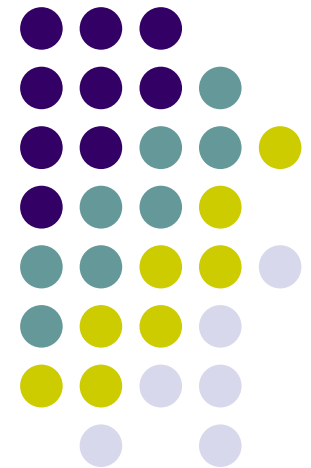# Bring your VIs to your LabVIEW User Group

Presented by Nick Golas

Telephonics Corp.

& Charlie Spitaleri

ALE System Integration

IEEE I&M Society LI Section &

Long Island LabVIEW Users Group (LILUG)

Thursday 3/13/2008

Add Date & Time to Filename

➢ Creates a path & a filename incorporating the Date & Time from the PC to date and timestamp the filename

```
Context Help

                    Time (Date & Time)
                    Date (Abbreviated)
      Initial Folder Path (c:\test)                          Complete Filename Path
          Filename Prefix (empty)                            Complete Filename Path as S...
          Filename Suffix (empty)                            error out
        Filename Extension (.txt)
              error in (no error)
         Field/Word Separator [Space...

                    Add Date & Time to Filename LV7.1.vi

This VI creates a path & a filename incorporating the Date & Time from the Computer to date/
timestamp the filename.
The  complete filename path consists of the following 6 sections:
  <Initial Folder Path>+ <Filename Prefix>+ <Date>+ <Time>+ <Filename Suffix>+ <Filename
Extension>
The Date enum control (default=abbreviated) supports the following formats:
Short - 1/21/94
Long - Friday, January 21, 1994
Abbreviated - Fri, Jan 21, 1994
The Time boolean control (default=Date & Time) allows to add both Date & Time to the filename or just
the Date Only
Field/Word Separator enum control [default=Space ()] makes the filename more readable. Besides the
Space () other separators can be utilized including the Underscore (_), Dash (-), Period (.) or None ()
Filename Prefix string control (default=empty) is an optional string to precede the date and time and is
usually used to identify the Program or the Function of the unit e.g. Ground Radar or ABC Program
etc.
Filename Suffix string control (default=empty) is an optional string that follows the date and time and
is usually used to identify the type of test performed e.g. ESS Test or Vibration Test etc.
Filename Extension string control (default=.txt) is usually a three or more character extension that is
appended to the end of the filename
Initial Folder Path (default=c:\test) path control is the folder to be appended to the beginning of the
Filename and the folder where the Filename will be saved
```
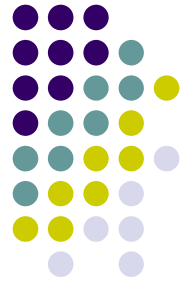
VI Documentation

