quartz
logger.1.level=ERROR

logger.2.name=com.webratio.rtx
logger.2.level=ERROR

logger.3.name=com.webratio.units
logger.3.level=ERROR

logger.4.name=com.webratio.units.bpm
logger.4.level=ERROR
logger.4.appenderRef.0.ref=BPMLogFile
logger.4.additivity=false
```

StrutsLogConfig.properties

```
status=WARN
packages=com.webratio.rtx.log

# Struts debug appender
appender.StrutsLogFile.name=StrutsLogFile
appender.StrutsLogFile.type=RollingFile
appender.StrutsLogFile.fileName=${logDirectory}/WRStruts.log
appender.StrutsLogFile.filePattern=${logDirectory}/WRStruts.log.%i
appender.StrutsLogFile.layout.type=PatternLayout
appender.StrutsLogFile.layout.pattern=%d{dd MMM yyyy HH:mm:ss,SSS}  %-
5p [%t]  (%c:%L) - %m%n
appender.StrutsLogFile.policy.type=SizeBasedTriggeringPolicy
appender.StrutsLogFile.policy.size=1 MB
appender.StrutsLogFile.filter.threshold.type=ThresholdFilter
appender.StrutsLogFile.filter.threshold.level=TRACE

# Struts error appender
appender.StrutsErrorLogFile.name=StrutsErrorLogFile
appender.StrutsErrorLogFile.type=RollingFile
appender.StrutsErrorLogFile.fileName=${logDirectory}/WRStrutsError.log
ap-
pender.StrutsErrorLogFile.filePattern=${logDirectory}/WRStrutsError.log
.%i
appender.StrutsErrorLogFile.layout.type=PatternLayout
appender.StrutsErrorLogFile.layout.pattern=%d{dd MMM yyyy HH:mm:ss,SSS}
%-5p [%t]  (%c:%L) - %m%n
appender.StrutsErrorLogFile.policy.type=SizeBasedTriggeringPolicy
appender.StrutsErrorLogFile.policy.size=1 MB
appender.StrutsErrorLogFile.filter.threshold.type=ThresholdFilter
appender.StrutsErrorLogFile.filter.threshold.level=ERROR

################################################################
########
```

```
# Use one of the following log levels: ERROR, WARN, INFO, DEBUG, TRACE


# Specific loggers


logger.struts0.name=com.webratio.struts
logger.struts0.level=INFO
logger.struts0.appenderRef.0.ref=StrutsLogFile
logger.struts0.appenderRef.1.ref=StrutsErrorLogFile
logger.struts0.additivity=false
```

AppLogConfig.properties

```
status=WARN
packages=com.webratio.rtx.log


# RTX debug appender
appender.RTXLogFiile.name=FullLogFile
appender.RTXLogFiile.type=RollingFile
appender.RTXLogFiile.fileName=${logDirectory}/RTX.log
appender.RTXLogFiile.filePattern=${logDirectory}/RTX.log.%i
appender.RTXLogFiile.layout.type=PatternLayout
appender.RTXLogFiile.layout.pattern=%d{dd MMM yyyy HH:mm:ss,SSS}  %-5p
[%t]  (%c:%L) - %m%n
appender.RTXLogFiile.policy.type=SizeBasedTriggeringPolicy
appender.RTXLogFiile.policy.size=1 MB
appender.RTXLogFiile.filter.threshold.type=ThresholdFilter
appender.RTXLogFiile.filter.threshold.level=TRACE


# RTX error appender
appender.RTXErrorLogFile.name=RTXErrorLogFile
appender.RTXErrorLogFile.type=RollingFile
appender.RTXErrorLogFile.fileName=${logDirectory}/RTXError.log
appender.RTXErrorLogFile.filePattern=${logDirectory}/RTXError.log.%i
appender.RTXErrorLogFile.layout.type=PatternLayout
```

```
appender.RTXErrorLogFile.layout.pattern=%d{dd MMM yyyy HH:mm:ss,SSS}
%-5p [%t]  (%c:%L) - %m%n
appender.RTXErrorLogFile.policy.type=SizeBasedTriggeringPolicy
appender.RTXErrorLogFile.policy.size=1 MB
appender.RTXErrorLogFile.filter.threshold.type=ThresholdFilter
appender.RTXErrorLogFile.filter.threshold.level=ERROR

# BPM appender
appender.BPMLogFile.name=BPMLogFile
appender.BPMLogFile.type=RollingFile
appender.BPMLogFile.fileName=${logDirectory}/BPM.log
appender.BPMLogFile.filePattern=${logDirectory}/BPM.log.%i
appender.BPMLogFile.layout.type=PatternLayout
appender.BPMLogFile.layout.pattern=%d{dd MMM yyyy HH:mm:ss,SSS}  %-5p
[%t]  (%c:%L) - %m%n
appender.BPMLogFile.policy.type=SizeBasedTriggeringPolicy
appender.BPMLogFile.policy.size=1 MB
appender.BPMLogFile.filter.threshold.type=ThresholdFilter
appender.BPMLogFile.filter.threshold.level=TRACE

# Struts debug appender
appender.StrutsLogFile.name=StrutsLogFile
appender.StrutsLogFile.type=RollingFile
appender.StrutsLogFile.fileName=${logDirectory}/WRStruts.log
appender.StrutsLogFile.filePattern=${logDirectory}/WRStruts.log.%i
appender.StrutsLogFile.layout.type=PatternLayout
appender.StrutsLogFile.layout.pattern=%d{dd MMM yyyy HH:mm:ss,SSS}  %-
5p [%t]  (%c:%L) - %m%n
appender.StrutsLogFile.policy.type=SizeBasedTriggeringPolicy
appender.StrutsLogFile.policy.size=1 MB
appender.StrutsLogFile.filter.threshold.type=ThresholdFilter
appender.StrutsLogFile.filter.threshold.level=TRACE

# Struts error appender
appender.StrutsErrorLogFile.name=StrutsErrorLogFile
```

```
appender.StrutsErrorLogFile.type=RollingFile
appender.StrutsErrorLogFile.fileName=${logDirectory}/WRStrutsError.log
ap-
pender.StrutsErrorLogFile.filePattern=${logDirectory}/WRStrutsError.log
.%i
appender.StrutsErrorLogFile.layout.type=PatternLayout
appender.StrutsErrorLogFile.layout.pattern=%d{dd MMM yyyy HH:mm:ss,SSS}
%-5p [%t]  (%c:%L) - %m%n
appender.StrutsErrorLogFile.policy.type=SizeBasedTriggeringPolicy
appender.StrutsErrorLogFile.policy.size=1 MB
appender.StrutsErrorLogFile.filter.threshold.type=ThresholdFilter
appender.StrutsErrorLogFile.filter.threshold.level=ERROR


####################################################################
########
# Use one of the following log levels: ERROR, WARN, INFO, DEBUG, TRACE

# Root logger (always used, unless additity=false)

rootLogger.level=ERROR
rootLogger.appenderRef.0.ref=RTXLogFiile
rootLogger.appenderRef.1.ref=RTXErrorLogFile

# Specific loggers

logger.0.name=com.webratio.struts
logger.0.level=INFO
logger.0.appenderRef.0.ref=StrutsLogFile
logger.0.appenderRef.1.ref=StrutsErrorLogFile
logger.0.additivity=false

logger.1.name=org.hibernate.SQL
logger.1.level=ERROR

logger.2.name=org.quartz
```

```
logger.2.level=ERROR


logger.3.name=com.webratio.rtx
logger.3.level=ERROR


logger.4.name=com.webratio.units
logger.4.level=ERROR


logger.5.name=com.webratio.units.bpm
logger.5.level=ERROR
logger.5.appenderRef.0.ref=BPMLogFile
logger.5.additivity=false
```

Custom Components

You must read this section if you have custom components using directly the log4j2 library.

The code of you custom components must be migrated according to the official documentation of log4j2 available at this link:
https://logging.apache.org/log4j/2.x/manual/migration

Scripts using Authorization Helper

If you have Script Components in your project using the AuthorizationHelper class, then you must change your code as following:

OLD CODE

