# NativeSPMetadataFilter

The `<MetadataFilter>` element configures a filter that examines metadata supplied by a metadata provider and adds, modifies, or deletes information depending on the filter's behavior.

Filters are generally used to impose additional security requirements on metadata, or limit the metadata consumed. Others are more advanced and work in conjunction with other software features.



## Common Attributes

- `type`(string)
  - Name of plugin type.

---

## Signature MetadataFilter

Identified by `type="Signature"`, validates any XML Signatures found in the metadata according to trust information configured into the filter. Embedded signatures are checked, but a primary signature over the metadata instance as a whole **MUST** be present.

```
<MetadataFilter type="Signature" certificate="signer.pem"/>
```

A variety of configuration options can be used, but they are mutually exclusive.

### Attributes

- `certificate`(local pathname)
  - Path to a certificate containing a public key to use to verify signature(s). The certificate's other content is ignored.

Version 2.1 and Above

- `verifyRoles`(boolean) (defaults to false)
  - If true, every entity's nested role or affiliation descriptor elements will be examined, and if signed, also verified. This introduces significant overhead to loading large metadata files, and such signing is unusual, so this is off by default.
- `verifyName`(boolean) (defaults to true)
  - If false, and a trust engine is configured for verification (see below), then the name of the signing certificate is ignored in the case of trust engines that would otherwise require checking of credential names. This is usually a dangerous option to disable.

Version 2.6 and Above

- `verifyBackup`(boolean) (defaults to true)
  - If false, then the backing file will not be verified at startup. Setting this option to false will speed up system startup, particularly if the metadata file is large. In any case, it is the deployer's responsibility to ensure that the file stored at the backup location is safe to use. In particular, do not manually replace the backing file with an unverified copy.

### Child Elements

- `<CredentialResolver>`(optional)
  - Used to resolve public keys to use while verifying signatures. The shorthand attribute syntax above is simpler to use for a single key, but a Chaining resolver can be used to supply multiple signing keys to the filter.
- `<TrustEngine>`(optional)

- Allows signatures to be validated using the more comprehensive trust engine interface, which allows for a richer interpretation of signature and key information. By default, the name of the entity over which a signature is being verified is used as the required certificate name for trust engines that verify credential names.

---

## Whitelist MetadataFilter

Identified by `type="Whitelist"`, deletes metadata for any entity not matched inside the plugin's configuration.

**Simple Example**

```
<MetadataFilter type="Whitelist">
    <Include>https://idp.goodguy.com/shibboleth</Include>
</MetadataFilter>
```

**Extended Example**

```
<MetadataFilter type="Whitelist" matcher="EntityAttributes">
    <saml:Attribute Name="http://macedir.org/entity-category-support" NameFormat="urn:oasis:names:tc:SAML:2.0:
attrname-format:uri">
        <saml:AttributeValue>http://refeds.org/category/research-and-scholarship</saml:AttributeValue>
    </saml:Attribute>
</MetadataFilter>
```

### Attributes

- `matcher` (string) (Version 2.5 and Above)
    - Specifies an EntityMatcher plugin to use to identify the entities to whitelist, allowing more flexible matching. Additional content will be included based on the type of plugin (see associated documentation).

### Child Elements

- `<Include>`(zero or more)
    - The element's content is matched against each `entityID` found in the source metadata to identify entities to keep.

---

## Blacklist MetadataFilter

Identified by `type="Blacklist"`, filters out metadata for any entity or entity group listed inside the plugin's configuration.

```
<MetadataFilter type="Blacklist">
    <Exclude>https://idp.badguy.com/shibboleth</Exclude>
    <Exclude>urn:evil:empire:entities</Exclude>
</MetadataFilter>
```

### Attributes

- `matcher` (string) (Version 2.5 and Above)
    - Specifies an EntityMatcher plugin to use to identify the entities to blacklist, allowing more flexible matching. Additional content will be included based on the type of plugin (see associated documentation).

### Child Elements

- `<Exclude>`(zero or more)
    - The element's content is matched against each `entityID` or group `Name` found in the source metadata and only matching entities are kept. When groups are excluded, all children of the group are excluded without further examination by any filters.

---

## RequireValidUntil MetadataFilter (Version 2.1 and Above)

Identified by `type="RequireValidUntil"`, rejects metadata whose root element does not contain a `validUntil` attribute, or whose validity period exceeds a threshold.

```
<MetadataFilter type="RequireValidUntil" maxValidityInterval="604800"/>
```

### Attributes

- `maxValidityInterval`(time in seconds) (defaults to 604800, seven days)
    - Maximum permitted delta between the current time and the expiration of the metadata.

---

## EntityRoleWhiteList MetadataFilter (Version 2.2 and Above)

Identified by `type="EntityRoleWhiteList"`, removes unneeded/irrelevant role information from metadata to save memory.

```
<MetadataFilter type="EntityRoleWhiteList">
    <RetainedRole>md:IDPSSODescriptor</RetainedRole>
</MetadataFilter>
```

### Attributes

- `removeRolelessEntityDescriptors`(boolean) (defaults to true)
    - If true, then any subordinate EntityDescriptor objects are removed if they don't contain any roles after filtering.
- `removeEmptyEntitiesDescriptors`(boolean) (defaults to true)
    - If true, then any subordinate EntitiesDescriptor objects are removed if they don't contain any child entities or groups after filtering.

### Child Elements

- `<RetainedRole>`(one or more)
    - Contains the qualified element or type name of a role to retain.

---

## EntityAttributes MetadataFilter (Version 2.5 and Above)

Identified by `type="EntityAttributes"`, adds `<mdattr:EntityAttributes>` extension content to entities in order to drive software behavior based on them. Entity attributes are `<saml:Attribute>` elements that annotate entities in metadata. They can be used to populate user attributes at runtime, drive other metadata filters, or impact discovery interfaces.

```
<MetadataFilter type="EntityAttributes">
    <saml:Attribute FriendlyName="state" Name="urn:oid:2.5.4.8" NameFormat="urn:oasis:names:tc:SAML:2.0:
attrname-format:uri">
        <saml:AttributeValue>Ohio</saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute FriendlyName="locality" Name="urn:oid:2.5.4.7" NameFormat="urn:oasis:names:tc:SAML:2.0:
attrname-format:uri">
        <saml:AttributeValue>Columbus</saml:AttributeValue>
    </saml:Attribute>
    <Entity>urn:mace:incommon:osu.edu</Entity>
</MetadataFilter>
```

### Child Elements

- `<saml:Attribute>`(one or more)
    - A SAML attribute to attach in an `<mdattr:EntityAttributes>` extension.
- `<Entity>`(one or more)
    - Contains the entityID of an entity to attach all the preceding attributes to.

All of the `<saml:Attribute>` tags preceding an `<Entity>` element will be attached to that entity.

---

# Examples

Additional examples are also available. These provide more complete examples and are contributed by users of the software.

| | |
|---|---|
| Example 1 | Refresh InCommon Federation metadata every hour |