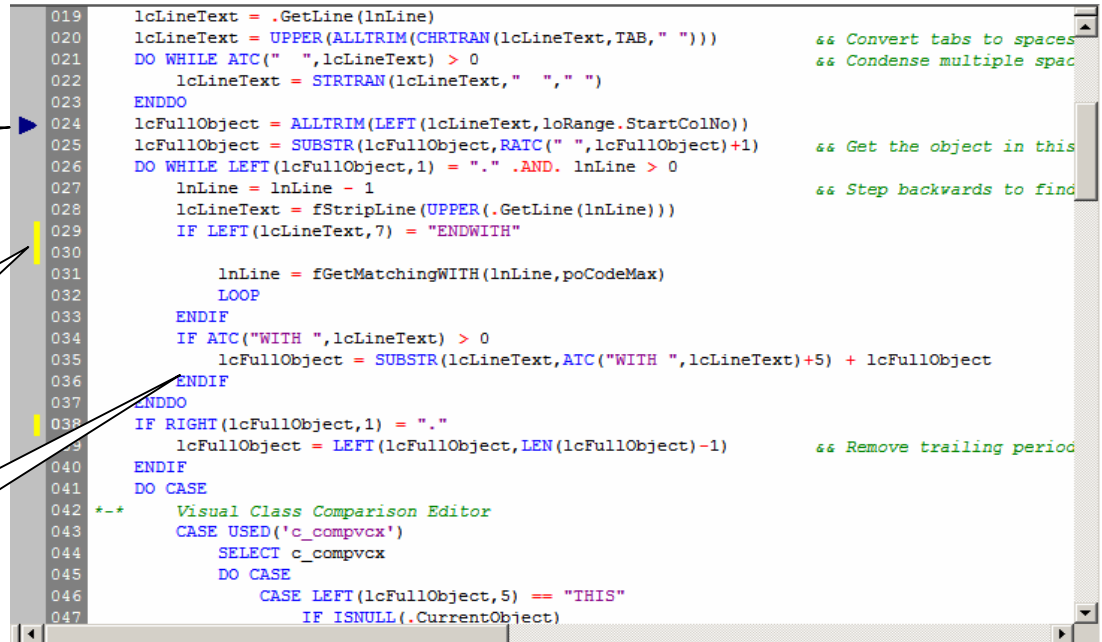


The GKK Tools Editor Feature Set

- **Color Syntax Highlighting.** VFP-specific keywords, comments, and strings are colored differently to set them apart from plain text. Bold, italics, and underline are supported per keyword set.
- **Change Control.** Lines that have changes are indicated with a yellow change bar in the left margin.
- **Bookmarks.** User can toggle on/off bookmarks for fast navigation.
- **Fully Customizable Keyboard Mappings.** Keystrokes can be added and removed to emulate popular keyboard mappings like Brief and Epsilon.
- **Keystroke Macros.** Users can record a series of keystrokes and assign a keystroke to play back the keystrokes repeatedly. Up to 10 macros may be recorded.
- **Multiple Split Views.** Users can create up to four separate views of the same edit buffer. Each view can be scrolled independently.
- **Unlimited Undo/Redo.** All edit actions are fully undoable and redoable. A limit can be set on the number of undoable actions.
- **Regular Expression Searching.** Perform regular expression searching and replacing using Perl-like regular expressions.
- **Auto Indentation.** Editor automatically indents lines to following the scoping rules of the active language.
- **Drag and Drop Text Manipulation.** Highlighted text can be dragged and dropped between any window supporting OLE text Drag and Drop. Text may be copied or moved.
- **Clipboard Support.** Text can be cut, copied and pasted onto the clipboard using the right-click menu or a keystroke.
- **RTF Clipboard support.** Text copied or dragged from the edit buffer can be pasted into other applications with all formatting information intact.
- **Column Selection and Manipulation.** Select text as a column rather than a paragraph. Empty columns (columns with a width of zero characters) can be selected, causing subsequent typing and deletion to occur over multiple lines at the same time.
- **Printing Support.** Print portrait or landscape modes, with or without line numbering, intelligent word-wrapping of long lines. Colored text supported. Print preview mode.
- **Custom Color Settings.** All foreground and background colors used in a Editor control are configurable by the end-user.
- **Built-In Right Click Menu.** A set of commonly used commands are available from a default right-click menu.
- **Microsoft IntelliMouse Support.** Scrolling, word selection, and line selection is easily accomplished with the Microsoft IntelliMouse.
- **Microsoft Windows XP Theming Support.** All elements of the Editor user interface automatically adapt to the current Windows theme when running on Windows XP.
- **Over 150 Separate Edit Commands.** More than 150 separate edit commands can be assigned to a keystroke and invoked by the user.
- **Line Numbering.** Lines can be numbered using any radix and the first line may start with any positive number.
- **Keyword Case Normalization.** Keywords can be converted to the proper case once a line is entered.
- **Virtual Space.** The text caret can optionally be confined to the bounds of the text or allowed to move beyond the end of a line.
- **Hyperlinks.** Hyperlinks embedded in code are highlighted differently and can be navigated by the user.
- **Line Sorting.** Lines can be sorted alphabetically in ascending or descending order.
- **Code Completion.** Display memberlist popup windows, parameter info tooltips, QuickInfo tooltips, and auto-complete partially typed identifiers.
- **Code Tips.** Command tips are displayed by hovering the mouse over a command keyword or during keyboard entry.
- **Automatic Brace Match Highlighting.** Matching braces are highlighted as users type.
- **Command block highlighting.** Command blocks such as IF ... ENDIF are highlighted for matching blocks for quick reference.
- **Incremental Search.** Users can search for text while Editor automatically selects the best match after each character is typed.
- **Comment/Uncommenting Support.** Blocks of code can be commented and uncommented via built in commands.
- **Automatic Font Resizing.** Users can change the font size simply by holding the control key down and spinning the mousewheel (Ctrl + MouseWheel).

