



BNCS Custom Controls

BBC_CC05 Keyboards

Contents

Contents	2
Custom Control Library 5	3
1.1 Introduction	3
1.2 Controls	3
1.3 Keyboard Type 1 [style code 1] 	4
1.4 Keyboard Type 2 [style code 2] 	6
1.5 Edit Buffer Control [style code 4] 	8
1.6 Keyboard Type 3 [style code 8] 	9
1.7 Keypad Type 4 [style code 9] 	18
Application Notes	24
Version History	25

Custom Control Library 5

1.1 Introduction

This custom control library complements the existing controls, providing keyboard function for use with ApplCore panels.

This custom control is automatically loaded and initialised by CSI and ApplCore applications, provided that BBC_CC.DLL is present (Note: V1/V2 requires BBC_CC.DLL, V3 requires BNCS_CC.DLL).

1.2 Controls

The following controls are common to all keyboards styles and are accessed via the ApplCore "Text Put" (TP) command. The colour selection values comply with the standard values specified for ApplCore and BBC Custom Controls. See document: CC_Attribs.pdf (Custom Control Attributes, 1999 BBC Resources Ltd).

Example – panel ID is 1, keyboard ID is 101:

```
"TP '/__A1' 1 101 :EI light green face, dark red characters"
```

Colour controls (follows Custom Control Attributes): "/ABCD"

- "/" - escape sequence character
- "A" - keyboard button colour character selector (0-F)
- "B" - keyboard button text colour character selector (0-F)
- "C" - keyboard edit buffer colour character selector (0-F)
- "D" - keyboard edit buffer text colour character selector (0-F)

Special Characters:

The "&" character is a special processing character, within CC05, it is used as the "editor cursor" character. When used as a mnemonic-prefix character "&" it is a directive to underscore the character that follows. Using TextPrint (TP) of "&&" is a directive to print a single "&" to the keyboard edit buffer.

Example: "TP '&&' 1 KeyboardID" writes a single "&" to the keyboard edit buffer.

1.3 Keyboard Type 1 [style code 1]



This control is a complete keyboard with a single control Id.

The Base Control Id + 0 executes when the <Return> key is pressed. A TG (Text Get) on the keyboard returns the contents of the keyboard edit buffer.

Text can be input and edited using the cursor, backspace and delete keys.

When the key is pressed the StringTable line corresponding to the control Id is executed and the text in the control can be retrieved using Text Get.

Pressing toggles the alpha keys between upper and lower case.

Default: Shift state is set Shifted.

Shifted Keyboard 1 (style code 1)



Unshifted keyboard 1 (style code 1)





This control is similar to keyboard 1. It generates 4 control Id messages. The Base Control Id + 0 executes when the <Return> key is pressed. A TG (Text Get) on the keyboard returns the contents of the keyboard edit buffer.

Base Control Id + 1 executes every time a key is pressed.

Base Control Id + 2 executes when the 'Up Arrow' is pressed.

Base Control Id + 3 executes when the 'Down Arrow' is pressed.

Default: Shift state is set Shifted.

Text Get (TG) Support:

The Text Get (TG) ApplCore command support is different from that of Style 1 keyboard.

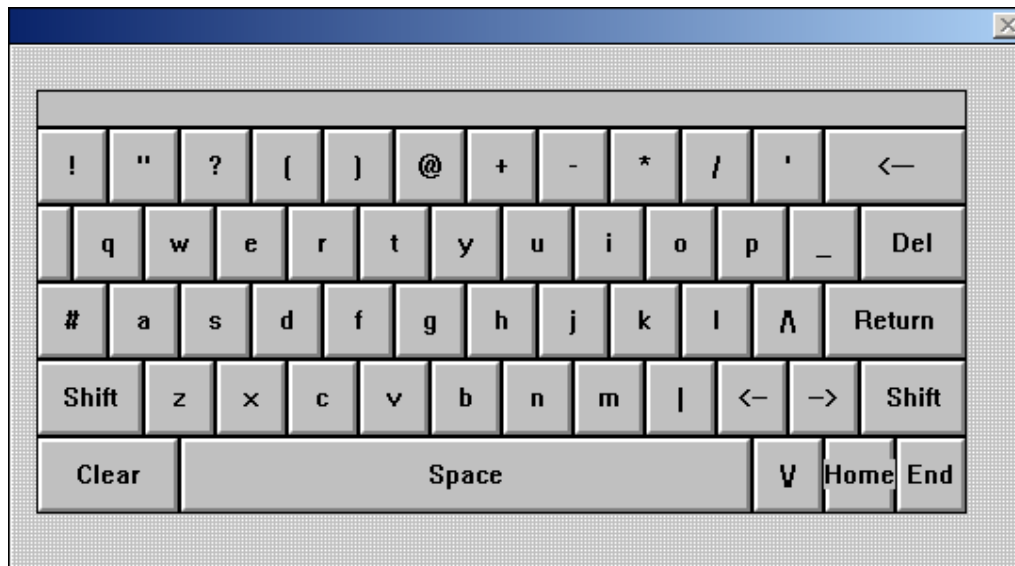
When a key is pressed the StringTable line corresponding to the control Id is executed and the text in the control can be retrieved using Text Get. The text retrieved will be the last single character pressed.

