



ZENworks Mobile Workspace

BRMS Developer Guide

Version 3.16.3 - January 2018

Copyright © Micro Focus Software Inc. All rights reserved.

Table of Contents

Overview.....	1
Introduction.....	1
Rules development.....	1
Available conditional elements	1
User information model.....	2
Mobile context model.....	2
Time context model.....	2
Location context model.....	2
Network context model.....	2
Available action elements	3
Session configuration (in SessionContextHolder).....	3
Security policies configuration (in SessionContext).....	3
Access control configuration (in SessionContext).....	3
Drools Rule Language (DRL).....	3
Drools workbench.....	4
Excel spreadsheet (XLS, XSLX).....	4
ZENworks Mobile Workspace configuration.....	6
Upload rules	6
Enable rules.....	6

Overview

This configuration guide helps developer to create business rules that can be interpreted by ZENworks Mobile Workspace rule engine. Therefore, access rights and security policies can be modified on the fly based on the mobile context and user’s metadata.

Introduction

ZENworks Mobile Workspace security server embeds a business rule engine (BRMS) called **Drools**. Modified to fit ZENworks Mobile Workspace requirements, it allows to define security policies that cannot be defined directly with ZENworks Mobile Workspace. This is useful when:

- Security settings must change depending of the context
- Security settings must be set at a different level than ZENworks Mobile Workspace Security Group (e.g. user)
- Security settings must be applied depending of external parameter (e.g. user’s calendar)

For ZENworks Mobile Workspace, the following concept has been developed: As an input, the application gather all information it can retrieved on the device and in the server configuration to create a request context. On the other hand, developers or business analysts create rules, based on this context, to modify on the fly security policies and access rights.

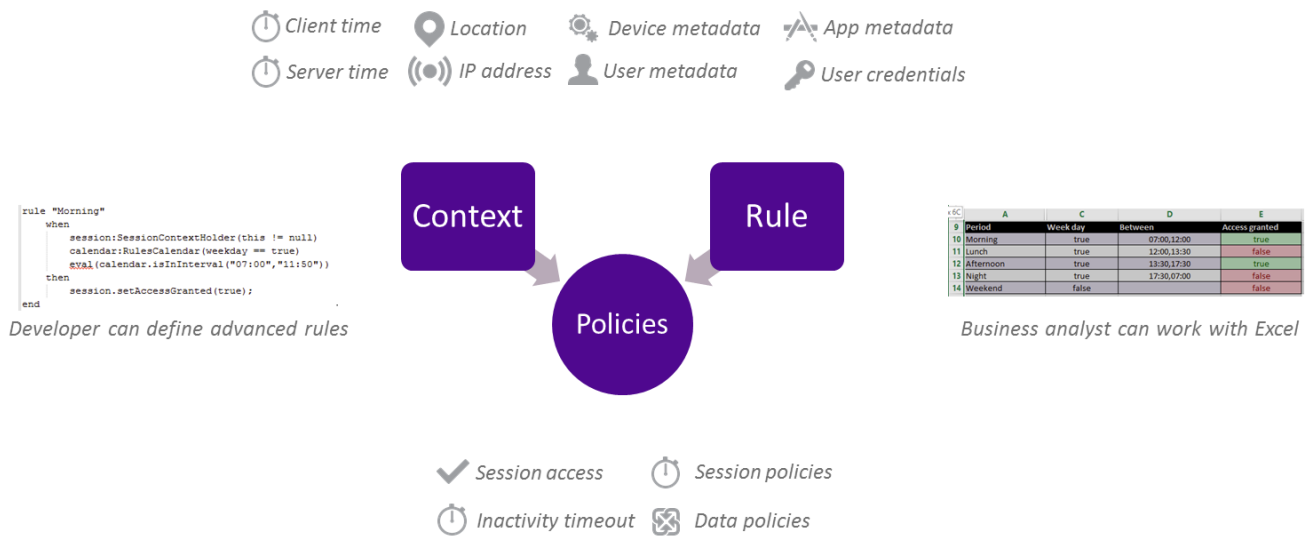


Figure 1. Concept

Rules development

Available conditional elements

This section list all objects/attributes that can be used to take decision.

