

wIntegrate®

**MISCELLANEOUS TIPS  
AND  
TROUBLESHOOTING**

**U2 Data Management Solutions**

1099 18<sup>th</sup> St., Suite 2500

Denver, CO 80202

Phone: 303.294.0800

FAX: 303.293.8880

Contributors

Kenny Brunel, Alan Buckley, Trevor Grover, Tim Rasmussen, Ed Robb,  
David Robertshaw, Susie Siegesmund

## Contents

Basic Usage of the wIntegrate Editor .....	1
Function Keys Don't Retains Settings When Remapped.....	12
How to Change the Keyboard Definitions .....	14
How to Force wIntegrate to Only Send Uppercase Characters.....	18
How wIntegrate Handles the Break Key .....	21
Mapping a Key to Display the Current Date.....	22
Numeric Keys Don't Map Correctly Using VT Emulation.....	24

## WINTEGRATE TECH TIPS

### BASIC USAGE OF THE WINTEGRATE EDITOR



wIntegrate®

A new screen editor WintEdit was introduced with wIntegrate 98. It is installed if you check the "Developer Additions" during installation. This is also put on the Start menu as Editor in the wIntegrate 98 group. Associations are created for .WIS and .WIT files. You can also specify WintEdit.exe as your TextEditor in wIntegrate.

Some of the features found in WintEdit include:

- Color-coded command syntax
- Edit Host BASIC programs
- Compile Host BASIC programs

There is on-line help available with WintEdit but this Tech Tip will highlight some of the more significant features.

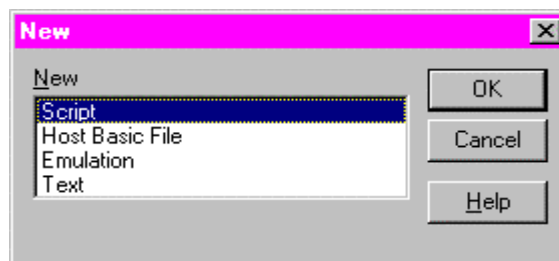
### ***EDITING A FILE***

When you first start the editor, select **File** to display the file options. You can create a new PC file, open an existing PC file, or open a text file on your host. You may open multiple files and easily switch between them.



*New*

Selecting **File | New** presents you with a choice of file types to create:



The only differences between each type are the colors used to display various commands and keywords, and the options available on the Tools menu.

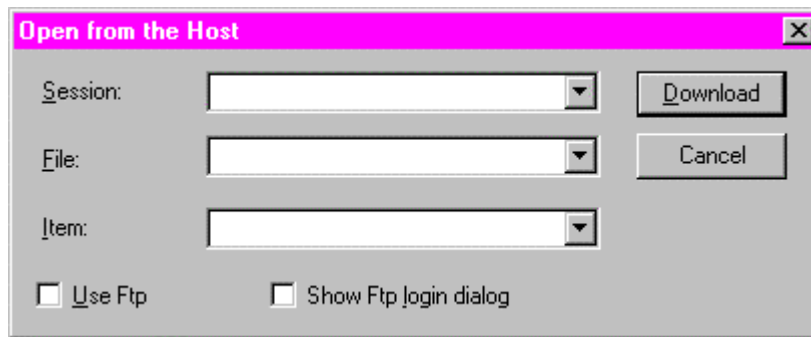
### *Open*

Opens an existing file. Defaults to the wIntegrate directory. When you have WintEdit configured to be the default editor in wIntegrate, it starts using the **Open** option.

### *Open From Host*

Opens a text file on the host. You must have a session connected to your host and be logged to an account where the host routines are accessible.

When you select **Open From Host**, the following dialog box is displayed.



#### Session

Enter the name of the session to open the file from. The drop down box will list all of the WIC files in the SESSION subdirectory.

#### File

Enter the name of the host file in which the item to be opened resides, for example, BP, \_HOLD\_, etc.

#### Item

Enter the name of the item in the file to be opened. The item must be text, for example, a BASIC program or saved report.

#### Use FTP

Check if you wish to use FTP to perform the transfer from the host. You must have FTP enabled on the host. This will be checked if you clicked **File | Use FTP** from the main menu.















### Show FTP login dialog







When checked, displays dialog box to prompt for username and password.

## ***TOOLBARS***

After creating or opening a file, the full toolbar is displayed.