The GKK Editor Window Features



019 lcLineText = .GetLine(lnLine)
020 lcLineText = UPPER(ALLTRIM(CHRTRAN(lcLineText,TAB," "))) *Convert tabs to spaces*
021 DO WHILE ATC(" ",lcLineText) > 0 *Condense multiple spaces*
022 lcLineText = STRTRAN(lcLineText," ", " ")
023 ENDDO
024 lcFullObject = ALLTRIM(LEFT(lcLineText,loRange.StartColNo))
025 lcFullObject = SUBSTR(lcFullObject,ATC(" ",lcFullObject)+1) *Get the object in this*
026 DO WHILE LEFT(lcFullObject,1) = "." .AND. lnLine > 0 *Step backwards to find*
027 lnLine = lnLine - 1
028 lcLineText = fStripLine(UPPER(.GetLine(lnLine)))
029 IF LEFT(lcLineText,7) = "ENDWITH"
030
031 lnLine = fGetMatchingWITH(lnLine,pcCodeMax)
032 LOOP
033 ENDIF
034 IF ATC("WITH ",lcLineText) > 0
035 lcFullObject = SUBSTR(lcLineText,ATC("WITH ",lcLineText)+5) + lcFullObject
036 ENDIF
037 ENDDO
038 IF RIGHT(lcFullObject,1) = "."
039 lcFullObject = LEFT(lcFullObject,LEN(lcFullObject)-1) *Remove trailing period*
040 ENDIF
041 DO CASE
042 *Visual Class Comparison Editor*
043 CASE USED('c_compvcx')
044 SELECT c_compvcx
045 DO CASE
046 CASE LEFT(lcFullObject,5) == "THIS"
047 IF ISNULL(.CurrentObject)