Pressing the <Return> key executes Base Control Id + 0 and executing a TG command applied to Base Control Id + 0 returns the current keyboard edit buffer. Subsequence TG on Base Control Id + 0 return a string length of zero (as the <Return> key empties the edit buffer). Additional key presses continue to populate the edit buffer and execute Base Control Id + 1.

Shifted keyboard 2 (style code 2)



Unshifted keyboard 2 (style code 2)





This control has been in the CC05 library from the inception but only recently documented.

Editing of the Edit Buffer is supported ONLY by the <BackSpace> (<BS>) character.

Control Id's:

Base Control Id + 0 executes when the <Enter> key is pressed.

Executing a Text Get (TG) command on the keyboard Base Id + 0 returns the contents of the keyboard edit buffer.

1.6 Keyboard Type 3 [style code 8]

This control was added to both the V2/16bit and V3/32bit libraries starting in versions BBC_CC05.dll (1.3.0) and BNCS_CC05.dll (3.1.6), dated: 28-Dec-2010.

Keyboard 3 is based on keyboard 2. However, it generates SEVEN Control Id messages instead of THREE (four new controls) and offers run-time colour controls, font selection and access to the external keyboard.

Relative to Keyboard 2, the button colour controls are reorganized and have supplementary functions as well.

Text Get (TG) Support:

The Text Get (TG) ApplCore command is supported via two operating modes:

1. "normal / default" mode returns the edit buffer on <Return> key press and subsequent characters as pressed one at a time.

Pressing the <Return> key executes Base Control Id + 0 and executing a TG command applied to Base Control Id + 0 returns the current keyboard edit buffer. Subsequence TG on Base Control Id + 0 return a string length of zero. Additional key presses continue to populate the edit buffer and execute Base Control Id + 1.

2. "Always Return String" mode always returns the whole of the edit buffer.

Key presses execute ApplCore line Base Control Id + 1.

Control Id's:

Base Control Id + 0 executes when the <Return> key is pressed.

Executing a Text Get (TG) command on the keyboard Base Id + 0 returns the contents of the keyboard edit buffer.

Base Control Id + 1 executes every time a key is pressed.

Executing a TG command on the Base Id + 0 then returns the character of the currently pressed key.

Base Control Id + 2 executes when the 'Up Arrow' is pressed.

Base Control Id + 3 executes when the 'Down Arrow' is pressed.

New Control Id's:

Base Control Id + 4 executes when the "Clear" key is pressed.

Base Control Id + 5 executes when the "Home" key is pressed.

Base Control Id + 6 executes when the "End" key is pressed.

Base Control Id + 7 executes when the "←" (Backspace) key is pressed.

Base Control Id + 8 executes when the "Esc" (Escape) key is pressed.

Base Control Id + 9 executes when the "Ctrl" (Control) key is pressed.

Base Control Id + 10 executes when the "Alt" (Alternate) key is pressed.

Base Control Id + 11 executes when the "Alt Gr" (Menu / Alt Graphics) key is pressed.

Base Control Id + 12 executes when the "Page Up" key is pressed (external keyboard only).

Base Control Id + 13 executes when the "Page Down" key is pressed (external keyboard only).

Base Control Id + 200 + Fn number executes when the "Fn" keys are pressed. Specifically Base control id + 201 executes when F1 is pressed and base control id + 211 executes when F11 is pressed.

Key Repeat:

Key repeat function is supported. 1 second initial, with ½ second repeat rate.