DEMO

Filename: Add Date & Time to Filename LV7.1.vi

Add Date & Time to Filename Front Panel

Add Date & Time to Filename Block Diagram
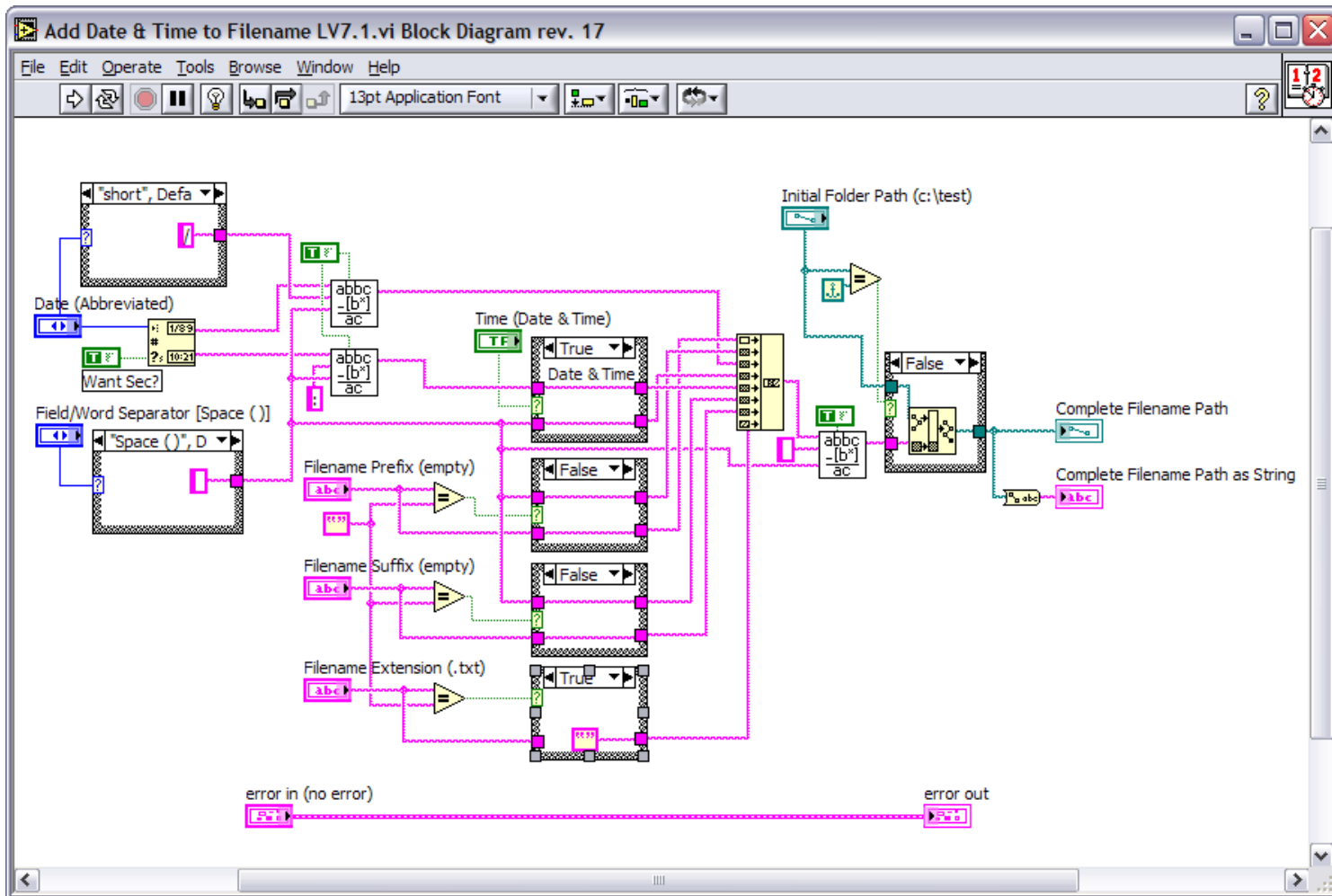
Check & Remove Illegal Characters from Filename
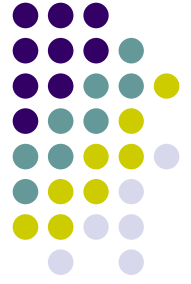
➢ Checks and removes any illegal characters that are not allowable as part of a filename by the Operating System. The default value [~0-9a-zA-Z.-_]) for the **Safe Characters to Keep** string control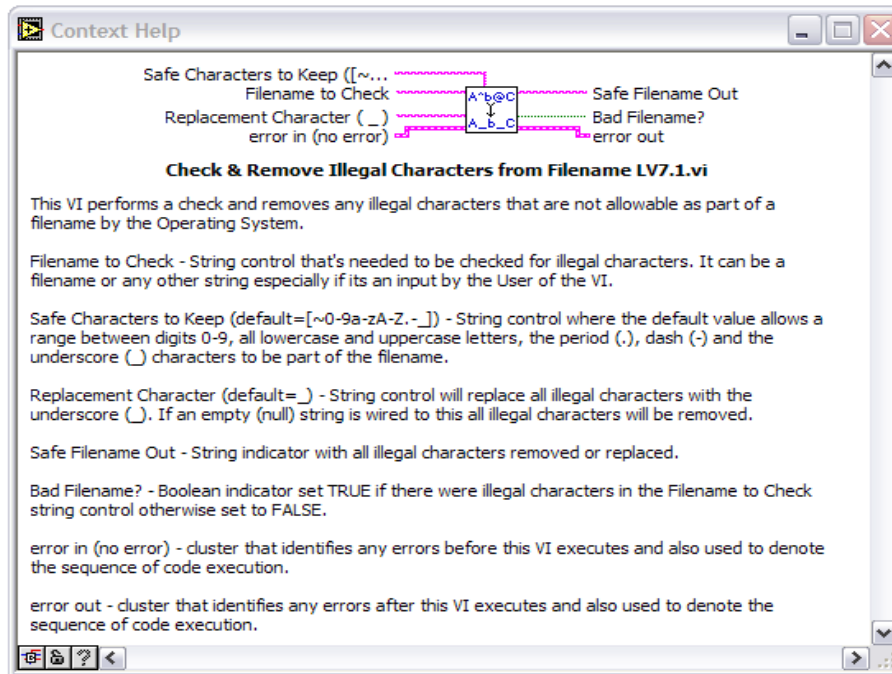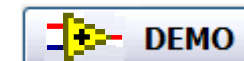 allows a range between digits 0-9, all lowercase and uppercase letters, the period (.), dash (-) and the underscore (_) characters to be part of the filename.

```
Context Help                                          _  □  ✕

        Safe Characters to Keep ([~...
                Filename to Check  ┈┈┈┈  A^b@C  ┈┈┈┈  Safe Filename Out
        Replacement Character ( _ )  ┈┈┈  A_b_C  ┈┈┈┈  Bad Filename?
                error in (no error)  ┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈  error out

        Check & Remove Illegal Characters from Filename LV7.1.vi

This VI performs a check and removes any illegal characters that are not allowable as part of a
filename by the Operating System.

Filename to Check - String control that's needed to be checked for illegal characters. It can be a
filename or any other string especially if its an input by the User of the VI.

Safe Characters to Keep (default=[~0-9a-zA-Z.-_]) - String control where the default value allows a
range between digits 0-9, all lowercase and uppercase letters, the period (.), dash (-) and the
underscore (_) characters to be part of the filename.

Replacement Character (default=_) - String control will replace all illegal characters with the
underscore (_). If an empty (null) string is wired to this all illegal characters will be removed.

Safe Filename Out - String indicator with all illegal characters removed or replaced.

Bad Filename? - Boolean indicator set TRUE if there were illegal characters in the Filename to Check
string control otherwise set to FALSE.

error in (no error) - cluster that identifies any errors before this VI executes and also used to denote
the sequence of code execution.

error out - cluster that identifies any errors after this VI executes and also used to denote the
sequence of code execution.
```
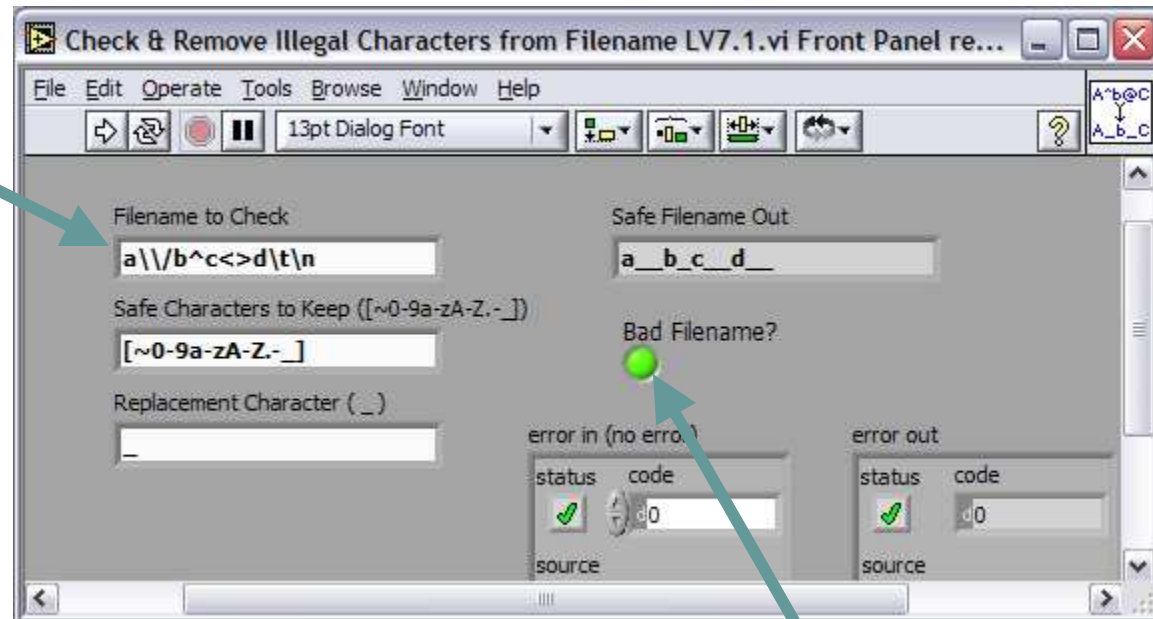
VI Documentation

