



# LabVIEW Developer Days

Build Code. Form Communities. Gain Confidence.



# User Interface Tips and Tricks

# Agenda

1. Definitions, rules, and advice  
(not specific to LabVIEW, but important)
2. Some cool UI techniques for LabVIEW  
(and why you would consider using them in your application)
3. Where to go to download some reusable components  
(because everybody loves free stuff)

# What Is a UI?

- **UI: User Interface**
  - How user interacts with the program
  - First thing the user notices
- **UX: User Experience**
  - Sometimes used interchangeably with UI
  - Broader, covers workflow

# Some General Rules

1. Do not be innovative
2. Less is more
3. Think about your user



# 1. Do Not Be Innovative

## Use familiar elements

- Buttons
- Icons
- Terminology
- Dialogs
- Menus





# 1. Do Not Be Innovative

## Still some creative license

- Do not change the way similar looking things behave
- Polish, do not reinvent

# Style Guidelines

- Make UI decisions once and record them
  - Often a living document
- Be consistent
  - Inconsistency robs your users of productivity
  - VI Analyzer can help enforce
- Not everyone needs to be a UI/UX expert



- References:
  - [LabVIEW Style Guide](#)
  - [Windows Application UI Development Guidelines](#)
  - [Apple OS X Human Interface Guidelines \(HIG\)](#)



# Style Guidelines - Example

Dev Center - Desktop > Docs > Desktop app development documentation > Windows Application UI Development > Windows User Experience Interaction Guidelines > Guidelines > Controls > Command Buttons

## Command Buttons

- Windows Desktop App Development
- Desktop app development documentation
- Windows Application UI Development
- Windows User Experience Interaction Guidelines
- Guidelines
  - ▾ Controls
    - Balloons
    - Check Boxes
    - Command Buttons
    - Command Links
    - Drop-down Lists & Combo Boxes
    - Group Boxes
    - Links
    - List Boxes
    - List Views
    - Progress Bars
    - Progressive Disclosure

79 out of 108 rated this helpful - [Rate this topic](#)

[Is this the right control?](#)

[Design concepts](#)

[Usage patterns](#)

[Guidelines](#)

[General](#)

[Split buttons](#)

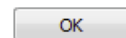
[Default values](#)

[Recommended sizing and spacing](#)

[Labels](#)

[Documentation](#)

With a *command button*, users initiate an immediate action.



A typical command button.

The *default command button* is invoked when users press the Enter key. It is assigned by the developer, but any command button becomes the default when users tab to it.

**Note:** Guidelines related to [layout](#) are presented in a separate article.

**Is this the right control?**

To decide, consider these questions:

## 2. Less Is More

- Too much on screen at once is distracting
- Allow your user to focus on what is important
- Animations, decorations have their place, but use sparingly

# 3. Think About Your User

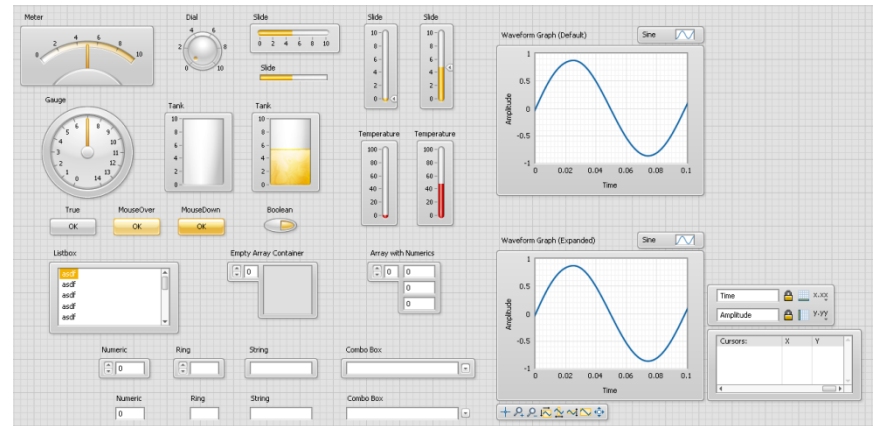
- They probably do not know as much as you
  - Explain what buttons do
  - Keep them informed about what your program is doing
- Know how the user plans on using your application
  - Mouse, keyboard?
  - Touch screen → large buttons
  - Outdoors → high contrast

# Better Yet...

Great UI design takes talent, training, and/or experience

If you have access to experts, use them

They don't have to be LabVIEW users – PPT, PDF, Photoshop work well to iterate on designs



Silver controls initial design in Photoshop

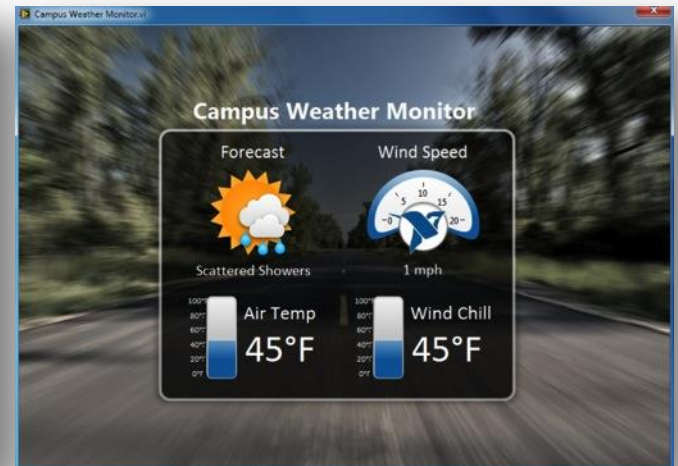
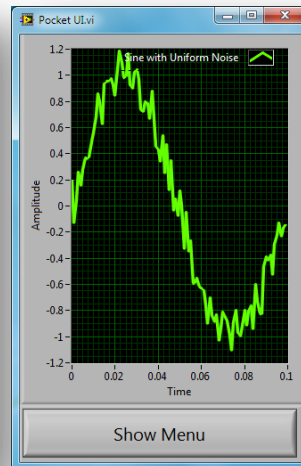
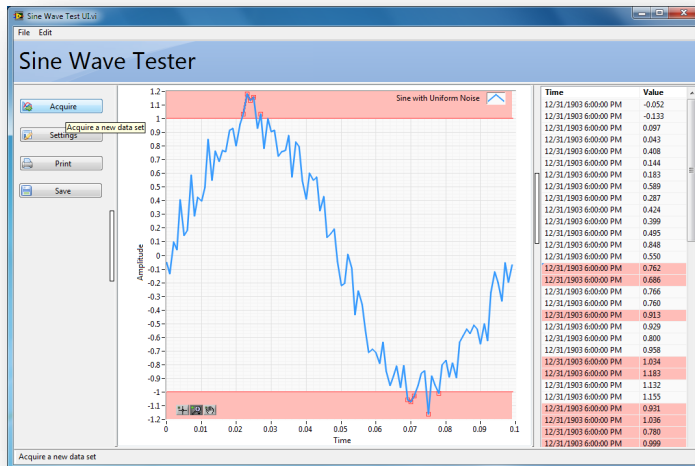