Bookmarks

Change line indicator

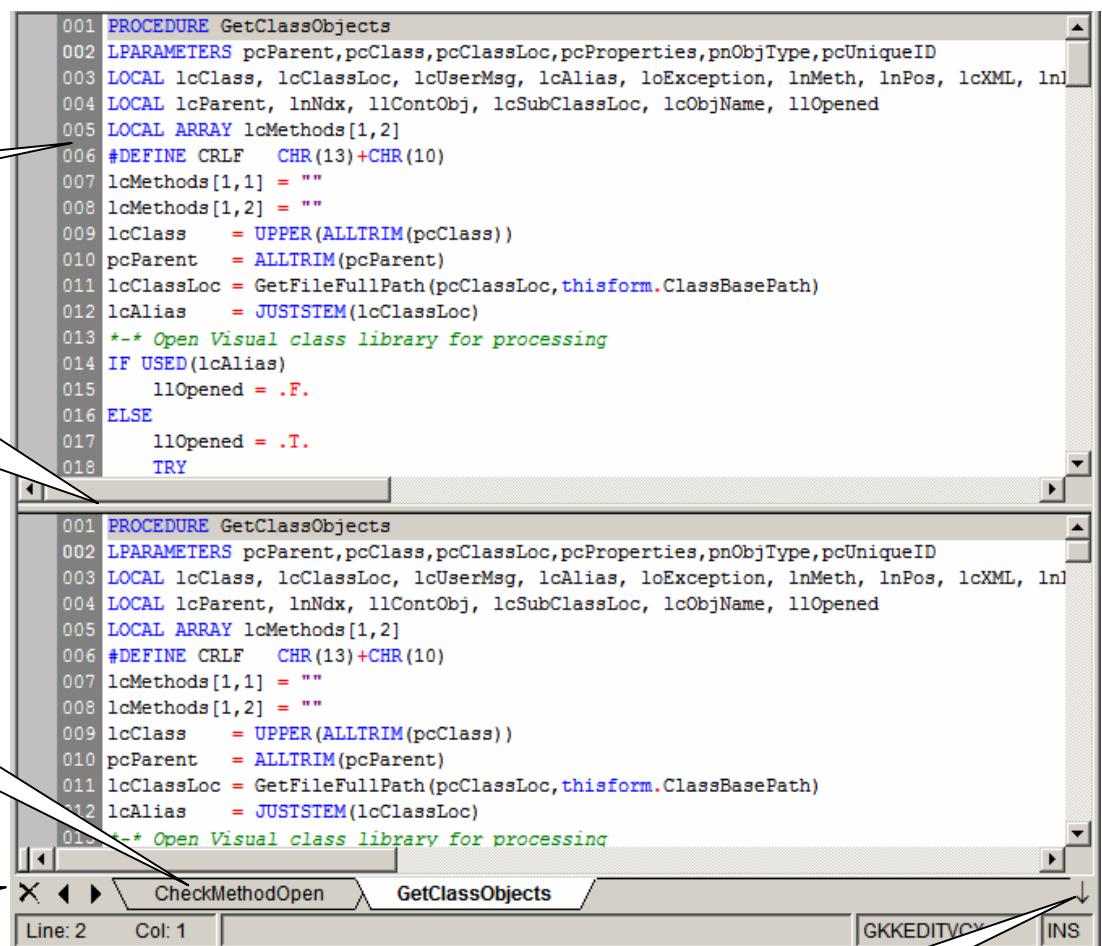
Code Highlighting

Line numbering

Split screen mode
(horizontal and vertical) for
independent navigation
within a single method

MDI Tabs for
selection between
methods

Close tabs and scroll
tabs for display



001 PROCEDURE GetClassObjects
002 LPARAMETERS pcParent,pcClass,pcClassLoc,pcProperties,pnObjType,pcUniqueID
003 LOCAL lcClass, lcClassLoc, lcUserMsg, lcAlias, loException, lnMeth, lnPos, lcXML, lnI
004 LOCAL lcParent, lnNdx, llContObj, lcSubClassLoc, lcObjName, llOpened
005 LOCAL ARRAY lcMethods[1,2]
006 #DEFINE CRLF CHR(13)+CHR(10)
007 lcMethods[1,1] = ""
008 lcMethods[1,2] = ""
009 lcClass = UPPER(ALLTRIM(pcClass))
010 pcParent = ALLTRIM(pcParent)
011 lcClassLoc = GetFileFullPath(pcClassLoc,thisform.ClassBasePath)
012 lcAlias = JUSTSTEM(lcClassLoc)
013 *Open Visual class library for processing*
014 IF USED(lcAlias)
015 llOpened = .F.
016 ELSE
017 llOpened = .T.
018 TRY
001 PROCEDURE GetClassObjects
002 LPARAMETERS pcParent,pcClass,pcClassLoc,pcProperties,pnObjType,pcUniqueID
003 LOCAL lcClass, lcClassLoc, lcUserMsg, lcAlias, loException, lnMeth, lnPos, lcXML, lnI
004 LOCAL lcParent, lnNdx, llContObj, lcSubClassLoc, lcObjName, llOpened
005 LOCAL ARRAY lcMethods[1,2]
006 #DEFINE CRLF CHR(13)+CHR(10)
007 lcMethods[1,1] = ""
008 lcMethods[1,2] = ""
009 lcClass = UPPER(ALLTRIM(pcClass))
010 pcParent = ALLTRIM(pcParent)
011 lcClassLoc = GetFileFullPath(pcClassLoc,thisform.ClassBasePath)
012 lcAlias = JUSTSTEM(lcClassLoc)
013 *Open Visual class library for processing*

Drop-down Menu for
quick tab selection

Code block highlight
when cursor placed
on block

```

013  *-- Open Visual class library for processing
014  IF USED(lcAlias)
015      llOpened = .F.
016  ELSE
017      llOpened = .T.
018      TRY
019          USE (lcClassLoc) IN 0 AGAIN ALIAS alcAlia
020      CATCH TO loException
021          IF FILE(lcClassLoc)
022              lcUserMsg = 'Visual Class Library in :
023              =WMSGBOX(lcUserMsg,"GKK Visual Class :
024          ELSE
025              lcUserMsg = 'Visual Class Library not
026              =WMSGBOX(lcUserMsg,"GKK Visual Class :
027          ENDIF
028      ENDTRY
029  ENDIF
030  IF USED(lcAlias)
031  *-- Get the class object

