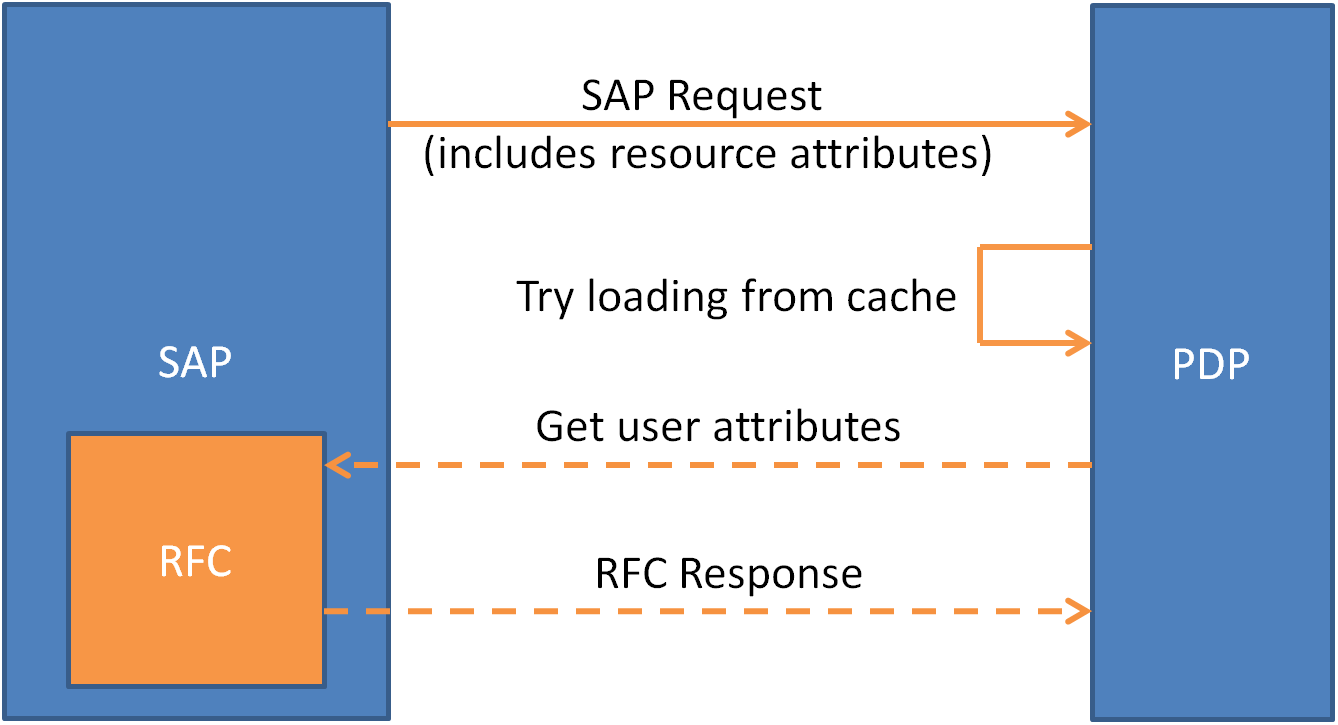
Cache SAP User Attribute Provider Plugin

User Guide

# Overview

## User Attribute Provider with Cache



## Invoking Sequence

1. Initial SAP request does not include user.mrparea.
2. Policy Controller requires value of user.mrparea for policy evaluation.
3. Policy Controller invokes user attribute plugin to obtain user.mrparea value.
4. User Attribute Provider plugin checks cache for user.mrparea.
5. If available, cached value of user.mrparea is returned.
6. Otherwise, the User Attribute Provider plugin makes Remote Function Call to SAP system to retrieve user.mrparea.

## Behavior

This User Attribute Provider plugin, as with the normal attribute provide plugins, is triggered by the Policy Controller when a required attribute value is unknown. Triggering the plugin is not in the control of the plugin itself.

However, this plugin will only process request for attribute of mrparea, in which proper values will be returned to the Policy Controller. Otherwise, the plugin will return an empty Multivalue to the Policy Controller.

Also, by default, the plugin will return an empty Multivalue to the Policy Controller when encountering Errors, Exceptions and Failures from calling the SAP Remote Function Calls, retrieving the values from the SAP system or parsing the retrieved values.

Policies should be designed and written in a way that takes care of the condition of an empty Multivalue to cater for unexpected errors and exceptions.

The flow of the User Attribute Provider plugin can be summarized in the diagram below.