UI Design ver2.pdf

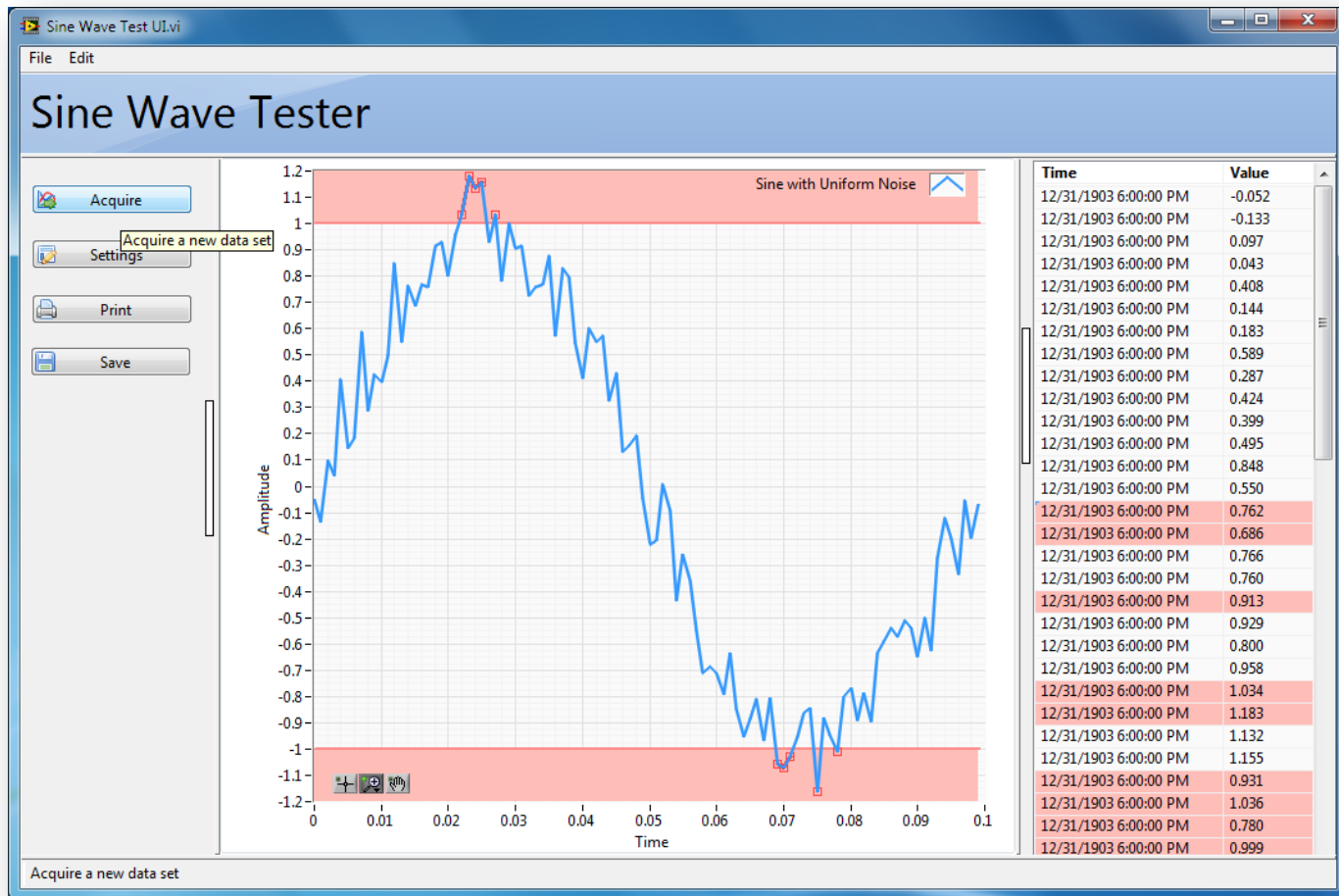


# Let's Take a Look at Some UIs

- Windows Desktop App
- Small Touch Screen App
- Informative Kiosk Display



# Windows Desktop App



# Applying the Rules

## Desktop Windows OS Application

### Do not be innovative

- Use system controls
- Add familiar icons to task buttons
- Use X to close application

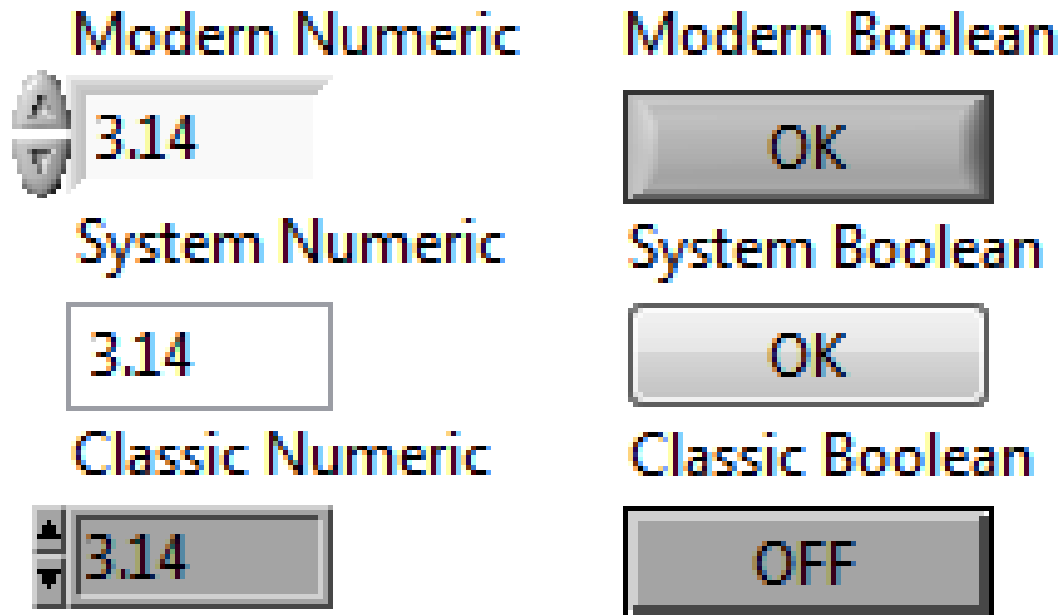
### Less is more

- Allow user to hide less important displays
- Hide the LabVIEW toolbar
- Do not persist one-time configuration controls for no reason → use temporary dialogs
- Customize the run-time menu

### Think about your user

- Create a status bar and use the busy cursor to update user
- Use tooltips to clarify functionality
- Allow the user to cancel long tasks
- Use panes to let the user resize your application

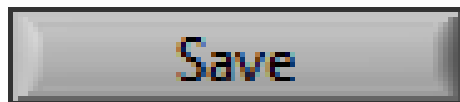
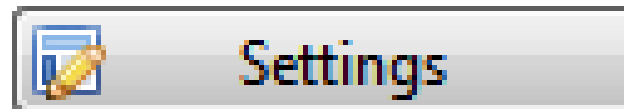
# Use Appropriate Controls



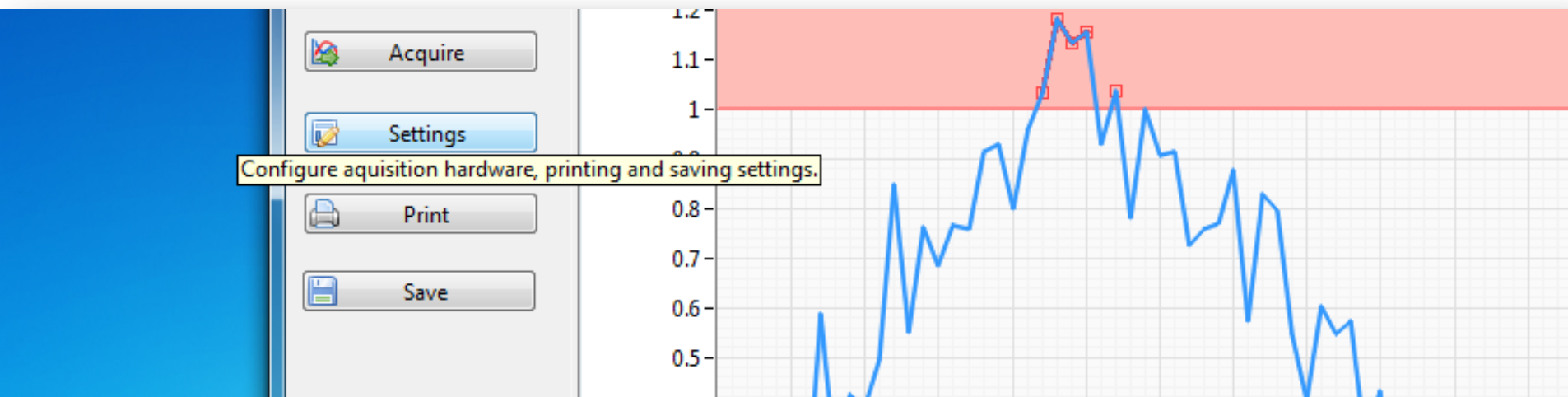
You can change your default type in **Tools»Options»Front Panel**