🔧 **DEMO**

Filename:

Check & Remove Illegal Characters from Filename LV7.1.vi

Check & Remove Illegal Characters from Filename Front Panel
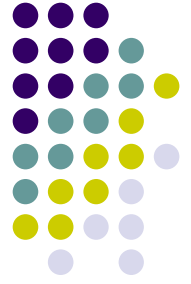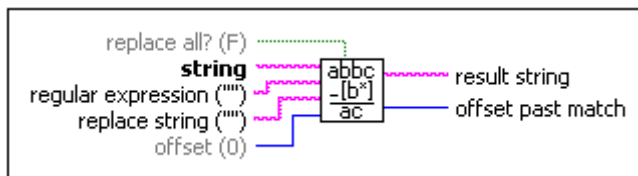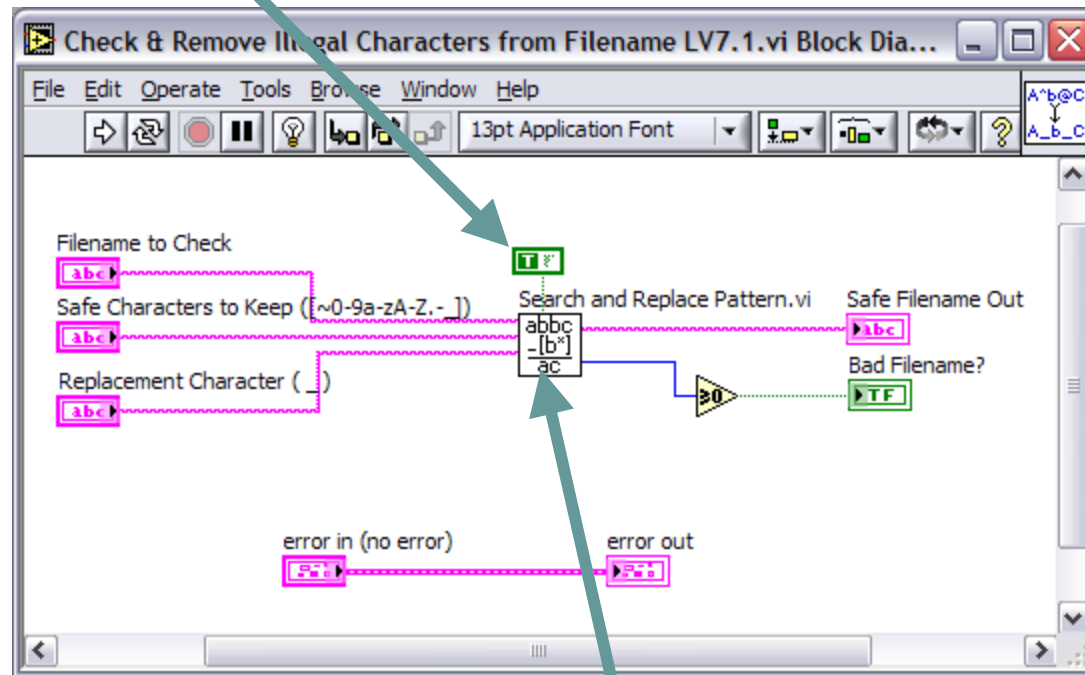
'\' Codes Display enabled

Can be used to verify any string for illegal characters

# Check & Remove Illegal Characters from Filename Block Diagram

**replace all?** if TRUE, then the VI replaces all substring occurrences

Check & Remove Illegal Characters from Filename LV7.1.vi Block Dia...

File Edit Operate Tools Browse Window Help

13pt Application Font

Filename to Check

Safe Characters to Keep (|~0-9a-zA-Z.-])

Replacement Character (_)

Search and Replace Pattern.vi

Safe Filename Out

Bad Filename?

error in (no error)

error out

replace all? (F)
string
regular expression ("")
replace string ("")
offset (0)
result string
offset past match

Search and Replace Pattern.vi
Searches string for substrings that match regular expression and replaces those substrings with replace string

Special Characters for Match Pattern
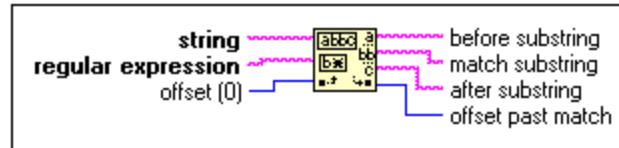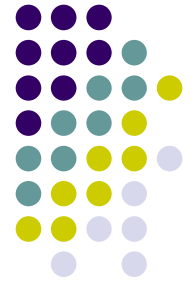
Use these special characters in the regular expression input of the Match Pattern function.