```
#input String username, String password
import com.webratio.rtx.RTXConstants
import com.webratio.struts.AuthorizationHelper
return AuthorizationHelper.computeAuthorization(username, password)



NEW CODE

#input String username, String password
import javax.servlet.http.HttpServletRequest
import com.webratio.rtx.RTXConstants
```

```
import com.webratio.struts.AuthorizationHelper
def request = localContext.get(RTXConstants.HTTP_SERVLET_REQUEST_KEY)
return AuthorizationHelper.computeAuthorization(username, password,
request)
```

MOVING FROM 7.2.16

This section aims to help you to migrate your project from version 7.2.16 to version 7.2.17. You must read this section to ensure that all security fixes are applied to your Web projects.

Password encryption

You must read this section if your Web project did not have encrypted passwords. A security warning detects this situation and suggest to apply a Crypt Algorithm on the attribute. If your Web application was deployed and there's a user account list, you have to manage the migration of current passwords, which are not encrypted.

The best way is to associate to the password an expiration date, which forces the user to change the password periodically. In this case you must force the expiration date to a date in the past, so that just after the deploy of the application version, the user must change the password.

The login procedure must be updated handling the password expiration case. This means that, when the user provides username and password, and the login fails, you should check whether the user account exists (using the username) and whether the password is expired. If this is true, you should compare the password provided by the user, to the one stored in the database for the specific username. The comparison must consider whether the password in the previous release were stored using an encryption algorithm or not. The best solution is to use a script operation handling all the possible cases. If not, the user should be redirected to the login page since he provided wrong credentials.

Here is the sample code of the Script operation comparing the passwords:

```
#input String databasePassword, String providedPassword
import com.webratio.rtx.core.CryptHelper;

// Encrypt provided password using all DES algorithms
def NullableDES3UnicodePassword = CryptHelp-
er.getCryptAlgorithm("com.webratio.rtx.core.NullableDES3UnicodeCryptImp
lementation").encrypt(providedPassword);
def DES3UnicodePassword = CryptHelp-
er.getCryptAlgorithm("com.webratio.rtx.core.DES3UnicodeCryptImplementat
ion").encrypt(providedPassword);
def DES3Password = CryptHelp-
er.getCryptAlgorithm("com.webratio.rtx.core.DES3CryptImplementation").e
ncrypt(providedPassword);
```

```
// check password match in plain text
if (databasePassword == providedPassword){
    return ["resultCode": "success"]
}
//check password match using NullableDES3Unicode algorithm
if (databasePassword == NullableDES3UnicodePassword){
    return ["resultCode": "success"]
}
//check password match using DES3Unicode algorithm
if (databasePassword == DES3UnicodePassword){
    return ["resultCode": "success"]
}
//check password match using DES3 algorithm
if (databasePassword == DES3Password){
    return ["resultCode": "success"]
}
//check password match using PBKDF2WithHmacSHA1 algorithm
if (CryptHelp-
er.getCryptAlgorithm("com.webratio.rtx.core.PBKDF2WithHmacSHA1CryptImpl
ementation").verify(providedPassword,databasePassword)){
    return ["resultCode": "success"]
}

return ["resultCode": "error"]
```

Web application deployment

You must check the configuration of the deployment environment to ensure that the new security checks against the "cache poisoning" and "arbitrary URL redirect" do not cause Web application errors.

You can verify that the Web application does not work due to these new security checks by looking at the StrutsError.log file. This log will contain a warning like the following:

```
                  Validation failed for target host
```

You must add some configurations to the deploy environment to solve this issue.

- In case of reverse proxy you have two options: either you configure the list of valid virtual hosts in the RTXConfig.properties file setting the "hostWhitelist" property or you set the "hostAlwaysAllowed" property in the same file and check that the reverse proxy is configured with a white list of valid domains.

- In all other cases you have to check that the application server is able to execute a reverse lookup DNS of their address. Otherwise you can set the "hostWhitelist" property in the RTXConfig.properties file.

You must check the deployment plans contained in your Web project to ensure that all changes done to fix the security warnings are also reported in each deployment plan.

Web.xml

You must read this section if your project contains a custom web.xml. You can check this condition by looking in the WebContent/WEB-INF folder of your Web project or searching for the web.xml file as a resource in the Web project folder.

The standard web.xml now contains a filter that removes the jsessionid parameter from the URL. You must adapt your custom web.xml to include this filter. This filter must be the first declared in the file. See the example code:

```
<filter>
     <filter-name>security-filter</filter-name>
     <filter-class>com.webratio.struts.filters.SecurityFilter</filter-
class>
</filter>
<filter-mapping>
     <filter-name>security-filter</filter-name>
     <url-pattern>/*</url-pattern>
</filter-mapping>
```

Style project

You must read this section if you have a custom style project for your Web application.

- You must check the JavaScript resources declared in the Layout.xml of you custom style project. In particular any jQuery resource having version less than 1.12.4 must be removed from the declared resource.

  All other JavaScript resources having a dependency on jQuery must be reviewed to understand if they are compatible with the current 1.12.4 version. You can choose either to update the third party resource to a newer version or use the plugin. The same solution applies to JavaScript containing custom code. If you want to manually adapt your JavaScript code please refer to the jQuery Release Notes.

- Custom templates for the Form Fields must be adapted to include the code fragment managing the new ReCaptcha field. You can add the code at the end of your template. See the following example:

```
[% def captchaField = getById(field["field"])?.name == "Cap-
tchaField" %]

………

[% else if (captchaField) {%]
   <div class="g-recaptcha" data-sitekey="[%= publicKey
%]"></div>
   <script
src='https://www.google.com/recaptcha/api.js'></script>
[% } %]
```

- Custom templates for the Form Fields must be adapted to include the code fragment managing the client side of the new Password Strength validation rule. Add the following code just below the input type password field:

```
[% def passwordStrengthness =
field.selectSingleNode("VRules/PasswordStrengthValidationRule") %]

<input type="password"….

[% if(passwordStrengthness != null){ %]data-security-validation="[%=
htmlId %]_validation"[% } %]/>
[% if(passwordStrengthness != null){ %]
     <wr:RequireResource ref="wr-password-strengthness-js"/>
     <wr:RequireResource ref="wr-password-strengthness-css"/>
<script id="[%= htmlId %]_validation" type="application/json">
{
     "securityLevel": "[%= passwordStrengthness["securityLevel"] %]",[%
     if (passwordStrengthness["securityLevel"] == "custom") { %]
          "customOptions": {
               "minLength": [%= passwordStrengthness["minLength"] ==
Integer.toString(NumberUtils.toInt(passwordStrengthness["minLength"]))
? passwordStrengthness["minLength"] : 0 %],
               "maxLength": [%= passwordStrengthness["maxLength"] ==
Integer.toString(NumberUtils.toInt(passwordStrengthness["maxLength"]))
? passwordStrengthness["maxLength"] : Integer.MAX_VALUE %],
               "useNumbers": [%= passwordStrengthness["useNumbers"]
== "true" %],
               "useLowerLetters": [%= passwordStrength-
ness["useLowerCaseCharacters"] == "true" %],
               "useCapitalLetters": [%= passwordStrength-
ness["useUpperCaseCharacters"] == "true" %],
               "useSpecialCharacters": [%= passwordStrength-
ness["useSpecialCharacters"] == "true" %]
          },[%
     } %]
```

