



R2i. INTEGRATED

September 27, 2007
Durga Prasad

Product: NukeDK.Listx

Topic: Create XHTML and SEO Friendly Navigation, Sitemap and Breadcrumb

NukeDk.ListX can work together with DNN pages to create useful site elements like Navigation, a Sitemap or Breadcrumbs, to name a few. DNN Uses a Tabs Table to store all of its pages. It then refers to the page with a tab id. That is why you see typical DNN URL Ends with /Tabid/{tab id}/Default.aspx. As most NukeDk.ListX users know, it is very easy to use NukeDk.ListX to query the table and display the records in your own html.

Here is an example of how to use NukeDk.ListX to create NukeDk.ListX Navigation, a Sitemap and a Breadcrumb

Quick Step-by-Step Run Through

1. Run the first two General Functions in your DNN SQL or SQL server Query Analyzer
2. Drop a NukeDk.ListX module on a page.
3. Go to 'View Options'
4. In General Tab Check 'Show All'
5. Enter code from the samples given into the Query and Format areas.
6. Enter class names from sample into portal.css or skin.css and re-style for your needs.
7. For Breadcrumb, pay attention to the extra procedure given in the sample for you to run.

Section Reference:

[General Functions](#)

[Sitemap](#)

[Root Only Navigation](#)

[Child Navigation](#)

[Breadcrumb](#)

General Functions

The following two functions are common to all three elements. These functions allow DNN's linking system to work correctly, **so you don't want to skip this step!** Run these one at a time either through DNN SQL or SQL server Query Analyzer.

----- Function 1 -----

```
Create FUNCTION [dbo].[ISNUMBER](@STRINGIN VARCHAR(50), @STRLEN BIGINT)
RETURNS SMALLINT

BEGIN
    DECLARE @X INT, @RET SMALLINT, @CHR VARCHAR(1), @CH2 VARCHAR(1)
    SET @X=1
    SET @RET =1
    WHILE (@X <= @STRLEN)
        Begin
            SET @CHR=""
            SET @CH2=UPPER(SUBSTRING(@STRINGIN, @X, 1))
            IF @CH2<>'1' AND @CH2<>'2' AND @CH2<>'3' AND @CH2<>'4' AND @CH2<>'5' AND
@CH2<>'6' AND @CH2<>'7' AND @CH2<>'8' AND @CH2<>'9' AND @CH2<>'0'
                Begin
                    SET @RET = 0
                    Break
                End
            SET @X=@X+1
        Continue
    END
    RETURN @RET
END
```

----- Function 2 -----

```
Create FUNCTION [dbo].[ReWriteURL](@STRINGIN VARCHAR(256), @TABID int)
RETURNS VARCHAR(256)

BEGIN
    Declare @url varchar(256), @folder varchar(256), @file varchar(256), @pid int

    if (dbo.ISNUMBER(@STRINGIN, len(@STRINGIN))=1 and @STRINGIN<>")
        set @url = '//[ALIAS, System]/tabid/'+@STRINGIN+'/Default.aspx'

    else if (Substring(@STRINGIN, 0, 8)='FileID=' and @STRINGIN<>")
        Begin
            select @Folder = Folder from Files where FileID=SUBSTRING(@STRINGIN, 8, LEN(@STRINGIN))
            select @file = fileName from Files where FileID=SUBSTRING(@STRINGIN, 8, LEN(@STRINGIN))
            select @pid = PortalID from Files where FileID=SUBSTRING(@STRINGIN, 8, LEN(@STRINGIN))
            set @url = '/Portals/'+convert(varchar(10), @pid)+'/'+@Folder+@file
        End
    else if (dbo.ISNUMBER(@STRINGIN, len(@STRINGIN))=0 and @STRINGIN<>")
        set @url = @STRINGIN
    else
        set @url = '//[ALIAS, System]/tabid/'+convert(varchar(10), @TABID)+'/'Default.aspx'

    return @url
END
```

Sitemap

Query

```
declare @UserID int,@bIsSuper bit

set @userID = [UserID,System]
SELECT @bIsSuper = IsSuperUser FROM Users WHERE UserID = @UserID

SELECT      TabOrder, TabName, [Level], dbo.ReWriteURL(url,tabid) as NewURL
FROM        Tabs AS t where tabid IN

(
  SELECT DISTINCT x.tabid from tabs x INNER JOIN TabPermission TP ON x.TabID =
TP.TabID
      INNER JOIN Permission P ON TP.PermissionID = P.PermissionID
      LEFT JOIN (SELECT @UserID AS UserID, -1 AS RoleID UNION SELECT @UserID,
RoleID FROM UserRoles WHERE UserID = @UserID) UR ON TP.RoleID = UR.RoleID
      WHERE x.IsVisible = 1 AND x.IsDeleted = 0 and x.portalid=[portalid,system]
and x.taborder<10001
      AND P.PermissionCode = 'SYSTEM_TAB' AND P.PermissionKey = 'VIEW'
      AND ( UR.UserID = @UserID OR @bIsSuper = 1 )
)
ORDER BY TabOrder
```

