

# Extend DataStoreFactorySpi.Param

<b>Motivation:</b>	Allow client code to query more DataStore parameter metadata
<b>Contact:</b>	Gabriel Roldán
<b>Tracker:</b>	<a href="http://jira.codehaus.org/browse/GEOT-1850">http://jira.codehaus.org/browse/GEOT-1850</a>
<b>Tagline:</b>	DataStore parameter isPassword? isUser?

This page represents the **current** plan; for discussion please check the tracker link above.

- [Motivation](#)
- [Scope](#)
  - [Status](#)
  - [Discussion](#)
  - [Proposal](#)
- [API Changes](#)
  - [BEFORE](#)
  - [AFTER](#)
- [Documentation Changes](#)
  - [Tasks](#)

## Motivation

DataStore password parameters can't easily be identified as for using a mask character in ui widgets or storing them in an encrypted form.

GeoServer and Udig can benefit of being able to query DataStore parameters are special in some way. For instance, knowing if a given parameter is a database password would allow them to store the passwords encrypted.

Hence, the resolution of <http://jira.codehaus.org/browse/GEOS-1793> is holding back on this proposal.

## Scope

The scope of this proposal is just to have this sort of extensions to DataStore parameters for client code to query on, both for 2.4.x and trunk, so we don't want to break API at all, but just adding some minor stuff.

## Status

Proposal has been positively voted and is being implemented

Voting status:

- [Andrea Aime](#) +1
- [Ian Turton](#)
- [Justin Deoliveira](#)
- [Jody Garnett](#) +1
- [Martin Desruisseaux](#)
- [Simone Giannecchini](#) +1

## Discussion

We need to integrate this in the GeoServer 1.6.x series, based on GeoTools 2.4.x. Hence, the alternative to choose shall be equally useful for GeoTools 2.4.x and trunk.

There's similar work in progress in GeoTools trunk as part of the (WPS oriented) Process API: [Parameter.java](#).

One possibility would be to abstract out the before mentioned `Parameter` class from the `process` artifact to the `api` one. This would break API for 2.4.x and since the process proposal is not finished yet we're not that sure `Parameter` will serve all purposes.

## Proposal

We're looking for a cheaper alternative, which may just be adding an extra `isPassword` field to `DataStoreFactorySpi.Param`, or rather use the process api `Parameter` as inspiration and provide `DataStoreFactorySpi.Param` with a `Map<String, Object>` of extra metadata

accompanied of well known keys.

## API Changes

### BEFORE

```
public interface DataStoreFactorySpi{
    ....
    class Param {
        public Param(String key, Class type, String description, boolean required,
Object sample) {
            ....
        }
        ....
    }
}
```

### AFTER

The idea is to introduce a Parameter interface that is used to document a key used in a Map of parameters. This interface gives user interface writers a common target to work from - to be used by both data store connection parameters and process parameters.

```
package org.geotools;
...

public class Parameter<T> {
    // mandatory information as fields
    public final String key;
    public final InternationalString title;
    public final InternationalString description;
    public final Class<T> type;
    public final boolean required;
    public final int minOccurs;
    public final int maxOccurs;
    public final Object sample;

    // optional information as metadata
    public static final String FEATURE_TYPE = "featureType";
    public static final String LENGTH = "length";
    public static final String ELEMENT = "element";
    public static final String MIN = "min";
    public static final String MAX = "max";
    public static final String IS_PASSWORD = "IS_PASSWORD";
    public final Map<String, Object> metadata;

    ...
}
```

The DataAccessFactory.Param class then extends this class (providing the traditional api for backwards compatibility):

```

package org.geotools.data;
...
public interface DataStoreFactorySpi{
    ....
    class Param extends Parameter{
        ...
    }
}

```

Example use:

```

if( Boolean.TRUE.equals( param.metadata( IS_PASSWORD ) ) ){
    ...encode password before sending it in plain text...
}

```

## Documentation Changes

Website:

- Update [Upgrade to 2.5](#) instructions





Developer Guide:

- We may want to indicate in the developers guide how to provide this sort of extra Parameter metadata for DataStore implementors.

User Guide and User Manual:

- User guide page on Parameters will need to be updated
- User guide page on connecting to datastores will need to be updated (to follow best practice)
- No special additions may be needed to the user guide other than mentioning how to get to the well known Param metadata fields, such as `isPassword()`

## Tasks

no progress		done		impeded		lack mandate/funds/time		volunteer needed
Extend the Param class in 2.4.x and trunk								
Update all supported DataStore implementations that use passwords to create their Parameter object with <code>isPassword() == true</code>								