

XMLdb – V3.5

XMLdb is a revised version of DotNetNuke (DNN) Core XML/XSL module that supports database information retrieval. Version 3.0 was developed for DNN Version and higher. While previous versions supported the existing XML/XSL module functionality, this version **only** supports database source for XML data. Please use the existing module where file sourcing is required.

No data provider is required for this module, however the **underlying database must support the "FOR XML" extension to SQL.** (MS-SQL does).

XMLdb's SQL command must have "for XML" as part of the command. For example:

```
select * from users users, userportals P
  where users.userid = P.userid
  and P.PortalID = [PortalID]
  order by users.lastname, users.firstName
  for xml auto, elements
```

This select command will return a single XML string of all users for current portal. The Users.xsl file is included in the Private Assembly.

XMLdb Settings

XML Data Source:

SQL Command: `select * from users users, userportals portals
where users.userid = portals.userid
and portals.PortalID = [PortalID]
order by users.lastname
for xml auto, elements`

Connection String:

Prefix: `<root>`

Suffix: `</root>`

Query String Checking: 30 Maximum Parameter Size

Numeric Only Parameters

Enable SQL Injection Filtering

Test Query String:

Test Command

Add SQL Command

Add CR/LF

XSL Transformation Source:

Show XSL source (after substitution)

Link Type:

URL (A Link To An External Resource)

File (A File On Your Site)

File Location: Root

File Name: users.xsl

Upload New File

Update Cancel

Since the for XML extension returns a XML fragment, a Prefix and Suffix must be applied to the results so that a complete XML document is returned. If no values are supplied, then a prefix of "<root>" along with the corresponding suffix of "</root>" is applied.

An optional connection string is available to allow connecting to another database. If the connection string is empty, then the current database is used.

Several special Meta-strings will be replaced with the current value PRIOR to execution of the SQL command:

[PortalID] = is replaced with the current portalID (see example above)

[ModuleID] = is replaced with the current ModuleID

[UserID] = is replaced with the current UserID

[TabID] = is replaced with the current TabID (new to v1.4)

A Test Command button exists on the Edit Screen. This allows the user to see the resultant XML document. Additionally, you may add the SQL command. Lastly, if required, the user can add CR/LFs to the document. This is helpful when copying the document to a flat file for testing.

Caution:

The initial use of the XMLdb module is for a Intranet site where all uses of the module are under control of a single Admin. This leads to several issues if not used in such an environment. The SQL command has **complete access** to the current database tables (all portals)

Examples

Simple - Users

After adding the XMLdb module to a page, edit the settings to add a select statement as shown below

```
select * from users users, userportals P
  where users.userid = P.userid
  and P.PortalID = [PortalID]
  order by users.lastname, users.firstName
for xml auto, elements
```

Copy the **Users.XSL** from the \desktopmodules\XMLdb directory to the root directory the portal. You can then reference the file in the XSL Transformation File Section of XMLdb Settings page. Your Settings page should look like the example page shown above.

This will create a simple listing of the users on your site.

Paged - Northwind Customers

See www.tressleworks.ca to view a Word document on the details of this example. There is a download available with all the files need to generate this example.

Revision History

Version 1.1

Now supports QueryString Substitution. If the SQL command contains a meta-string of the same name as the querystring name, the value will be substituted in the SQL command. The QueryString name must be surrounded by '[' and ']'. For example, the if the QueryString contained "Username=Admin" then the SQL command requires "[Username]" to perform the substitution.

Note: XMLdb does a exact substitution, so if the resulting value is a string, the SQL command must supply the needed quotes. The above SQL command example may look like:

```
... where Username = '[Username]' ....
```

and would result in

```
... where Username = 'Admin' ...
```

Version 1.2

This version performs XSL QueryString Substitution – similar to above on the SQL command.

Version 1.3

This version performs 3 types of QueryString validation and injection filtering. You may specify the maximum length of a QueryString Parameter, whether the parameters are numeric, and lastly perform filtering of possible command injection attacks. The default length is 30 characters with numeric only disabled and Injection Checking enabled.

Version 1.4

This version supports output paging. Specifically, XMLdb adds Default Query string values that are applied after the supplied Query string variables are substituted. Additionally, two new substitution are performed against the XSL file. [CurrentPageUrl] return the current page name (all characters up to the "?") and [TabID] returns the current TabID. Using these two values and a supporting Stored procedure, XMLdb will support paging.

Another feature of XMLdb V1.4 is the ability to process multiple SQL Select results. For example consider the following SQL snippet.

```
Select '<count>' + count(code) + '</count>' from CodeCountry  
Select * from CodeCountry from CodeCountry order by Country for XML auto, elements
```

The two separate select statements will generate XML output in two separate results. XMLdb will combine all results into one XML string for processing by the XSL translator.

This new feature is used in the CodeCountry example.

Version 3.0

This rename version developed for DNN Version 3.0 and above using DNN 3.0.11 beta as a development base. XMLdb supports localization by does **not** support the new search feature of DNN v3.0 as the database data may change based on the time of searching.

Version 3.0.2

This version contains the ability to see the substituted XSL used to perform the translation. This new feature will help in debugging issues with XSL. Also corrects a problem associated with paging. See Examples

Version 3.5

This version has been revised to work with V5.0 of the DotNetNuke Framework.

Revision History Release Dates:

V1.0.0 - 15-Jun-2004	- Initial Release
V1.1.0 - 29-Aug-2004	- Revised to allow QueryString Variable substitution
V1.2.0 - 30-Aug-2004	- Revised to allow XSL Variable substitution
V1.3.0 - 13-Oct-2004	- Revised to perform Injection testing
V1.4.0 - 11-Nov-2004	- Revised to support output paging
V3.0 - 27-Feb-2005	- Revised for DNN V3.0
V3.0.2 - 01-May-2005	
V3.5 - 22-Dec-2009	- Revised to work with DNN V5.0

Support

If you have any questions or comments, please send them to XMLdb@tressleworks.ca