Special Character	Interpreted by the Match Pattern Function As
.	Matches any character. For example, l.g matches lag, leg, log, and lug.
?	Matches zero or one instances of the expression preceding ?. For example, be?t matches bt and bet but not best.
\	Cancels the interpretation of any special character in this list. For example, \? matches a question mark and \. matches a period. You also can use the following constructions for the space and non-displayable characters:
	\b backspace
	\f form feed
	\n newline
	\s space
	\r carriage return
	\t tab
	\xx any character, where xx is the hex code using 0 through 9 and upper case A through F

**Note: For LabVIEW v7.1. Newer versions of LabVIEW provide additional capabilities**

Special Characters for Match Pattern (Continued)

Special Character	Interpreted by the Match Pattern Function As
^	If ^ is the first character of **regular expression**, it anchors the match to the **offset** in **string**. The match fails unless **regular expression** matches that topic of **string** that begins with the character at **offset**. If ^ is not the first character, it is treate
[ ]	Encloses alternates. For example, [abc] matches a, b, or c. The following character has special significance when used within the brackets in the following manner.
	-(dash) Indicates a range when used between digits, or lowercase or uppercase letters; for example, [0-5], [a-g], or [L-Q].
	The following characters have significance only when they are the first character within the brackets.
	~ Matches any character, including non-displayable characters, except for the characters or range of characters in brackets. For example, [~0-9] matches any character other than 0 through 9.
	^ Matches any displayable character, including the space and tab characters, except the characters or range of characters enclosed in the brackets. For example, [^0-9] matches all displayable characters, including the space and tab characters, except 0 t

Note: For LabVIEW v7.1. Newer versions of LabVIEW provide additional capabilities
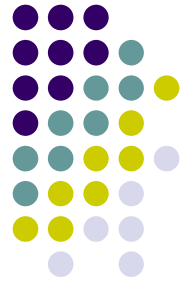
Special Characters for Match Pattern (Continued)
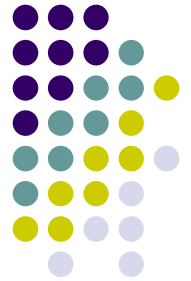
| Special Character	Interpreted by the Match Pattern Function As
+	Matches the longest number of instances of the expression preceding +; there must be at least one instance to constitute a match. For example, be+t matches bet and beet but not bt.
*	Matches the longest number of instances of the expression preceding * in **regular expression**, including zero instances. For example, be*t matches bt, bet, and beet.
$	If $ is the last character of **regular expression**, it anchors the match to the last element of **string**. The match fails unless **regular expression** matches up to and including the last character in the string. If $ is not last, it is treated as a regular char

Note: For LabVIEW v7.1. Newer versions of LabVIEW provide additional capabilities

Display Day and Time on the Front Panel Window Title

➢ **Displays the day and time in different formats on the front panel window title**

- **Short:** 1/21/94

- **Long:** Friday, January 21, 1994

- **Abbreviated:** Fri, Jan 21, 1994
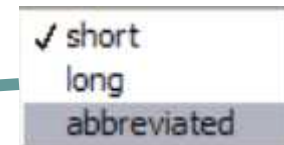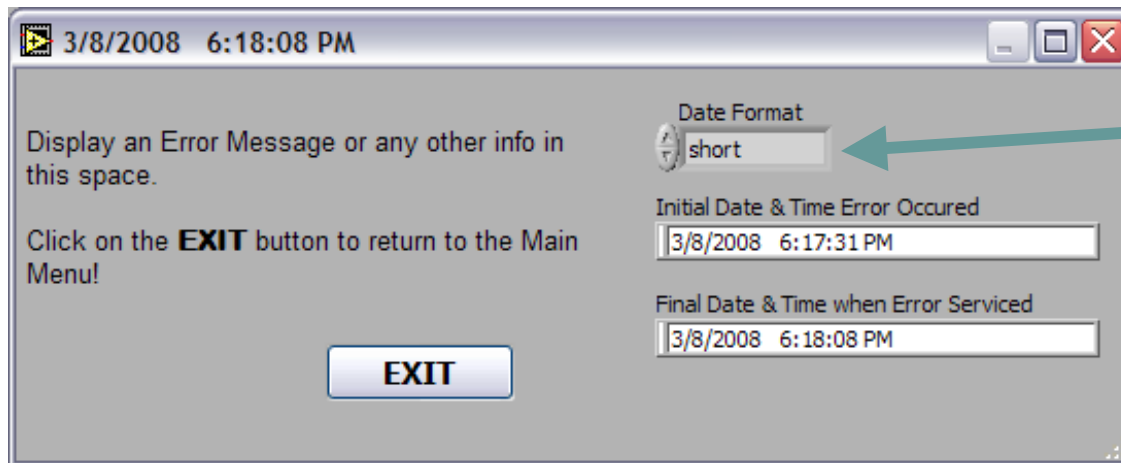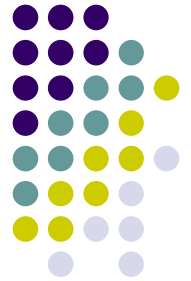
- **Time:** HH:MM:SS AM or PM

