```
hi,


i did have rewrite my "one Class" to follow Xbase++ OOP Concept

here now DXE_TreeView and DXE_TreeViewItem
just #include "DXE_CLONE.CH" into your Source Code

most of it now react like Xbase++ "original"
so send me a Sample if "your" Code does not work like before.

i have include some Sample which work fine but some other
like c:\ALASKA\XPPW32\Source\samples\apps\ImgView\ImgView.prg
need some Workaround when Node are create dynamic using OOP
( see also : "DXE Array for Xbase++ Object" )

DXE_TreeViewItem Class include all API Function but i have just
written Wrapper for those we have it Xbase++

ImageList is still limited to "one only"

DXE "native" Control will NOT convert ANSI <-> OEM automatic
so you can use chinese BIG5 Sign.


have fun
greetings by OHR
Jimmy

*******************************************************************************
*
* DXE_TreeView
*
*******************************************************************************

Xbase++ Syntax
--------------
        init(p1,p2,p3,p4,p5,p6)
        Create(p1,p2,p3,p4,p5,p6)
        Destroy()

        rootItem() -> DXE_TreeViewItem

        lockUpdate(lOnOff)
        Show()
        Hide()
        setdata(oMarkedItem)
        getdata()

        itemExpanded (mp1 , mp2)
        itemCollapsed(mp1 , mp2) // unused
        itemMarked    (mp1 , mp2)
        itemSelected (mp1 , mp2)

DXE Workaround
--------------
        CheckExpandNode()       -> Check for Expand "+" Flag

DXE Debug
---------
        Get_All_Node() -> write Debug Logfile


DXE internal iVAR
-----------------

        VAR hTVbase              -> DXE_TreeView ::GetHWnd()
        VAR lCheckNode          -> use CheckExpandNode()        DEFAULT .T.
        VAR SingleExpand        -> single click to Expand Node  DEFAULT .T.
        VAR drawmode            -> XBP_DRAW_OWNERADVANCED -> OnCustomDraw(oh)


DXE missing (under Construction )
-----------------------------

        :itemFromPos( <aPos> )  -> oTreeViewItem | NIL
        :configure()            -> self
```

```
****************************************************************************
*
* DXE_TreeViewItem
*
****************************************************************************


Xbase++ Syntax
--------------
        init(p1,p2,p3,p4,p5,p6)
        Create(p1,p2,p3,p4,p5,p6)
        Destroy()
        getData()
        EditBuffer()
        setData( xValue )
        delItem( oItem )
        getChildItems()
        getParentItem()
        addItem(          cText,xNormalImage,xMarkedImage,xExpandedImage,cDLL,xValue)
        insItem(oInsAfter,cText,xNormalImage,xMarkedImage,xExpandedImage,cDLL,xValue)

        setExpandedImage( nResId )      // under Construction
        setImage( nResId )              // under Construction
        setMarkedImage( nResId )        // under Construction

DXE Workaround
--------------

        isExpanded()                    // hm ... Workaround
        expand(lTF)                     // call API
        Collapse()                      // call API


DXE new, not in Xbase++
-----------------------
        AddImage(nResource,cDLL)        // add to Imagelist (only 1 yet )
        ClearAll()                      // reset all Node / Array

DXE API direct
--------------

        APIinsertItem(cText,hParent,hInsertAfter,cChildNo,iImage)
        APIItemState(hItem,nState)      // Set-Get Item State
        APIItemChildren(hItem,cChildren)// Set-Get HAVE_NO_ITEM / HAVE_SUB_ITEM
        APIItemCaption(hItem,cCaption)  // Set-Get Caption
        APIgetItem()                    // Get TreeItem "Text"
        APIgetNextItem(item,flag)
        APIgetNext(item)                // get the next sibling item
        APIgetPrevious(item)            // get the previous sibling item
        APIgetRoot()                    // get the topmost item
        APIgetParent(item)              // get the parent item
        APIgetChild(item)               // get the first child item
        APIgetSelected()                // get the selected item
        APIgetDropHilight()             // get the target item of a drag-and-drop operation.