Figure 1 - Plugin Flow

Table of Contents

[Cache SAP User Attribute Provider Plugin 1](#_Toc461543875)

[Overview 1](#_Toc461543876)

[User Attribute Provider with Cache 1](#_Toc461543877)

[Invoking Sequence 1](#_Toc461543878)

[Behavior 1](#_Toc461543879)

[Installation & Deployment 4](#_Toc461543880)

[Uninstallation 4](#_Toc461543881)

[Usage Instruction 5](#_Toc461543882)

[Configuration 5](#_Toc461543883)

[Policies 7](#_Toc461543884)

[Test Instruction 7](#_Toc461543885)

[Invoking the Attribute Provider 7](#_Toc461543886)

[Reading Log 7](#_Toc461543887)

[Cache Status 8](#_Toc461543888)

[Cache Configuration 8](#_Toc461543889)

[Loading from SAP or Cache 8](#_Toc461543890)

[Raw MRP Area List from SAP (XML) 8](#_Toc461543891)

[Parsed MRP Area List from SAP or Cache 9](#_Toc461543892)

[Processed MRP Area Multivalue for Policy Controller Evaluation 9](#_Toc461543893)

[Time Taken 9](#_Toc461543894)

[Attribute Handling 10](#_Toc461543895)

# Installation & Deployment

Each attribute plugin artifact is packed as a zip file. The zip file contains the following files:

* Jar file (CacheSAPUserAttributeProvider.jar)
* Configuration file (CacheSAPUserAttributeProvider.properties)
* This readme document (ReadMe.docx)

Follow these steps to Install and Deploy the user attribute provider plugin into the Policy Controller.

1. Stop the Policy Controller.
2. Copy CacheSAPUserAttributeProvider.jar to [PC Home]\jservice\jar\
3. Copy CacheSAPUserAttributeProvider.properties to [PC Home]\jservice\config\
4. Update the configuration in the Configuration file to match the deployment system.
5. Start the Policy Controller.
6. The user attribute provider plugin is now installed and deployed.

# Uninstallation

Follow these steps to uninstall the user attribute provider plugin from the Policy Controller.

1. Stop the Policy Controller.
2. Remove CacheSAPUserAttributeProvider.jar from [PC Home]\jservice\jar\
3. Remove CacheSAPUserAttributeProvider.properties from [PC Home]\jservice\config\
4. Start the Policy Controller.
5. The user attribute provider plugin is now uninstalled.

# Usage Instruction

## Configuration

The user attribute plugin can be configured by editing the CacheSAPUserAttributeProvider.properties file.

The available configurations and their effects are listed in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties** | **Description** | **Possible Values** | **Value Effect** |
| jar-path | Sets the path for the user attribute provider Jar file. | Absolute or relative path from [Nextlabs] to the jar file. | The Jar file in the path provided will be used. |
| use\_cache | Sets whether cache will be used in the user attribute provider plugin. This may impact the evaluation response time and the total memory consumption of the Policy Controller. | YES | Cache will be used. |
| NO | Cache will not be used. |
| time\_to\_idle\_cache\_duration | Sets the duration before the cache item getting expired and evicted from the cache starting from the point it is left idle (cache item not accessed). This value is defined in number of seconds as an Integer. This value has to be SMALLER or EQUAL (<=) to time\_to\_live\_cache\_duration. | Integer (No decimal point).  E.g. 60 | Cache item will be expired and evicted 60 seconds after last action performed on the item. |
| time\_to\_live\_cache\_duration | Sets the maximum valid duration of cache item before getting expired and evicted from the cache, regardless of whether the cache item is idle. This value is defined in number of seconds as an Integer. This value has to be GREATER or EQUAL (>=) to time\_to\_idle\_cache\_duration. | Integer (No decimal point).  E.g. 300 | Cache item will be expired and evicted 300 seconds after getting inserted into the cache. |
| sap\_handler | The SAP Remote Function Call handler. Consult the SAP administrator for the handler value. | String.  E.g. /NXLDFPS/GET\_USER\_ATTRS | The user attribute provider plugin will call the /NXLDFPS/GET\_USER\_ATTRS handler over the Remote Function Call when retrieving the user MRP Area list. |
| sap\_server\_prefix | The server prefix for SAP server configuration defined in SAPJavaSDKService. Refer to SAPJavaSDKService.properties for the server prefix.  This value should contain only a single value, commonly suffixed **with** underscore (‘\_’) and **without** semicolon (‘;’). This value has to be exact or the plugin may not work correctly. | String.  E.g. SERVSWAF\_ | The SERVSWAF\_ server configuration will be used to communicate with the SAP system. |
| sap\_auth\_value | The Authorization Object for the Remote Function Call. Consult the SAP administrator for the authorization object value. | String.  E.g. DF\_BERID | The value DF\_BERID will be passed alongside the query to the SAP system during the Remote Function Call. |

## Policies

The user attribute provider plugin will only be triggered when the user attribute user.mrparea is required and not found in the original request from the Policy Enforcer.

