Job and Task Analysis for:
Certified LabWindows / CVI Developer

Terminal Objective: Given a moderately complex specification or measurement need, the developer is able to create a solution using LabWindows / CVI.

Task	Objective
Examine the CVI environment	Identify the CVI Windows • Workspace window • User Interface window • Source window • Output window • Variable watch window
Navigate the CVI environment	• Menus • Toolbars • Options
Demonstrate the use of Function Panels	• Access on-line documentation • Declare variables interactively • Execute the function panel using interactive execution
Demonstrate debugging tools	• Apply source code stepping and debugging • Demonstrate the use of variables, watch and other debug windows • Set next statement
Demonstrate compiling of a program	• Demonstrate program compilation • Compare debug version versus release version
Develop User Interface for the application	Demonstrate the UI editor • Demonstrate the UI tools • Demonstrate code builder
Demonstrate User Interface components	• Panels • Controls • Custom Controls – create and use • Menus o Toolbars

Demonstrate User Interface properties	• Define Callback functions
	• Apply general settings for a panel or control
	  o Modes – use and explain
	  • Hot
	  • Normal
	  • Validate
	  • Indicator

Identify files associated with User Interface	• Identify the use of:
	  o .uir file
	  o .h file

Task	Objective

Develop Code for the application	• Define the input and output data
	  ○ Demonstrate CVI scanning functions
	  ○ Demonstrate CVI formatting functions
	  ○ Demonstrate the use of File IO functions
	  ○ Demonstrate the functions to manipulate custom controls
	  ○ Demonstrate UI library functions
	  • Get/Set attribute functions
	  • Get/Set values functions
	  • Event processing functions
	  • Loading/Unloading functions

Develop the code in the Callback functions	• For user interface components • For non-user interface components (example: DataSocket)

Define the threading model for the application	• Define multithreading terminology
	• Create threads
	  ○ Define thread pools
	  ○ Define thread safe queues and variables

Define the inter-application communication model	• Controlling software server through ActiveX automation
	• Transferring data through a DataSocket connection
	• Interfacing with LabVIEW
	• TCP communication
	• Static libraries for third party compilers
	  ○ .lib files
	  ○ .obj file
	• Dynamic link libraries
	  ○ Creating / debugging a Dll using LabWindows / CVI
	  ○ Calling functions inside a Dll
	• Calling Windows API (SDK)

Develop Instrument Driver	• Identify Instrument driver components
	  ○ Source code
	  ○ Include file
	  ○ Function panels
	  ○ Documentation
	• Create an instrument driver
	  ○ Create classes and function trees
	• Editing an instrument driver
	  ○ Debug instrument driver
	• Accessing an instrument driver

Task	Objective
Deploy and distribute the application	• Create release executable • Create distribution kit with installer • Explain the need for the run time engine • Explain backward compatibility of run time