Shift State:

Default: The "Shift state" is set to non-shifted (different from keyboard styles 1 & 2).

Keyboard Controls:

The Keyboard controls are selected via the standard ApplCore interface using control characters sequences embedded in a ApplCore TP command (s). The TP command string definitions are as follows: "/ABCDEFGH". See the examples following.

- "/" - escape sequence character
- "A" – Keyboard button colour (0-F)
- "B" – Keyboard button foreground/text colour (0-F)
- "C" – Button down colour (0-F)
- "D" – Button down foreground/text colour (0-F)
- "E" – Text Put Block Switch:
 - 0 – Main keys group (default, resets all keys colours groups - following)
 - 1 – Edit buffer (only "A" and "B" affect edit buffer)
 - 2 – Shift / CapLock keys group
 - 3 – Arrow, Home, End, Del & Clear keys group
- "F" – Edit buffer Control - visibility -- Enable: 1 / Disable: 0
- "G" – Text Get Control - TG support - what is returned in response to a TG command.
 - 1 - Always Return String
 - 0 - Normal (Default: 0)
- "H" – Keyboard Source Control – source of keyboard input.
 - 0 - Disable physical keyboard capture (Default: 0)
 - 1 – not supported mode
 - 2 - Enable physical keyboard capture mode 2 (reference ApplCore KR mode 2)
 - 3 - Enable physical keyboard capture mode 3 (reference ApplCore KR mode 3)

External Keyboard Support:

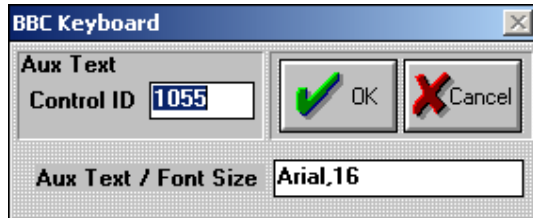
The external keyboard controls offer keycode mapping behaviour as ApplCore KR command modes 2 and 3.

However, certain keys / keycodes are excluded include the extra keys not defined on keyboard style 3. These keys / keycodes are not offered from the external keyboard (If a key / keycode is not represented on Keyboard Style 3, it is ignored, i.e. no-op).

Please note: The character "&" (Ampersand) is not offered because it is the edit buffer "editor cursor" control character. See the note about "&" in Keyboard Style 1's description.

Font support:

Keyboard supports selection of fonts for the keyboard key button faces. The default font is "System, 12". The desired font is specified at "design time" using Borland Workshop. Use of "Arial, 16" is shown. The edit buffer is automatically resized to accommodate the specified font size.



Note: The above results in the string "\$Arial,16\$" being written in the specifier field in the RC control specification. The default is "" (zero length string).

Run-time font change support:

Change Font and Font size (the same as BBC CC17)

Syntax: TP 'font=new font,size' <Pnl ID%> <Ctl ID%>

Example: TP 'Font=Arial,14' 1 1055

PLEASE NOTE:

Currently, there are certain limitations in size (height – width) of the keyboard panel versus the font size (typeface and point size used). These limitations cause graphical errors regarding the rendering of the font on the keyboard key button face. Two of the most susceptible keys are the "Home" and "CapLock" keys – with the large number of character on the key button face. Please see the Keyboard 3 example images following -----

Currently, the only solution is to select another typeface and or font size or resize the keyboard window.

Examples:

The keyboard test panel examples following use an ID of 1, and keyboard ID of 101.

Keyboard main keys colour controls:

TP '/A1__0__' 1 101 - light green button, dark red foreground/text
TP '/70__0__' 1 101 - dark gray button, black foreground/text
TP '/80__0__' 1 101 - gray button, black foreground/text
TP '/__7F0__' 1 101 - dark gray button, white foreground/text DOWN

Edit Buffer colour controls:

TP '/A1__1__' 1 101 :EI light green button, dark red foreground/text
TP '/70__1__' 1 101 :EI dark gray button, black foreground/text
TP '/80__1__' 1 101 :EI gray button, black foreground/text

Shift / CapLock colour controls:

Set light green button, dark red foreground/text, yellow down button, dark green foreground/text Shift / CapLock:

TP '/A1B22__' 1 101

Arrows, Home, End, Del & Clear colour controls:

Set light green button, dark red foreground/text, light blue button, brown foreground/text Arrows, Home, End, Del & Clear

TP '/A1C33__' 1 101

Edit Buffer enable/disable control:

TP '/____0' 1 101 -- disable
TP '/____1' 1 101 -- enable

Edit Buffer TG function control:

TP '/_____0' 1 101 -- normal

TP '/_____1' 1 101 -- set always return string mode

External Keyboard control:

TP '/_____0' 1 101 -- disabled

TP '/_____1' 1 101 -- not supported mode

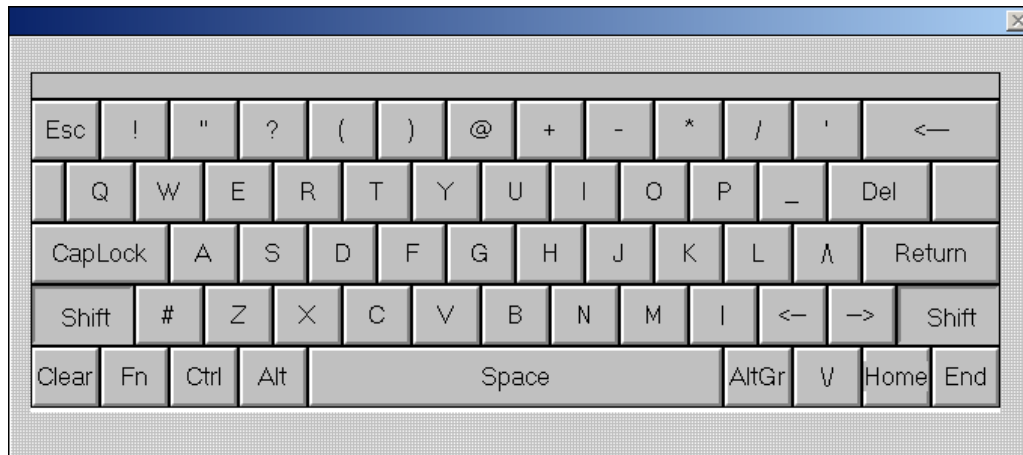
TP '/_____2' 1 101 -- enable external keyboard mode 2 (ref. ApplCore KR 2)

TP '/_____3' 1 101 -- enable external keyboard mode 3 (ref. ApplCore KR 3)

Important note:

Setting of the "A" & "B" **button colour** and **button text colour** will reset all of the other key button colours, i.e. **Shift / Caplock / Arrow, etc. groups button, foreground/text and down press colours.**

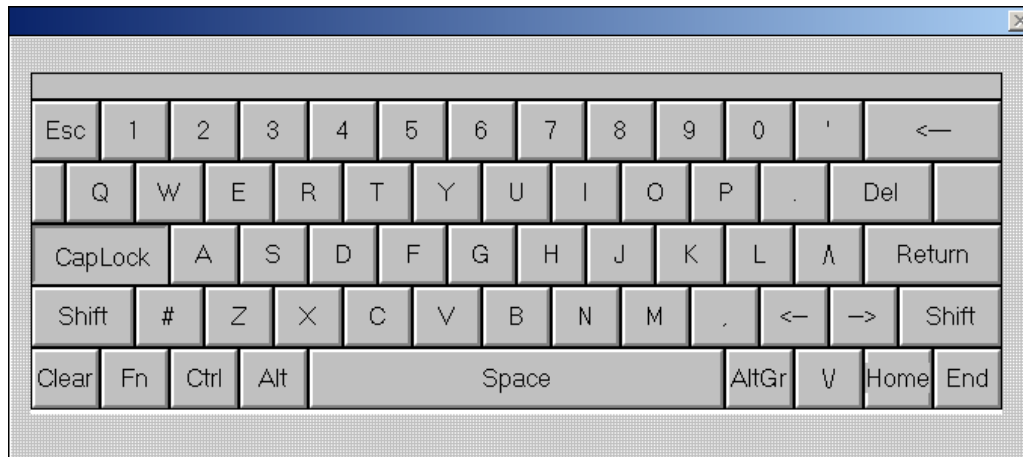
Shifted keyboard 3 (style code 8)



Unshifted keyboard 3 (style code 8)



CapsLock keyboard 3 (style code 8)



Function Lock keyboard 3 (style code 8)



Keypad Type 4 [style code 9]



This control was added to both the V2/16bit and V3/32bit libraries starting in versions BBC_CC05.dll (1.3.0) and BNCS_CC05.dll (3.1.6), dated: 28-Dec-2010.

