

## An array browser with class

### Data Based Advisor

December 01, 1991 | Schinkel, Mike

**Clipper** 5.01 provides a powerful browsing mechanism known as the `tbrowse` object. A `tbrowse` object, combined with multiple `tbcolumn` objects, can be used to implement an on-screen browse of practically any tabular data, the two most common forms being databases and arrays. To create a `tbrowse` object configured for browsing databases in the current work area, **Clipper** provides the `tbrowse` class function `TBrowseDB()`. However, **Clipper** doesn't provide an equivalent for browsing arrays; you must call the `tbrowse` class function `TBrowseNew()` and roll your own.

Doing this is simple; in fact, you can create a `tbrowse` "class function" for browsing arrays. Below is `TBrowseArray()`; it creates a `tbrowse` configured for browsing a one- or two-dimensional **array**.

```
FUNCTION TBrowseArray( ; // Create a generic TBrowse
    nTr,; // Top Row
    nLc,; // Left Column
    nBr,; // Bottom Row
    nRc,; // Right Column
    aArray,; // Array to browse
    bGetSetIndex ) // Code block to get / set index

LOCAL tb := TBrowseNew( nTr, nLc, nBr, nRc ) // Return Value : Builded TBrowse Object

IF Eval( bGetSetIndex ) == NIL // Initialize Current Element
    Eval( bGetSetIndex, 1 )
ENDIF

// Top of array: element == 1

tb:goTopBlock:= { || Eval( bGetSetIndex, 1 ) }

// Bottom of array: Element == array length
tb:goBottomBlock:= { || Eval( bGetSetIndex, Len( aArray ) ) }

// Movement in array: based on amount to move and current element
tb:skipBlock:= { | nMove | SkipElement( nMove,;
    aArray,bGetSetIndex ) }

RETURN tb // Return, ready to browse array

// Static function only visible in same
// source file as TBrowseArray()

STATIC FUNCTION SkipElement( nMove, aArray,bIndex )

IF nMove > 0 // Move down requested amount or
    // as much as possible
    nMove:= Min( nMove, Len( aArray ) - ;
        Eval( bIndex ) )
ELSE // Move up requested amount . . .
    nMove:= Max( nMove,1 - Eval( bIndex ) )
ENDIF

// Set current element to new position
Eval( bIndex, Eval( bIndex ) + nMove )

RETURN nMove // Return amount moved
```

`TBrowseArray()` accepts six required parameters: the four viewport coordinates, a reference to the **array** to browse, and a Get/Set codeblock for a variable that contains the current element number. `TBrowseArray()` assigns codeblocks to the `tb:goTopBlock`, `tb:goBottomBlock`, and `tb:skipBlock` instance variables that use the sixth parameter of `TBrowseArray()`--the Get/Set codeblock--to manipulate the current **array** element variable. Here's an example that shows how to use `TBrowseArray()`. Note that the `TBKeyProcess()` function is a simple `tbrowse` key processor:

```

// Example use of TBrowseArray()
// Browse names of files in current directory

#include "directry.ch"
#xtrans GetSetBlock( ) ; => ; { | x | IIf( x==NIL,,:= x ) }

LOCAL aDir:= Directory( "*.*" )
LOCAL i, tba

// Create TBrowse configured for an array
tba:= TBrowseArray( 10,20,15,32, aDir,;
                   GetSetBlock( i ) ) // Define a column for viewing file names

c := TBColumnNew( "File Name", ;
                 { || aDir[ i ] [ F_NAME ] } )

c:width := 12 // Force width to 12

tba:addColumn( c )
@ 09,18 CLEAR TO 16,33
@ 09,18 TO 16,33 DOUBLE

TBKeyProcess( tba )

RETURN

// Process user keystrokes

FUNCTION TBKeyProcess( b )
LOCAL nKey
WHILE .T.
  WHILE !b:stabilize()
    IF NEXTKey() <> 0
      EXIT
    ENDIF
  END
  nKey:= InKey( 0 )
  DO CASE
    CASE nKey == K_UP;   b:up()
    CASE nKey == K_DOWN; b:down()
    CASE nKey == K_PGUP; b:pageUp()
    CASE nKey == K_PGDN; b:pageDown()
    CASE nKey == K_ESC;  EXIT
  ENDCASE
END

RETURN nKey

```

Mike Schinkel is the founder and president of Expert Education, Inc. of Atlanta, GA. He's the author of Programming in **Clipper 5**, published by Addison-Wesley, a frequent contributor to several **Clipper** journals, and was the primary author of Nantucket's authorized training materials for **Clipper** Summer '87. You can contact Mike at (404) 953-3698, or on CompuServe (73567,3612).

[http://www.accessmylibrary.com/coms2/summary\\_0286-9245180\\_ITM](http://www.accessmylibrary.com/coms2/summary_0286-9245180_ITM)