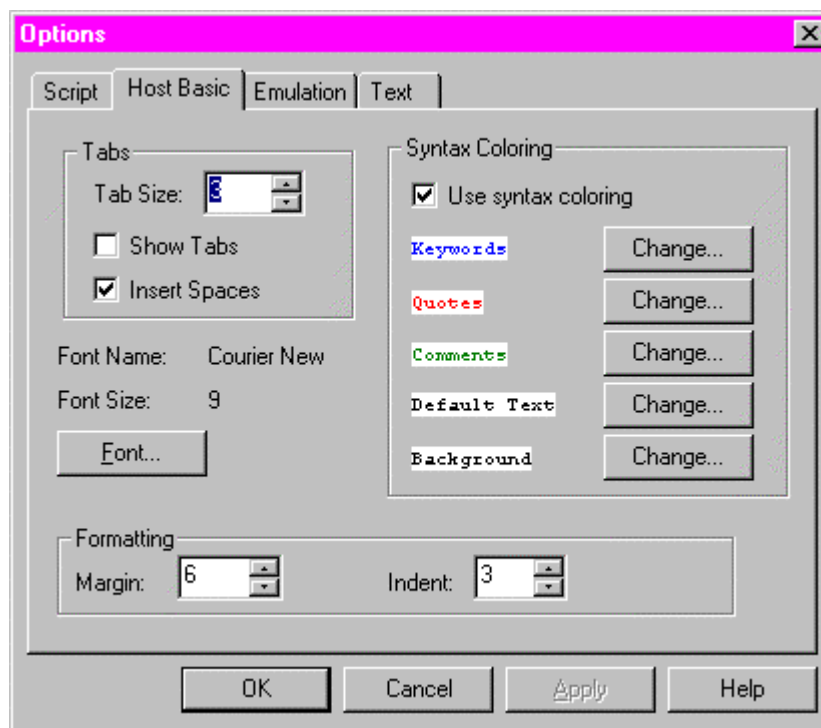


-  Open a new document.
-  Open an existing document. wIntegrate Editor displays the Open dialog box, in which you can locate and open the desired file.
-  Save the active document or template with its current name. If you have not named the document, wIntegrate Editor displays the Save As dialog box.
-  Open (download) basic program from host.
-  Save (upload) basic program to host.
-  Print the active document.
-  Remove selected data from the document and stores it on the clipboard.
-  Copy the selection to the clipboard.
-  Insert the contents of the clipboard at the insertion point.
-  Reverse the last editing. Note: You cannot undo some actions.
-  Undo last edit.
-  Redo result from last undo operation.
-  Find text.
-  Find next occurrence of text.

-  Toggle bookmark.
-  Locate next bookmark.
-  Locate previous bookmark.
-  Clear all bookmarks.
-  About wIntegrate.
-  Help.

### ***FORMATTING TEXT***

Select **Tools | Options** to modify the text formatting. The following dialog box is displayed.



The **Options** dialog box allows you to set the options used for displaying and editing files. There is one tab for each file type supported by the editor.

#### *Tab Size*

Sets the number of spaces to display for tab characters

### *Show Tabs*

Shows a double chevron symbol (») wherever a tab character appears in the file.

### *Insert Spaces*

Insert spaces instead of tab characters when the tab key is pressed. Overrides **Show Tabs**.

### *Font Name/Font Size*

The current font name and size used to display this file type. Click on the **Font** button to choose a different fixed pitch font to use.

### *Use Syntax Coloring*

Use syntax coloring to display this file type. This must be checked for the colors to be displayed. If 'Text' is selected all text will be displayed in black.

### *Keywords*

Sets the foreground color for keywords appropriate to this file type.

### *Quotes*

Sets the foreground color for text that is shown between single or double quotes.

### *Comments*

Sets the foreground color for comments.

### *Default Text*

Sets the foreground color for the all the text that does not fit into one of the above categories.

### *Background*

Sets the background color to be used for all text.

### *Margin*

Set the size of the left-hand margin when formatting the source text of a file.



### *Indent*

Set the size of the indent to use when indenting blocks of code when formatting the source text of a file.

### ***DEFINING KEYWORDS***

For each type of file - emulation, script, or host BASIC, there is an INI file containing a list of keywords. These keywords are displayed using the color selected for keywords under **Tools | Options**. Each list is composed of the commands available for the file type. For example, the keyword list for scripts contains all of the script commands and aliases. While quite complete, you may wish to add additional words to one or more of the lists.