User information model

- **Username:** MobileUserLogin.name
- **Group name:** UserGroup.name
- **Domain name:** UserDomain.domainIdentifier
- **Password:** SecurityPassword.password

Mobile context model

- **OS name:** DeviceInformation.Name.SystemName
- **OS version:** DeviceInformation.Name.SystemVersion
- **Device model:** DeviceInformation.Name.Model
- **Device Name:** DeviceInformation.Name.Name

Time context model

- **Current hour:** RulesCalendar.hour
- **Current day:** RulesCalendar.day
- **Check interval:** RulesCalendar.isInInterval
- **Check weekend:** RulesCalendar.isWeekend
- **Check weekday:** RulesCalendar.isWeekday

Location context model

- **ISO 3166 2-letter code:** RulesCountry.code
- **Full Country Name:** RulesCountry.name
- **Latitude:** GeoLocation.coordA
- **Longitude:** GeoLocation.coordB
- **Altitude:** GeoLocation.coordC
- **Accuracy:** GeoLocation.uncertainty
- **Coordinate system:** GeoLocation.crs

Network context model

- **Host address:** IpAddress.hostAddress
- **Check if public address:** IpAddress.isPublicAddress
- **Check if private address:** IpAddress.isPrivateAddress
- **Check if lookback address:** IpAddress.isLoopbackAddress
- **Check if link local address:** IpAddress.isLinkLocalAddress
- **Check if site local address:** IpAddress.isSiteLocalAddress

Available action elements

This section list all objects/attributes that can be modified depending on the condition.

Session configuration (in SessionContextHolder)

- **Grant access (boolean):** SessionContextHolder.accessGranted
- **Access denied reason (AccessDeniedReason):** SessionContextHolder.accessDeniedReason
 - **UNTRUSTED_DEVICE:** Set when the access is denied because of a forbidden device.
 - **UNTRUSTED_NETWORK:** Set when the access is denied because of the network from where the request is coming.
 - **DISALLOWED_APPLICATION:** Set when the access is denied because the application is not allowed for the user.
 - **LOCATION_REQUIRED:** Set when the access is denied because the location of the user is required.
 - **NO_REASON:** Set when above reasons does not fit the situation.
- **Session context (SessionContext):** SessionContextHolder.sessionContext

Security policies configuration (in SessionContext)

- **User inactivity timeout (long):** SessionContext.userInactivityTimeout
- **Session inactivity timeout (long):** SessionContext.sessionInactivityTimeout
- **Background execution (boolean):** SessionContext.backgroundExecutionEnabled
- **Max. wrong credentials (int):** SessionContext.maximumAuthenticationAttempts
- **Offline access (boolean):** SessionContext.offlineAccessEnabled
- **Offline access timeout (long):** SessionContext.offlineAccessValidity
- **Location required (boolean):** SessionContext.locationRequired
- **Storage availability (boolean):** SessionContext.localStorageEnabled
- **Push notification (boolean):** SessionContext.pushNotificationsEnabled

Access control configuration (in SessionContext)

- **Workspace components (List):** SessionContext.authorizedApps
- **Proxy URL (List):** SessionContext.serviceEndpoints

Drools Rule Language (DRL)

Drools is the rule engine embedded in ZENworks Mobile Workspace which understand its own “native” rule language. This document does not aims to teach developers how to write rules as details document is available here: <https://docs.jboss.org/drools/release/5.2.0.Final/drools-expert-docs/html/ch05.html>.

Using drools language is for expert that have knowledge in Java development and want to create complex rules that may require external libraries. Please refer to existing samples for quick start.