```

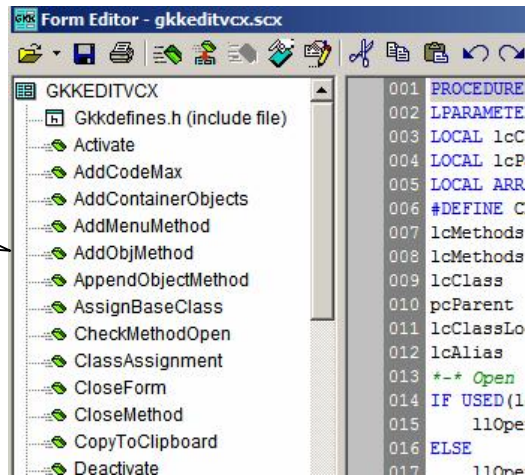
Column selection
mode (hold Alt key
with mouse selection)

```

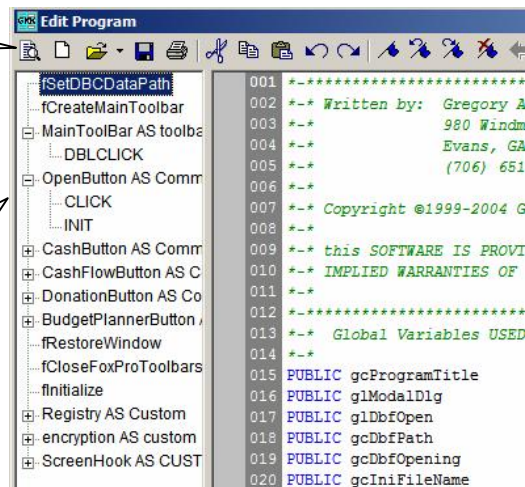
013  *-- Open Visual class library for process
014  IF USED(lcAlias)
015      llOpened = .F.
016  ELSE
017      llOpened = .T.
018      TRY
019          USE (lcClassLoc) IN 0 AGAIN ALIAS
020      CATCH TO loException
021          IF FILE(lcClassLoc)
022              lcUserMsg = 'Visual Class Lib
023              =WMSGBOX(lcUserMsg,"GKK Visua
024          ELSE
025              lcUserMsg = 'Visual Class Lib
026              =WMSGBOX(lcUserMsg,"GKK Visua
027          ENDIF
028      ENDTRY
029  ENDIF
030  IF USED(lcAlias)

```

GKK Form Editor provides
Treeview for fast
navigation of methods and
objects in a form



DocView is opened/closed
(refreshed when opened)



GKK Procedure Editor
provides DocView for fast
navigation of classes and
procedures in a file
(references updated
dynamically as code is
edited)

Tooltips – hover mouse over VFP keyword for tip

```

083
084 CASE UPPER(LEFT(lcCodeText,4)) = "NEXT"
085 lnTabInc = 0
086 lnTab = lnTab - 1
087 CASE INLIST(UPPER(LEFT(lcCodeText,5)), "ENDDO", "ENDIF")
088 lnTabInc = 0
089 CASE UPPER(LEFT(lcCodeText,11)) = "ENDPRINTJOB"
090 lnTabInc = 0
091

```

During scroll with mouse, displays line number in tooltip

```

090 CASE UPPER(LEFT(lcCodeText,11)) = "ENDPRINTJOB"
091 lnTabInc = 0
092 lnTab = lnTab - 1
093 CASE INLIST(UPPER(LEFT(lcCodeText,6)), "ENDTRY", "ENDSCAN")
094 lnTabInc = 0
095 lnTab = lnTab - 1
096 CASE UPPER(LEFT(lcCodeText,7)) = "ENDCASE"
097 lnTabInc = 0
098 IF poBeautify.IndentCase
099 lnTab = lnTab - 2
100 ELSE
101 lnTab = lnTab - 1
102 ENDIF
103 CASE INLIST(UPPER(LEFT(lcCodeText,7)), "ENDTEXT", "ENDSCAN", "ENDWITH")

```

Line 78

Command Completion

Diagram illustrating Command Completion:

Left side (Initial state):

```

01 DO CASE

```

Right side (After Command Completion):

```

01 DO CASE
02 CASE
03
04 OTHERWISE
05
06 ENDCASE

```

Member Lists:
Displays properties, methods, and objects for the object. Notice that the period is on a new line and displaying members for thisform

Two screenshots of the Member Lists window in Visual FoxPro:

Left screenshot (Initial state):

```

00 PROCEDURE Init
01 LPARAMETERS llMode
02 WITH thisform
03 .cm
04
05 Click
06 Clipcontrols
07 Closable
08 Cls
09 cmdCancel
10 cmdIssue
11 Colorsources
12 Comment
13 ContinuousScroll
14 Controlbox