```
     "messages": {
          "title": "<bean:message
key="PasswordStrengthValidationRule.boxTitle"/>",
          "useNumbers": "<bean:message
key="PasswordStrengthValidationRule.numbersCriteria"/>",
          "useLowerLetters": "<bean:message
key="PasswordStrengthValidationRule.lowerCharCriteria"/>",
          "useCapitalLetters": "<bean:message
key="PasswordStrengthValidationRule.capitalCharCriteria"/>",
          "useSpecialCharacters": "<bean:message
key="PasswordStrengthValidationRule.specialCharCriteria"/>",
          "noRepetitions": "<bean:message
key="PasswordStrengthValidationRule.noRepetitionCriteria"/>",
          "charactersWord": "<bean:message
key="PasswordStrengthValidationRule.charactersWord"/>"
          }
     }
</script>
[% } %]
```

- Custom templates for the Multiple Form Fields must be adapted if they contain the option "Use checkboxes" for boolean fields. This is the code to change:

OLD CODE

```
[% if ((type == "boolean")) { %]
     [% if (params["boolean-field-rendering"] == "checkbox") { %]
     ……………..
     [% } else { %]
     …….
     <c:if test="[% printJSPTagValue(readOnly)%]">
          <html:hidden indexed="true" name="[%=unitId%]" proper-
ty="value([%=fieldProp%])"/>
     </c:if>
     [% } %]
[% } %]
```

NEW CODE

```
[% if ((type == "boolean")) { %]
     [% if (params["boolean-field-rendering"] == "checkbox") { %]
          ……………..
     [% } else { %]
          …….
```

```
        [% } %]

                <c:if test="[% printJSPTagValue(readOnly)%]">

                        <html:hidden indexed="true" name="[%=unitId%]"
property="value([%=fieldProp%])"/>

                </c:if>

        [% } %]
```

- Custom templates for Form and Multiple Form fields including or referencing JavaScript code that dynamically change the value of a hidden fields must be changed because the CRC check will fail. The solution is to change the model setting the field **not** hidden and hide the field through HTML/CSS.

- Custom templates for Multiple Form managing the fields in dynamic mode (by adding and removing rows) must be changed because the CRC check will fail. The solution is to change the model setting the field **not** hidden and hide the field through HTML/CSS.

Components Project

You must read this section if you have a custom Components project that includes a form based custom component for your Web application.

You must implement to additional methods in the component Java class, which are requested by the RTXFormUnitService interface.
These methods are `extractRequestedServerFormValues` and `extractCurrentServerFormValues`. Both methods retrieve the list of server fields (i.e. hidden fields) to be protected with the CRC.
The `extractCurrentServerFormValues` method is used to compose the CRC before the page is shown to the user.
The `extractRequestedServerFormValues` method is used after the submit of the form to compute a new CRC to be compared with the previous one and thus validating the request.
You can decide to leave the implementation empty returning an empty map, if your component does not manage hidden fields. Otherwise you must implement those methods. Please refer to the `EntryUnitService` class for examples.
If your component uses modeled fields, you must also change the Logic.template adding to the Field node the information about hidden fields that must be considered server fields. See the following example:

```
[% if (field["hidden"] == "true") { %]serverOnly="true"[% } %]
```

MOVING FROM 7.2.15

This section aims to help you to migrate your project from version 7.2.15 to version 7.2.16. You must read this section if you use a crypted password for password type attributes in your projects. You should remove from your model, if it is present, the operations chain terminating with a Selector Component based on "User" Entity, which sends the "password" value to a Login Component.
This pattern is no longer granted for security issues.

You must read this section if you use customizations for Form Component or Multiple Form Component in your project. You should change the "Output.template" of your component, if you have one, to benefit from the fix that removes passwords from redirect URLs and query strings. To do so, follow these guidelines:

It is need to add the attribute confidential="true" to the outputs generated by the password input fields. For example, related to the Form Component:

NEW CODE

```
<% for (field in unit.selectNodes("Field | SelectionField | MultiSelec-
tionField")) { %>

    ...

    <OutputParameter ... confidential="<%= field["type"] == "password"
%>"/>

    ...
<% } %>
```

You must read this section if your applications are deployed behind a reverse proxy. They may start responding with the "*Cross-site request forbidden*" error with the form submission. Moreover, in these cases, a "*warning Struts/MVC Blocked cross-site request from <IP>: different origin*" is also logged.
The issue is caused by the new security checks on the origin of form submissions, which are sometimes incompatible with deployments behind the reverse proxy.
You should change the configuration of the reverse proxy, if any, by following these guidelines:
* Forward the header Host or the X-Forwarded-Host to the destination.
  For example, it is possible to configure the Apache with
  "`ProxyPreserveHost On`". Alternatively, but with less security, you can

turn off the control by including the following property in the *WEB-INF/conf/RTXConfig.properties* file: `checkRequestOrigin=false`.

This release introduces HTTPS communication to the activation server. In the next months, the previous releases to 7.2.16, will only be activated via manual activation because the HTTP requests will be disabled.

MOVING FROM 7.2.14

This section aims to help you to migrate your project from version 7.2.14 to version 7.2.15. You must read this section if you use the following methods of the *wr.nav.PropagationRouter* constructor in your custom js scripts. They were moved to instances of *mergeLinkData*, *clearLinkData*, *getLinkData*, *setKeepHashes* and *extendParameters*.

If you want to continue to use them, you need to retrieve the application's specific propagation router. For example:

OLD CODE

```
wr.nav.PropagationRouter.extendParameters(a, b, c);
```

NEW CODE

```
var propagRouter =
app.getPlugin("nav").getRouterOfType(wr.nav.PropagationRouter);
propagRouter.extendParameters(a, b, c);
```

Fixed in 7.2.20

- Fixed a regression that caused the impossibility to use the visual debug. Projects having breakpoints in the model were shown as corrupted and the modeler could not add/remove any breakpoint.

- Fixed a runtime exception raised when the session is invalidated at user log out having the Visual Debug enabled through the `com.webratio.rtx.debug=true` java system property.

FIXED IN 7.2.19

Security Fixes

- Upgraded log4j library to log4j 2.17 due to several vulnerabilities.

- Extended the chance to ignore specific hidden fields when CRC check is performed. You can add the custom property with name "`unsignable`" to these fields as well as you can already do with the Multiple Form and Dynamic template.

- Fixed the xml external entity processing vulnerability in published REST web services. Now if you try to invoke a rest web service using application/xml as content type and injecting a DTD, the web service does not more process the received data anymore.

- Fixed a problem that cause a CRC error when an event is triggered from a popup.

- Fixed the management of the `nonProxyHosts` feature. The list of "non-proxy" hosts defined as system property is correctly validated now.

- Added the new "File Types" Valiation Rule to check the content, extension and mime type of file loaded through blob field type.

Others

- Fixed the generation of the token necessary to invoke a protected web services when the user password is encrypted. The generated token was not valid. You can now generate the correct token starting from the user credentials. Please refer to the <u>Moving from 7.2.18 section</u> for further information.

- You can now customize the duration of the token required to invoke protected web services. This time was previously fixed to 15 minutes. You can use the `authorizationTimeout` property in the RTXConfig.properties file. For example: authorizationTimeout=180000 which means that the token is valid for 3 minutes.

- Improved generation performance when the web project contains complex and nested modules.

- Fixed the behavior of the web service related to events of versioned BPM processes. All instances of the versioned process were wrongly retrieved. Now you can specify which version of the process you want through the ""version"" request parameter lets. You must manually add this parameter in the request message.

- If the value is current, only the current process version is used.

- If the value is one of the non-current versions (e.g., v1), only that process version is used

- If the value is one of the non-current versions (e.g., v1), only that process version is used.

- If the value is omitted (either version is blank or it is completely missing), all process versions are used at once.

For example:

http://localhost:8080/BPMWeb/wr-bpm/REST/Pool1/StartMessage1?version=v1

