ZENworks Mobile Workspace

BRM Developers Guide

May 2017

Legal Notice

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see https://www.novell.com/company/legal/.

Copyright © 2017 Micro Focus Software Inc. All Rights Reserved.

TABLE OF CONTENTS

1 Overview 3

2 Introduction 3

2.1 Overview 3

3 Managing rules 4

3.1 Upload rules 4

3.2 Enable rules 4

4 Rules development 4

4.1 Available conditional elements 4

4.2 Available action elements 6

4.3 Drools Rule Language (DRL) 7

4.4 Excel spreadsheet (XLS, XSLX) 8

1 OVERVIEW

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

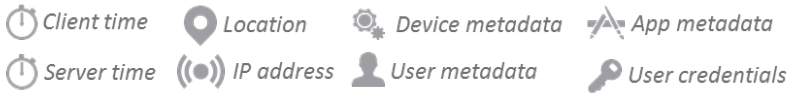
2 INTRODUCTION

ZENworks Mobile Workspace security server embeds a business rule engine (BRMS) called Drools (https://www.drools.org/). 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 other than the ZENworks Mobile Workspace Security Group (e.g. user)
- Security settings must be applied depending on external parameters (e.g. user's calendar)

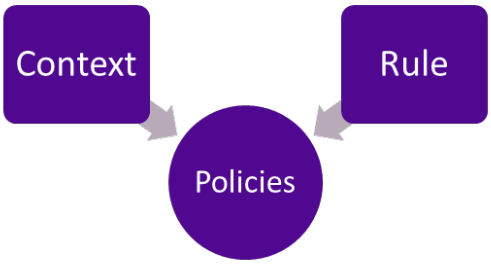
2.1 Overview

For ZENworks Mobile Workspace, the following concept has been developed: As an input, the application gathers all the information that it can retrieve from the device and 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.




```
rule "Morning"
    when
        session:SessionContextHolder(this != null)
        calendar:RulesCalendar(weekday == true)
        eval(calendar.isInInterval("07:00","11:50"))
    then
        session.setAccessGranted(true);
end
```

Developer can define advanced rules



	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

Business analyst can work with Excel

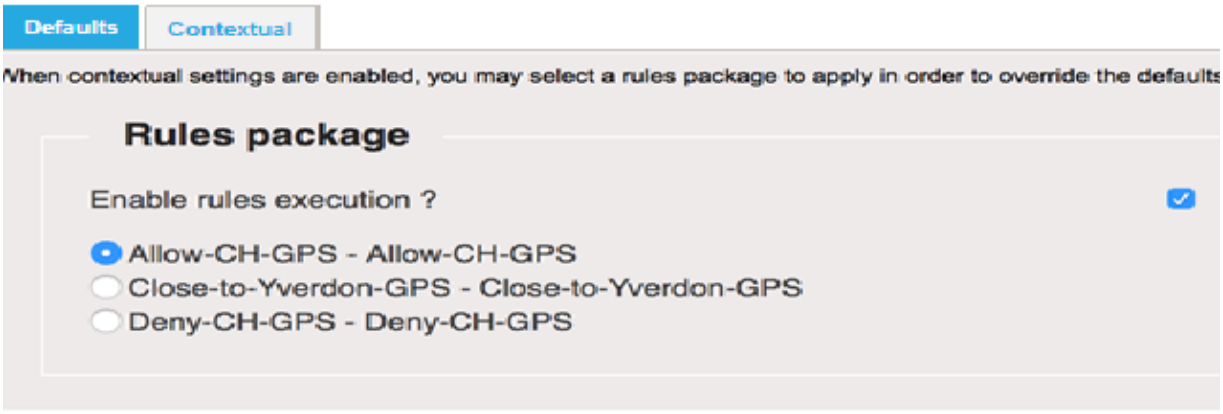
3 MANAGING RULES

3.1 Upload rules

1. Connect to the security server and go to SETTINGS -> Contextual rules
2. Click the add icon and enter the name and the version
3. Select the drl file and create the rule

3.2 Enable rules

1. Connect to the security server and go to SETTINGS -> Security settings
2. Edit the security settings that must use contextual rule and go to Contextual tab
3. Check “Enable rules execution” and select which rule must be executed



4 RULES DEVELOPMENT

4.1 Available conditional elements

This section lists all objects and attributes that can be used to take decisions.