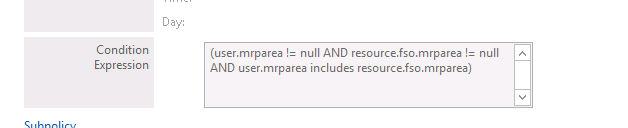
This plugin will return NULL when invoked with other user attribute query.

To create policy that will invoke the user attribute plugin, an advanced condition expression that evaluates user.mrparea needs to be used.

For example, the following policy can be used.

**FOR** TRUE **ON** TRUE **TO** TRUE  
**BY** user.adgroup = "Restricted"  
**WHERE**   
(TRUE AND (TRUE AND (user.mrparea != null AND resource.fso.mrparea != null AND user.mrparea includes resource.fso.mrparea)))  
**DO** allow **ON** deny **DO** log

From the Policy Studio, the following can be seen when implementing the policy example shown above.



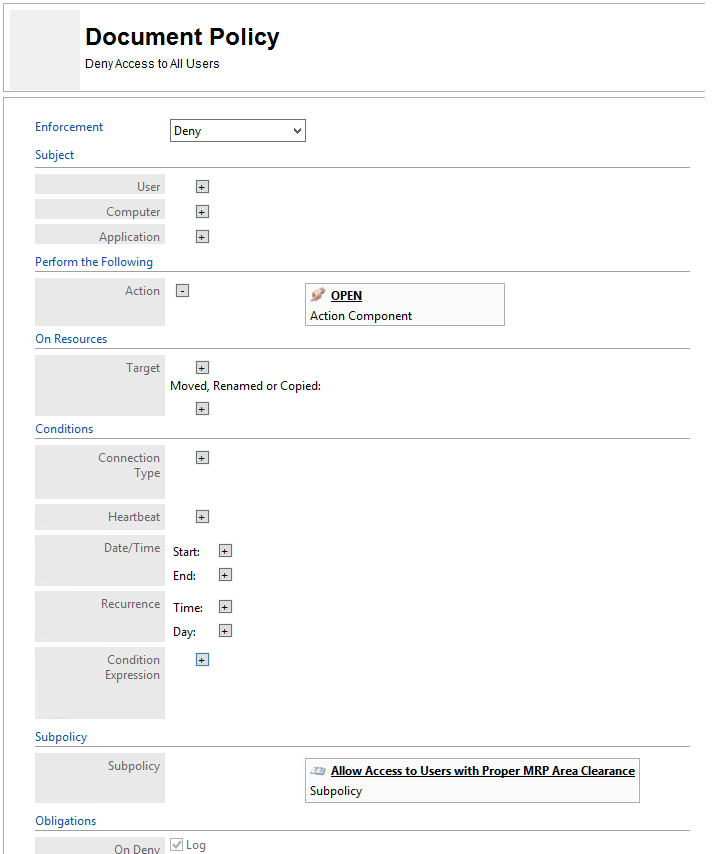
# Policy Example

As an example to demonstrate the usage of the plugin, the following policies can serve as a guide.