VI Documentation

DEMO
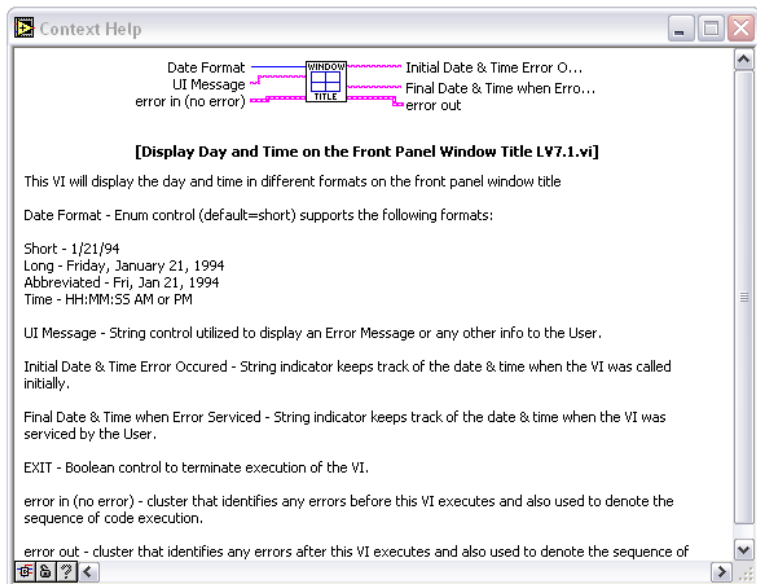
Filename:

Display Day and Time on the Front Panel Window Title LV7.1.vi

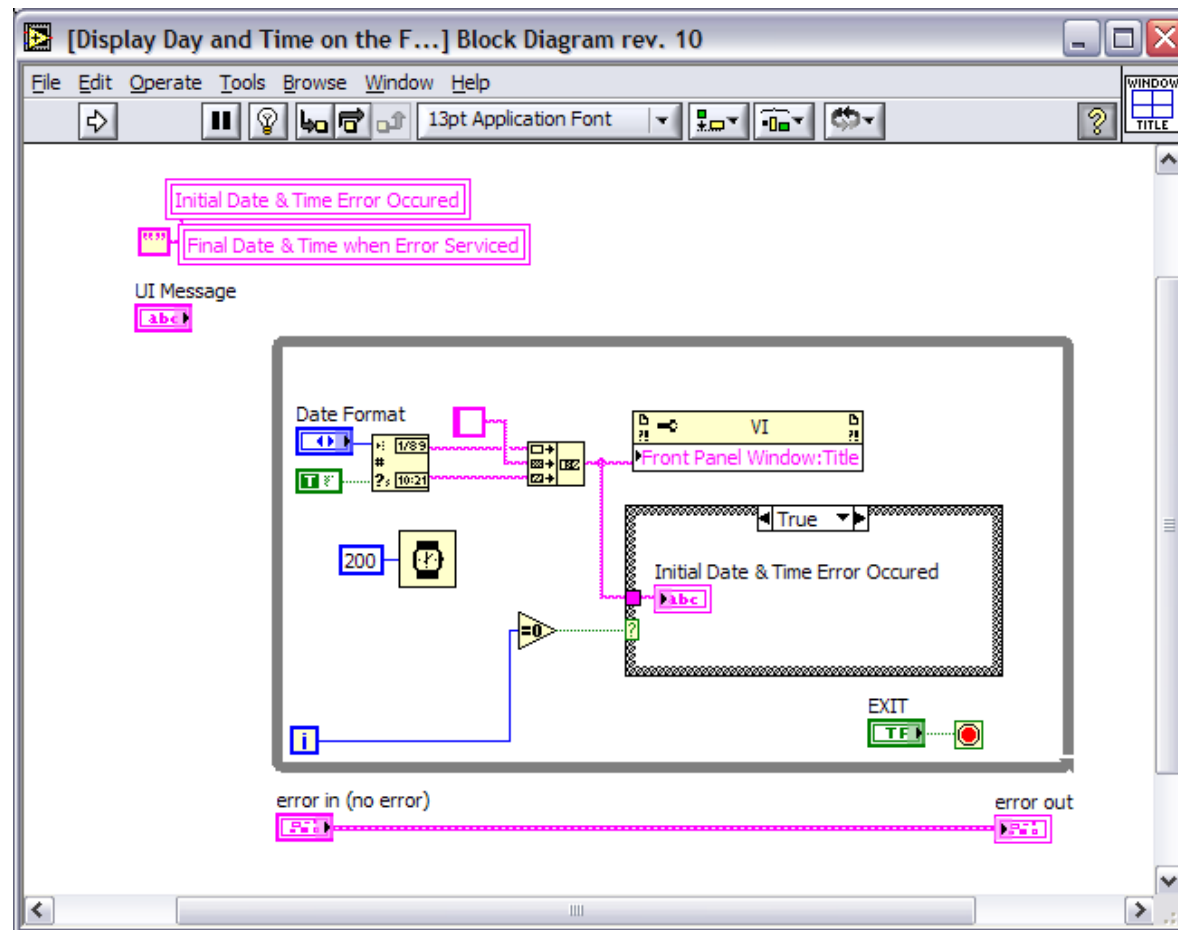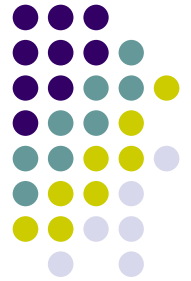# Display Day and Time on the Front Panel Window Title
Front Panel