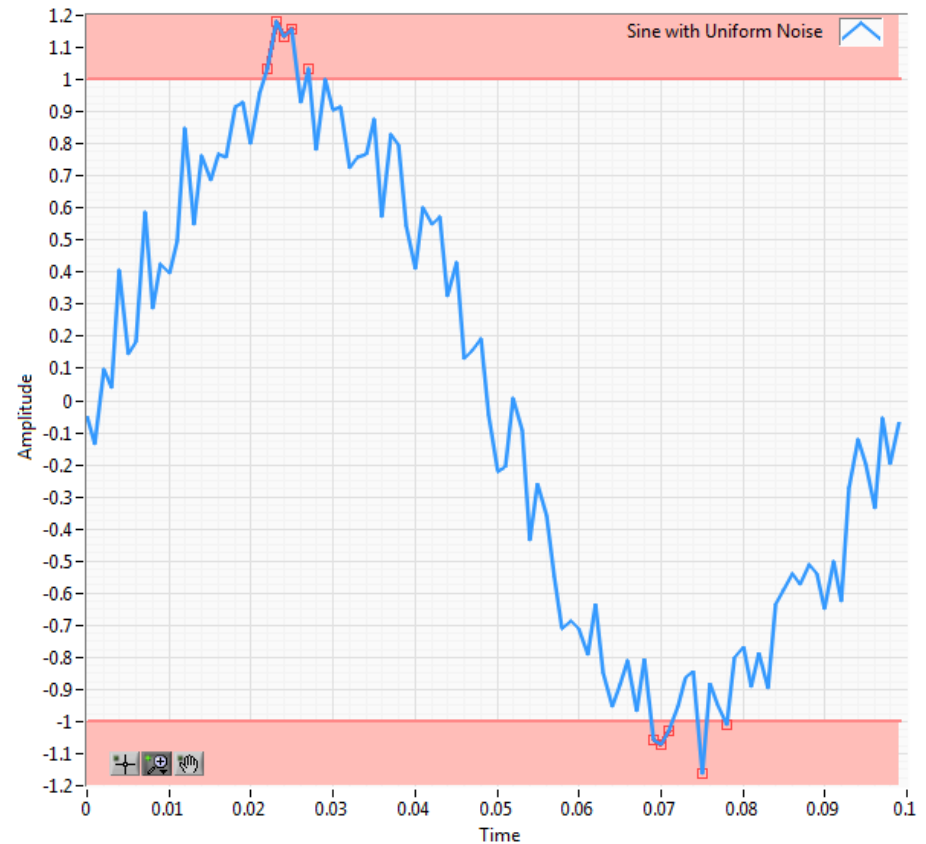
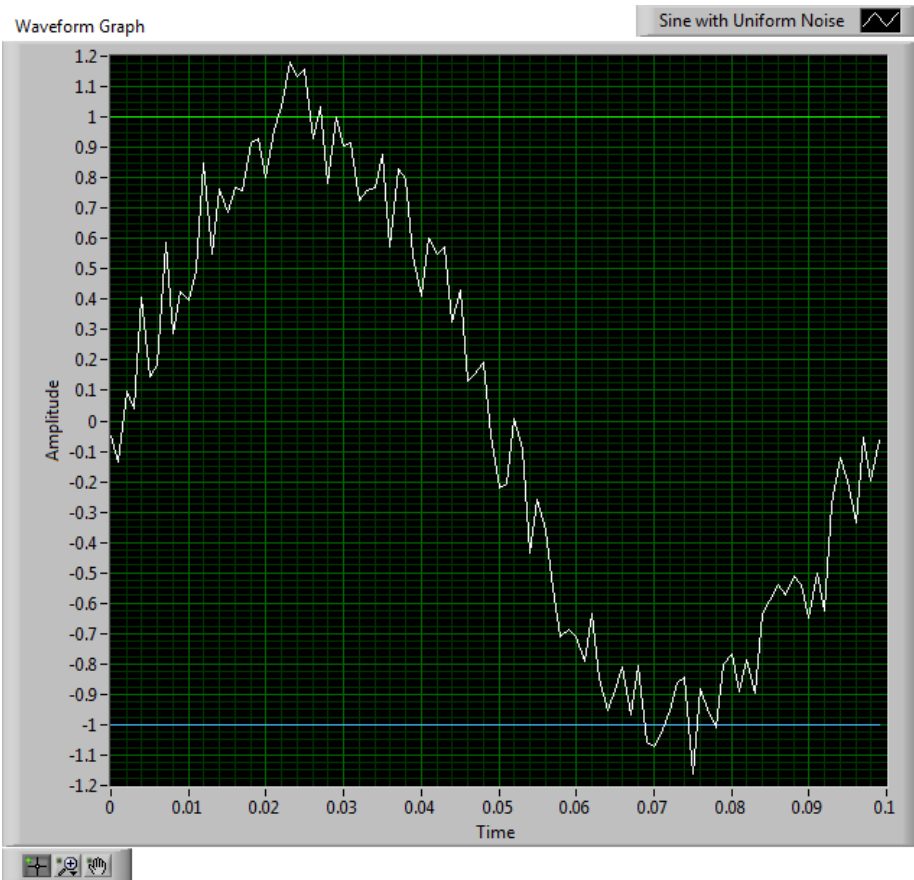
# Add Decals to Buttons