There are 3 files containing keywords located in the WintEdit subdirectory of the wIntegrate folder: Emul.wis, Script.ini and HBasic.ini. These may be modified using any text editor. Each uses the same format:

```
[Keywords]
AB=1
Abort=1
Abs=1
AC=1
Activate=1
```

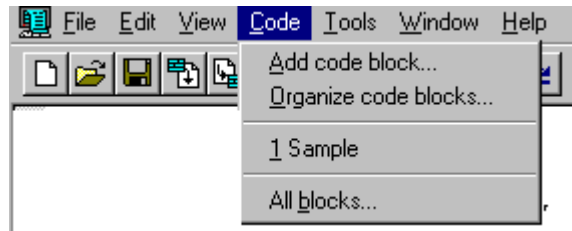
The first line must be [Keywords]. Add any new keywords anywhere in the list, followed by =1. The list is not case-sensitive and keywords do not have to be in alphabetical order. For example, if you wanted to add Programming to the top of the list in Script.ini, the entry would be:

```
[Keywords]
Programming=1
AB=1
Abort=1
Abs=1
AC=1
Activate=1
```

### ***ORGANIZING CODE BLOCKS***

Frequently when writing scripts or BASIC programs you may have blocks of code you reuse in multiple subroutines or other programs. Rather than retyping the code, or trying to cut and paste it to other programs, you can save the code as blocks with distinct names to be called up whenever you need to.

When you create or open an emulation file, script or host basic program, you will have the **Code** menu option displayed at the top of the editor.



### *Add code block*

Highlight the lines of code you wish to save then click **Code | Add code block**. Enter the name to use to identify this block. The name will be added to the list appearing under **Organize code blocks**.

### *Organize code blocks*

Organize code blocks allows you to delete a code block from a list of saved code blocks.

### *1,2...*

Inserts a code block at the location specified by the cursor. To insert one of the code blocks listed, click on its name.

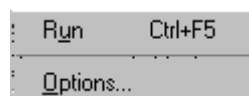
### *All blocks*

A total of 10 code block names can be displayed on the **Code** pull down menu. Click **All blocks** to display all code blocks. You may then select any block to insert into the file.

## ***TOOLS MENU***

The options for the **Tools** menu vary depending on the type of file being edited, from just **Options** when editing an emulation or text file, to numerous host-related options when editing a host program.

### *Script Files*



## Run

Runs the script.

## Options

Displays the **Options** dialog box (previously described).

## Host BASIC File

C <u>ompile</u>	F7
C <u>atalog</u>	
R <u>un</u>	Ctrl+F5
C <u>ompile and R</u> un	F5
C <u>ompile and C</u> atalogue	
H <u>ost Options</u> ...	
O <u>ptions</u> ...	

## Compile

Copies the program back to host then compiles it.

## Catalog

Catalogs the program on host.

## Run

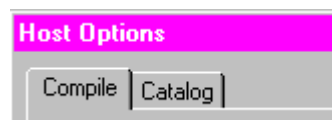
Compile and Run the program on the host.

## Compile and Catalog

Compile and Catalog the program on the host.

## Host Options

This option allows you to set the options used when compiling or cataloging the program on the host. There are 2 tabs - **Compile** and **Catalog**.



## Compile

### Machine Type

By default, this will already be set to the machine type used to install the wIntegrate host routines. The drop down box lists all machine types available with the host routines.

### Options

Enter any options to be included when compiling a program. For example, you may use -D, -G or -Z with UniData. The default is no options.

## Catalog

### Machine Type

By default, this will already be set to the machine type used to install the wIntegrate host routines. The drop down box lists all machine types available with the host routines.

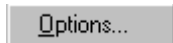
### Options

Enter any options you wish to use when cataloging a program. Depending on the machine type, this field may already have FORCE defined. You may type in an option or click on one of the buttons (FORCE, LOCAL, DIRECT, COMPLETE).

### Options

Displays the **Options** dialog box (previously described).

## *Emulation and Text Files*

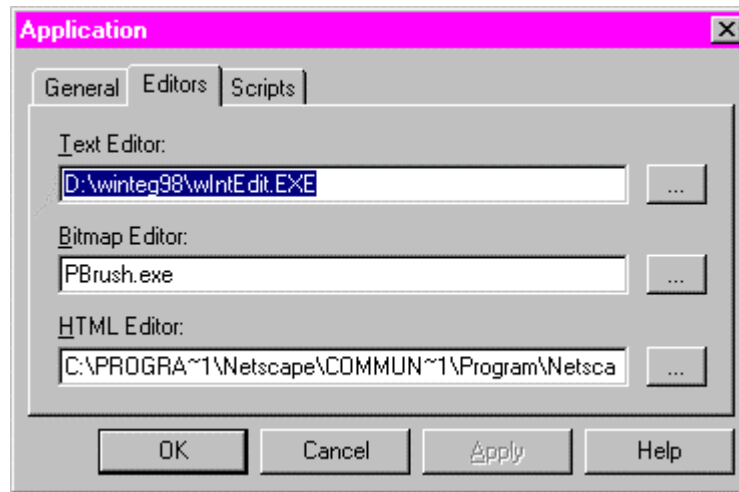


### Options

Displays the **Options** dialog box (previously described).

