ScriptConfiguration

Overview

The `Script` (`basic:Script` prior to V3.2) type allows definition of complex filtering with a scriptlet as either a Mapper or a PolicyRule depending on the location. The script is either a Policy Rule or a Mapper depending on its location:

- If the script is specified within the scope of an `<AttributeRule>` element then the script has to be Mapper, returning a `Set<IdPAttributeValue>`, which is added to the permit or deny list for the attribute in question.
- If the script is specified within the scope of a `<PolicyRequirementRule>` element then the script has to be a PolicyRule (returning a `Boolean`), which defines whether the rule is active or not.

Schema Name

The `Script` type is defined in the `urn:mace:shibboleth:2.0:afp` namespace, the schema for which can be located at http://shibboleth.net/schema/idp/shibboleth-afp.xsd

The deprecated `basic:Script` type was defined in the `urn:mace:shibboleth:2.0:afp:mf:basic` namespace, the schema for which can be located at http://shibboleth.net/schema/idp/shibboleth-afp-mf-basic.xsd

Attributes

Two optional attributes may be specified:

Name	Type	Default	Description
`language`	String	javascript	The language of the script
`customObjectRef` 3.2	String		The name of a Spring Bean defined elsewhere. This bean will be made available to the script with the name "`custom`". See the ScriptedAttributeDefinition for more details.

Child Elements

One of two child elements can be provided

Name	Description
`<ScriptFile>`	The path of a resource (usually a file) which contains the script
`<Script>`	The script. It is usual to specify this within a CDATA

Data available to the script

The script has the following variables available

Name	Type	Description
`filterContext`	AttributeFilterContext	The AttributeFilter context provides some information about the request, and a mechanism to navigate to other contexts in the tree
`profileContext`	ProfileRequestContext	The root context for the request
`attribute` (Matcher Only)	IdPAttribute	The attribute being filtered
`custom` 3.2	Object	Contains whatever was provided by the `customObjectRef` attribute (see above)
`subjects` 3.3	Array of Subject	The Subjects associated with this authorization. Note that these will only be present if the attribute resolution has been associated with an Authentication (and so this will not work for back channel requests).

Examples

Inline Matcher

```
<AttributeRule attributeID="email">
       <PermitValueRule xsi:type="Script">
              <Script>
       <![CDATA[
           hashSetType = Java.type("java.util.LinkedHashSet");
           result = new hashSetType();
           result.add(attribute.getValues().iterator().next());
           result;
       ]]>
              </Script>
       </PermitValueRule>
</AttributeRule>
```

This simple rule just adds the first value of the attribute "email" to its permit list.

Externally specified PolicyRule

```
<AttributeFilterPolicy id="Example">
       <PolicyRequirementRule xsi:type="Script" language="JavaScript">
              <ScriptFile>%{idp.home}/conf/scripts/simple.js</ScriptFile>
       </PolicyRequirementRule>
</AttributeFilterPolicy>
```

Simple JavaScript PolicyRule

```
boolType = Java.type("java.lang.Boolean");
if (/* Some sort of condition */) {
  result = new boolType(false);
} else {
  result = new boolType(true);
}
result;
```

V2 Compatibility

No compatibility with V2 is provided..