4.1.1 User information model

- **Username**: MobileUserLogin.name

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

4.1.2 Mobile context model

Device information (DeviceInformation):

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

4.1.3 Time context model

Calendar rules (RulesCalendar)

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

Client time

4.1.4 Location context model

IP based (RulesCountry)

- **ISO 3166 2-letter code**: RulesCountry.code
- **Full Country Name**: RulesCountry.name

GPS based (GeoLocation)

- **Latitude**: GeoLocation.coordA
- **Longitude**: GeoLocation.coordB
- **Altitude**: GeoLocation.coordC
- **Accuracy**: GeoLocation. uncertainty
- **Coordinate system**: GeoLocation.crs

4.1.5 Network context model

IP (IpAddress)

- 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

4.2 Available action elements

This section lists all objects and attributes that can be modified depending on the condition.

4.2.1 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 the above reasons do not fit the situation.
- **Session context (SessionContext)**: SessionContextHolder.sessionContext

4.2.2 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

4.2.3 Access control configuration (in SessionContext)

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

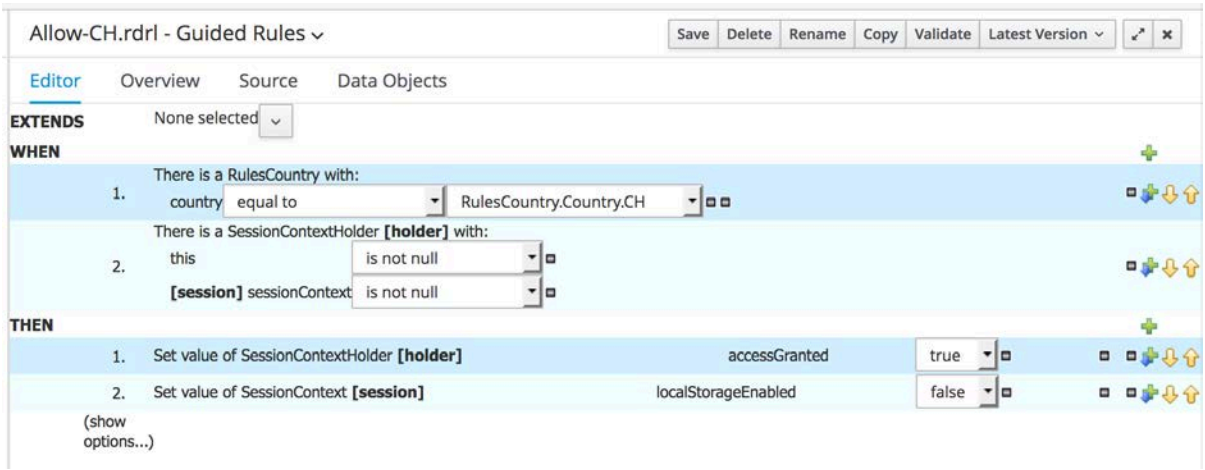
4.3 Drools Rule Language (DRL)

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

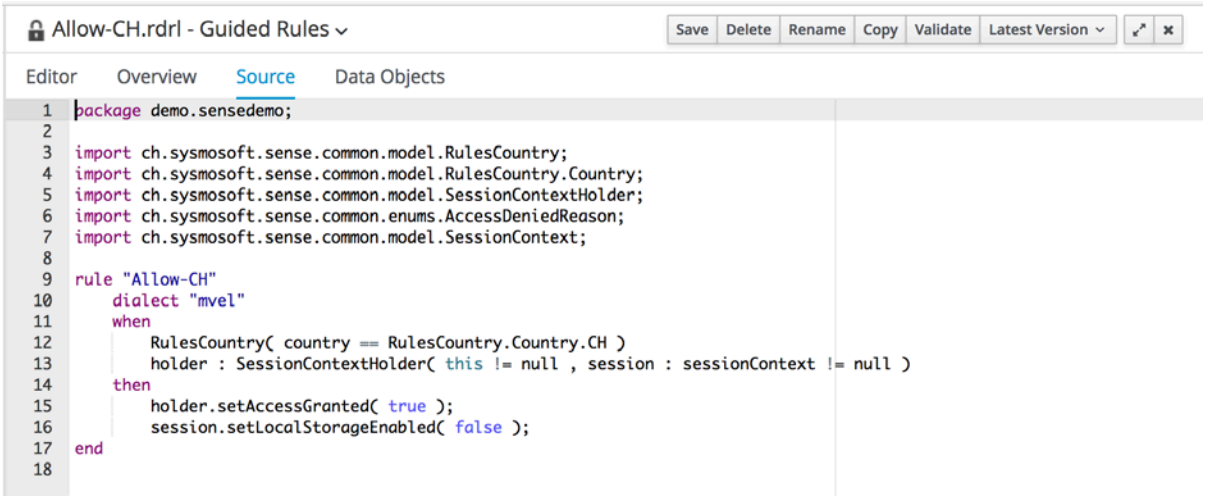
Using the drools language is for experts that have knowledge in Java development and want to create complex rules that may require external libraries. Please refer to the existing samples for a quick start.