## ***USING WINTEDIT AS THE DEFAULT EDITOR***

When wIntegrate is first installed Windows Notepad is configured as the default text editor. To change this to WintEdit click **Setup | Application** then the **Editors** tab.



In the **Text Editor** box, type in the path to WintEdit.exe or click on the ... button to locate WintEdit manually. After clicking **OK**, be sure to click **File | Save**.

### ***STARTING WINTEDIT FROM THE HOST***

Typically, you would start WintEdit from the PC, whether by selecting the wIntegrate program group or from the wIntegrate **File** menu. You can also start it from the TCL/ECL command line or from a BASIC program. Included with the host routines shipped with wIntegrate 98 is a new subroutine WIN.EDIT.

#### *Command Line*

To start the editor from the TCL/ECL command line, use the syntax:

```
WIN.EDIT file item
```

where,

*file* = the name of the file to open, e.g. BP, \_HOLD\_, etc.

*item* = the name of the item to be edited. This must be text, for example, a BASIC program or saved report.

For example, to edit the wIntegrate host routine WIN.TRANSFER, in the WIN.PROGS file, the command would be:

```
WIN.EDIT WIN.PROGS WIN.TRANSFER
```

*BASIC Program*

You can also start the editor from within a BASIC program using the above syntax and the EXECUTE command.

```
EXECUTE "WIN.EDIT WIN.PROGS WIN.TRANSFER"
```

## WINTEGRATE TECH TIPS

### FUNCTION KEYS DON'T RETAIN SETTINGS WHEN REMAPPED



wIntegrate®

When you use one of the VT emulations in wIntegrate, the function keys are mapped much as they would be if you were using an actual VT terminal. However, keys F1 – F5 are not mapped as they would be on a DEC VT terminal because these keys perform functions specific to the VT terminal. wIntegrate maps F1 – F4 to be PF1 – PF4; nothing for F5. F6 – F12 are mapped as on the VT terminal; Shift-F1 – Shift-F4 are mapped as F13 – F16; Shift-F5 – Shift-F10 are mapped as F16 – F20. The default codes mapped are:

F1 = <esc>OP	S_F1 = <esc>[25~
F2 = <esc>OQ	S_F2 = <esc>[26~
F3 = <esc>OR	S_F3 = <esc>[28~
F4 = <esc>OS	S_F4 = <esc>[29~
F5 =	S_F5 = <esc>[28~
F6 = <esc>[17~	S_F6 = <esc>[29~
F7 = <esc>[18~	S_F7 = <esc>[31~
F8 = <esc>[19~	S_F8 = <esc>[32~
F9 = <esc>[20~	S_F9 = <esc>[33~
F10 = <esc>[21~	S_F10 = <esc>[34~
F11 = <esc>[23~	
F12 = <esc>[24~	

Under normal circumstances, these should be fine for your environment. However, if your host application depends on the use of the function keys, and expects them to be mapped differently than the default settings, or you like to use the function keys for keystroke shortcuts, you will need to clear the defaults.

The default settings for the function keys are stored in the emulation file `\winteg\wintsys\wit\vt***.wit`, where \*\*\* can be 100, 220 or 420. When you start wIntegrate the specific emulation file is read and cached into memory.

## ***MAPPING NEW SETTINGS***

1. Select **Setup | Keyboard**. The **Keyboard** dialog is displayed.
2. Click the **Function Keys** radio button under **Key List**. The escape sequences currently mapped to the function keys will be displayed in the white box at the right.
3. If you wish to clear the definitions for all the function keys, click **Clear**. This will clear the current settings for all the function keys. You can also change the definitions for selected function keys. In this case, enter the new definition without clicking **Clear**.
4. Map the new definitions you want for the function keys. Be sure to click **Set** after each key definition you remap. You can map different escape sequences, macros to run scripts, macros to perform a series of keystrokes, etc.
5. Click **OK**. This returns you to your session window.
6. Click **File | Save**. This stores the new definitions in your configuration file (.WIS if using wIntegrate 3.0 or .WIC if using wIntegrate 98), and will override settings in the emulation file. The next time you start wIntegrate using this configuration file the new definitions for your function keys will be loaded.



## WINTEGRATE TECH TIPS

### HOW TO CHANGE THE KEYBOARD DEFINITIONS



wIntegrate®

When you set up your terminal emulation and select a keyboard extension, wIntegrate maps some of the keys on your keyboard to perform the most commonly used functions. For example, when you use vt220 emulation and the key\_norm extension, the unshifted UpArrow key scrolls up through backpages one line at a time. You can use the default settings for the function keys or customize the settings to perform other functions.

