

IdPConfigConfig

IdP Configuration Configuration

This page provides information about various ways in which you can influence how the IdP loads/reads its configuration.

Loading Configuration Files

The `service.xml` configuration file provides information about all the other the configurations loaded into the IdP. Each `service` that reads a configuration may read any number of them. The result of reading more than one configuration is equivalent to merging all the configurations into one large file.

A configuration is loaded using the `<ConfigurationResource>` element. This element is a [input resource type element](#).

Enabling Configuration Reloading

Most of the IdP configuration files can be watched and reloaded if a change is detected.



Caution

Do **NOT** enable configuration reloading in a production environment unless you have a rigorous configuration testing process in place and used.

Reloadable Services



To configure automatic reloading of Service Provider metadata please refer to the arguments for [MetadataProvider within relying-party.xml](#). Reloading **relying party configuration manager** service is not intended to be used for reloading changes to metadata.

The IdP contains four services which can be reloaded

- **attribute resolver** - responsible for fetching and creating attributes, controlled by `$IDP_HOME/conf/attribute-resolver.xml`
- **attribute filtering engine** - responsible for filtering attributes based on policy, controlled by `$IDP_HOME/conf/attribute-filter.xml`
- **profile handler manager** - responsible for defining IdP endpoints (profile handlers), controlled by `$IDP_HOME/conf/handler.xml`
- **relying party configuration manager** - responsible for managing per relying party configurations, controlled by `$IDP_HOME/conf/relying-party.xml`

To enable service, and hence configuration, reloading you edit the service definition in the service configuration file, `$IDP_HOME/conf/service.xml`. Each `Service` element has two optional attributes that control service reloading:

- **configurationResourcePollingFrequency** - the frequency with which the service's configuration(s) are polled for changes, expressed [XML duration](#) notation (e.g., "PT15M" for "every 15 minutes")
- **configurationResourcePollingRetryAttempts** - number of times the IdP will attempt to reload a failed configuration before giving up, default value of 3.

If **configurationResourcePollingFrequency** is not specified services are never reloaded. If a configuration resource fails to load because it can't be read, is invalid, or for any other reason the IdP will continue to use the previous configuration and try to read the file again after **configurationResourcePollingFrequency**. If the configuration file fails to load a number of times equal to **configurationResourcePollingRetryAttempts**, the IdP will stop trying to reload the file. A restart will be required to re-enabling polling.