4.3.1 Drools workbench

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



Automatically generating drool rule file:



To quickly install the workbench, the best option is to install Docker (https://www.docker.com/) and install the drools workbench showcase image: https://hub.docker.com/r/jboss/drools-workbench/

4.4 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. A Business Analyst for example, knows the business needs but is unlikely to develop a rule. Therefore, developers need to provide them with an easy way to create rules.

Here again, this document does not aims to teach developers how to write rules in Excel as a detailed 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 a Java developer but when it is ready, the Excel file can be provided to the business analyst. Please refer to existing samples for a quick start.

4.4.1 Setup of the decision table

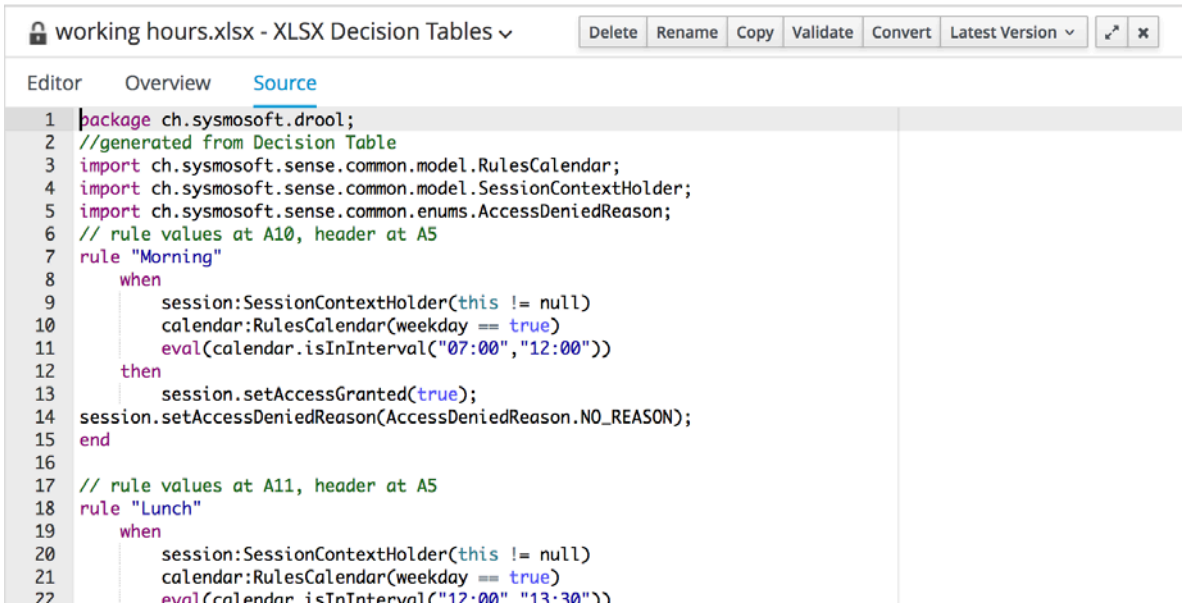
	A	C	D	E
1	**RuleSet**	ch.sysmosoft.drool		
2	**Import**	ch.sysmosoft.sense.common.model.RulesCalendar, ch.sysmosoft.sense.common.model.SessionContextHolder, ch.sysmosoft.sense.common.enums.AccessDeniedReason		
3	**Notes**	Working hours		
4				
5	**RuleTable Access during working hours**			
6	**NAME**	**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

4.4.2 Hide complexity for business analysts

	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

4.4.3 Generated rules

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




```
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"))
```