Keypad 4 generates TEN Control Id messages.

The button colour controls are the same as Keyboard Type 3 & 4.

Text Get (TG) support:

The Text Get (TG) ApplCore command is support via two operating modes:

“normal / default” mode which returns the edit buffer on <Return> key press and subsequent characters as pressed one at a time.

While a <Ent> key press returns the edit buffer contents to a TG command – the next TG executed at keyboard Base Control Id + 0 returns a zero length string.

Pressing the <Ent> key executes Base Control Id + 0 and executing a TG command applied to Base Control Id + 0 returns the current keyboard edit buffer. Subsequence TG on Base Control Id + 0 return a string length of zero. Additional key presses continue to populate the edit buffer and execute Base Control Id + 1.

“Always Return String” mode – where the whole of the edit buffer is always returned.

Key presses execute ApplCore line Base Control Id + 1.

Control Id's:

Base Control Id + 0 executes when the <Ent> key is pressed.

Executing a Text Get (TG) command on the keyboard Base Id + 0 returns the contents of the keyboard edit buffer.

Base Control Id + 1 executes every time a key is pressed.

Executing a TG command on the Base Id + 0 then returns the character of the currently pressed key.

Base Control Id + 2 executes when the 'Up Arrow' is pressed.

Base Control Id + 3 executes when the 'Down Arrow' is pressed.

Base Control Id + 4 executes when the "Clear" key is pressed.

Base Control Id + 5 executes when the "Home" key is pressed.

Base Control Id + 6 executes when the "End" key is pressed.

Base Control Id + 7 executes when the "←" (Backspace) key is pressed.

Base Control Id + 8 executes when the "PgUp" key is pressed.

Base Control Id + 9 executes when the "PgDn" key is pressed.

Base Control Id + 10 executes when the "Del" key is pressed.

Base Control Id + 11 executes when the "Ins" key is pressed.

Base Control Id + 12 executes when the "<--" Left Arrow key is pressed.

Base Control Id + 13 executes when the "-->" Right Arrow key is pressed.

Key repeat:

Key repeat function is supported. 1 second initial, with ½ second repeat rate.

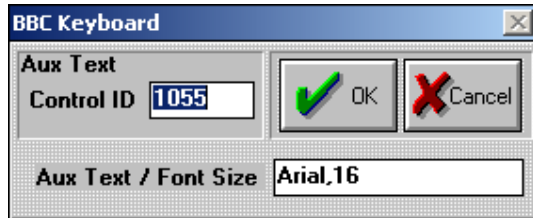
Keypad controls:

The Keypad controls are selected via the standard ApplCore interface using control characters sequences embedded in a ApplCore TP command (s). The TP command string definitions are as follows: "/ABCDEFG". See the examples following.

- "/" - escape sequence character
- "A" – Keyboard button colour (0-F)
- "B" – Keyboard button foreground/text colour (0-F)
- "C" – Button down colour (0-F)
- "D" – Button down foreground/text colour (0-F)
- "E" – Text Put Block Switch:
 - 0 – Main keys group (default, resets all keys colours groups - following)
 - 1 – Edit buffer (only "A" and "B" affect edit buffer)
 - 2 – NumLock keys group
 - 3 – Arrow, Home, End, Del keys group
- "F" – Edit buffer Control - visibility -- Enable: 1 / Disable: 0
- "G" – Text Get Control - TG support - what is returned in response to a TG command. Always Return String : 1 / Normal : 0 (Default: 0)

Font support:

Keypad supports selection of fonts for the keypad key button faces. The default font is "System, 12". The desired font is specified at "design time" using Borland Workshop. Use of "Arial, 16" is shown. The edit buffer is automatically resized to accommodate the specified font size.



Run-time font change support:

Change Font and Font size (the same as BBC CC17)

Syntax: TP 'font=new font,size' <Pnl ID%> <Ctl ID%>

Example: TP 'Font=Arial,14' 1 1055

PLEASE NOTE:

Currently, there are certain limitations in size (height & width) of the keyboard panel versus the font size selected (typeface and point size used). These limitations cause graphical errors regarding the rendering of the font on the keyboard key button face. Two of the most susceptible keys are the "Home", "CapLock" and "NumLk" keys – which have a large number of characters on the key button face. Please see the Keyboard 3 example images following.