        APIselectItem(item,flag)        // select active Item
        APIdeleteItem(item)             // delete Item
        APIeditLabel(item)
        APIgetEditControl()
        APIgetCount()
        APIexpand(item,flag)
        APIcollapse(item)
        APIsetIndent(indent)
        APIgetIndent()
        APIsetImageList(imageListHandle, imageListType)
        APIgetImageList(imageListType)
        APIcreateDragImage(item)
        APIgetItemRect(rect,textOnly)
        APIhitTest(hitTestInfo)         // under Contruction, need Structure
        APISetBkColor(clrBk)
        APIGetBkColor()
        APISetItemHeight(nHeight)

DXE internal iVAR
-----------------
```

```
        VAR ID_hTVitem              -> DXE_TreeViewItem ::GetHWnd()
        VAR ID_hParent              -> hItem from APIinsertItem()
        VAR ID_hItemRoot            -> Parent (ROOT) Node Handle ::GetHWnd()
        VAR lFlagExpand             -> Expand Flag .T. / .F.


DXE missing
-----------
        none


*******************************************************************************
*
* DXE Array for Xbase++ Object
*
*******************************************************************************

each Element using o:AddItem() / o:InsItem() create a DXE_TreeViewItem Object.
all Xbase++ Object are hold in a internal Array.

#define A_DXEobj            1      // Xbase++ Object
#define A_cText             2      // Caption
#define A_hTVitem           3      // ID_hTVitem   GetHWnd()
#define A_hItem             4      // ID_hParent   Item Handle from ::insertItem()
#define A_nItemRoot         5      // ID_hItemRoot Parent (ROOT) Node Handle ::GetHWnd()

using DXE_TreeView:Get_All_Node() will write into Debug Logfile

      OnDummy(aNode[i][A_hTVitem]  ,;
              aNode[i][A_hItem]    ,;
              aNode[i][A_nItemRoot],;
              aNode[i][A_cText]    )


if DXE_TreeView:lCheckNode = .T. it will use DXE_TreeView:CheckExpandNode()

      IF aNode[i][A_hTVitem] = aNode[i][A_nItemRoot]
         IF ::HaveRealChild(hTVitem) > nnn

if ID_hTVitem = ID_hItemRoot you might have a "Root" Node so we must check
if we have real Child Node if you build a Tree dynamic

      nPosi := ASCAN(aNode,{|x| x[A_nItemRoot] = hTVItem},nStart )

if DXE_TreeView:lCheckNode = .T. ( DEFAULT )

 DXE_TreeView:Show() / DXE_TreeView:itemMarked()

will use DXE_TreeView:CheckExpandNode() to check aNode Elements


*******************************************************************************
*
* using Debug Logfile
*
*******************************************************************************

you have to modify your MAIN this Way :

    PROCEDURE MAIN
    SET ALTERNATE TO cFileName [ADDITIVE]
    SET ALTERNATE on
    //
    // your Code
    //
    SET ALTERNATE off
    SET ALTERNATE TO
RETURN


now OnDummy() will write into your <cFileName>
Parameter can be any VALTYPE()

... and YES it will work in GUI Environment


PROCEDURE OnDummy()                                // write into Debug Logfile
LOCAL iMax := PCount()
```

```
LOCAL i
LOCAL cText := ""

  FOR i := 1 TO iMax -1
      cText += Var2Char( PValue( i ) ) + CHR(9)
  NEXT
  cText += Var2Char( PValue( iMax ) )

  SET CONSOLE OFF
  SET ALTER ON

   ? cText

  SET ALTER OFF
  SET CONSOLE ON
RETURN

*****************************************************************************
*
* DXE_TreeView:SingleExpand
*
*****************************************************************************


while i found it usefull a single Click will expand a Node (if possible)
you can override DXE_TreeView:SingleExpand with .F. to use DblClick


*****************************************************************************
*
* known Bugs
*
*****************************************************************************

when using c:\ALASKA\XPPW32\Source\samples\apps\ImgView\ImgView.prg
and you change from HDD "C" to "D" you have to clear all "C" Node.

i do not understand how Alaska did so i wrote DXE_TreeView:ClearAll()

Glinde 28. Januar 2013
```