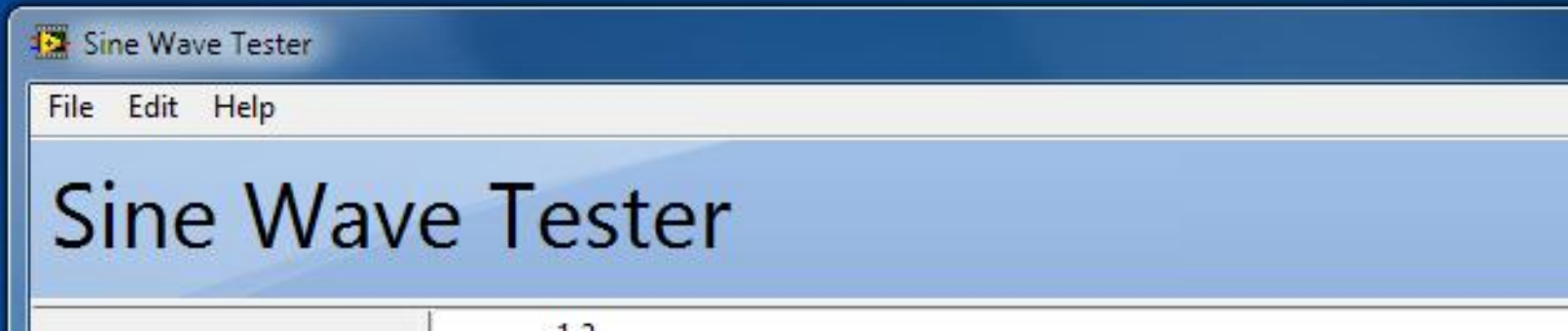
# Tooltips



# Recolor Graphs

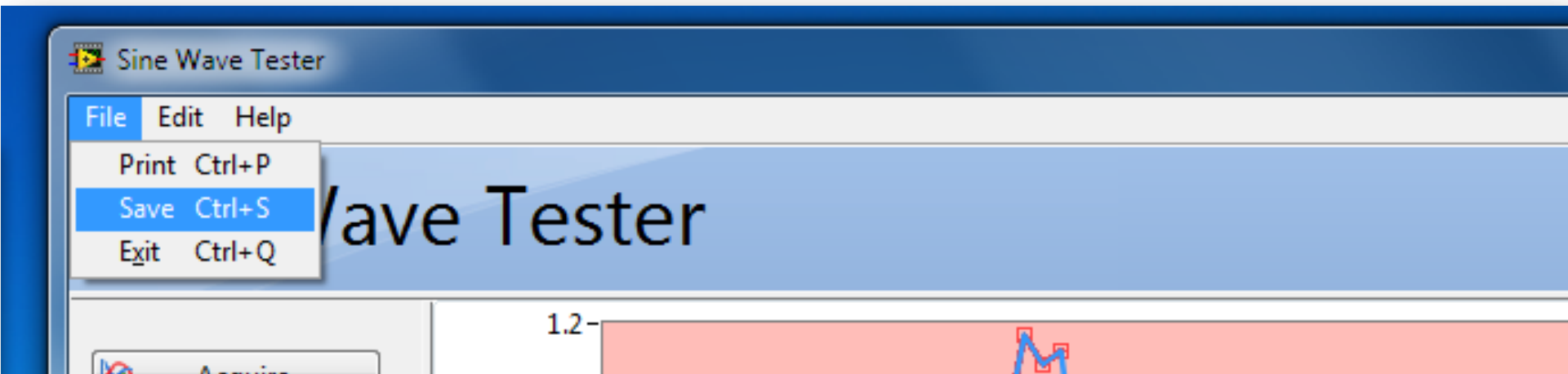


# Hide the LabVIEW Toolbar

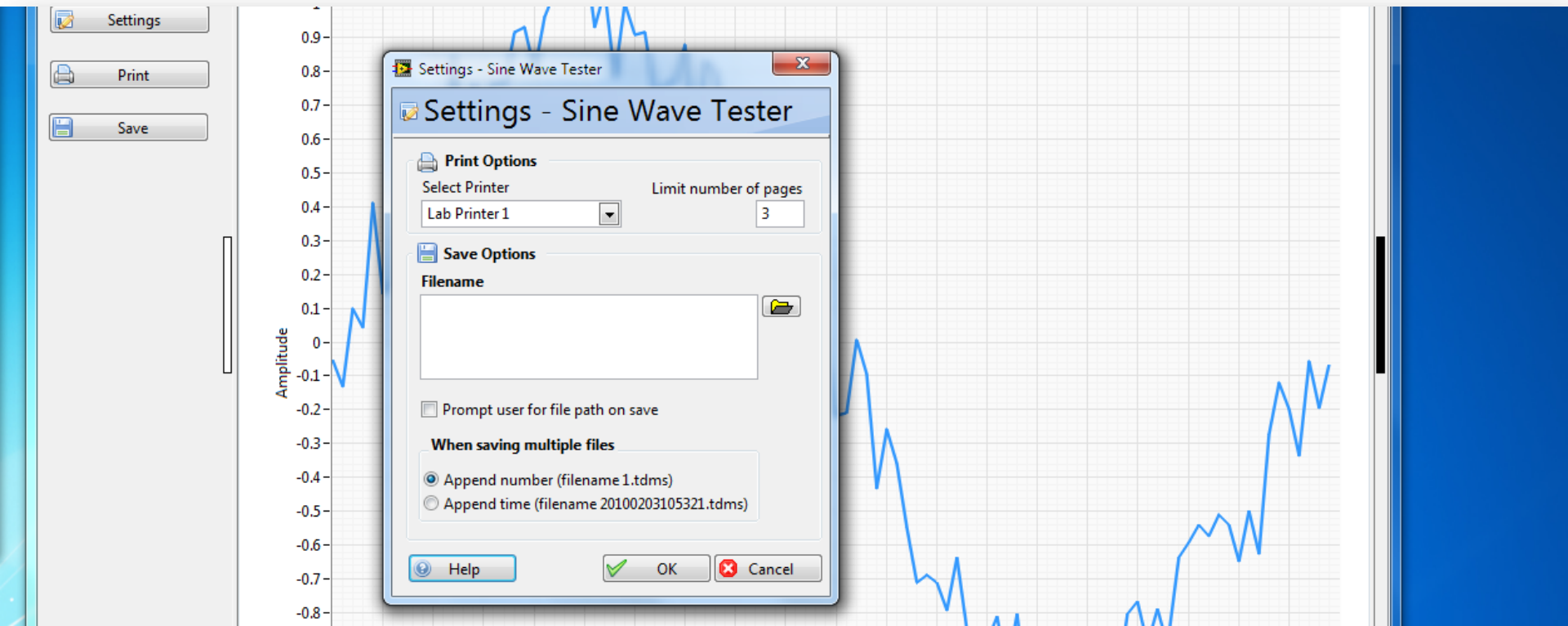




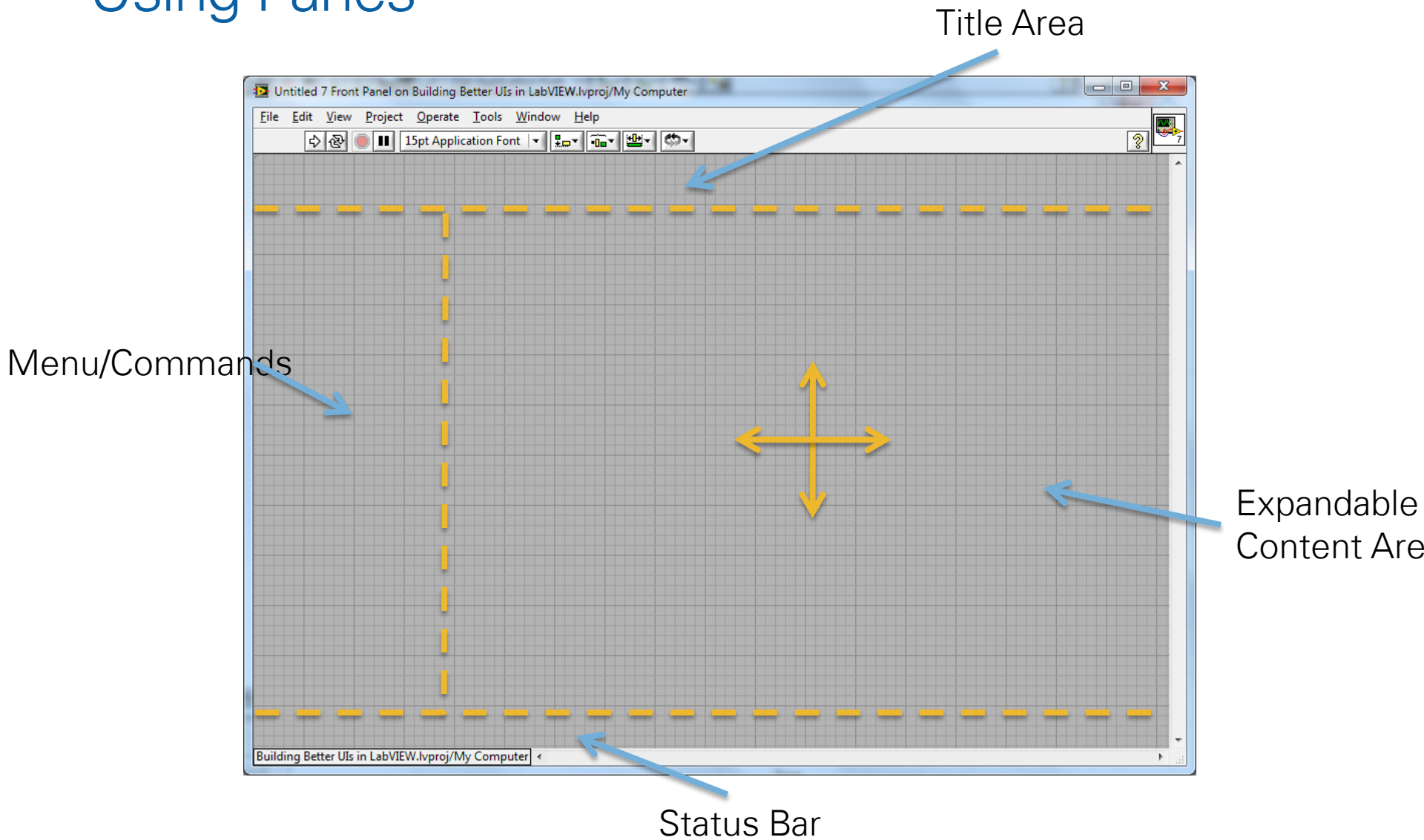
# Customizing the Run-Time Menu



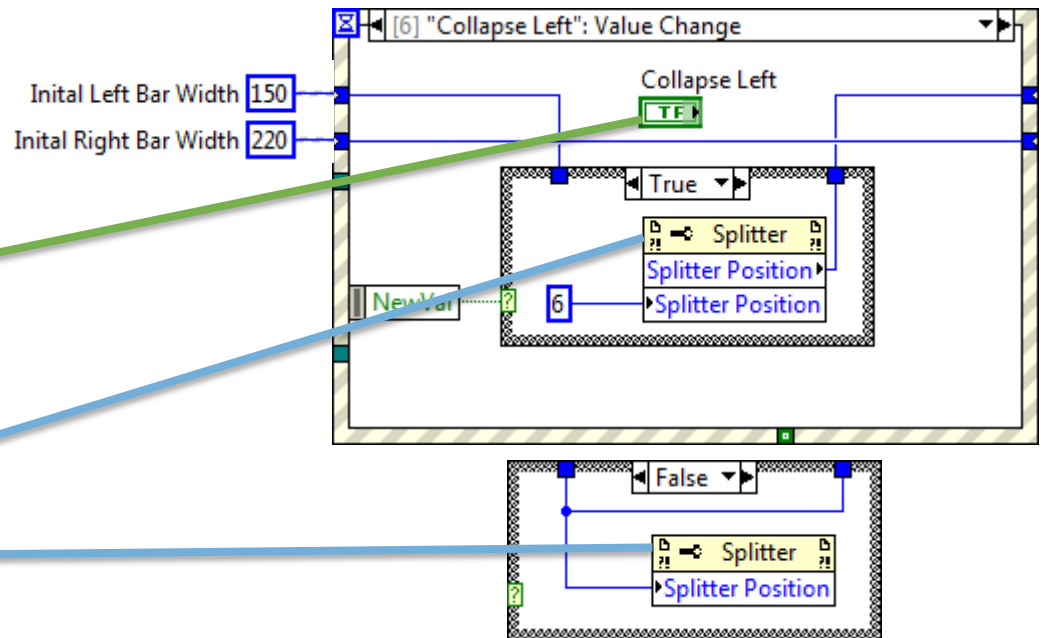
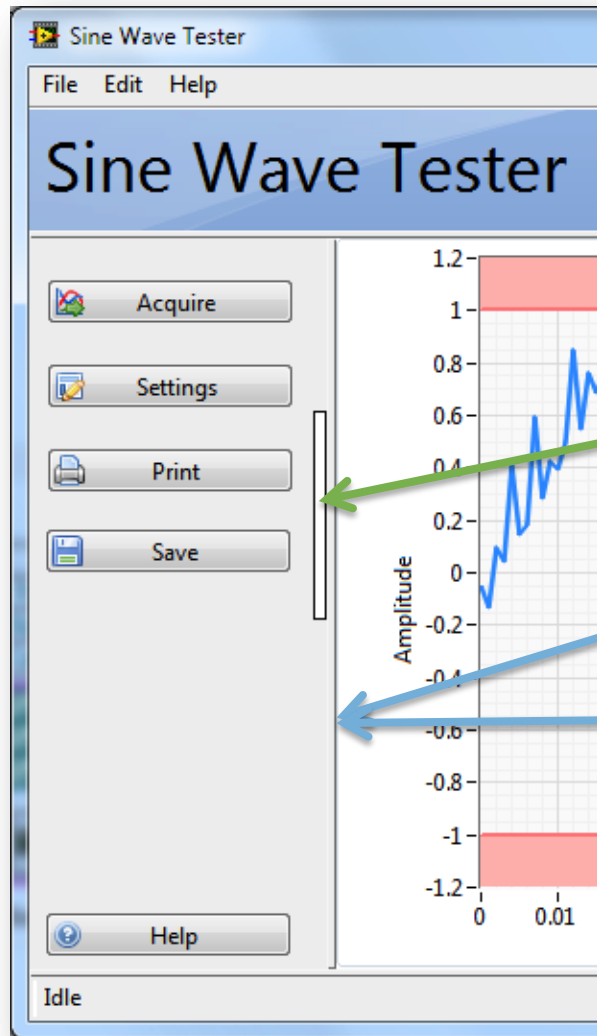
# Spawning Dialogs