Drools workbench

Drool comes with a workbench allowing developer to create and edit rules with a developer friendly interface:

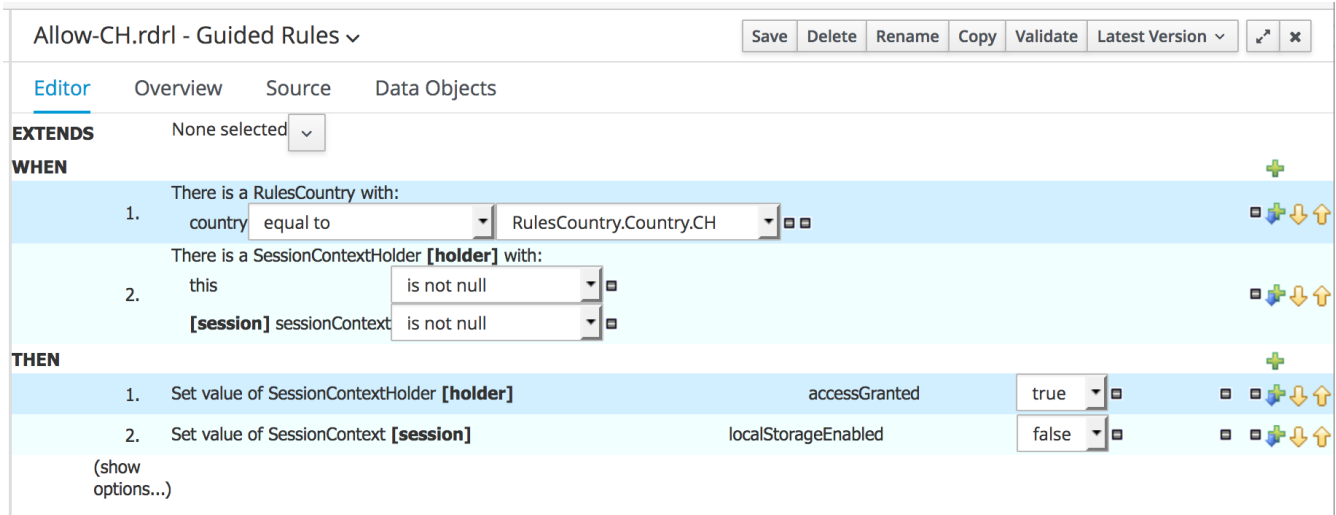


Figure 2. Drools workbench

Automatically generating drool rule file:

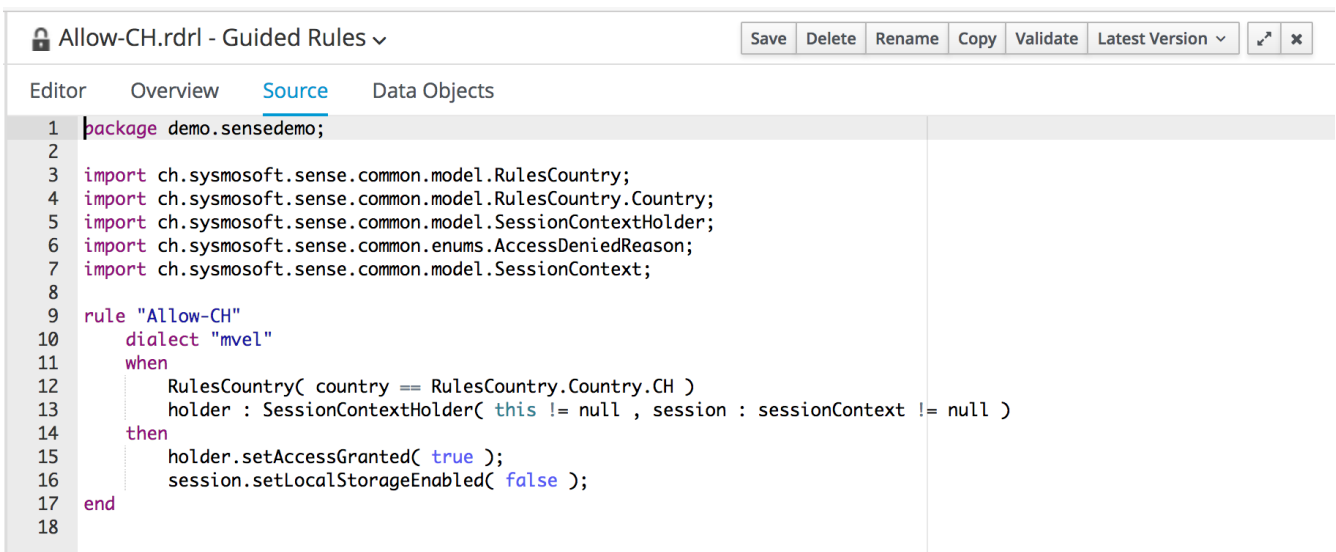


Figure 3. Drools rule file

To quickly install the workbench, the best is to install [Docker](#) and install the [drools workbench showcase](#) image.