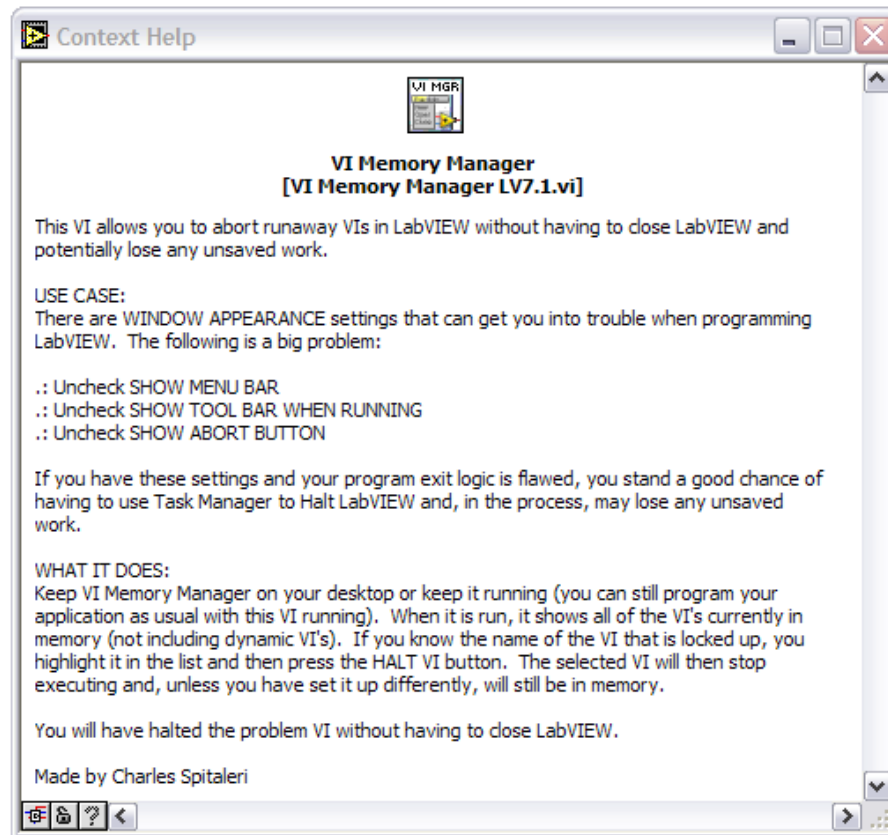**Resize Front Panel to appear as this**

Display Day and Time on the Front Panel Window Title Block Diagram

VI Memory Manager

➤ Abort runaway VIs in LabVIEW without having to close LabVIEW and potentially lose any unsaved work
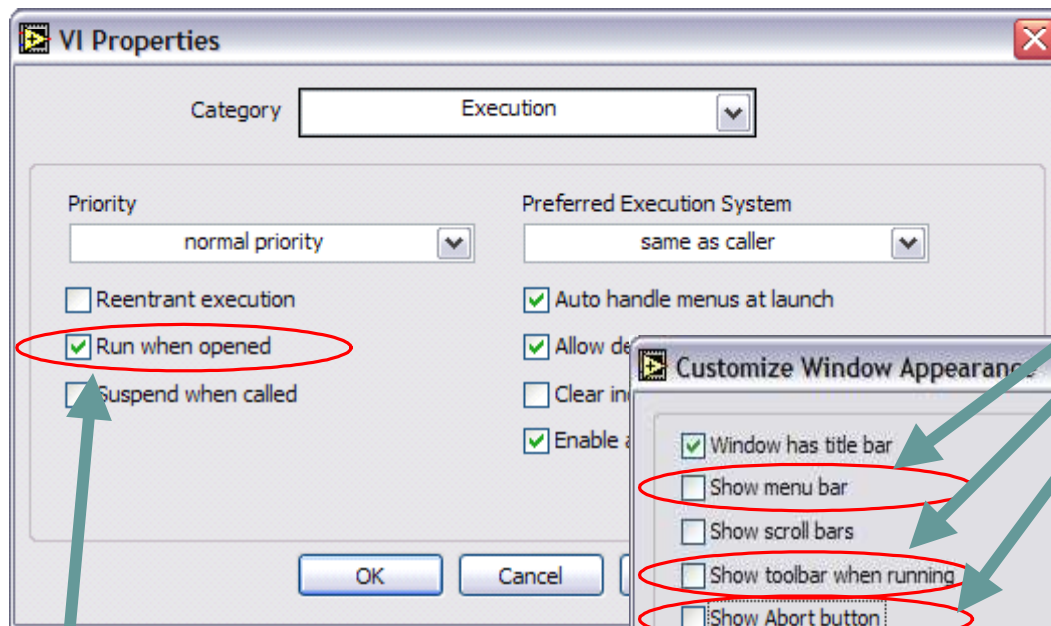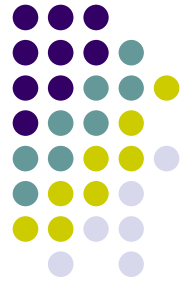

Context Help

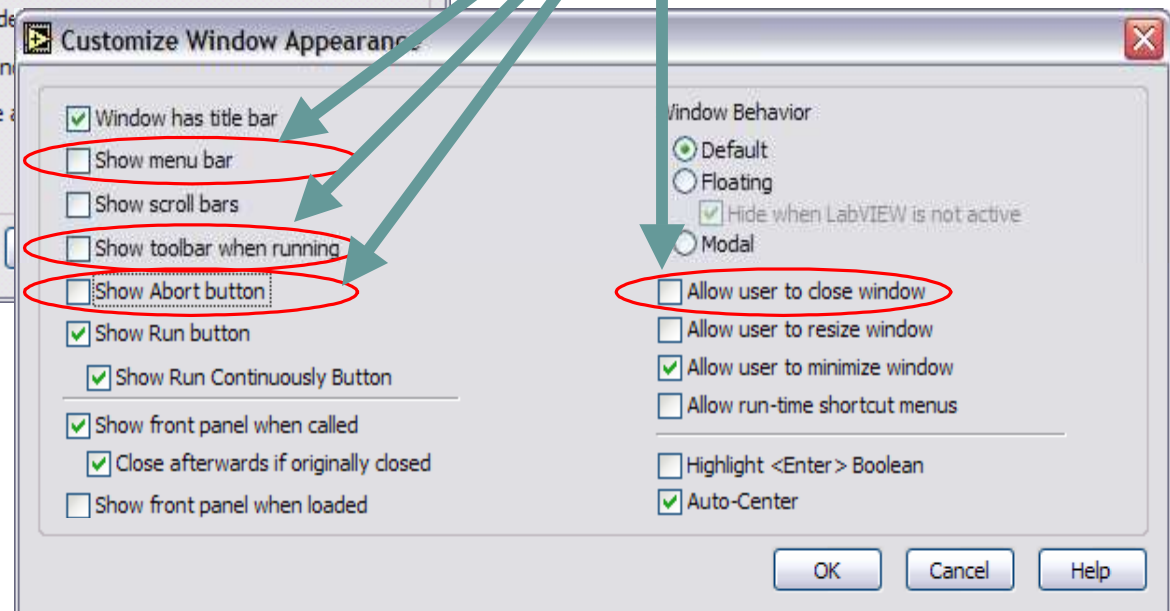**VI Memory Manager**
[VI Memory Manager LV7.1.vi]