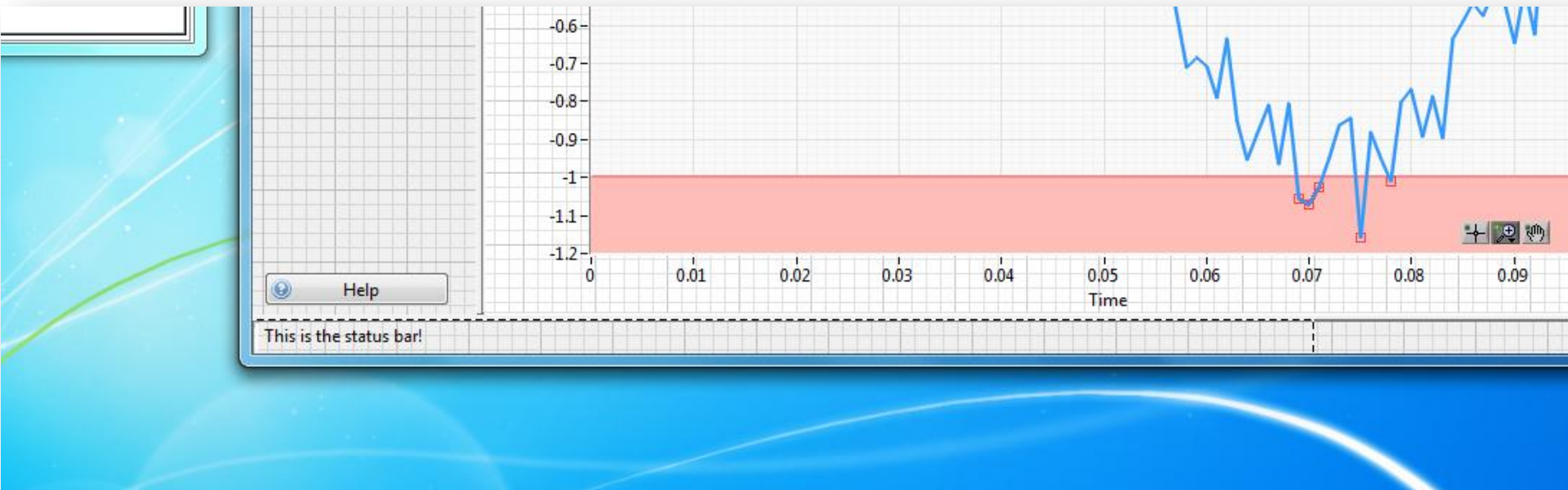
# Using Panes




# Hiding Panes



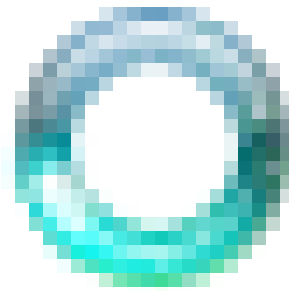
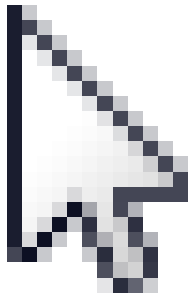
# Status Bar



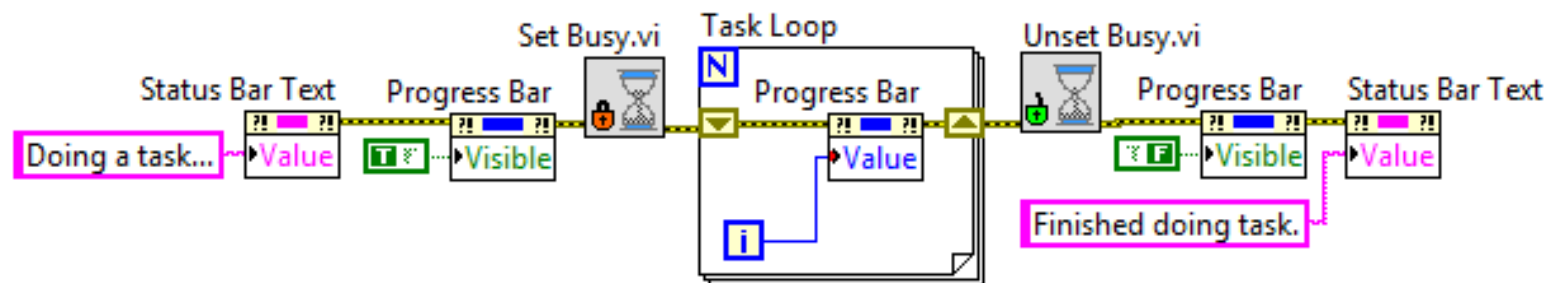
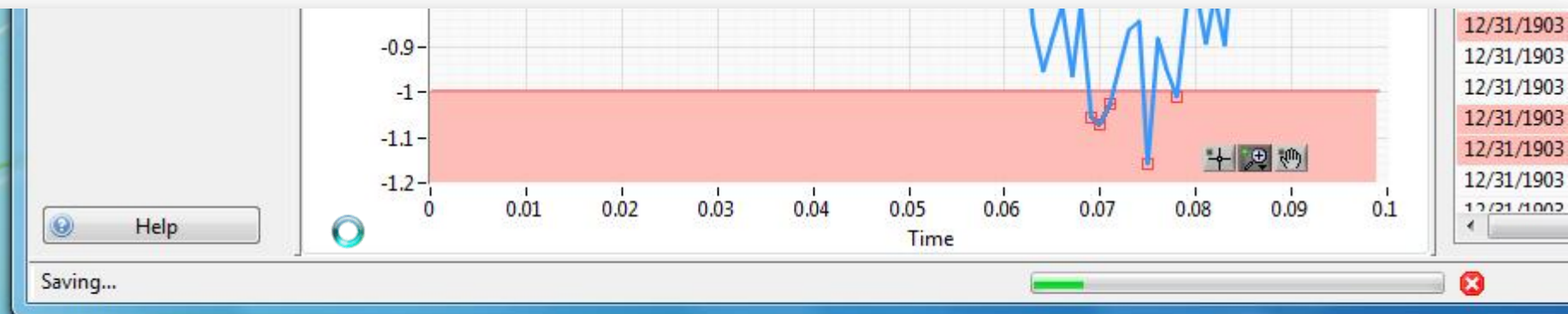
 Status Bar Text

# Busy Cursors

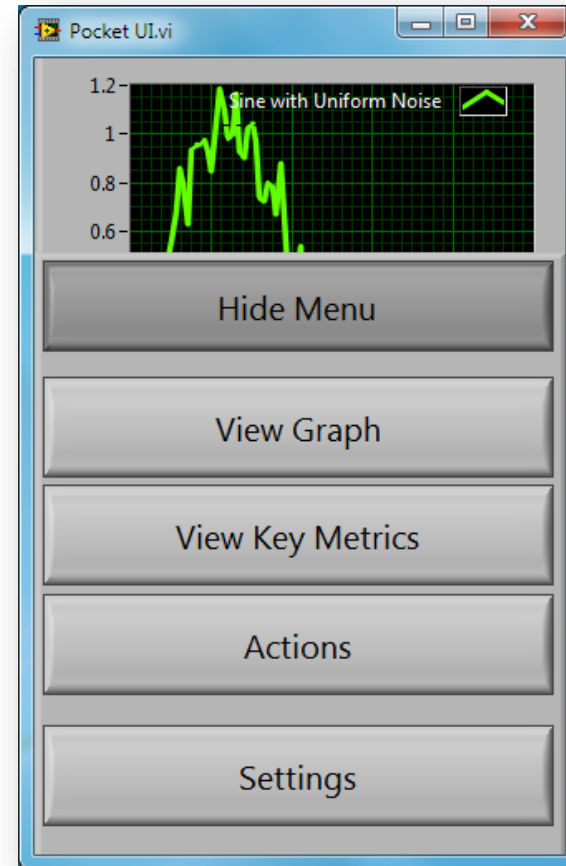
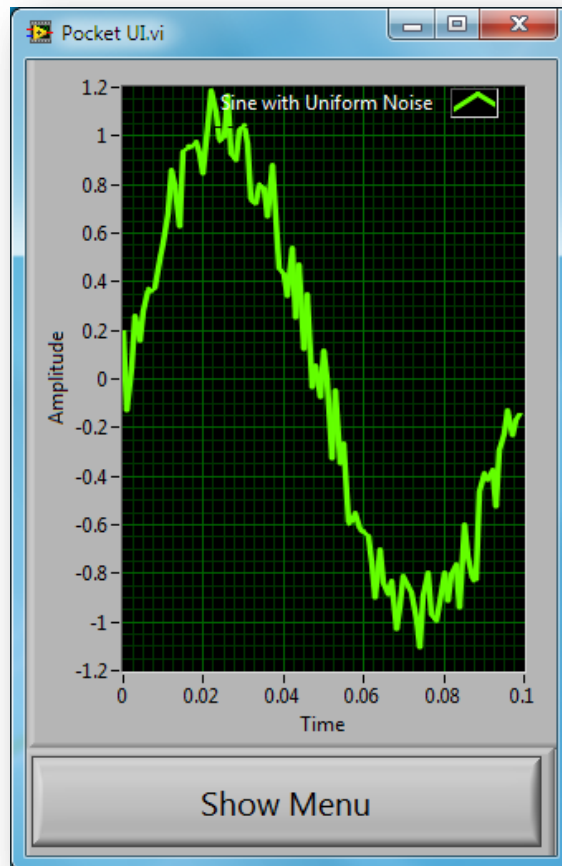