```
(SOAP) http://localhost:8080/BPMWeb/wr-
bpm/SOAP/Pool1/StartMessage1
<soapenv:Envelope
xmlns:soapenv=""http://schemas.xmlsoap.org/soap/envelope/""
xmlns:bpm=""http://www.webml.org/webservices/bpm"">
   <soapenv:Header/>
   <soapenv:Body>
     <bpm:StartMessage1>
        <Version>v1</Version>
     </bpm:StartMessage1>
   </soapenv:Body>
</soapenv:Envelope>
```

- Fixed an issue about default "Sort Attributes" that were ignored in the sorting process when they are different from the sort attributes. This unexpected behavior was wrongly introduced in WebRatio 7.2.18.

- Fixed an issue about the multisession management. The request of a new session when the user jump to another web model of the Enterprise Project is correctly created now.

- Fixed an issue about the default sort attributes order. Now when the user access to the page at the first time can see the attribute's sorting properly set as default.

- When you install WebRatio for windows the java-home path is automatically set in the WebRatio.ini file. In this way WebRatio will start with the expected

and set java even if some others java versions are already installed and java-home system property is pointing to another java version.

- Now it is possible to debug the runtime execution of WebModel and Module projects embedded in the Enterprise Project. To enable it you have to:

  - Configure the Web Server hosting the application.
  - Prepare a Debug configuration for each project.
  - Add breakpoints into the projects wherever needed.
  - Run any Debug Configuration involved in the execution flow.
  - Debug the Web application.

  NOTE: To debug Action and Module definitions inside embedded projects you have to define breakpoints inside the target embedded project.

- Now sequences in SQL Server 2012 are supported. You can define the created (and available in the database) sequence as ID Generator for Primary key Attributes.

- Fixed an issue about redirect URL and listening port (performed by the Jump component) when the webapp is hosted in an environment behind a Proxy server.

- Fixed a Domain Model error about management of primary keys when the target database does not support them.

- Solved an issue about wrong redirect URL when the application is behind a reverse proxy.

- Fixed an issue that prevented the Solicit operation to retrieve data in the expected parameter order if several calls are performed in parallel.

- Updated the Apache POI libraries to improve performances of the Excel component.

- Fixed a problem in the StringCache class that may cause a thread block in some situations.

- Fixed an issue that may cause the ajax requests of the web application to be interrupted due to a parse error.

FIXED IN 7.2.18

Security Fixes

- WebRatio Platform now communicates with the external services used for activations, downloads and so on using the HTTPS protocol. Previous WebRatio Platform versions cannot be activated anymore using the automatic service. Only manual activation will be available for that version.

- Fixed the Web application runtime so that now it addressed properly the requested REST web service when there are two published REST web services having the same base URL but different parameter fragments.

- Fixed an issue related to the publishing of SOAP web services. Now when the web service fails no exception trace is included in the response, preventing an "Information Disclosure" vulnerability.

- Fixed a problem related to the CRC check on form hidden fields. Now the check works properly in presence of an empty Multiple Form inside the page.

- Fixed a problem related to the CRC check on form hidden fields. Now the check works properly in presence of a Form with visibility conditions applied.

- Fixed a problem related to the CRC check on form hidden fields. Now WebRatio Platform lets you exclude from the check hidden fields belonging to Multiple Forms having the Dynamic template. You must add a custom property with name "unsignable" to these fields. You can use this XPath search on the project to find these fields:

  ```
  //layout:Unit[@layout:unitLayout = 'WRDefault/Dynamic' or ancestor-
  or-self::*[layout:UnitConfiguration[@unitType = 'MultiEntryUnit' and
  @layout:unitLayout = 'WRDe-
  fault/Dynamic']]][id(@unitId)/Field[@hidden = 'true']]
  ```

- Fixed a problem related to the CRC check of form hidden fields that prevented the user to see the proper login page when the session expires.

- Improved the security of the Web application so that now it's possible to mark a field as confidential by adding a custom property with name "confidential". The effect is that the information contained in this field is never appended into the URL.

Others

- Improved the Compare View so that now it's possible to import also single elements in composite properties such as Display Attributes, Sort Attributes, Flow Order and so on.

- Fixed an issue of the JSON I/O operation that prevented to properly import a JSON document having NULL nodes on children's elements.

- Fixed a problem of the Find Model Problems command which could raise the exception "Marker property value is too long" when running on very big projects.

- Fixed a problem of the Hierarchy component. Now the component bean contains the sorting information also when the default sorting is applied.

- Fixed a problem of the Hierarchy component. Now the component bean contains the current sorting information also when the Hierarchy has the pagination enabled.

- Fixed a problem of the generated JSP pages when the "Split subpages" option is used. Now the proper content type "text/xml" is appended to AJAX JSP pages. This problem caused WebLogic to return a "Parse error" in the deployed Web application.

- Fixed a problem related to the field validation when the field is related to an attribute having an XSD attached. Now the field properly validates the value using the restriction specified in the XSD, if any.

FIXED IN 7.2.17

Security Fixes

- Changed the WebRatio Addon login mechanism to meet the new privacy requirements. Since this change involve external web services, previous WebRatio versions cannot access anymore the WebRatio Addon.

- Extended the CRC check to form hidden fields. This fix avoids a maliciously third party to change the value of one or more hidden fields. Check the "Moving from 7.2.16 – Components Project" section to see the migration guide.

- Fixed an issue that allowed a "Cookie Manipulation" attack using the "cbck" query string parameter. Now this parameter is strictly validated to accept a specific name.

- Fixed an issue that allowed to acquire limited knowledge about the web application folder structure by manipulating the "res" query string parameter. This parameter is used when the Web project publishes web services. Now the value of this parameter is validated to accept files belonging to the proper directory containing XSD schema files.

- Fixed an issue related to the publishing of REST web services. Now when the web service fails no exception trace is included in the response, preventing an "Information Disclosure" vulnerability.

- Fixed an issue that allowed to manipulate the "Host" header parameter. This issue could be exploited to carry out a "Cache Poisoning" attack. Now the "Host" header parameter is always validated using a whitelist coming from the web application or from the reverse proxy/application server.

- Fixed an issue that allowed to manipulate the "lastURL" parameter to arbitrary redirect the user to an external site. Now this parameter is validated using the host white list.

- Fixed an issue that caused the Web application to show to the user different error pages depending to the error types. Now the error page is unique and it is the one set in the Style Project applied to your Web project.

- Added the HTML Sanitizer validation rule that removes from the HTML code coming from a rich text editor all the malicious code, such as <script> fragments. It's mandatory on fields having "text/html" content type.

- Added the Password Strength validation rule that ensure that the provided password is a secure one. This validation rule works both on client side and

on server side. It's mandatory on password fields. Check the "Moving from 7.2.16" section to see the migration guide.

- Added the ReCaptcha field that prevents a form to be filled in by robots. It's mandatory on public forms. Check the "Moving from 7.2.16" section to see the migration guide.

- Added a filter servlet that removes the jsessionid from the URL. This prevents to share by mistake the session cookie with other users. Check the "Moving from 7.2.16 – Web.xml" section to see the migration guide.

- The Apache Commons File Upload library has been upgraded to version 1.3.3 to fix the vulnerabilities "CVE-2016-3092" and "CVE-2016-100031".

- Fixed the "Display All" property of the Details Component so that password attributes now are automatically excluded.

- The Apache Commons IO library has been upgraded to version 2.2 as a requirement for the Apache Commons File Upload to be updated to 1.3.3.

- The jQuery library has been upgraded to version 1.12.4 to fix the vulnerability "CVE-2012-6708" and the following security bugs: #11290 , #2432. Check the "Moving from 7.2.16 – Style Project" section to see the migration guide.

Security Warnings

This release includes several new warnings related to the security. Launching the Find Model Problems on your Web project will report these problems; some also have the "Quick Fix" option to easily solve them. Fix the problems to get a more secure Web application.

- A security warning is reported when the Web project contains a logout operation without the "Invalidate Session" flag enabled.

- A security warning detects a public form which is missing a captcha field. The captcha field prevents the form to be filled in by robots.

- A security warning detects a password field which is missing the Password Strength validation rule.

- A security warning is reported when a context parameters has the "Use URL" property enabled.

- A security warning detects when the Web project contains at least a protected view container and the "Protect URL" property is not enabled.

- A security warning detects when the Web project has the "Protect URL" property enabled but not the "Protected BLOBs" one.

- A security warning detects when the Web project has the "Protect URL" property enabled and the "Digest Id" one is empty.

- A security warning detects when the Web project has the "Digest Type" property set to CRC, which is not secure.

- A security warning detects when the Web project has the "Admin Area" enabled. This area must not be included in the deploy.

- A security warning is reported when the Web project contains orphan pages. An orphan page is a page that cannot be reached in any way and it is intended as a test one that must not be included in the deploy.

- A security warning is reported when a field with content type "text/html" does not have the HTML Sanitizer validation rule.

- A security warning detects all public pages that shows user information which must be private. In addition, for all pages a warning detects whether a password attribute has been added as display one.

- A security warning detects all not encrypted password attributes in the Domain Model or if they are not using the secure "PBKDF2WithHmacSHA1" algorithm. Check the "Moving from 7.2.16 – Password Encryption" section to see the migration guide.

Others

- Now it is possible to Generate and Run the Web applications on a local test environment also in the Community Edition.

- Removed the CumuLogic Cloud Account and Deploy Configuration since the third party service is not available anymore.

- Fixed an issue that caused the presence of some context parameters on the URL when they are passed through an operation chain.

- Fixed an issue that prevented to call a protected SOAP Web Service in case of encrypted password.

- Fixed an issue that prevented the Restful Request Response Operation to properly call a Rest service in which the Content-Type response header is missing. Now the response is treated as a Text response.

- Fixed an issue related to XSD synchronization in the Domain Model. In particular the presence of both xsd:element and xsd:complex-type in the XSD produced duplicated elements in the domain model.

- Fixed a memory saturation issue due to a wrong management of the TLS-SSL protocol in combination with HttpClient 3.x during a REST Service invocation.

- Fixed an issue that prevented a REST web service request to be accepted when the endpoint contains the ".do" string.

- Fixed the field template of the Multiple Form which didn't print a hidden field for boolean fields having the style set to checkbox. This caused the field to be missing in the POST at the submit. Check the "Moving from 7.2.16 – Style Project" section to see the migration guide.

- Fixed an error of the Create operation that prevented to properly save an incoming date having the ISO format.

- Fixed an issue that caused field values to be posted twice in some situations. This might have caused an error about URL size and polluted the field values.

- Fixed an issue that prevented a REST Web Service to work properly when it includes query string parameters in its request.

- Solved a memory leak in the generation process that caused the generation of the layout pages to be slow in some situations.

- Fixed an issue of the Restful Request Response operation that prevent to call successfully some web services. In particular, some web servers may reject the user-agent used by the operation. Now a "safe" user-agent is sent for the request.

- Fixed the PBKDF2WithHmacSHA1 algorithm so that now the white space character can be used inside passwords.

FIXED IN 7.2.16

- Improved the provided algorithms used to crypt a password attribute. A new strong password hashing implementation is now available named "PBKDF2WithHmacSHA1", that you can find in the dropdown list of the "Crypt Algorithm" property for a password attribute.