This VI allows you to abort runaway VIs in LabVIEW without having to close LabVIEW and potentially lose any unsaved work.

USE CASE:
There are WINDOW APPEARANCE settings that can get you into trouble when programming LabVIEW. The following is a big problem:

.: Uncheck SHOW MENU BAR
.: Uncheck SHOW TOOL BAR WHEN RUNNING
.: Uncheck SHOW ABORT BUTTON

If you have these settings and your program exit logic is flawed, you stand a good chance of having to use Task Manager to Halt LabVIEW and, in the process, may lose any unsaved work.

WHAT IT DOES:
Keep VI Memory Manager on your desktop or keep it running (you can still program your application as usual with this VI running). When it is run, it shows all of the VI's currently in memory (not including dynamic VI's). If you know the name of the VI that is locked up, you highlight it in the list and then press the HALT VI button. The selected VI will then stop executing and, unless you have set it up differently, will still be in memory.

You will have halted the problem VI without having to close LabVIEW.

Made by Charles Spitaleri

VI Documentation

DEMO

Filenames: **VI Memory Manager LV7.1.vi &**
Runaway LV7.1.vi

Note: If you are using **LabVIEW 8.0** or later use **VI Memory Manager LV8.0.vi**
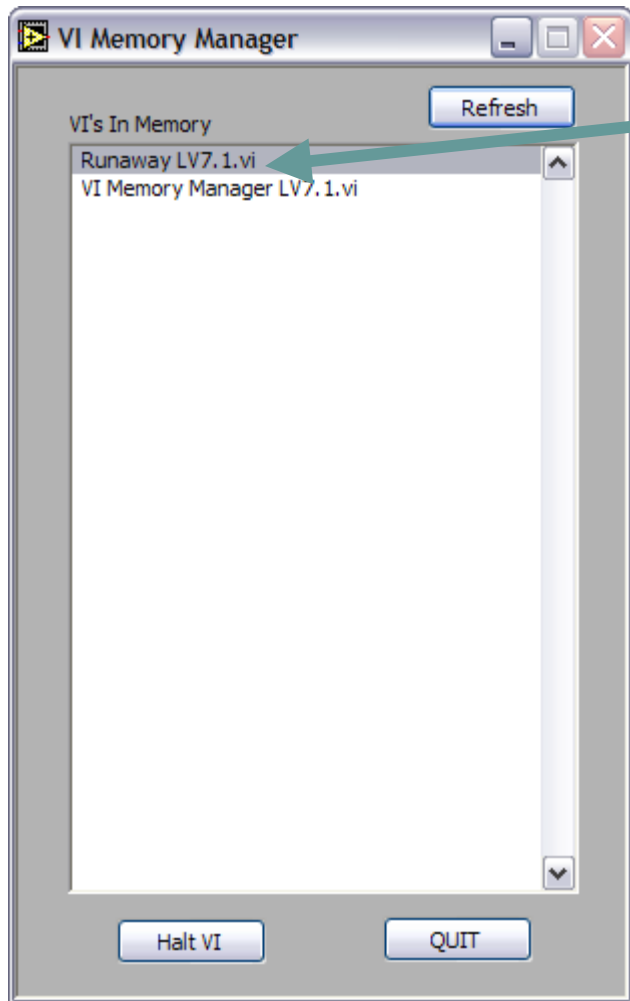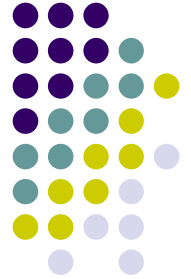
VI Memory Manager
"VI Properties" Pitfalls

Unchecking these **Window Appearance** settings can get you into trouble when programming **LabVIEW**

Having the VI **Run when opened** might create problems if it cannot be stopped

VI Memory Manager
Front Panel

VI Memory Manager

VI's In Memory   Refresh

Runaway LV7.1.vi
VI Memory Manager LV7.1.vi

Halt VI   QUIT

Highlight runaway VI
and click on **Halt VI**
button

Can not use **Close**
button because its
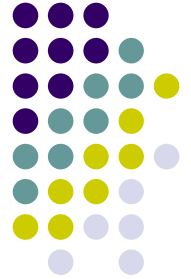disabled