ListItem

```
<div Style="padding-left:[Level]0Px;" class="normal"><a href="[NewURL]">[TabName]</a></div>
```

Root Only Navigation

Query

```
declare @tid int, @pid int,@tpid int, @lvl int,@UserID int,@bIsSuper bit

set @tid = [PortalSettings.ActiveTab.TABID,System]
set @userID = [UserID,System]
SELECT @bIsSuper = IsSuperUser FROM Users WHERE UserID = @UserID

select @lvl = level from tabs where tabid=@tid

while (@lvl > 0)
Begin
    set @lvl = @lvl -1
    select @pid = ParentID from Tabs where tabid = @tid
    set @tid = @pid
    if (@lvl = 0)
        Break
    ELSE
        Continue
END
select      distinct @tid as pid,
              x.tabid as tid,
              x.tabname,
              dbo.RewriteURL(x.url,x.tabid) as NewURL,
              x.taborder

FROM Tabs x INNER JOIN TabPermission TP ON x.TabID = TP.TabID
      INNER JOIN Permission P ON TP.PermissionID = P.PermissionID
      LEFT JOIN (SELECT @UserID AS UserID, -1 AS RoleID UNION SELECT @UserID,
RoleID FROM UserRoles WHERE UserID = @UserID) UR ON TP.RoleID = UR.RoleID
      WHERE x.IsVisible = 1 AND x.IsDeleted = 0 and x.Level=0 and x.portalid=0 and
x.taborder<10000
      AND P.PermissionCode = 'SYSTEM_TAB' AND P.PermissionKey = 'VIEW'
      AND ( UR.UserID = @UserID OR @bIsSuper = 1 )
```

Order By TabOrder

ListItem

```
<div class='{IIF,"[pid]=[tid]","RootmenuC","Rootmenu"}'><a href="[NewURL]">
[FORMAT,"[TABNAME]",{REPLACE:<br>,&nbsp;}} </a></div>
```

Child Navigation

Query

```
declare @qry varchar(max), @tid int, @lvl int, @pid int, @cid int,@UserID
int,@bIsSuper bit

set @userID = [UserID,System]
SELECT @bIsSuper = IsSuperUser FROM Users WHERE UserID = @UserID

set @tid=[PortalSettings.ActiveTab.TABID,System]
set @qry = '
declare @UserID int,@bIsSuper bit

set @userID ='+convert(char(10),@UserID)+'
SELECT @bIsSuper = IsSuperUser FROM Users WHERE UserID = @UserID

select DISTINCT * from (
select @lvl = level from tabs where tabid=@tid
set @cid = @tid

while (@lvl > 0)
Begin
    set @lvl = @lvl -1
    select @pid = ParentID from Tabs where tabid = @tid

    set @qry = @qry + 'select
                                t.tabid,
                                t.url,
                                t.tabname,
                                t.isVisible,
                                t.isDeleted,
                                t.taborder,
                                t.level,
                                dbo.RewriteURL(t.url,t.tabid) as NewURL
                                from tabs

t where ParentID=' + convert(varchar(10),@pid) + ' Union '
    set @tid = @pid

    if (@lvl = 0)
        Break
    ELSE
        Continue
END
set @qry = @qry + 'select
                                t.tabid,
                                t.url,
                                t.tabname,
                                t.isVisible,
                                t.isDeleted,
                                t.taborder,
                                t.level,
                                dbo.RewriteURL(t.url,t.tabid) as NewURL
                                from tabs t where ParentID=' +
convert(varchar(10),@cid) + ' ) tbs where '
set @qry = @qry + '
tbs.tabid IN
```

```
(
  SELECT DISTINCT x.tabid from tabs x INNER JOIN TabPermission TP ON x.TabID =
TP.TabID
  INNER JOIN Permission P ON TP.PermissionID = P.PermissionID
  LEFT JOIN (SELECT @UserID AS UserID, -1 AS RoleID UNION SELECT @UserID,
RoleID FROM UserRoles WHERE UserID = @UserID) UR ON TP.RoleID = UR.RoleID
  WHERE x.IsVisible = 1 AND x.IsDeleted = 0 and x.portalid=0 and
x.taborder<10000
  AND P.PermissionCode = ''SYSTEM_TAB'' AND P.PermissionKey = ''VIEW''
  AND ( UR.UserID = @UserID OR @bIsSuper = 1 )
)
```

Order By TabOrder'

exec (@qry)