Currently, the solution is to select another typeface and or font size or resize the keyboard window.

Examples:

The keypad test panel examples following use an ID of 1, and keypad ID of 101.

Keypad main keys colour controls:

TP '/A1__0__' 1 101 - light green button, dark red foreground/text
TP '/70__0__' 1 101 - dark gray button, black foreground/text
TP '/80__0__' 1 101 - gray button, black foreground/text
TP '/__7F0__' 1 101 - dark gray button, white foreground/text DOWN

Edit Buffer colour controls:

TP '/A1__1__' 1 101 :EI light green button, dark red foreground/text
TP '/70__1__' 1 101 :EI dark gray button, black foreground/text
TP '/80__1__' 1 101 :EI gray button, black foreground/text

NumLock colour controls:

Set light green button, dark red foreground/text, yellow down button, dark green foreground/text NumLock:

TP '/A1B22__' 1 101

Arrows, Home, End, Del colour controls:

Set light green button, dark red foreground/text, light blue button, brown foreground/text Arrows, Home, End, Del.

TP '/A1C33__' 1 101

Edit Buffer enable/disable control:

TP '/____0' 1 101 -- disable
TP '/____1' 1 101 -- enable

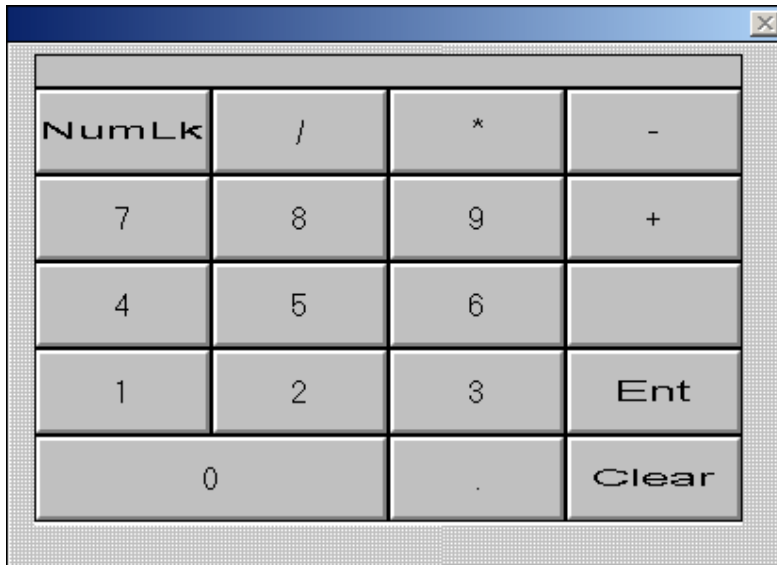
Edit Buffer TG function control:

TP '/____0' 1 101 -- normal
TP '/____1' 1 101 -- set always return string mode

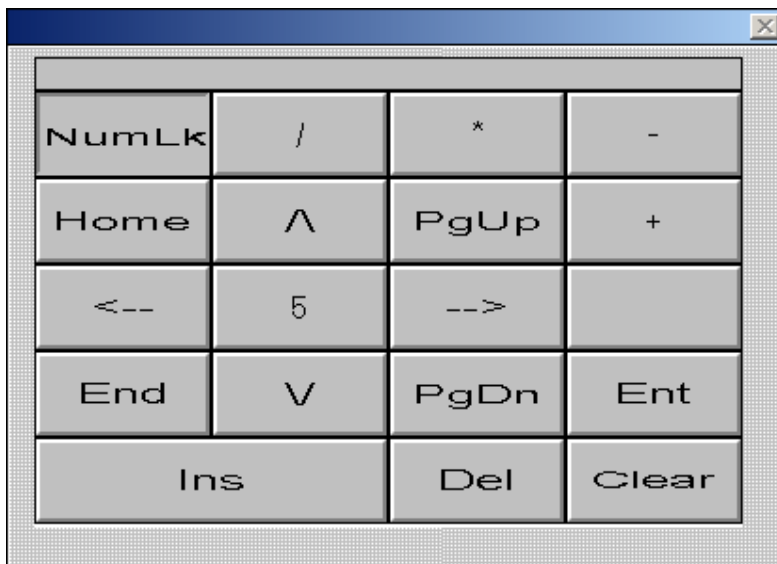
Important note:

Setting of the "A" & "B" **button colour** and **button text colour** will reset all of the other key button colours, i.e. **NumLock / Arrow, etc. groups button, foreground/text and down press colours.**

NON-NumLock keypad 4 (style code 9)



NumLock keypad 4 (style code 9)



Application Notes

Normally the library is used in support of ApplCore. Please note when the library is used with a Windows program other than ApplCore it can be useful to know what Windows messages the library responds to and utilises.