# Keeping the User Updated





# Small Touch Screen App



# Applying the Rules

## Small Touch Screen

### Do not be innovative

- Use large controls and indicators that resemble their physical equivalents
- Simple is best

### Less is more

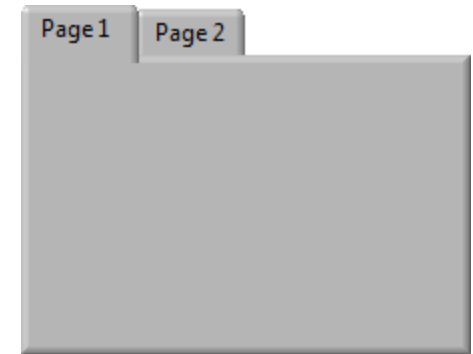
- Screen real estate is valuable; use it wisely
- Use trays, tabs, or different screens to stretch screen space

### Think about your user

- Glare may be an issue → use more contrast
- Touch screens require more spacing
- Users' fingers may obscure part of the screen

# Tab Controls

- Tab controls are a familiar way to put more information on a screen
- Because the tabs can be hidden and changed programmatically, they are also useful for some less obvious UI techniques



Tab Control



Tab Control

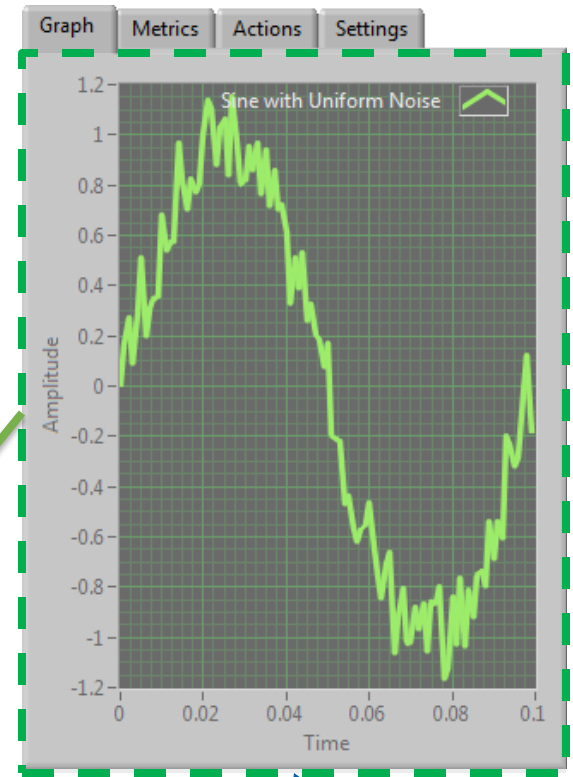
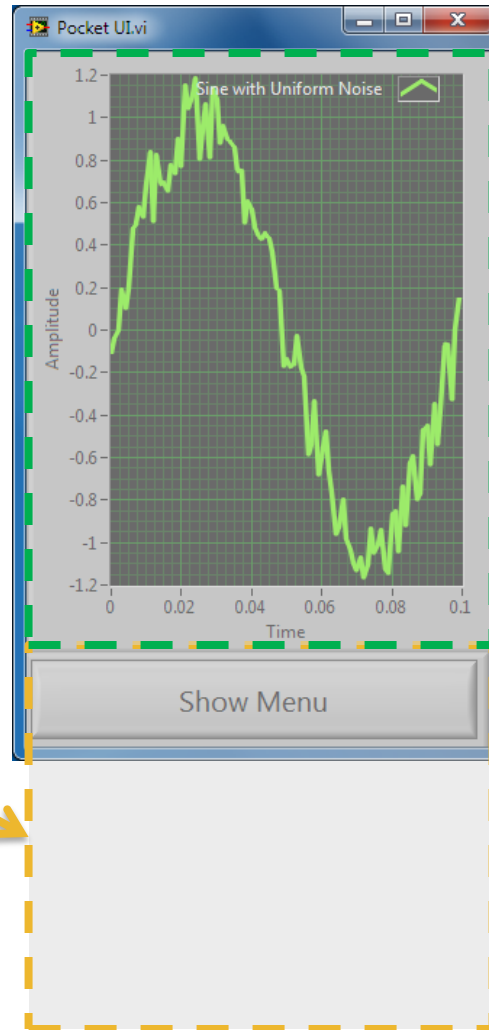
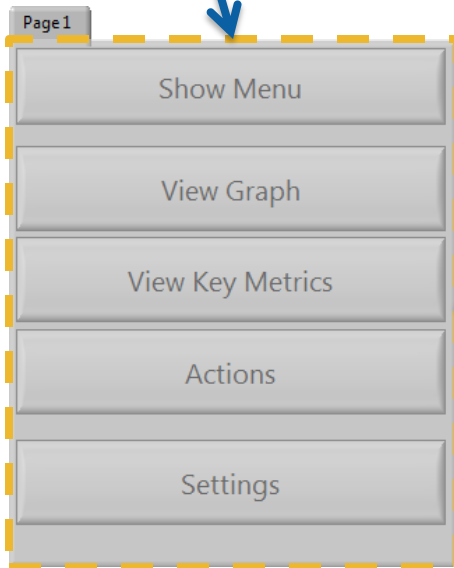