```

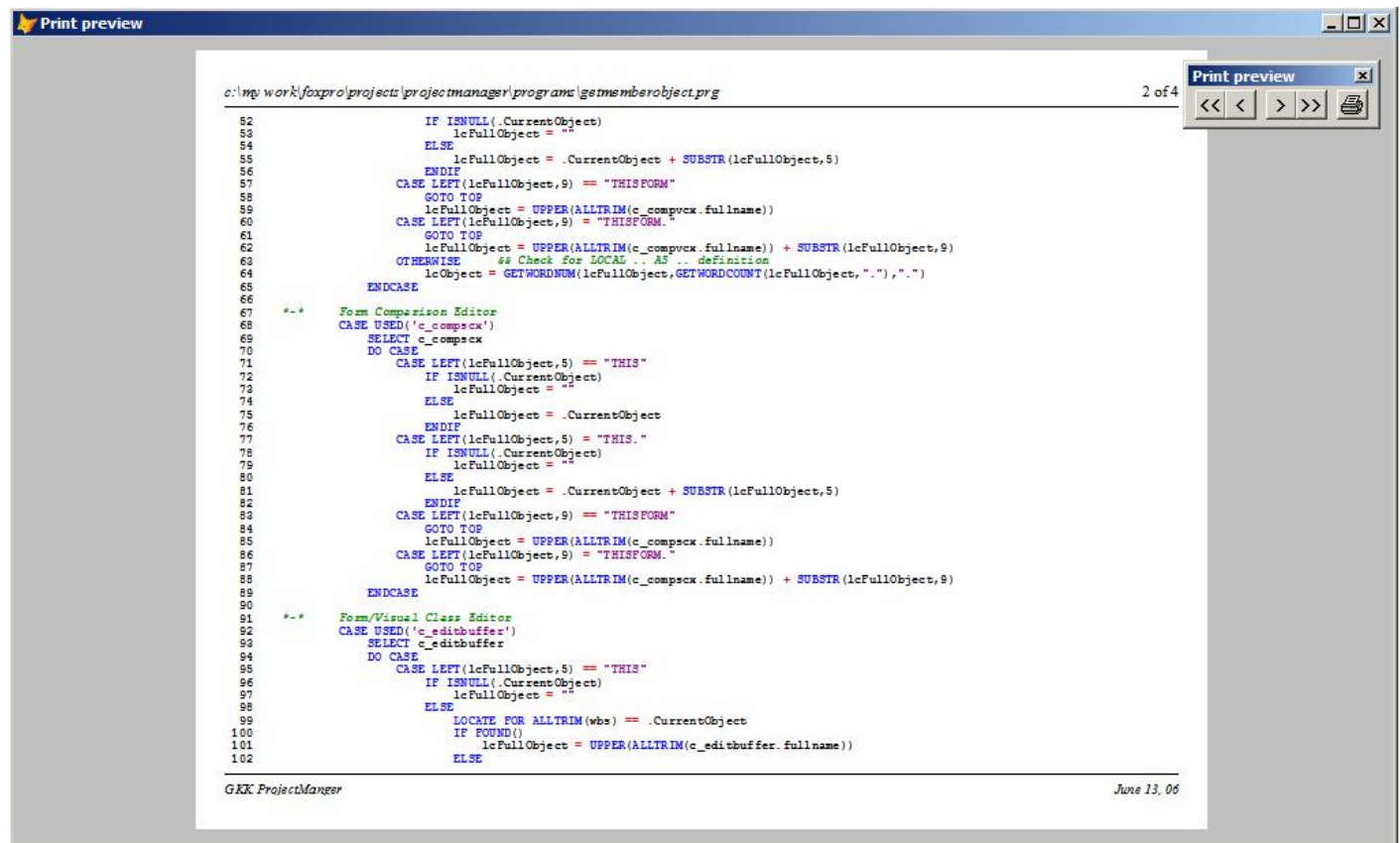
Right screenshot (After Command Completion):

```

00 PROCEDURE Init
01 LPARAMETERS llMode
02 WITH thisform
03 .cmdCancel.
04 .Account
05 .Account
06 .caption
07 IF !glB
08 .As
09 .Lal
10 .Sh
11 .cm
12 .cm
13 .he
14 ENDIF
15 ENDWITH
16 ENDPROC

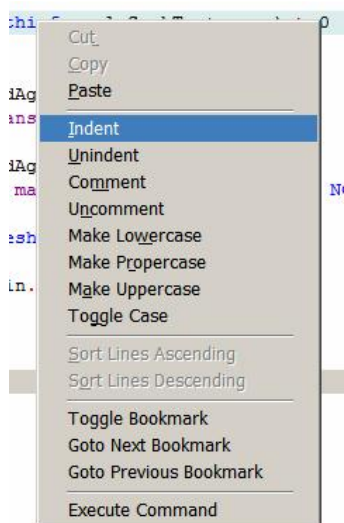
```

The Print Preview



The Editor Context Menu

Editor has a right-click menu.



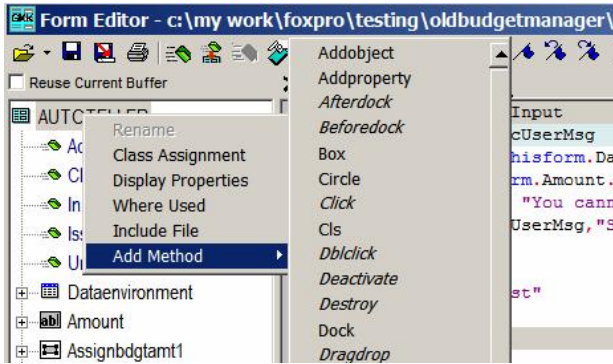
The GKK Form and Visual Class Editor Toolbar



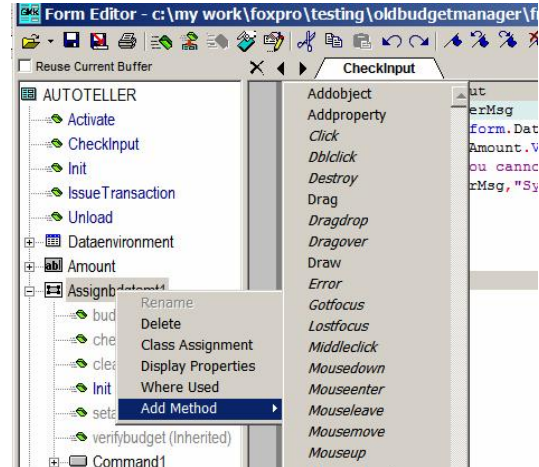
Description of command buttons from left to right:

- Open Form – provides for a File Open dialog and a drop-down MRU list of files
- Save command
- Save-As command
- Print command – allows for printing current method, selection of text in method, or any/all methods. Provides portrait or landscape, page numbers, smart word-wrap, font selection, and support for color.
- Add method – allows adding custom or stock methods for forms and objects.
- Add method from... – allows adding custom or stock methods from another form or visual class
- Display/Edit inherited method – toggles between display of the inherited method in a pop-up form (if inherited method is edited) and display of inherited method.
- Edit custom properties – provides form for defining/editing custom properties. Included are property types with builder for color, font, etc.
- Memberdata editor – allows editing the memberdata information (provides support for problem of oversize limit of 8K)
- Cut current method to clipboard
- Copy current method to clipboard
- Paste method from clipboard
- Undo command
- Redo command
- Toggle bookmark on/off
- Goto next bookmark from current position
- Goto previous bookmark from current position
- Clear bookmarks
- Go back line – provides navigation based on user's historical cursor movement
- Go forward line – provides navigation based on user's historical cursor movement
- Toggle case of selected text
- Goto matching pair – provides navigation between matching parenthesis and code blocks (Ctrl-I)
- Set tab order – form for setting the tab order of objects
- Find – quick search for current selected text
- Find – allows search for text in current method, object, or form (support for regular expressions)
- Find Increment – Toggles on incremental search
- Replace text – replaces text
- Goto line – popup for line entry
- Show whitespace – toggles whitespace display on/off
- Get syntax for Win32 API DECLARE DLL command
- Beauty command (indentation, case of keywords/symbols, and in-line comment alignment to column)
- Modify form – executes the MODIFY FORM command and hides the GKK Form Editor; re-reads form for changes when MODIFY FORM closed
- User preferences for editor
- About dialog

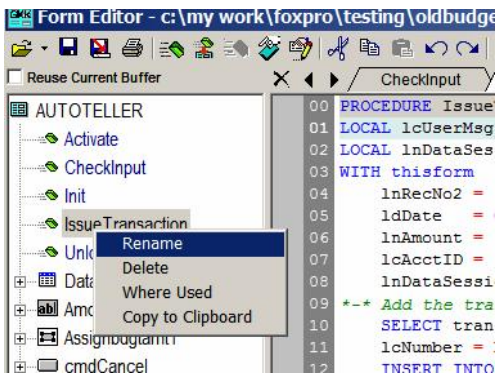
The GKK Form Editor Treeview Right-Click Menu



Form Right-Click Menu



Object on Form Right-Click Menu



Method Right-Click Menu

The Right-Click menu options:

Rename	Allows for renaming of object or method. Provides for prompted replace function to update references in form or visual class
Class Assignment	Allows changing the class defined for the object; cannot change the base class
Display Properties	Provides for display of the selected object properties. This can be on the editor or in a pop-up form (configuration option).
Where Used	Provides for a 'Where Used' list. This can be on the editor or in a pop-up form (configuration option). Quick Navigation to code location is supported on the Where Used list by double-clicking on the item.
Include File	Allows adding or deleting an Include file. Include file is listed with the form or visual class on the treeview display.
Add Method	Allows adding a stock method based on the object. Events are italicized.
Delete	Allows deleting the object or the method.
Copy to clipboard	Copies the method code to the standard windows clipboard.

The GKK Procedure Editor Toolbar



Description of command buttons from left to right:

- DocView Toggle – opens or closes the DocView navigation Treeview on left; when opening it will refresh the navigation list
- Create – provides for creating a new procedure (PRG) file
- Open file – provides for a File Open dialog and a drop-down MRU list of files
- Save command
- Save-As command
- Print command – allows for printing current procedure. Provides portrait or landscape, page numbers, smart word-wrap, font selection, and support for color.
- Cut to clipboard
- Copy to clipboard
- Paste from clipboard
- Undo command
- Redo command
- Toggle bookmark on/off
- Goto next bookmark from current position
- Goto previous bookmark from current position
- Clear bookmarks
- Go back line – provides navigation based on user's historical cursor movement
- Go forward line – provides navigation based on user's historical cursor movement
- Indent selection
- Unindent selection
- Toggle case of selected text
- Goto matching pair – provides navigation between matching parenthesis and code blocks (Ctrl-])
- Find – quick search for current selected text
- Find – allows search for text in current method, object, or form (support for regular expressions)
- Find Increment – Toggles on incremental search
- Replace text – replaces text
- Goto line – popup for line entry
- Show whitespace – toggles whitespace display on/off
- Get syntax for Win32 API DECLARE DLL command
- Beauty command (indentation, case of keywords/symbols, and in-line comment alignment to column)
- Compile procedure – displays compile errors in list at bottom of editor
- User preferences for editor
- About dialog