- Fixed an issue related to the caching of the SOAP Web Service Provider definition when the WSDL XML Document contains encoding different from UTF-8. The "Invalid byte x of y-byte UTF-8 sequence Exception" with WSDL .cache file associated to the SOAP Web Service Provider is now solved.

- Improved the security of the webapp in order to prevent cross site request forgery vulnerability (CSRF). Now, the framework checks the origin and verifies the CSRF token submitted by the form.

- Improved the security of administrator password management. If you enable the "Admin Area" in the Properties View of the project and the password is flagged as "crypted", the password is now hashed with a robust algorithm.

- Improved the security of cookie management in order to avoid user impersonation vulnerability. The cookie related to the default Cookie Storage Provider associated with the Remember Credentials feature is now automatically encrypted. You can also continue to apply the encryption on custom Cookie Storage Providers by filling the "Encryption Key" Property available on it with your custom private key.

- Improved security related to management of xml requests in order to avoid xml external entity (XXE) processing vulnerability. The malicious SOAP requests are now processed and the undesired behavior is rejected.

- Fixed an issue related to the header of a SOAP request. According to SOAP 1.1 specification the SoapAction header (for SOAP 1.1 only) is now quoted.

- Fixed an issue related to the security of the password value that was wrongly reported in the LasUrl hidden field or in the redirect url. The password parameter is now excluded from the LastUrl hidden field and from the redirect url.

- Improved the runtime behavior of the REST services. Now it is handled the multiple-parameter notation (e.g. oid=1&oid=2).

- Improved security in order to solve the Apache Commons HttpClient 3.x hostname vulnerability.

- Fixed a problem related to the CKEditor that prevented to showing an image correctly when setting the RTXConfig.properties file with a custom upload folder.

FIXED IN 7.2.15

- Fixed an error of the JSON I/O Operation in "Export" mode that caused the truncation of the decimal values in the columns with decimal numbers.

- Fixed an issue related to the generation of Enterprise Project by Command line and Ant script. Referring to the "Deploying from CommandLine" feature you can automatically run the desired deploy configuration for an Enterprise Project with success.

- Fixed a problem related to a Multiple Form Component that did not pass the Checked Objects on the outgoing navigation flow after an AJAX event (such as OnChange event).

- Improved the algorithm used to generate the logic of a Web Project. Now the full generation of a Web Project is faster.

- Fixed a problem related to the usage of a Response Operation in combination with the useSoftReference property. Now the Application Server properly manages the service disposal when the property is set to true.

- Fixed a problem related to the Mandatory Validation Rule applied on a field of a Form Component. Now, the Validation Rule also fires when it has a companion field with the predicate property not defined.

- Fixed a problem related to the Restful Request Response Operation that caused a Null Pointer Exception when the operation was associated with a Web Service Provider.

- Fixed a problem related to a Context Parameter associated with the Cookie Provider. Now, a value with the underscore character is properly saved in the parameter.

- Fixed a problem of the Jump Operation that prevented the "jump back" mode from working properly when the operation was used in Module Projects of an Enterprise Project.

- Fixed a problem related to a Multiple Form Component. Now the Component is correctly updated with one row of data, after being updated with multiple rows of data.

FIXED IN 7.2.14

- Improved the JSON Import/Export operation. The operation can now able to import documents whose property names have the first letter capitalized. Simply set the "Mapping Type" property to "Model Names".

- Fixed a problem of the Script Operation used in a job. Now in the generated descriptor of the operation, the list of operation parameters is properly given.

- Fixed the Solicit Operation that has a Form Data Parameter and is used with the REST protocol. In particular, the operation can now properly handle the input of a form data parameter file type, even if the incoming request not appended to the request in the "filename" information.

- Fixed a problem related to the Mandatory Validation Rules with a boolean field as companion field. Now, the validation works properly with the Equal predicate as long as the when has enabled the "ignore case" property and is set to false.

- Fixed a problem related to the Restful Request Response operation. Now, the operation properly handles the web service response in case is without a charset declared in the response header. The default charset is UTF-8.

- Fixed the SQL dialect used when the project uses a SQL Server 2012 database. Now, the proper dialect is used, which allows for the use of the sequences defined in the database as ID generator for entities.

- Fixed the behavior of the Restful Request Response so that now considered response content type is always the one present in the model, despite the one contained in the web service response.

- Fixed the generation of a web project. It is now possible to overwrite the default *.prepare.sql files with a set of custom file.

- Improved the definition of Attributes on Volatile Entities. You can set the Length Property (in the Mapping tab) in order to manage the size or scale of the data type.

- It's now possible to centralize some libraries, the hibernate-all.jar file is now split because previously caused exception on the application server.

- Fixed a problem related to the autocomplete event applied to a field of a Multiple Form. Now the generated JavaScript related to the autocomplete attached to the field uses correct ID.

FIXED IN 7.2.13

- Fixed a problem related to the Restful Request Response component. Now the component with form-url-encoded properly encodes the JSON parameters.

- Fixed a problem that prevented, in some situations, the administration section of a BPM application from properly updating the process metadata.

- Fixed the behavior of a Form component based on an entity bounded to an XSD. Now a restriction validation rule is evaluated only if the related field contains a value.

- Fixed the sorting of a Hierarchy component when the master entity is a volatile one. Now the alphabetical order is used and not the ASCII one.

- Improved the Request Response component. Now, to call PayPal services, the "TLSv1.2" protocol is used.

- Fixed issues related to the propagation of parameters between scripts and modules.

- Fixed the behavior of a PayPal component that prevented one from redirecting correctly after the payment operation when an ajax event was executed before the component.

- Fixed a problem related to a passing value with the NoOp component that prevented the propagation of the value between pages.

- Fixed issues related to the propagation of parameters between Get components and other operations.

- Fixed the Hierarchy component so that now it is possible to bind the "block index" parameter both in input and output for the component.

- The Apache Commons Collections has been upgraded to version 3.2.2 to fix the vulnerability "CVE-2015-6420".

FIXED IN 7.2.12

- Fixed a runtime problem that rewrote the URL of links containing JavaScript code. Now those links are no longer rewritten and their URL remains properly written on the page.

- Improved the Request Response Component. It is now supported the NTLM Authentication.

- Fixed a problem with the Excel Component that prevented one from properly importing a column containing numeric values but had the "string" type and the "text" formatting.

- Fixed a problem that prevented the simulation of the BPMN process from working properly when the process contained a message event published through SOAP.

- Fixed an issue involving the loading of an application deployed on JBoss EAP 6.4.0. Now you can run Enterprise Applications on this version of JBoss.

- Improved the algorithm of the logic generation of an operation group in a Web Project, which is now faster.

- Fixed a problem with the "Move To" option used in the Domain Model. Now all bindings involving entities that are moved inside or outside of packages are properly maintained.

- Fixed the "RestSummary" page so that Form Data Parameters defined on the Solicit Operation are now properly included in the Web Service description.

- Fixed the behavior of the Delete Operation working on objects having a composite key. Now the operation works properly, including when a composite key contains a column with a fixed length.

- Fixed a problem with the "Find Model Problem" option, which showed an error involving the context parameters, saying that it "is not implicitly set in a non-localized context" even if the "Localize" flag was properly set.

- Fixed a problem related to the logic generation of a Web Project caused by the excessive use of the flow analysis performed on the operation chains that returned a "Heap Space" error.

- Fixed a runtime behavior related to the propagation of parameters from a page to a Parameter Collector Component caused by a wrong key used to retrieve the form-bean in the presence of modules within an Enterprise Project.

- Fixed a problem related to the propagation of the values of Fields with a Visibility Condition that caused the wrong propagation of the page.

- Improved the support of the database used in a Web Project. Now it is available the "Oracle 12c" type.

- Improved the "Find Model Problems" function. It can identify and return an error related to a broken Derived Attribute defined on Volatile Entities.