Tab Control



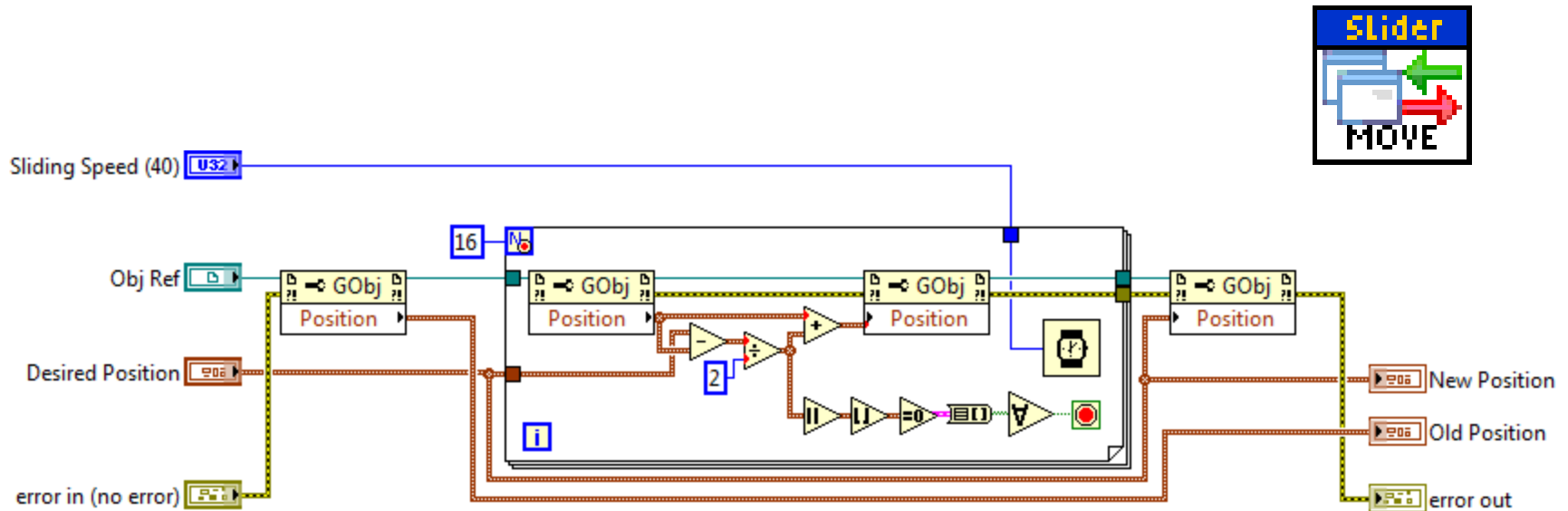
# "Hidden" Tab Controls

Sliding Tab Control



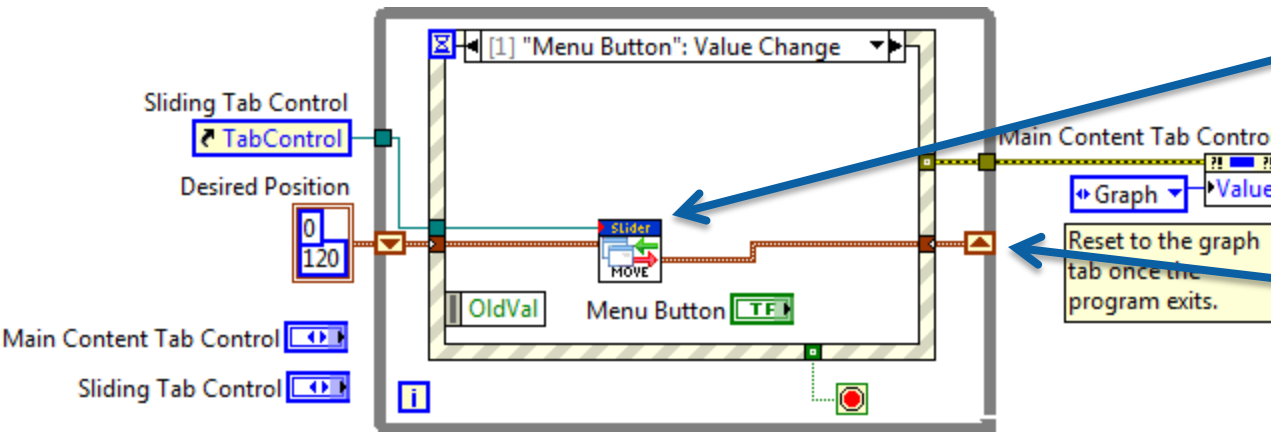
Main Content Tab Control

## Sliding a Control – Move.vi



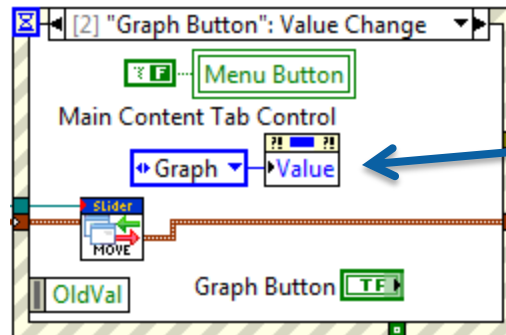
- Moves an object to the Desired Position
- Moving half the remaining distance in each loop iteration gives a natural sliding appearance

# Putting It Together



When the Menu Button is clicked, slide the invisible tab control into view

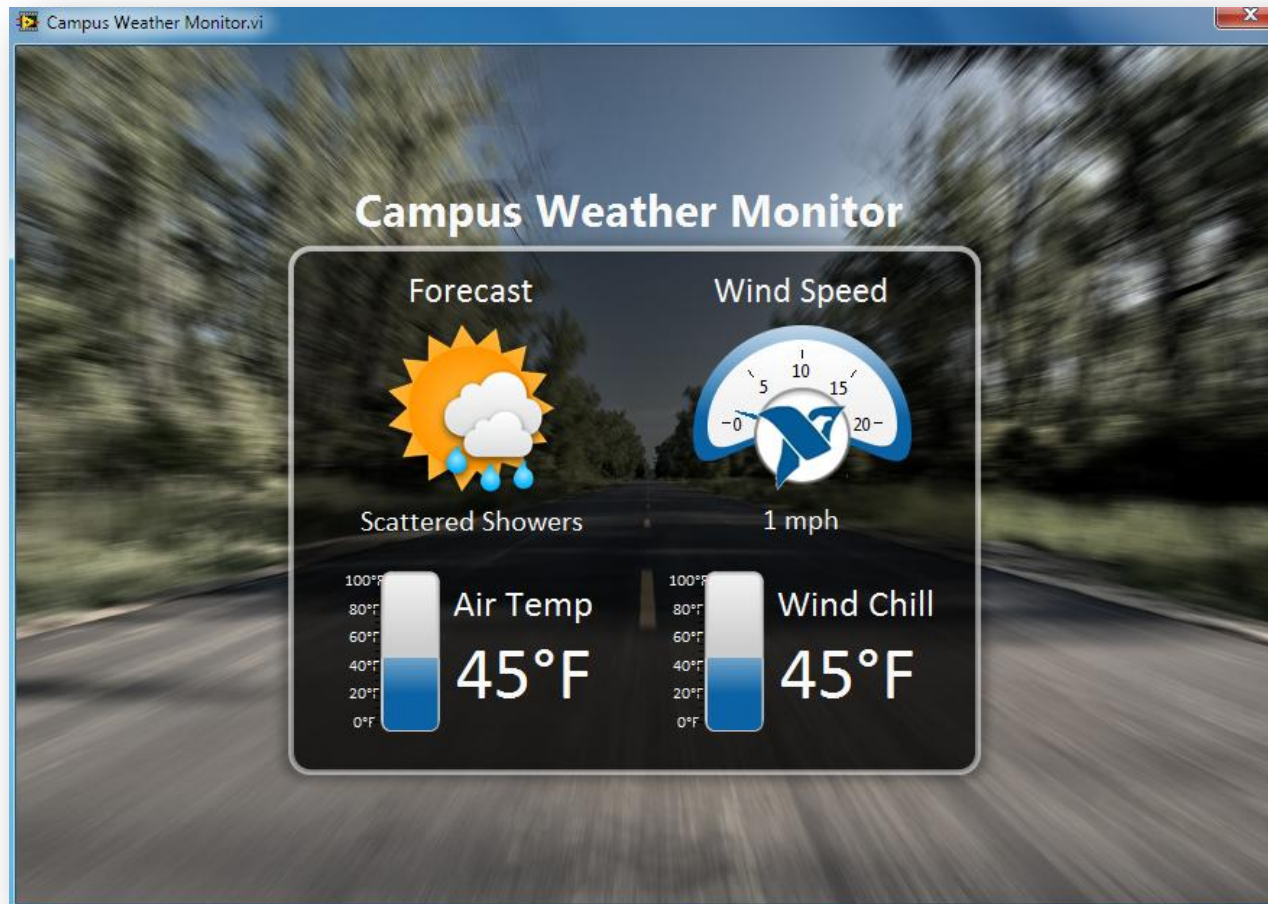
Store the old position of the menu so we can slide it back



When a new view is selected, change the Main Content Tab Control to the selected page