The GKK Menu Editor

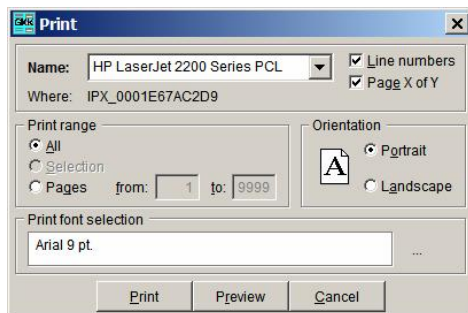
The GKK Menu Editor provides for all menu editing functions in an easy to use interface. Features included are easy treeview display of menu structure, menu branch copy/cut and paste (can be copied between two menus being edited), menu icon assignments (corrects bug in VFP), and easy access to all menu editing commands/features.

GKK Menu Editor Toolbar

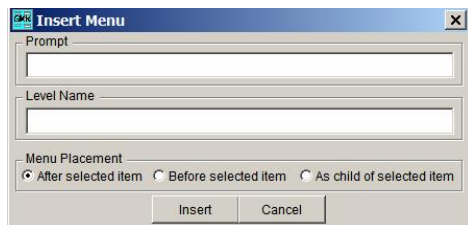
Provides access to all menu edit functions:



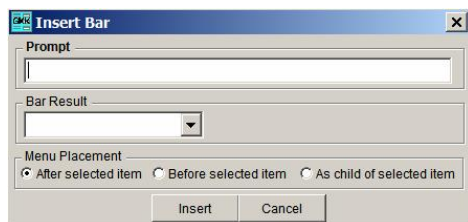
- Create – provides for creating a new menu (MNU) file
- Open file – provides for a File Open dialog and a drop-down MRU list of files
- Print menu – provides for printing a report of the menu



- Save menu
- Cut menu node/branch to clipboard (internal managed clipboard – shared between instances of the GKK Menu Editor)
- Copy menu node/branch to clipboard (internal managed clipboard – shared between instances of the GKK Menu Editor)
- Paste menu node/branch from clipboard (internal managed clipboard – shared between instances of the GKK Menu Editor)
- Insert menu pad – inserts a menu pad at the selected position on the Treeview menu structure



- Insert menu bar – inserts a menu bar at the selected menu pad on the Treeview menu structure



- Delete menu item (bar or pad) – deletes the select menu item

The GKK Comparison Editor

The comparison editor can be executed with or without command line parameters. If no parameters are provided, then a prompt for the type of comparison will be provided. To call with parameters:

```
DO GKKCompare WITH "c:\my files\firstfile.scx", "c:\my files\secondfile.scx", PRJTYPE_FORM
```

Where the third parameter is defined by the following:

```
#DEFINE PRJTYPE_CLS          "C"
#DEFINE PRJTYPE_FORM        "K"
#DEFINE PRJTYPE_PROGRAM     "P"
#DEFINE PRJTYPE_TABLE       "D"
#DEFINE PRJTYPE_VCX         "V"
```

Using the Mouse

Editor supports the following mouse actions:

Mouse Action:	Result:
L-button click over text	Changes the caret position
R-button click	Displays the right click menu
L- button down over selection, and drag	Moves text
Ctrl + L-button down over selection, and drag	Copies text
L-Button click over left margin	Selects line
L-Button click over left margin, and drag	Selects multiple lines
Alt + L-button down, and drag	Select columns of text
L- button double click over text	Select word under cursor
Spin IntelliMouse mouse wheel	Scroll the window vertically
Single click IntelliMouse mouse wheel	Select the word under the cursor
Double click IntelliMouse mouse wheel	Select the line under the cursor
Ctrl + Spin IntelliMouse mouse wheel	Increase or decrease text font size
L-button down and drag splitter bar	Split the window into multiple views or adjust the current splitter position
Double click splitter bar	Split the window in half into multiple views or unsplit the window if already split
L-button click or Ctrl+ L-button click over hyperlink	Navigate a hyperlink

Keyboard Navigation Mappings

Command	Keystroke
Move between edit buffers	Alt + PgUp or Alt + PgDn
Move between procedures/ functions/classes in editor buffer	Ctrl + PgUp or Ctrl + PgDn