Runaway LV7.1.vi rev. 7

DO NOT PRESS!
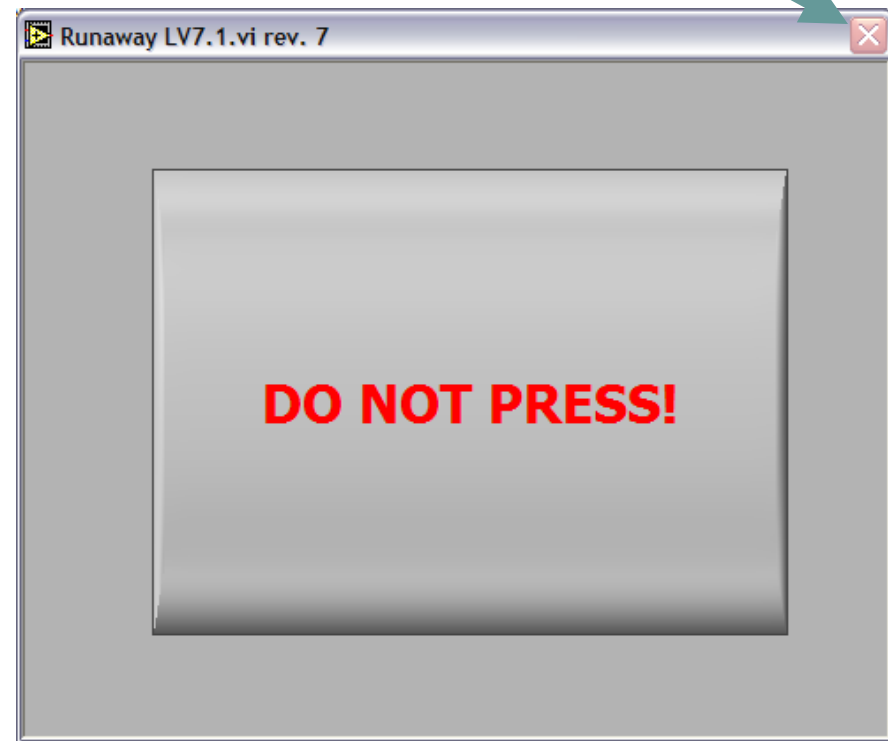
VI Memory Manager
Block Diagram
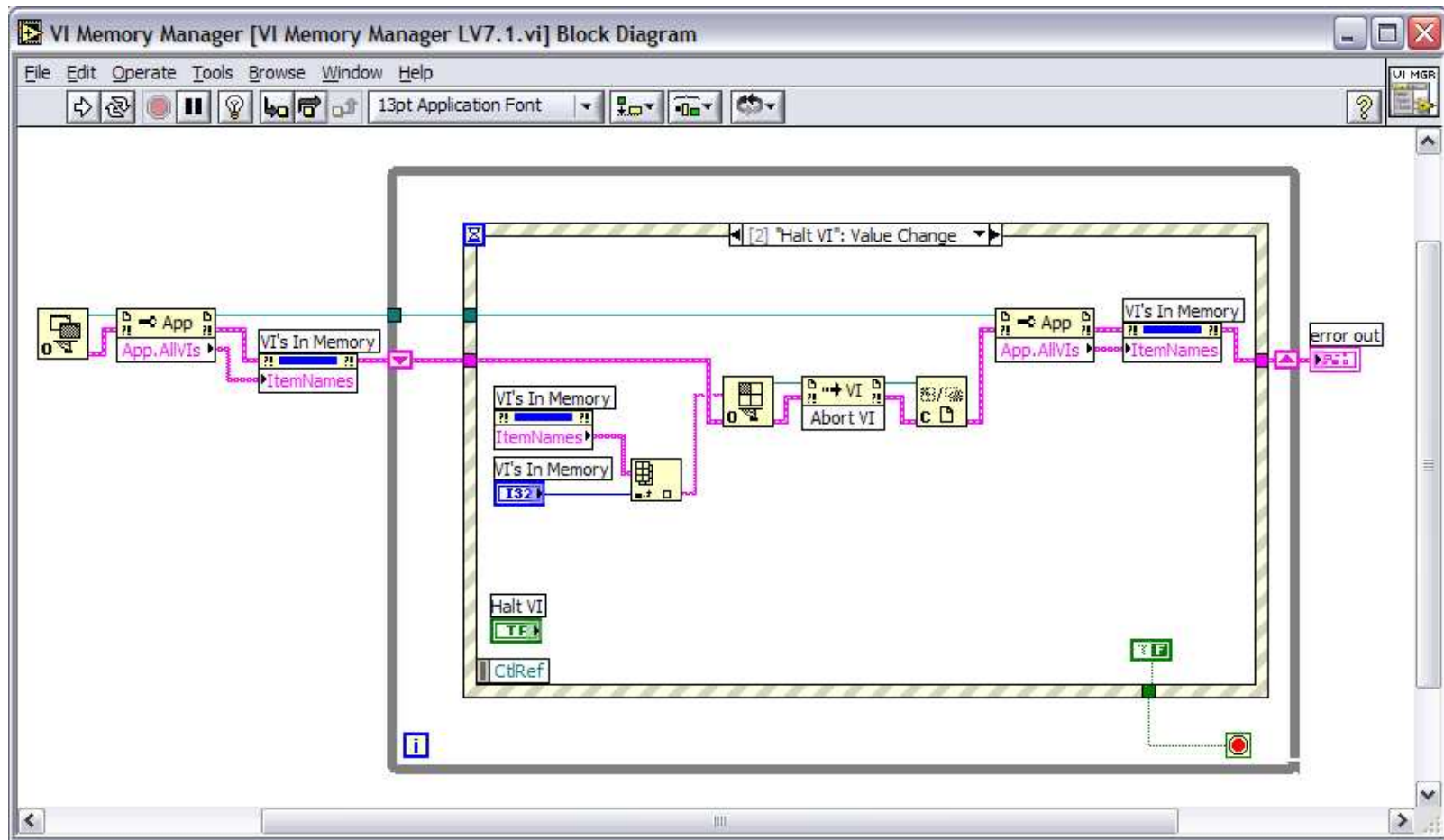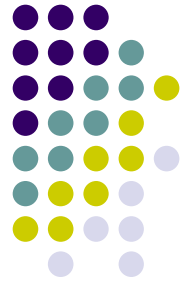
Viewgraphs & Examples

The Viewgraphs and Examples of this presentation will be available in a few days at:

- ➢ IEEE Long Island Section website, Instrumentation & Measurement Society page

    www.IEEE.LI

- ➢ ALE System Integration website User Group page

    www.aleconsultants.com