ListItem

```
{IIF, "[LEVEL]=1", "<div
\\IIF, \"'[PortalSettings.ActiveTab.TABID, System]'=[TabID]\", \"class=GroupHov\", \"class=GroupStat\""}
onmouseover=\"this.className='GroupHov';\"
onmouseout=\"\\{IIF, \"'[PortalSettings.ActiveTab.TABID, System]'=[TabID]\", \"this.className='GroupHov'\", \"this.
className='GroupStat'\"}\\}\"><a href='[NewURL]'
\\IIF, \"'[PortalSettings.ActiveTab.TABID, System]'=[TabID]\", \"class=GroupTextCurr\", \"class=GroupText\""}
target='[Target]'>[TabName]</a></div>\", \""}
{IIF, "[LEVEL]=2", "<div
\\IIF, \"'[PortalSettings.ActiveTab.TABID, System]'=[TabID]\", \"class=ItemHov\", \"class=ItemStat\""}
onmouseover=\"this.className='ItemHov';\"
onmouseout=\"\\{IIF, \"'[PortalSettings.ActiveTab.TABID, System]'=[TabID]\", \"this.className='ItemHov'\", \"this.cl
assName='ItemStat'\"}\\}\"><a href='[NewURL]'
\\IIF, \"'[PortalSettings.ActiveTab.TABID, System]'=[TabID]\", \"class=ItemTextCurr\", \"class=ItemText\""}
target='[Target]'>[TabName]</a></div>\", \""}
{IIF, "[LEVEL]=3", "<div
\\IIF, \"'[PortalSettings.ActiveTab.TABID, System]'=[TabID]\", \"class=ItemHovC\", \"class=ItemStatC\""}
onmouseover=\"this.className='ItemHovC';\"
onmouseout=\"\\{IIF, \"'[PortalSettings.ActiveTab.TABID, System]'=[TabID]\", \"this.className='ItemHovC'\", \"this.
className='ItemStatC'\"}\\}\"><a href='[NewURL]'
\\IIF, \"'[PortalSettings.ActiveTab.TABID, System]'=[TabID]\", \"class=ItemTextCurr\", \"class=ItemText\""}
target='[Target]'>[TabName]</a></div>\", \""}

```

Breadcrumb

Procedure for Bread Crumb: Run this in your DNN SQL or SQL server Query Analyzer

Create PROC [dbo].[BreadcrumbTabs]

```
(
    @TabID int,
    @UserID int
)
AS
CREATE TABLE #tmpTabs
(
    tempID INT IDENTITY,
    Level [int],
    TabID [int],
        ParentID [int],
    TabName [varchar](200),
    Title [varchar](200),
        url [varchar](200),
        NewURL [varchar] (256),
        TabOrder [int],
        isVisible [bit],
        isDisabled [bit]
)

DECLARE @ParentID int, @blsSuper bit
        SELECT @blsSuper = IsSuperUser FROM Users WHERE UserID = @UserID

WHILE NOT @TabID IS NULL
BEGIN
    INSERT INTO #tmpTabs (Level, TabID, ParentID, TabName, Title, url, NewURL, TabOrder, isVisible,
IsDisabled)
        SELECT DISTINCT Tabs.Level, Tabs.TabID, Tabs.ParentID, Tabs.TabName, Tabs.Title,
        Tabs.url, dbo.ReWriteURL(Tabs.url, Tabs.TabID), Tabs.TabOrder, Tabs.isVisible, Tabs.DisableLink
        FROM Tabs INNER JOIN TabPermission TP ON Tabs.TabID =
TP.TabID
        INNER JOIN Permission P ON TP.PermissionID =
P.PermissionID
        LEFT JOIN (SELECT @UserID AS UserID, -1 AS RoleID
UNION SELECT @UserID, RoleID FROM UserRoles WHERE UserID = @UserID) UR ON TP.RoleID =
UR.RoleID
        WHERE Tabs.TabID = @TabID AND Tabs.IsDeleted = 0
        AND P.PermissionCode = 'SYSTEM_TAB' AND
P.PermissionKey = 'VIEW'
        AND ( UR.UserID = @UserID OR @blsSuper = 1 )

        SELECT @ParentID = ParentID FROM Tabs WHERE TabID = @TabID
        SET @TabID = @ParentID
END
SELECT * FROM #tmpTabs order by tempID desc
DROP TABLE #tmpTabs
```

Query

```
exec BreadcrumbTabs [PortalSettings.ActiveTab.TabID], [userID,System]
```

ListX Format

HEADER:

```
<span class=breadcrumb>
```

FOOTER

```
</span>
```

List Item

```
{IIF,"[LEVEL]=0","", "&nbsp;<img src='[PortalSettings.ActiveTab.SkinPath,System]_i/bread_sep.gif'>&nbsp; "}  
<a href="http://[*ALIAS]/tabid/[tabid]/Default.aspx"  
class='{IIF,"[PortalSettings.ActiveTab.TabID]=[tabid]","breadcrumbC","breadcrumb"}>[FORMAT,"[TABNAME]",{  
REPLACE:<br>,&nbsp; }}</a>
```