- Fixed a problem with the Email Operation. Now images embedded in text/html content are properly handled and shown in the email message.

- Improved the "Find Model Problems" function. It can identify and return an error related to a key attribute selected with the distinct property checked on the Selector Component.

FIXED IN 7.2.11

- Fixed a problem of Excel Component placed in a Hybrid Module of an Enterprise project that created a file with only three records.

- Fixed a problem of page having the MergeReport template. Now the generated page works properly also when the CRC is enabled.

- Fixed a layout problem of the Rest Summary page. Now notes are properly visible.

- Fixed the behavior of the Solicit Operation when a malformed JSON is received. Now the operation properly returns a 500 Http Status Code.

- Fixed a problem related to RestFul Request Response Component that caused an infinite loop when the modeled response structure does not match the real response structure.

- Fixed a runtime behavior of the Enterprise Project when the session expires. Now the user is properly redirected to the login page, when the user tries to navigate to a different application of the Enterprise Project and the session is expired.

FIXED IN 7.2.10

- Improved the runtime behavior of the JobStatus Component. It is now possible to retrieve information about multiple scheduled jobs and their trigger instances.

- Improved the paging management with SQL Server databases in which the limit and offset parameters are not available in the Hibernate Dialect.

- Fixed a problem related to the structure of Request Response output containing a ComplexType with the extension node and nested sequence that caused an error when you invoke the Web Service.

- Improved the Solicit Operation. It is now possible to add a Soap Header element. The Soap Header element can refer to an XSD schema for its definition.

- Fixed a problem of blob imported attributes that prevented the file from being correctly seen in the generated Web Application.

- Fixed a problem of the Excel Component that prevented to write more than 6500 rows in the Excel file.

- Fixed a problem of the Solicit Operation that prevented a published REST Web Service from working properly when the request contains a query string parameter including the underscore character.

- Fixed the Find Model Problem command of the Domain Model so that it now reports relationship roles mapped on elements no longer present in the attached XSD schema.

- Fixed a problem related to Cookies that prevented their update from working properly when a null value was contained in them for certain information.

- Fixed the Visual Debug enabled on a Web Project that caused an exception when you applied a defined Deploy Configuration on the project.

- Fixed the runtime behavior for the Context Parameters stored in the Cookies. Now, the name of the context parameter is saved correctly in the cookies.

- Fixed a problem related to the XML I/O Component, associated with an Entity based on XSD that prevented importing a string longer than 255 characters.

- Fixed a problem of the Restful Request Response Component having "Binary Stream" value set for the "Response Content Type" property that prevented the Web Service from returning a PDF file.

- Fixed a problem of the "Database" bundle store type used in Enterprise Projects. Now, the "bundle.keys" file containing all the bundle keys is updated correctly with all keys.

- Fixed the file management related to blob attributes of volatile entities. Now, files are deleted when the session expires, as expected.

- Fixed the runtime behavior for a JSON I/O Operation with a subnode mapped on an entity in relation with the main entity. Now, the result JSON Document returns all nodes.

- Fixed the behavior of an Ajax page with more than one scrollable list. Now, the lists are properly refreshed when you change the status of one them.

- Fixed the runtime behavior for a selection field modeled in a Hybrid Module set as content page. Now, the onChange events are correctly fired and handled.

- Fixed a runtime behavior for the tooltip box applied on attributes of List Component. Now, the tooltip disappears correctly when the cursor moves beyond the attribute.

- Fixed a problem of the process diagram image in a BPM process that prevented the image from generating correctly in Linux and Mac OS environments.

- Fixed the runtime behavior for a form in a page with no focus on the fields. Now, if a popup is opened or an onChange event is run, the focus is not applied on the first field.

FIXED IN 7.2.9

- Fixed a problem of the JSON I/O Component that prevent to export an empty numeric attribute of the associated entity.

- Fixed a problem related to an entity in a package present in the Domain Model Project of the Enterprise Project when the project is synchronized with bpm. Now, if a business object is mapped on this entity works properly.

- Fixed the runtime behavior for an entity based Form. Now the "Validate" property is properly considered by the navigation flow when posting the form even when the entity used on the form is mapped on an XSD. If the "Validate" property is not checked, now validation rules are properly ignored when posting the form with the specific navigation flow.

- Improved the runtime behavior of the form validation. Now the validation patterns "\p{IsBasicLatin}" and "\p{IsBasicLatin}" contained in the XSD file, are properly supported.

- Fixed a problem of the Synchronize XSD command of the XSD Provider that prevent to import entities when the file is the result of the conversion from a DTD file to XSD.

- Fixed a problem of the Mail Component having attachments. Now when the "Multiple" value is set as Attachments Policy , files are sent each with its proper name.

- Fixed a problem of the "Complex Calculated Attribute" Derivation defined for an attribute to concatenate strings that caused an error during the export process to the database.

- Fixed a problem of the bundle data stored in the database used in Enterprise Projects that not always loaded the bundle set in the project, but sometimes loaded from file.

- Fixed a problem of the Hybrid Module modeled in the Module Project of the Enterprise Project that caused an error when you move from an Hybrid Module to another after opening a popup containing in other Hybrid Module.

- Fixed a problem of the Password Component that prevent to generate correctly password with old projects that used alphanumeric only.

- Fixed a problem related to the Restful Request Response Operation that returned a string instead of a Document for the "Response Message" Output Parameter.

- Fixed a problem related to a Web Service defined with protected URL and CRC enabled. If the custom property "protectURL=false" was set on the Service View, an error was reported when you called the Web Service.

- Improved the JSON I/O Component. Now the component supports class hierarchies.

- Fixed the behavior of the WebRatio Compare View when a Web Project is synchronized with the CVS repository. Now the Data Binding information of a Selector Component associated to a Class, contained in a package, are correctly shown.

- Fixed the problem related to the descriptor of Adapter and RequestResponse Components. The information about path of xsl transformation templates was wrongly generated when they are inherited from the Enterprise Project.

- Fixed a problem of the Synchronize XSD command of the XSD Provider that prevent to import the "choice" node defined in the XSD file.

THIRD PARTY LIBRARIES

Java libraries

Library	Link
Apache Commons BeanUtils Core 1.8.3 BeanUtils component provides easy-to-use wrappers around the Java language Reflection and Introspection APIs.	BeanUtils
Apache Commons Chain 1.1 Base API that facilitates using the "Chain of Responsibility" pattern, as described (among many other places) in the classic "Gang of Four" design patterns book, a popular technique for organizing the execution of complex processing flows is the.	Chain
Apache Commons Codec 1.4 Apache Commons Codec (TM) software provides implementations of common encoders and decoders such as Base64, Hex, Phonetic and URLs.	Codec
Apache Commons Collections 3.2.2 Commons-Collections builds upon the JDK data structures classes by providing new interfaces, implementations and utilities that accelerate development of most significant Java applications.	Collections
Apache Commons DBCP 1.3 This Commons package provides Database Connection Pools facilities.	DBCP
Apache Commons Digester 1.8.1 The Digester package lets configure an XML -> Java object mapping module, which triggers certain actions called rules whenever a particular pattern of nested XML elements is recognized.	Digester
Apache Commons FileUpload 1.3.3 The Commons FileUpload package makes it easy to add robust, high-performance, file upload capability to your servlets and web applications. FileUpload parses HTTP requests which conform to RFC 1867, "Form-based File Upload in HTML".	FileUpload
Apache Commons HttpClient 3.1 HttpClient provides an efficient, up-to-date, and feature-rich package implementing the client side of the most recent HTTP standards and recommendations.	HttpClient

Apache Commons IO 2.2	IO
Commons IO is a library of utilities to assist with developing IO functionality.	
Apache Commons Lang 2.5	Lang
Lang provides a host of helper utilities for the java.lang API, notably String manipulation methods, basic numerical methods, object reflection, concurrency, creation and serialization and System properties.	
Apache Commons Logging 1.1.1	Logging
The Logging package is an ultra-thin bridge between different logging implementations. A library that uses the commons-logging API can be used with any logging implementation at runtime.	
Apache Commons Net 1.4.1	Net
Apache Commons Net library implements the client side of many basic Internet protocols. The purpose of the library is to provide fundamental protocol access, not higher-level abstractions.	