Excel spreadsheet (XLS, XSLX)

As we have seen in the previous chapter, drool language must be created by developers. However, most of the time, they are not those who will match business needs and rules. Business analyst for example knows the business needs but are unlikely to develop a rule. Therefore, developer needs to provide them an easiest way to create rules.

Here again, this document does not aim to teach developers how to write rules in Excel as details document is available here: <https://docs.jboss.org/drools/release/5.2.0.Final/drools-expert-docs/html/ch06.html>

Setup of these files must still be done by java developer but when it is ready, the Excel file can be provided to business analyst. Please refer to existing samples for quick start.

	A	C	D	E
1	RuleSet	ch.sysmosoft.drool		
2	Import	ch.sysmosoft.sense.common.model.RulesCalendar, ch.sysmosoft.sense.common.model.SessionContextHolder, ch.sysmosoft.sense.common.enums.AccessDeniedReason		
3	Notes	Working hours		
4				
5	RuleTable Access during working hours			
6	NAME	CONDITION	CONDITION	ACTION
7		calendar:RulesCalendar		
8		weekday == \$param	eval(calendar.isInInterval("\$1","\$2"))	session.setAccessGranted(\$param); session.setAccessDeniedReason(AccessDeniedReason.NO_REASON);
9	Period	Week day	Between	Access granted
10	Morning	true	07:00,12:00	true
11	Lunch	true	12:00,13:30	false
12	Afternoon	true	13:30,17:30	true
13	Night	true	17:30,07:00	false
14	Weekend	false		false

Figure 4. Setup of the decision table

	A	C	D	E
9	Period	Week day	Between	Access granted
10	Morning	true	07:00,12:00	true
11	Lunch	true	12:00,13:30	false
12	Afternoon	true	13:30,17:30	true
13	Night	true	17:30,07:00	false
14	Weekend	false		false

Figure 5. Hide complexity for business analysts

The current version of ZENworks Mobile Workspace is not able to upload Excel spreadsheet directly. It must be converted with drools workbench into a DRL file.

```

working hours.xlsx - XLSX Decision Tables
Delete Rename Copy Validate Convert Latest Version

Editor Overview Source
1 package ch.sysmosoft.drool;
2 //generated from Decision Table
3 import ch.sysmosoft.sense.common.model.RulesCalendar;
4 import ch.sysmosoft.sense.common.model.SessionContextHolder;
5 import ch.sysmosoft.sense.common.enums.AccessDeniedReason;
6 // rule values at A10, header at A5
7 rule "Morning"
8     when
9         session:SessionContextHolder(this != null)
10        calendar:RulesCalendar(weekday == true)
11        eval(calendar.isInInterval("07:00","12:00"))
12    then
13        session.setAccessGranted(true);
14    session.setAccessDeniedReason(AccessDeniedReason.NO_REASON);
15    end
16
17 // rule values at A11, header at A5
18 rule "Lunch"
19     when
20        session:SessionContextHolder(this != null)
21        calendar:RulesCalendar(weekday == true)
22        eval(calendar.isInInterval("12:00","13:30"))

```

Figure 6. Excel based generated rule

ZENworks Mobile Workspace configuration

Once the rule file has been created, it must be uploaded in the security server administration console. Full description can be found in the **ZENworks Mobile Workspace Security Server: Administration Guide**. Here are the summary steps:

Upload rules

1. Log in the ZENworks Mobile Workspace administration console as an administrator.
2. Click on the **GENERAL** menu.
3. Go under the **Contextual rules** tab.
4. Create a new rule.
5. Put a **name** and a **version**.
6. Upload your rule **file**.

Enable rules

1. Log in the ZENworks Mobile Workspace administration console as an administrator.
2. Click on the **GENERAL** menu.
3. Go under the **Security settings** tab.
4. Edit the desired settings (or create a new one).
5. In the **Rules package** section, **Enable rules execution ?** by checking the box.
6. Select the desired rule by clicking in the radio button.