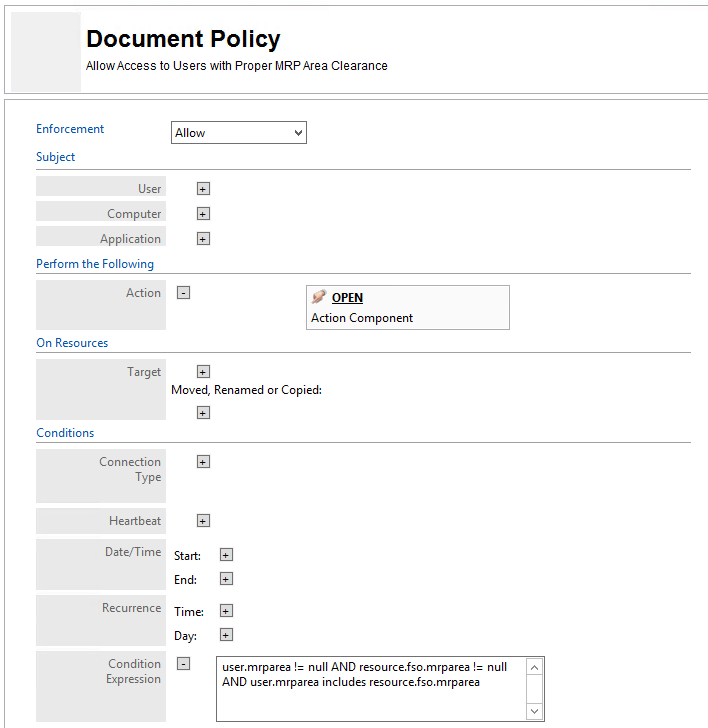
Suppose that the end goal is to **only allow users to access (Open) the MRP Area resource that they have clearance to access**, the following policies can be created.



## Parent Policy: Deny Access to All Users



## Sub Policy: Allow Access to Users with Proper MRP Area Clearance



# Test Instruction

## Invoking the Attribute Provider

To invoke the User Attribute Provider, a policy evaluation on the Policy Controller has to be invoked with a **non-existent** user.mrparea attribute on a policy which requires the user.mrparea attribute.

The policy evaluation can be invoked via a Policy Enforcer, for example, the SAP Entitlement Manager. Also, it may be invoked using the Nextlabs Policy Validator software with the appropriate configuration.

## Reading Log

To test and debug the User Attribute Provider plugin, the log serves as the primary tool to understand the condition, state and values returned by the plugin.

The following shows the list of important log statements provided by the User Attribute Provider plugin and their significant contribution in test and debug sessions.

To ensure a more thorough log, use the **FINEST** logging level.

### Cache Status

To check whether the cache is enabled and being used, the following logs can be referenced.



Figure - Cache Enabled



Figure - Cached Disabled

This should reflect the same configuration as configured in the CacheSAPUserAttributeProvider.properties file, under the use\_cache property.

### Cache Configuration

To check the cache configuration loading status, the following shows the values (in seconds) loaded into the plugin. It should match the value stated in the properties file.



Figure - Cache Configuration

### Loading from SAP or Cache

To ensure the Cache is working correctly, the plugin can be invoked multiple times at carefully selected intervals to check the source of user attribute data. The time (highlighted in the figures) can be used to check against the expected expiration time of the cached item for proper behavior. The source of data may be either the SAP system (via Remote Function Call) or the Cache as shown in figures below.



Figure - Loading from SAP



Figure - Loading from Cache

If working correctly, the duration between two calls to the SAP system should be more than the duration set in the configuration file, under the cache\_duration property. Do note that this property is defined in **number of seconds**.

### Raw MRP Area List from SAP (XML)

For checking the MRP area list returned from the SAP system, the following log item can be referenced.



Figure - Raw XML Data from SAP

This log item shows the raw XML data returned from the SAP system.  
The “MDLG\_OUT” field is expected to exist for the plugin to extract the list of MRP areas from this XML data.

### Parsed MRP Area List from SAP or Cache

After checking the raw XML data, the following figure shows the parsed MRP Area list. This can be used to check against the raw data for missing or incorrect values.



Figure - Parsed MRP Area

### Processed MRP Area Multivalue for Policy Controller Evaluation

Finally, to return the MRP Area list to the Policy Controller for evaluation, the data has to be converted to a Nextlabs proprietary Multivalue item.

The following figure shows the log item of the Multivalue content after the conversion from the generic Java ArrayList.



Figure - Processed MRP Area Multivalue

This is the final value that will be returned to the Policy Controller for evaluation.

### Time Taken

To determine the time taken by the User Attribute Provider plugin, the total time taken in the plugin is logged. This is measured from the start of the plugin to the end of the plugin, right before returning the final value to the Policy Controller.

Also, the attribute name requested by the Policy Controller is also included to assist debugging and fine tuning as the User Attribute Provider plugin will only handle the mrparea attribute while returning an empty Multivalue otherwise.

C:\Users\ypoh\AppData\Local\Microsoft\Windows\INetCache\Content.Word\time_taken_mrp_area.png

Figure - TIme Taken (mrparea)



Figure - Time Taken (other attributes)

### Attribute Handling

To further confirm that the attribute name requested is not mrparea, the following log item can be referenced. With this, the plugin will return an empty Multivalue to the Policy Controller.



Figure - Not MRP Area