Apache Commons Pool 1.5.5	Pool
Pool provides an Object-pooling API, with three major aspects: a generic object pool interface that clients and implementers can use to provide easily interchangeable pooling implementations, a toolkit for creating modular object pools, several general purpose pool implementations.	
Apache Commons Validator 1.3.1	Validator
This package addresses common issues when developing validation rules to verify the integrity of the data.	
Apache Derby 10.5.3.0	Derby
Apache Derby is an open source relational database implemented entirely in Java.	
Apache POI 3	POI
Apache POI is a facility to create and maintain Java APIs for manipulating various file formats based upon the Office Open XML standards (OOXML) and Microsoft's OLE 2 Compound Document format (OLE2).	
Apache Tika 1.9	Tika
The Apache Tika™ toolkit detects and extracts metadata and text from over a thousand different file types (such as PPT, XLS, and PDF). All of these file types can be parsed through a single interface, making Tika useful for search engine indexing, content analysis, translation, and much more.	
Apache Xalan 2.7.2	Xalan
Xalan-Java is an XSLT processor for transforming XML documents	

into HTML, text, or other XML document types. It implements XSL Transformations (XSLT) Version 1.0 and XML Path Language (XPath) Version 1.0.

cglib 2.2	[cglib](#)
cglib is a powerful, high performance and quality Code Generation Library used to extend JAVA classes and implements interfaces at runtime.	
Dom4j 1.6.1	[Dom4j](#)
dom4j is an open source library for working with XML, XPath and XSLT on the Java platform using the Java Collections Framework and with full support for DOM, SAX and JAXP.	
FontBox 1.5.0	[PDFBox](#)
FontBox is a Java library used to obtain low level information from font files. FontBox is a subproject of PDFBox.	
Groovy 1.5.6	[Groovy](#)
Groovy is an agile and dynamic language for the Java Virtual Machine that builds upon Java but has additional power features inspired by languages like Python, Ruby and Smalltalk and compiles straight to Java bytecode.	
Guava 11.0.1	[Guava](#)
The Guava project contains several of Google's core libraries: collections, caching, primitives support, concurrency libraries, common annotations, string processing, I/O, and so forth.	
H2 1.3.166	[H2](#)
H2 is a Java SQL in-memory databases database, very fast, open source, with JDBC API.	
Hibernate 3.5.6-FINAL	[Hibernate](#)
Hibernate is an object relational database persistence API that is the "de facto" standard for persistence in Java.	
iText 2.1.7	[iText](#)
iText is an open source library that allows to create and manipulate PDF documents.	
Jackson 1.9.10	[Jackson](#)
Jackson is a multi-purpose Java library for processing JSON data format.	
Jakarta BCEL 5.2	[BCEL](#)
The Byte Code Engineering Library (Apache Commons BCEL) is intended to give a convenient way to analyze, create, and manipulate (binary) Java class files.	

Jakarta Oro 2.0.8	Oro
The Jakarta-ORO Java classes are a set of text-processing Java classes that provide Perl5 compatible regular expressions, AWK-like regular expressions, glob expressions, and utility classes for performing substitutions, splits, filtering filenames, etc.	
JasperReports 4.0.1	Jasper
The JasperReports Library is the world's most popular open source reporting engine.	
JavaBeans Activation Framework 1.1.1	JavaBeans
JavaBeans Activation Framework offers services to determine the type of an arbitrary piece of data, encapsulate access to it, discover the operations available on it, and to instantiate the appropriate bean to perform said operation(s).	
JavaMail API 1.4.3	Mail
The JavaMail API is a set of abstract APIs that model a mail system. The API provides a platform independent and protocol independent framework to build Java technology based email client applications. The JavaMail API provides facilities for reading and sending email. Service providers implement particular protocols.	
Javassist 3.9.0.GA	Javaassist
Javassist (Java programming assistant) is a load-time reflective system for Java. It is a class library for editing bytecodes in Java.	
Jaxen 1.1.3	Jaxen
Jaxen is an open source XPath library written in Java. It is adaptable to many different object models, including DOM, XOM, dom4j, and JDOM.	
JCaptcha 1.0-RC6	Captcha
JCAPTCHA stands for Java Completely Automated Public Test to tell Computers and Humans Apart. It is the open source java framework for captcha definition and integration.	
JCommon 1.0.15	JCommon
JCommon is a Java class library that contains miscellaneous classes for chart displaying.	
JempBox 1.5.0	PDFBox
The Apache JempBox library is an open source Java tool that implements Adobe's XMP(TM) specification. JempBox is a subproject of Apache PDFBox.	
JFreeChart 1.0.12	FreeChart
JFreeChart is a Java chart library that makes it easy for developers to display professional quality charts in their applications.	

JSR 305 1.3.9	JSR305
JSR 305 works to develop standard annotations (such as @NonNull) that can be applied to Java programs to assist tools that detect software defects.	
Log4j 2.17	Log4j
Apache log4j is a logging library for Java.	
Lucene 2.3.2	Lucene
The Apache LuceneTM project develops open-source search software.	
MIMEpull 1.5	MIMEPull
Mimepull project provides a streaming API to access attachments parts in a MIME message.	
PDFBox 1.5.0	PDFBox
Apache PDFBox is an open source Java tool for working with PDF documents. This project allows creation of new PDF documents, manipulation of existing documents and the ability to extract content from documents.	
Quartz 2.1.6	Quartz
Quartz is a richly featured, open source job scheduling library that can be integrated within virtually any Java application - from the smallest stand-alone application to the largest e-commerce system.	
Slf4j API 1.6.0	Slf4j
The Simple Logging Facade for Java is a simple facade or abstraction for various logging frameworks (e.g. java.util.logging, logback, log4j) allowing the end user to plug in the desired logging framework at deployment time.	
Slf4j Log4j12 1.5.8	Slf4j
This project is a Slf4j wrapped implementation of Log4j.	
Standard Taglib 1.1.2	Taglib
Apache Standard Taglib is an implementation of the JSP Standard Tag Library specification.	
Stax API 1.0	Stax
Stax is a standard XML processing API that allows to stream XML data from and to your application.	
Struts Core 1.3.10	Struts
Apache Struts is a free, open-source, MVC framework for creating elegant, modern Java web applications.	
WSDL4j 1.6.2	WSDL4j
The Web Services Description Language for Java (WSDL4J) allows the creation, representation, and manipulation of WSDL	

documents. Is the reference implementation for JSR110 'JWSDL' (jcp.org).	
XMLBeans 2.3	XMLBeans
XMLBeans is a technology for accessing XML by binding it to Java types.	
XMLSchema 1.4.7	XMLSchema
XMLSchema is a lightweight Java object model that can be used to manipulate and generate XML schema representations.	

JavaScript libraries

Library	Link
CKeditor 4.3	CKeditor
CKEditor is a ready-for-use HTML text editor designed to simplify web content creation. It's a WYSIWYG editor that brings common word processor features directly into web pages.	
ExplorerCanvas 3	Canvas
ExplorerCanvas bring HTML5 canvas tag functionalities to Internet Explorer.	
Flotr2	Flotr2
Flotr2 is a plotting library to draw HTML5 charts and graph.	
jQuery Timepicker Addon 1.4.5	Timepicker
Timepicker extends jQuery UI Datepicker for entering times and timestamps.	
JQuery 1.7.2	JQuery
jQuery is a fast, small, and feature-rich JavaScript library with an easy-to-use API that works across a multitude of browsers.	
JQuery BBQ 1.2.1	BBQ
BBQ is a jQuery plug-in that allows simple, yet powerful bookmarkable #hash history.	
JQuery Cookie 1.3.1	Cookie
A simple, lightweight jQuery plugin for reading, writing and deleting cookies.	
JQuery DialogExtend 1.0	Dialog
A simple, lightweight jQuery plugin for Maximizing and Minimizing Buttons for UI Dialog.	
JQuery UI 1.9.2	UI
jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library.	
JsDump 1.0.0	JSDump
This script library can dump any type of Javascript data (or most), thus, generating a string out of the received information.	
JsHashtable 2.1	JSHashtable
JsHashtable is a standalone implementation of hash table in JavaScript.	
ParseURI 1.2.2	ParseURI
Split URLs in JavaScript.	
Respond.js 1.4.0	Respond
Respond is a fast & lightweight polyfill for min/max-width CSS3	

Media Queries (for IE 6-8, and more).

Underscore 1.3.3	Underscore
Underscore is a utility-belt library for JavaScript that provides a lot of the functional programming support like both the usual functional suspects: map, select, invoke - as well as more specialized helpers: function binding, javascript templating, deep equality testing, and so on.	
Underscore String 2.2.0rc	Underscore
Underscore String is a String manipulation extensions for Underscore.	
XRegExp 2.0.0	XRegExp
XRegExp is an open source JavaScript library that provides augmented and extensible regular expressions.	

SYSTEM REQUIREMENTS

WebRatio Development Platform

Operating System

- Microsoft Windows XP, Vista or higher, 32 or 64 bit

- Linux, 32 or 64 bit

- Mac OSX 10.7 (Lion) or higher, 32 or 64 bit

Installing WebRatio Development Platform also requires:

- Disk Space: 600 MB

- RAM: 4 GB minimum, suggested is 8 GB

Java

When running WebRatio Development Platform on a computer you will need the Java Runtime Environment (JRE) 1.7.
WebRatio 7.x does not support jdk version 1.8 or higher.
You can download JDKs here.

WebRatio Generated Applications

Application Server

When running WebRatio Generated Server Applications on a server you will need a Java Application Server compatible with Servlet 3.0 and JSP 2.2.

Java

When running WebRatio Generated Server Applications on a server you will need the Java Runtime Environment (JRE) 1.8 or higher.

You can download JDKs here.

learn.webratio.com