Again, store the old position of the menu so we can slide it back

# Informative Kiosk Display





# Applying the Rules

## Informative Console Display

### Do not be innovative

- Take inspiration from TV, Web sites or similar applications

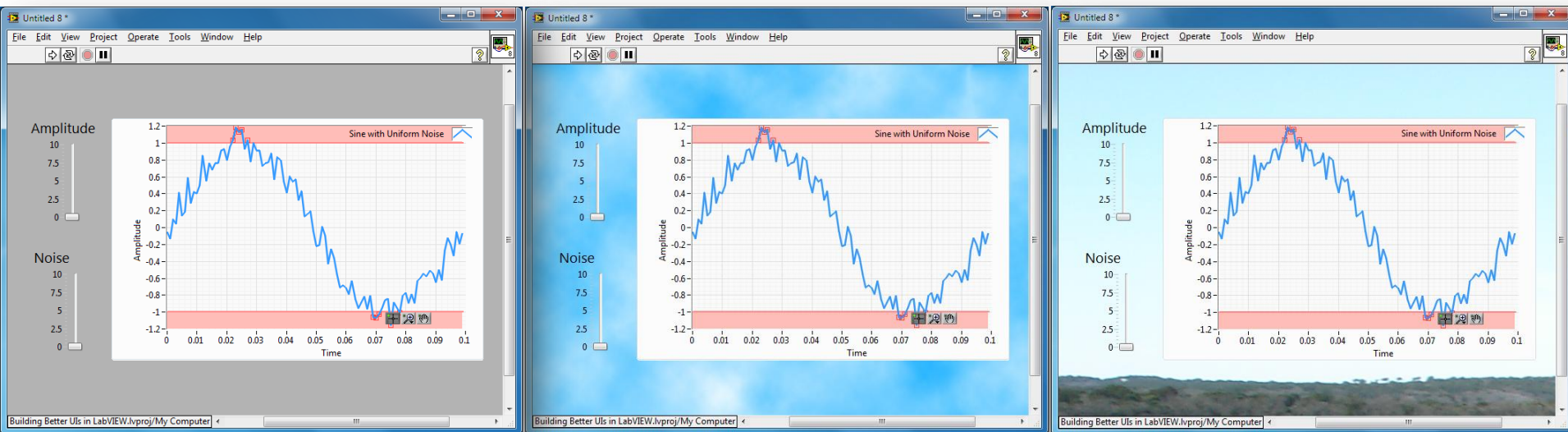
### Less is more

- Show only the important information in an instantly recognizable way

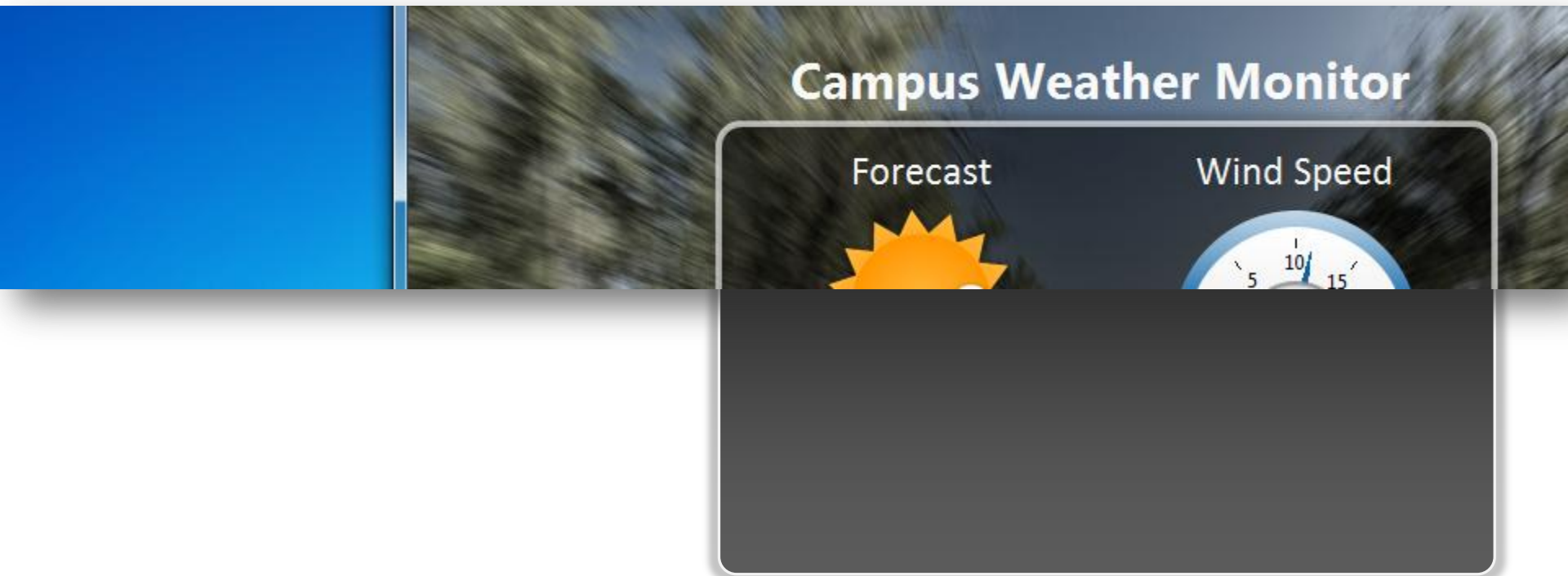
### Think about your user

- Passive audience → visual appeal is more important

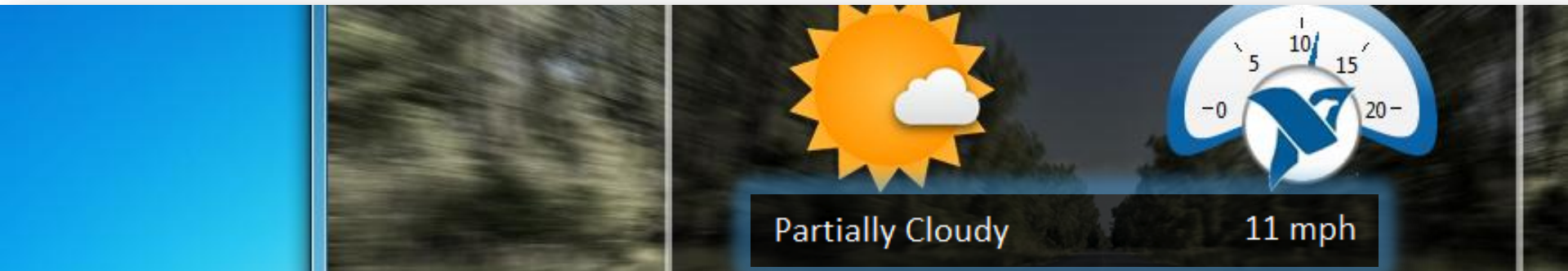
# Panel Background



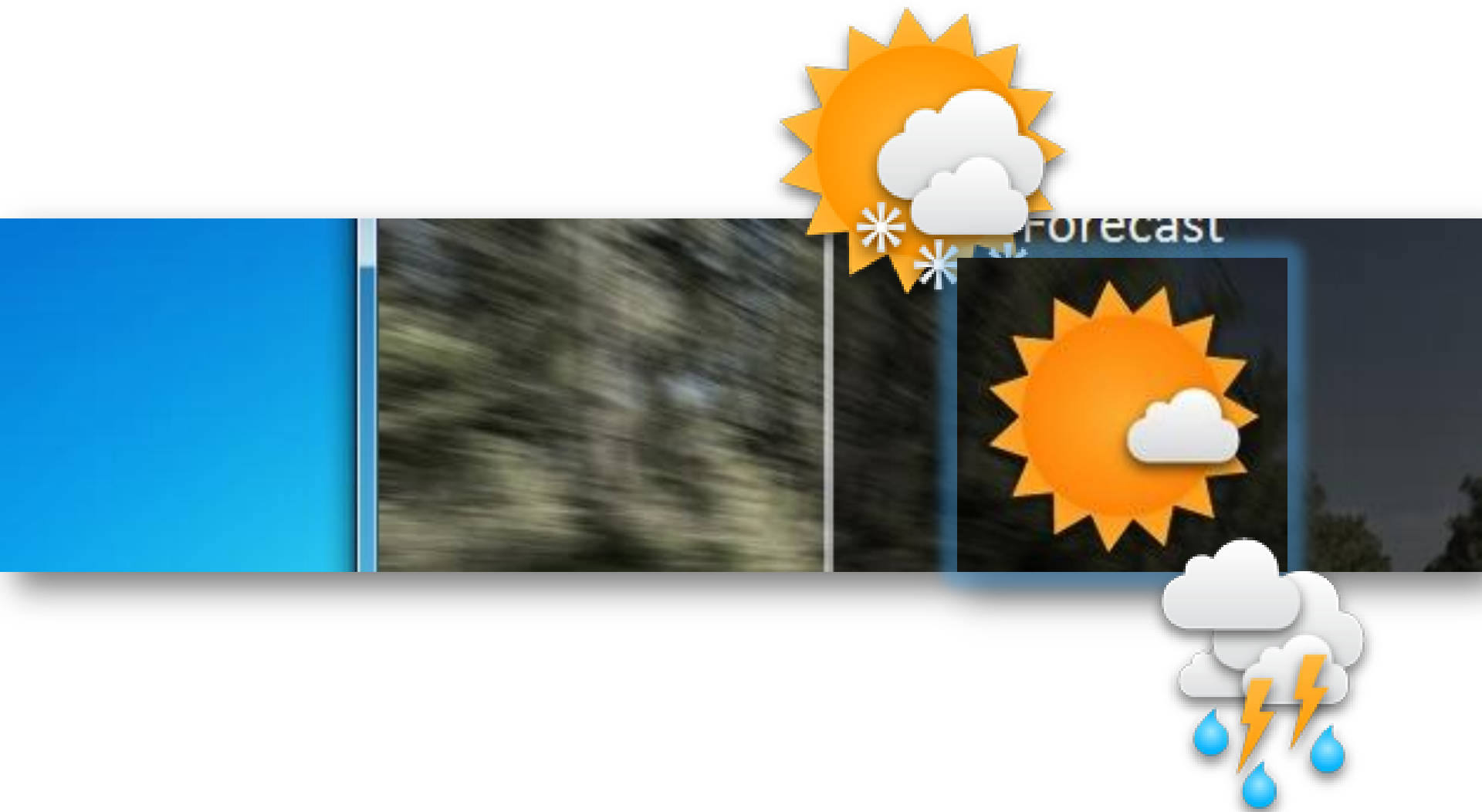
# Create Decorations in PowerPoint



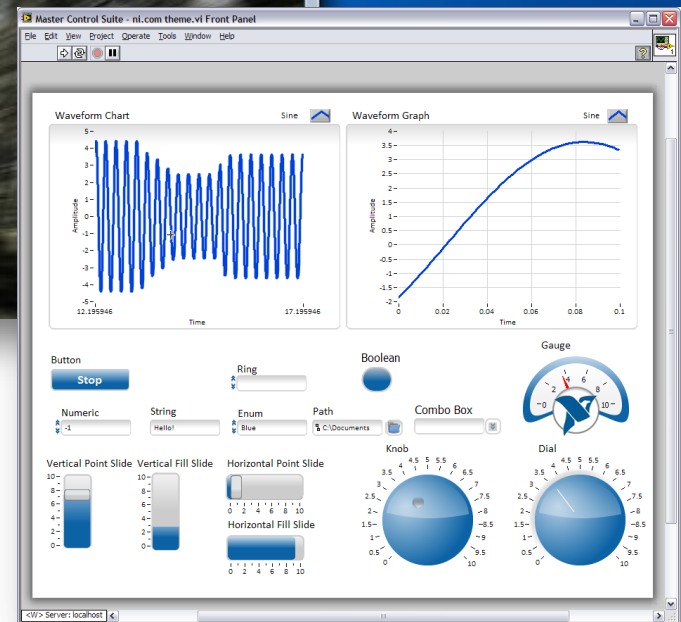
# Transparent Indicators



# Transparent PNGs in a Picture Ring



# Heavily Customized Controls



# Free Stuff – UI Interest Group



<http://decibel.ni.com/content/groups/ui>



# Key Takeaways

- The “Rules”
  1. Do not be innovative
  2. Less is more
  3. Think about your user
- Take advantage of what LabVIEW gives you
  - Transparency
  - Different controls/control customization
  - Panes/tabs
- UI interest group on the community
  - <http://decibel.ni.com/content/groups/ui>