You can remap, or assign a different function to, most keys or key combinations. For example, you can assign a new function to F1 and to the key combinations SHIFT+F1, CTRL+F1, and ALT+F1. However, you cannot remap the SHIFT, CTRL or ALT keys themselves and a few other keys reserved by Microsoft Windows for special uses. You can also remap keys to do specific tasks, such as send a keystroke or combination of keystrokes, or run a script.

We suggest you use caution when remapping keys, particularly any Alt-, Shift- or Ctrl-letters as the host CPU commonly utilizes these. Function keys are a different story, however. These tend to be application-specific so changing their functionality is not unusual.

#### ***REMAPPING A KEY BY ASSIGNING A SCRIPT***

The following steps demonstrate how to assign a script to a function key. In this example, we will remap the F5 key to open the **Preferences** dialog box by assigning the key a wIntegrate script that performs this task.

##### 1. Select the Keyboard Option

On the **Setup** pull-down menu, click **Keyboard**.

*wIntegrate 98*

The **Keyboard** dialog box appears with the **Key Definition Listing** tab open.

*wIntegrate 3.0*

The **Keyboard** dialog box appears.

## 2. Select a Function Key to Remap

Although you can select a key to remap from the first screen, you may find it easier to work from the keyboard view. To select the keyboard view,

*wIntegrate 98*

Click the **US Keyboard** tab.

*wIntegrate 3.0*

Click the **View** button. Keys with existing definitions will be displayed with red text.

Click the **F5** key. No script is currently assigned to this key, so the **Definition** box is blank..

## 3. Assign the new function to the key

In the **Definition** box, type **Show SetupPreferences**, the name of the script you are assigning to the F5 key. This script performs the function of opening the **Preferences** dialog box. Select the **Macro** check box so wIntegrate recognizes the script name you typed in the **Definition** box as a command. If you do not select the **Macro** check box, wIntegrate sends your input to the host as regular text.

Click the **Set** button to assign this definition to the F5 key. Click **OK** to save your changes. The wIntegrate main window appears.

## 4. View Results

Press the F5 key. If you have completed this remapping exercise successfully, the **Preferences** dialog box appears.

### ***REMAPPING A KEY BY CREATING A MACRO***

Now that you know how to remap a key by assigning it an existing wIntegrate script, you can apply some of the same concepts to create your own macro and assign it to a function key. As an example, assume that you use an accounting application to perform your daily work. On the main menu of your application, you might select option 2 to go to an Accounts Payable menu, then select option 4 for the Check Processing menu, and finally select option 1 to open the Print Checks dialog box.

So, to navigate to the Print Checks dialog box, you performed six keystrokes: one for each option and another for each time you pressed ENTER. If this is a common activity for you, you may want to map this series of keystrokes to a function key. The following steps show you how to create a macro that performs these six keystrokes and assign it to the SHIFT+F12 key combination.

## 1. Identify the Keystrokes

Make note of the keystrokes necessary to perform your task. In this example, the keystrokes to open the Print Checks dialog box are as follows:

## **2 ENTER 4 ENTER 1 ENTER**

### 2. Select the Keyboard Option

On the wIntegrate menu bar, select the **Setup** pull-down menu. Select **Keyboard**.

*wIntegrate 98*

The **Keyboard** dialog box appears with the **Key Definition Listing** tab open. Click the **US Keyboard** tab.

*wIntegrate 3.0*

The **Keyboard** dialog box appears. Click the **View** button.

### 3. Select the Key Combination

In this exercise, you'll use the keyboard graphic to select SHIFT+F12 as the key combination you are mapping.

First, select the **F12** key in the keyboard graphic. At the top of the dialog box, you'll see **Key: F12**. Now, select the **Shift** check box to specify the SHIFT+F12 key combination.

### 4. Define the Macro

Click in the **Definition** box and enter the script commands and keystrokes that make up the macro. For this example, type the following:

**ENTER "2";ENTER "4";ENTER "1"**

This macro says "type in a 2, press <CR>, type in a 4, press <CR>, type in a 1, press <CR>". You can adapt the same logic to map other key combinations just as easily.

Note: The wIntegrate script command **ENTER** sends text to the host, followed by a carriage return. When you enter the definition for a macro, use a semicolon to separate each command. Select the **Macro** check box. This tells wIntegrate to treat the series of commands and keystrokes as a macro instead of input to the host computer. All of the wIntegrate script commands can be found in the PC Scripts Reference manual, included on the wIntegrate CD.