Default Keyboard Mappings

Command	Keystroke
BookmarkNext	Ctrl + K + N
BookmarkNext	F2
BookmarkPrev	Ctrl + K + P
BookmarkPrev	Shift + F2
BookmarkToggle	Ctrl + K + K
BookmarkToggle	Ctrl + F2
CharLeft	Left
CharLeftExtend	Shift + Left
CharLeftExtendCol	Shift + Alt + Left
CharRight	Right
CharRightExtend	Shift + Right
CharRightExtendCol	Shift + Alt + Right
ChooseCommand	Ctrl + D
CommentSelection	Ctrl + K + C
CompleteWord	Ctrl + Space
Copy	Ctrl + C
Copy	Ctrl + Insert
Cut	Shift + Delete
Cut	Ctrl + X
CutSelection	Ctrl + Alt + W
DefineLocalVar	Alt + 6
Delete	Delete
DeleteBack	Backspace
DeleteLine	Alt + X
DeleteToEOL	Ctrl + Y
DocumentEnd	Ctrl + End
DocumentEndExtend	Ctrl + Shift + End
DocumentStart	Ctrl + Home
DocumentStartExtend	Ctrl + Shift + Home
Find	Alt + F3
Find	Ctrl + F
FindAgain	Ctrl + G
FindNext	F3
FindNextWord	Ctrl + F3
FindPrev	Shift + F3
FindPrevWord	Ctrl + Shift + F3
FindReplace	Ctrl + Alt + F3
FindReplace	Ctrl + H
GoToLine	Ctrl + L
GoToMatchBrace	Ctrl +]
Help (VFP help)	F1
Home	Home
HomeExtend	Shift + Home
IncrementalSearch	Ctrl + I
IndentSelection	Tab
JumpToNextMethod	Ctrl + PgDn
JumpToPreviousMethod	Ctrl + PgUp
JumpToNextEditBuffer	Alt + PgDn
JumpToPreviousEditBuffer	Alt + PgUp

Command	Keystroke
LineDown	Down
LineDownExtend	Shift + Down
LineDownExtendCol	Shift + Alt + Down
LineEnd	End
LineEndExtend	Shift + End
LineOpenAbove	Ctrl + Shift + N
LineUp	Up
LineUpExtend	Shift + Up
LineUpExtendCol	Shift + Alt + Up
ListMembers	Ctrl + J
LowercaseSelection	Alt + L
PageDown	Next
PageDownExtend	Shift + Next
PageUp	PRIOR
PageUpExtend	Shift + Prior
Paste	Ctrl + V
Paste	Shift + Insert
Properties	Alt + Enter
QuickInfo	Ctrl + K + I
RecordMacro	Ctrl + Shift + R
Redo	Ctrl + Shift + Z
Redo	Ctrl + Y
ReverseIncrementalSearch	Ctrl + Shift + I
Save	Ctrl + S
SelectAll	Ctrl + A
SelectLine	Ctrl + Alt + F8
SelectSwapAnchor	Ctrl + Shift + X
SentenceCut	Ctrl + Alt + K
SentenceLeft	Ctrl + Alt + Left
SentenceRight	Ctrl + Alt + Right
SetRepeatCount	Ctrl + R
ToggleCase	Alt + T
ToggleOvertyping	Insert
ToggleWhitespaceDisplay	Ctrl + Alt + T
UncommentSelection	Ctrl + K + U
Undo	Ctrl + Z
Undo	Alt + Backspace
UnindentSelection	Shift + Tab
UppercaseSelection	Alt + U
WindowScrollDown	Ctrl + Up
WindowScrollLeft	Ctrl + PageUp
WindowScrollRight	Ctrl + PageDown
WindowScrollUp	Ctrl + Down
WordDeleteToEnd	Ctrl + Delete
WordDeleteToStart	Ctrl + Backspace
WordLeft	Ctrl + Left
WordLeftExtend	Ctrl + Shift + Left
WordRight	Ctrl + Right
WordRightExtend	Ctrl + Shift + Right

Regular Expression Syntax Rules

The regular expression implementation in Editor is a partial implementation of Perl regular expressions. The following syntax rules are supported:

Metacharacters	
\	Quote the next metacharacter
.	Match any character
	Alternation (OR'd expression)
()	Grouping
[]	Set of characters. For example, "[0123a-z]" will match 0, 1, 2, 3 and all lowercase alphabet characters. If first character is '^', none of the specified characters must be present in order to match.
^	Match the beginning of the line.
\$	Match the end of the line

Quantifiers (greedy)	
*	Match 0 or more times
+	Match 1 or more times
?	Match 0 or 1 times
{n}	Match exactly <i>n</i> times
{n,}	Match at least <i>n</i> times
{n,m}	Match at least <i>n</i> but not more than <i>m</i> times

By default, a quantified subpattern is “greedy”, meaning it will match as many times as possible while still allowing the rest of the pattern to match. If you wish for a regular expression to match the minimum number of times possible, follow the quantifier with a question mark (“?”).

Quantifiers (non-greedy)	
*?	Match 0 or more times
+?	Match 1 or more times
??	Match 0 or 1 times
{n}?	Match exactly <i>n</i> times
{n,}?	Match at least <i>n</i> times
{n,m}?	Match at least <i>n</i> but not more than <i>m</i> times

Escape sequences	
\t	match a tab character
\w	Match a "word" character (alphanumeric plus "_")
\W	Match a non-"word" character
\s	Match a whitespace character
\S	Match a non-whitespace character
\d	Match a digit character
\D	Match a non-digit character
\b	Match a word boundary
\B	Match a non-(word boundary)
\a	Match only at beginning of line
\z	Match only at end of line