

# textxport

This plugin allows the enabling/disabling of the Smart Client menuitems that expose the generic text import/export logic.

## Property Summary

|         |                                                                            |
|---------|----------------------------------------------------------------------------|
| Boolean | <a href="#">exportEnabled</a><br>Enable the export feature of this plugin. |
| Boolean | <a href="#">importEnabled</a><br>Enable the import feature of this plugin. |

## Method Summary

|        |                                                                                                                              |
|--------|------------------------------------------------------------------------------------------------------------------------------|
| String | <a href="#">textExport</a> (foundSet, dataProviderIds)<br>Export to text 'separated value' data (*.                          |
| String | <a href="#">textExport</a> (foundSet, dataProviderIds, separator)<br>Export to text 'separated value' data (*.               |
| String | <a href="#">textExport</a> (foundSet, dataProviderIds, separator, exportHeader)<br>Export to text 'separated value' data (*. |

## Property Details

### exportEnabled

Enable the export feature of this plugin.

#### Returns

[Boolean](#)

#### Sample

```
plugins.textxport.exportEnabled = true;  
var isEnabled = plugins.textxport.exportEnabled;
```

### importEnabled

Enable the import feature of this plugin.

#### Returns

[Boolean](#)

#### Sample

```
plugins.textxport.importEnabled = true;  
var isEnabled = plugins.textxport.importEnabled;
```

## Method Details

### textExport

[String](#) [textExport](#) (foundSet, dataProviderIds)  
Export to text 'separated value' data (\*.tab/\*.csv)

#### Parameters

[JSFoundSet](#) foundSet - the foundset to export with  
[String\[\]](#) dataProviderIds - the ids of the dataproviders

#### Returns

[String](#)

---

**Sample**

```
//export with default separator(tab) and no header
var dataToBeWritten = plugins.textxport.textExport(forms.form1.foundset,['id','name']);
```

**textExport**

**String** **textExport** (foundSet, dataProviderIds, separator)

Export to text 'separated value' data (\*.tab/\*.csv)

**Parameters**

**{JSFoundSet}** foundSet - the foundset to export with  
**{String[]}** dataProviderIds - the ids of the dataprovers  
**{String}** separator - the separator of the data

**Returns**

**String**

**Sample**

```
//export with ';' separator and no header
var dataToBeWritten = plugins.textxport.textExport(forms.form1.foundset,['id','name'],';');
```

**textExport**

**String** **textExport** (foundSet, dataProviderIds, separator, exportHeader)

Export to text 'separated value' data (\*.tab/\*.csv)

**Parameters**

**{JSFoundSet}** foundSet - the foundset to export with  
**{String[]}** dataProviderIds - the ids of the dataprovers  
**{String}** separator - the separator of the data  
**{Boolean}** exportHeader - true for exporting with the table header, false for not

**Returns**

**String**

**Sample**

```
//export with ';' separator and header
var dataToBeWritten = plugins.textxport.textExport(forms.form1.foundset,['id','name'],';',true);
```