Click the **Set** button, then click **OK**. The SHIFT+F12 key combination is now remapped for the current session.

### 5. Save the Session

You can save the mapping for this key combination as a part of the configuration for this session. On the **File** pull-down menu, click **Save** to save the session with this keyboard macro, or click **Save As** if you want to save the session as a new wIntegrate configuration (.wic) file. The next time you start a session with this name, the SHIFT+F12 key combination is mapped as you saved it.

## WINTEGRATE TECH TIPS

### HOW TO FORCE WINTEGRATE TO ONLY SEND UPPERCASE CHARACTERS



wIntegrate®

#### *INTRODUCTION*

From time to time, users have expressed an interest in having wIntegrate send only uppercase characters to their host. There are a variety of reasons why one might want to do this, for example, consistency in your data, or, perhaps, your database may require uppercase command syntax.

uniVerse has a CASE command that can be used to force uppercase mode when using uniVerse, but it also causes problems with file transfers as wIntegrate uses lower-case commands. UniData has no such feature, but there are times when one needs to force uppercase commands, for example, when using ECLTYPE 'P'. Your particular database or application environment may have similar requirements.

This Tech Tip will show you how to adapt wIntegrate to send only uppercase characters without having to change the operating environment on your host. This is accomplished using the local variable, **CapsLock**. We will show you how to use this in the session script as well as defining the Caps Lock key to run a macro to set it.

By changing the value of **CapsLock** you can affect how wIntegrate handles the Caps Lock key. Though you can't completely disable the key, you can make it more difficult for the user to change and effectively force Caps Lock to be on.

***CAPSLOCK***

The **CapsLock** variable is used to set the value of the Caps Lock key. Its syntax is as follows:

`CapsLock=option`

<b>Option</b>	<b>Description</b>
<i>Normal</i>	No processing for the Caps Lock key.
<i>On</i>	Caps lock is switched on when the focus goes to wIntegrate.
<i>Off</i>	Caps lock is switched off when the focus goes to wIntegrate.
<i>Recall</i>	wIntegrate remembers the caps lock setting when wIntegrate loses the focus and restores it when the focus returns.
<i>Force</i>	Soft control of caps lock. Ignores caps lock light (forces it off). When caps lock is on, shift + letter gives lower case. This is similar to how a French vt emulation keyboard works.

***EDITING THE SESSION SCRIPT***

One way you can configure wIntegrate to send upper case characters is to modify the configuration file used for the session. For example, the following segment represents the commands found in SESSION1.WIS.

```
* Full configuration script
* Saved on Mon, 25 Jan 1999 10:28
Configure
Window="40,89,560,336,0"
.
.
.
BreakKey=3
CapsLock="Normal"
NumLock="Normal"
.
.
.
PrinterCharMap=1
Update
```

If you are using wIntegrate 4.0, create a WIS configuration file by clicking **File | Save As**, then save the session using a WIS extension. Edit this file, set **CapsLock** to the desired value, then use it when you start wIntegrate. You can save it as a WIC format by clicking **File | Save As**.

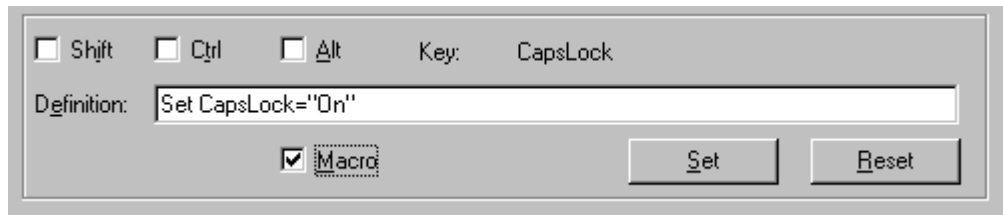
If you are using wIntegrate 3.0, simply edit the WIS file you are using. There is no conversion necessary.

By default, **CapsLock** is defined as **Normal**, meaning the Caps Lock key behaves as you would expect - uppercase when on, lowercase when off. The value defined for **CapsLock** only sets the initial condition of the Caps Lock key. The user still has the ability to use the Caps Lock key to change the case of the characters being used.

### ***REDEFINING CAPS LOCK KEY***

Each time you press the Caps Lock key you are either turning Caps Lock on or off. This is the default behavior. You can redefine the behavior of the Caps Lock key so that it sets the value of **CapsLock**, effectively preventing the user from changing the case of the characters being sent to the host.

Click **Setup | Keyboard**, then click the **US Keyboard** tab. This displays a map of the keyboard. Click the **CAPS** button on this display. In the Definition box, enter **Set CapsLock="On"**, check the **Macro** box, then click **Set**, as in the following example:



After clicking **Set**, click **OK** to exit the Keyboard dialog box. Be sure to click **File | Save** to store the changes.

