

# NativeSPAttributeFilter

The `<AttributeFilter>` element is used to configure plugins that filter incoming attributes to prevent applications protected by an SP from seeing data that violates whatever policies the filter implements.

While there are no specifically "mandated" points at which filters run, the SP generally invokes filtering immediately prior to the caching of a set of attributes into a user's session. Actually performing the filtering process is typically up to an [Assertion Consumer Service handler](#) (in the case of attributes delivered during SSO) or an [attribute resolver](#).

---

## Common Attributes

- `type(string)`
    - Plugin type name.
- 

## XML AttributeFilter

Indicated by `type="XML"`, implements an XML-based policy language that is equivalent to the language supported by the IdP for control over attribute release. Most (though not all) of the features of the IdP's Java implementation are supported with the following exceptions:

- Policy and rule references cannot span separate XML files, and must be limited to same-document references.
- The following functors are not implemented:
  - TBD

For the time being, refer to the [IdPAddAttributeFilter](#) topic.

The XML attribute filter's XML "portion" is a [reloadable resource](#), which means that the XML content can be supplied inline, in a local file, or a remote file, and can be monitored for changes and reloaded on the fly. The root of the XML instance MUST be an `<af:AttributeFilterPolicyGroup>` element.

## Attributes

Inherits attributes supported by [reloadable resources](#).

---

## Chaining AttributeFilter

Indicated by `type="Chaining"`, wraps a set of filtering plugins so that they execute in sequence. The input to each plugin depends on the output of the previous one.

With V2.4 and above, this is implied by any configuration with multiple `<AttributeFilter>` elements, so is no longer explicitly needed.

## Child Elements

- `AttributeFilter(one or more)`
    - Embedded plugins to instantiate.
-