The Windows used / responded to include:

1. WM_GETTEXT and interface GetDlgItemText() is used to capture input text.
2. WM_GETTEXTLENGTH and GetWindowTextLength() is used to ascertain the current text buffer text length. Support for message WM_GETTEXTLENGTH was added BBC_C05.dll version: 1.6.3, BNCS_C05.dll version: 3.2.7.
3. WM_GETDLGCODE
4. WM_CREATE
5. WM_SIZE
6. WM_SETTEXT
7. WM_ENABLE
8. WM_DESTROY
9. WM_PAINT
10. WM_MOUSEMOVE
11. WM_LBUTTONDOWNBLCLK
12. WM_LBUTTONDOWN
13. WM_LBUTTONUP
14. WM_RBUTTONDOWN
15. WM_TIMER – used for key / button repeat functions.

Version History

- Original version supporting BBC_CC05, this document was an html page which documented Keyboard Styles 1 and 2, dated: 12-January-2001.
- 09-Dec-2010:
BNCS_C05.dll version: 3.1.6, 06Dec2010, BBC_CC05.dll, version: 1.3.0, 09-Dec-2010 -- updated to include Keyboard 3.
Correction of buffer over-run problems, buffers now limited at 255 characters.
Added Shift, CapLock changes.
Added colour support for key down press colours.
Added colour support for colours groups.
Added line execution support for Clear, Home, End, Arrow Up & Down.
- 27-Dec-2010:
Added support for Numeric Keypad (Keyboard Type 4)
- 10-March-2011
Corrected a number of problems reported regarding shift caplock interactions and spacing issues.
- 21-March-2011
BNCS_C05.dll version: 3.02.01
Added support for external keyboard in keyboard type 3 and type 4 – uses KeyHKAC.DLL
Added run-time don't selection support in keyboard type 3 and type 4.
- 10-May-2011
BNCS_C05.dll version: 3.02.02
BBC_CC05.dll version: 1.5.1
Correction for interaction with 'F' key and 'Fn' keys.
Documentation updates.

- 07-Nov-2011
BBC_CC05.dll version: 1.6.2, dated: 07-Nov-2011
BNCS_C05.dll version: 3.2.5, dated: 07-Nov-2011
Corrections for bug 2141 documentation corrections, bug 2142 missing design-time icons, bug 2158 keyboard +1 vector and bug 2160 "1" key repeat issues.
- 08-Nov-2011
BNCS_C05.dll version: 3.2.6, dated: 08-Nov-2011
Corrections for bug 2158 +1 vector key repeat issues.
- 09-Nov-2011
BNCS_C05.dll version: 3.2.6, dated: 09-Nov-2011
Corrections for bug 2158 <BackSpace> key repeat issues.
- 26-Jan-2012
Document converted from Siemens to Atos.
Corrections for bugs 2185, 2187, 2188 & 2189
Bug 2185 "Edit buffer" control and external keyboard issues
Bug 2187 TextLength returns 0 for style 1 and 2, now all support WM_GETTEXTLENGTH
Bug 2188 GETTEXT buffer overrun on style 1 and 2
Bug 2189 <Clear> key does not remove text
Added <Clear> key to Style 4 keyboard.
Added Documentation for "Edit Buffer".
BBC_C05.dll version: 1.6.3, dated: 26-Jan-2012
BNCS_C05.dll version: 3.2.8, dated: 26-Jan-2012
- 13-Feb-2012
Support for bug 2199, added external keyboard events support for Page Up and Page Down keys.
Corrections for more issues associated with buffer over write
BBC_C05.dll version: 1.6.4, dated: 03-Feb-2012
BNCS_C05.dll version: 3.2.9, dated: 14-Feb-2012

Atos IT Solutions and Services Limited
Faraday House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
Tel.: +44 (0)1276 696000