Now, whenever the user presses the Caps Lock key, it will run the macro, **Set CapsLock="On"**. They will be unaware of this happening and will only know the Caps Lock key doesn't allow them to change to lowercase when in wIntegrate. Other applications will be unaffected by this change.

This is the most secure method to force uppercase characters but a user can still circumvent it by pressing shift+letter to get lowercase characters.

## WINGTEGRATE TECH TIPS

### HOW WINGTEGRATE HANDLES THE BREAK KEY



wIntegrate®

#### *TYPES OF BREAK*

There are two types of break, which we might call the soft break and the hard break. The soft break is a single ASCII character 3 sent to the host. Normally, this is entered by pressing <control>C. The hard break is (in simple terms) a very brief corruption of the communications connection. The specific action differs according to the connection type, for example serial or TCP/IP.

A host will often respond to one type of break and not to the other, or will respond differently. For example, if you connect to one Unix host and use the "telnet" command to connect to second host, the hard break acts on the first machine, but the ASCII character 3 is passed on to the second.

#### *CHANGING THE VALUE OF THE BREAK KEY*

In a standard wIntegrate installation <control>C is defined to send a hard break to the host. You can see this in **Setup | Keyboard**, where the BreakKey value is set to "3", the ASCII value for <control>C. So, with a standard install, if the <control>C doesn't break the host, change the BreakKey value to represent a different key, for example to "1" to represent <control>A. Now, when you press <control>C the single ASCII character 3, the soft break, will be sent. This usually works to resolve problems breaking the host.

If you want to send ASCII zero, you can either press <control><shift>@ or set a key to the value: \000.



## WINTEGRATE TECH TIPS

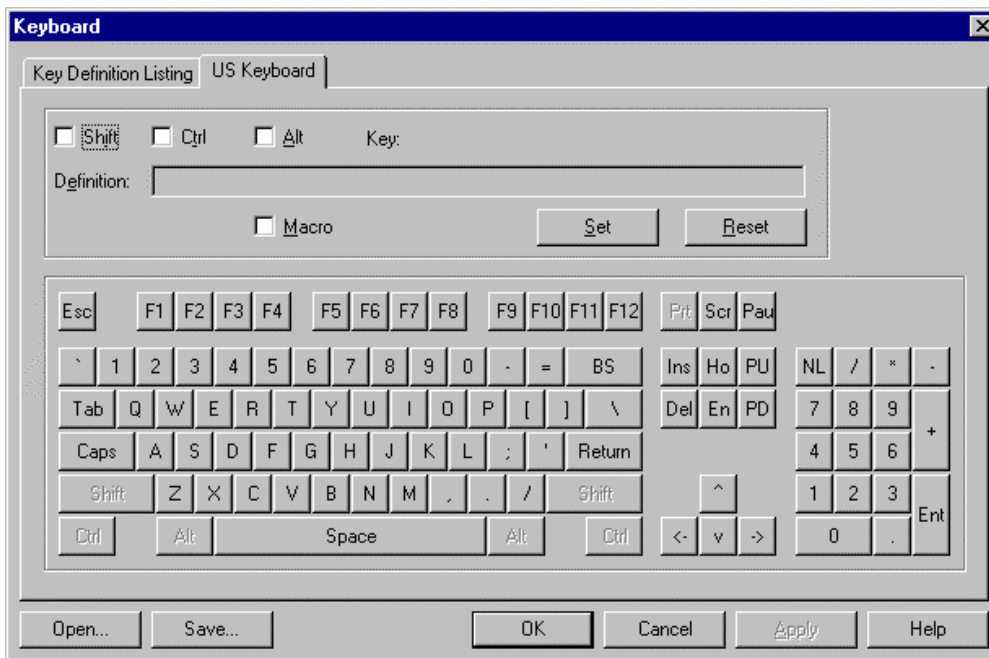
### MAPPING A KEY TO DISPLAY THE CURRENT DATE



wIntegrate®

Often in data entry applications, certain fields require that the current date be entered. Through a simple script command one can map this function to a key thereby reducing multiple keystrokes to one.

1. Open the **Setup | Keyboard** dialog and click on the **US Keyboard** tab.



Select the key you wish the map this function to. For this example, we'll use Shift-F12.

2. Check the **Shift** box, click on the **F12** button, and then check the **Macro** box. In the **Definition:** text box, enter the following line:

```
Type Oconv(Date(),"D2/")
```

3. Now click on the **Set** key and click **OK**.
4. From the wIntegrate **Main Menu** select **File | Save**. This will save the new key definition to the configuration file.



