

Data Filter Plugins

See Also: [Writing Registry Plugins](#) and [Registry Data Filters](#)



The interface for Data Filter Plugins is *experimental* and may change across minor releases.

The name of the plugin should match the format `FooFilter`.

Data Filter plugins operate within specified contexts. Currently, all Data Filter plugins are invoked in all contexts, though this may change in a future release. The following contexts are defined:

- **OrgIdentitySource:** [Registry Data Filters](#) run when the [Organizational Identity Source](#) record is retrieved from the source and converted to a Registry Organizational Identity. Available in Registry v4.1.0.
- **ProvisioningTarget:** [Registry Data Filters](#) run prior to provisioning data being passed to [Provisioner Plugins](#).

As of Registry v4.1.0, Data Filter Plugins must declare which context(s) they support in `$PLUGIN/Model/FooFilter.php`:

```
// The context(s) this filter supports
public $supportedContexts = array(
    DataFilterContextEnum::OrgIdentitySource,
    DataFilterContextEnum::ProvisioningTarget
);
```

Each Data Filter plugin must implement the following function in `$PLUGIN/Model/FooFilter.php`:

```
public function filter($context, $dataFilterId, $data) {}
```

where

- **context:** `DataFilterContextEnum` (currently will always be `ProvisioningTarget`)
- **dataFilterId:** The ID of the appropriate Data Filter instantiation (ie: `cm_data_filters:id`)
- **data:** An array of data to be filtered, depending on the context
 - **OrgIdentitySource:** Org Identity data, in the same format as returned by Organizational Identity Source Plugins
 - **ProvisioningTarget:** Provisioning data, in the same format as passed to Provisioner Plugins
 - This field was labeled `$provisioningData` prior to Registry v4.1.0.

The function should return an array of the same form passed in `$data`, or throw an `Exception` on error.