dox42 Custom Data Source How To

A dox42 Custom Data Source allows for integration of any data read logic implemented in .NET in dox42. This document shows how to develop a dox42 Custom Data Source.

Classes and Interfaces

A dox42 Custom Data Source consists of three classes:

1. **Configuration Dialog**
   a windows form that implements dox42.CustomDataSourceKit.IDataSourceConfigDialog
2. **Data Source Parser**
   implements dox42.CustomDataSourceKit.IParser
3. **Data Source**
   implements dox42.CustomDataSourceKit.IDataSource

How to implement it

Start with the sample solution CustomCSVDataSource. You may obtain this solution free of charge from dox42.

Follow these steps (location for changes in source code in the sample solution are marked as TODO 1,2,3,.. according to the list below.)

Implement your data source class

TODO 1: Define all configuration parameters as public properties of the data source class.

TODO 2: Implement your data read logic in the method: GetDataTable().

Implement your data source parser class

TODO 3: Define XML Names for your configuration parameters.

TODO 4: Alter the method buildCommandString to write your configuration parameters to the command string.

TODO 5: Alter the method createAndInitDataSource to extract configuration parameters from command string and initialize data source.

Implement your configuration dialog class

TODO 6: Create a windows form that will run in the dox42 Data Designer (or you may start with the dialog provided in the sample solution). The class must implement the interface dox42.CustomDataSourceKit.IDataSourceConfigDialog.

TODO 7: Alter the method getCommandString to write GUI values to the command string. Make use of your method buildCommandString in your parser class.

TODO 8: Alter the method setCommandString to populate GUI from command string.

Glue it together

TODO 9: Insert the class name of your configuration dialog in the method getConfigDialogClassName of your parser class.

Import Configuration

TODO 10: Rename and adapt the CSVDataSouceImport.config for your classes.

dox42RemoteController

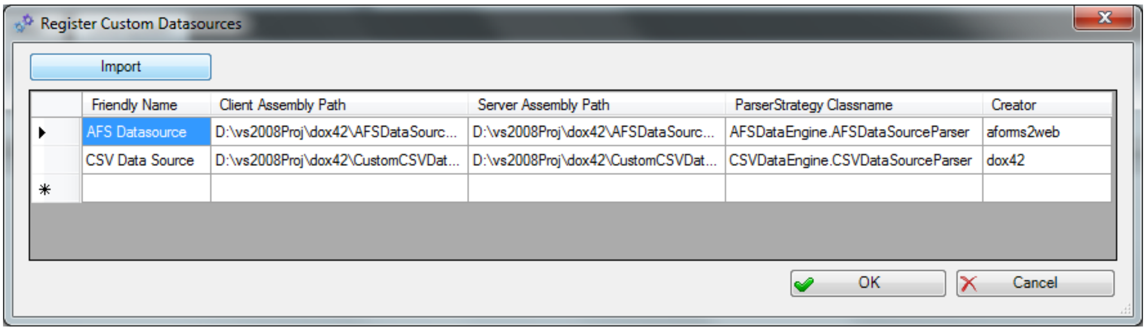
The dox42RemoteController provides you with functionality to interact with dox42 like choosing data fields, open up a data preview dialog or parsing and writing the command string.

In case your custom data source depends on data fields from other data source please don’t forget to tell dox42 about it by registering your data source as dependent:

```
dox42RemoteController.Instance.getDataSourceRC(this.Name).registerAsDependendFrom(dataSourceName);
```

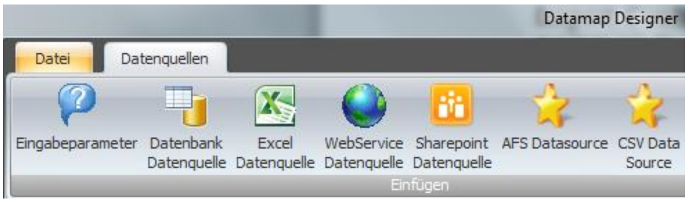
Import and Test

After compilation register your custom data source in the dox42 DataMap Designer > File > Custom Datasources.



	Friendly Name	Client Assembly Path	Server Assembly Path	ParserStrategy Classname	Creator
▶	AFS Datasource	D:\vs2008Proj\dox42\AFSDataSourc...	D:\vs2008Proj\dox42\AFSDataSourc...	AFSDataEngine.AFSDataSourceParser	aforms2web
	CSV Data Source	D:\vs2008Proj\dox42\CustomCSVDat...	D:\vs2008Proj\dox42\CustomCSVDat...	CSVDataEngine.CSVDataSourceParser	dox42
*					

Your custom data source shows up in the DataMap Designer ribbon and may be used as any other dox42 data source.



Distribution

For Distribution you may create a zip file containing the two assemblies and the import configuration file.

dox42 Server

In case you are using the dox42 Server custom data sources must be registered in a separate section in the web.config under the name “customDataSources“.

```
<customDataSources>
        <add key="CSVDataEngine.CSVDataSourceParser"
              value="CSV Datenquelle;c:\dox42\dox42WebService\bin\CSVDataEngine.dll;"/>
</customDataSources>
```

Support

Should you have any questions, please do not hesitate to contact support@dox42.com. We are happy to help you!

Good luck with dox42!

Your dox42 Team