- Now when you press Shift-F12, the internal system date is captured and converted to MM/DD/YY format and displayed on the host.

## WINTEGRATE TECH TIPS

### NUMERIC KEYS DON'T MAP CORRECTLY USING VT EMULATION



wIntegrate®

On a standard VT terminal, the numeric keypad is mapped as an application keypad. The actual codes sent by these keys perform tasks specific to the DEC environment. In order to provide a compatible VT emulation, wIntegrate maps the numeric keypad exactly like a standard VT terminal. The default codes mapped for these keys are:

```
NumPad0 = <esc>Op
NumPad1 = <esc>Oq
NumPad2 = <esc>Or
NumPad3 = <esc>Os
NumPad4 = <esc>Ot
NumPad5 = <esc>Ou
NumPad6 = <esc>Ov
NumPad7 = <esc>Ow
NumPad8 = <esc>Ox
NumPad9 = <esc>Oy
Decimal = <esc>On
Add = <esc>Ol
Enter = <esc>OM
Subtract = <esc>Om
```

Under normal circumstances, these should be fine for your environment. However, if your host application depends on the use of the numeric keys, and expects them to be mapped differently than the default settings, or you like to use the numeric keys like a 10-key pad, you will need to clear the defaults.

#### ***CLEARING DEFAULT SETTINGS***

1. Select Setup | Keyboard
2. Click 'Numeric Pad' under Key List
3. Click 'Clear', then 'OK'. This will clear the current settings for the numeric keypad.
4. Select File | Save to save the settings

Depending on the host, after you clear the default settings you may find that upon starting wIntegrate again, the defaults return. This is a result of the host sending a code that forces the numeric keys to be reset. Should this occur, you will need to comment out the lines in the emulation file in order to disable the reset.

### ***DISABLING RESET***

When wIntegrate receives <esc>=, the VT52 sequence to enter alternate keypad mode, a script is run which assigns the numeric keys their default VT values. <esc>> exits the alternate keypad mode. If your host sends any of these escape sequences some, or all, of your numeric keys will be remapped to their default values, as described previously. To disable this:

1. Using Notepad, or any text editor, open \winteg\wintsys\wit\vtxxx.wit, where 'xxx' is the specific VT emulation you are using (VT100, VT220, or VT420).

2. In the .wit file, locate the following lines:

```
TerminalCommand      = "\e=", "Script 'wintsys\\wit_scr\\vtnumkey',1"  
TerminalCommand      = "\e>", "Script 'wintsys\\wit_scr\\vtnumkey',0"
```

3. Add an asterisk ('\*') before each line:

```
*TerminalCommand     = "\e=", "Script 'wintsys\\wit_scr\\vtnumkey',1"  
*TerminalCommand     = "\e>", "Script 'wintsys\\wit_scr\\vtnumkey',0"
```

4. Add 2 lines:

```
Ignore = "\e="  
Ignore = "\e>"
```

5. Save the file as a text file.

Now, when you start wIntegrate, should your host send the escape sequences to set the numeric keys, wIntegrate will ignore them and not reset the numeric keys.

### ***ENABLING OR DISABLING THE NUMERIC KEYPAD***

When you are using a VMS editor some numeric keys (Num Lock, /, \*, -) and edit keys (Home, Delete, End, etc.) perform specific tasks when pressed. The VT numeric keypad has 4 special keys, PF1 (Gold) through PF4, corresponding to the top row on the keypad. By default, wIntegrate maps these keys to F1 through F4. The 6 named edit keys are not mapped.

If you need to have the numeric keys and edit keys mapped as they would be on a typical VT terminal you will need to add an additional extension file to your configuration:

1. Select Setup | Terminal

2. Under Extensions, locate and click on 'vtaltkey'

3. Select File | Save

When you use this extension the following keys will be mapped:

Insert = Find

Home = Insert Here

PageUp = Remove

Delete = Select

End = Prev

PageDown = Next

Shift\_F5 = Help

Shift\_F6 = Do

Shift\_F7 = F17

Shift\_F8 = F18

Shift\_F9 = F19

Shift\_F10 = F20

\* The top row of keys on the numeric keypad

NumLock = PF1

Divide = PF2

Multiply = PF3

Subtract = PF4

While using the vtaltkey extension, the Num Lock key remains on for that session and functions as the PF1 (Gold) key. If you wish to reset the numeric keypad to its default configuration, press Shift-NumLock. To return to the application configuration, press Alt-NumLock. The result of the use of either Shift- or Alt-NumLock will not be saved unless you select File | Save.