# Definitions

**FPGA Developer** – Individual responsible for the FPGA code for the device. This is mostly likely the card manufacturer.

**Custom Device Developer** – Individual responsible for the code for the VeriStand Custom Device. This is most likely the card manufacturer.

**User** – Individual responsible for executing the test. This is sometimes the Integrator.

**Integrator** – Individual responsible for the combining the FPGA VIs provided by the FPGA Developers and instantiating the Custom Devices in the System Explorer. This is sometimes the User.

**Device Prefix –** The Company ID appended with a period appended with the device name.

e.g. “NI.myDevice”

# Developing a Resource Manager-Compliant FPGA VI

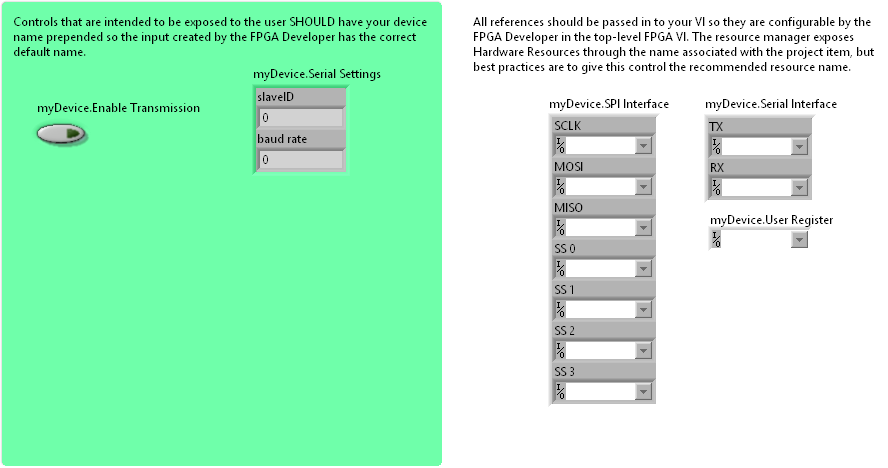
## Overview

To provide a simple experience for the Integrator, the FPGA Developer will create two VIs: The FPGA VI, and a Placer VI.

The FPGA VI is the core logic of your IP and must support the ability to be duplicated or mixed with FPGA IP from other vendors without modification. It defines the functionality but does not specify any resources that are target or implementation dependent, such as IO Channels, Registers, and DMA FIFOs. Instead, it provides controls for the Integrator to specify what resources should be used.

The Placer VI is a VI that contains your FPGA VI as well as any FPGA resource constants, parameter controls and indicators, timed loops, etc. that your FPGA VI needs to function. This VI will go on the palette in LabVIEW, and all the contents will be placed on the Block Diagram when the VI is selected. This simplifies building the top-level FPGA VI by providing one click to place the core IP and any external resources necessary. The Control names used by the Integrator in the top-level VI are what is used in the System Definition, so the names given to Controls in the Placer VI should serve as a recommendation to the Integrator.

## Front Panel Example



## 

## Documentation Templates

### VI Documentation Template

Use the following template for the description of your FPGA VI.

{Description}

<b>Associated Custom Device</b>

{Custom Device Name}

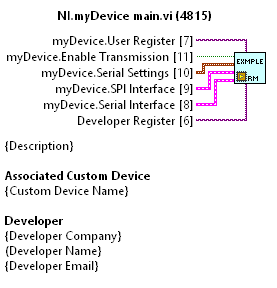
<b>Developer</b>

{Developer Company}

(Support Name}

{Support Email}

### VI Documentation Appearance Example



### Control Documentation Template

Use the following template for the Description of your controls.

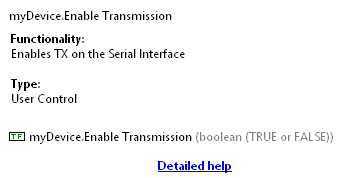
<b>Type:</b>

User Control/User Resource/Developer Resource

<b>Functionality:</b>

{Description}

### Control Documentation Appearance Example



## [FPGA 1] FPGA VI Front Panel Requirements

### [FPGA 1.1] Code Modularity

To ensure that the **FPGA Developer’s** code is easily integrated by the **Integrator**:

[FPGA 1.1.1] The code required for a given FPGA task **SHALL** be contained in one VI.

[FPGA 1.1.2] FPGA Controls accessible by the **User** **SHALL** provide an input on the VI’s Connector Pane.

[FPGA 1.1.3] FPGA Resources intended to be chosen by the **Integrator** **SHALL** provide an input on the VI’s Connector Pane.

[FPGA 1.1.4] Controls **SHOULD** have the appropriate requirement level set (Required, Recommended, Optional)

[FPGA 1.1.5] FPGA Controls accessible by the **User SHOULD** be grouped together.

### [FPGA 1.2] Grouping

[FPGA 1.2.1] To provide buses or logical groups of channels, related or similar resources **MAY** be grouped together in a cluster.

* NOTE: Items grouped together in a cluster will appear as one entity in the Resource Manager UI, and sub-items will not be individually selectable.

[FPGA 1.2.2] Resources **MAY NOT** be grouped together in an array.

### [FPGA 1.3] Naming Conventions

[FPGA 1.3.1] The Labels of FPGA Controls accessible by the **User** **SHOULD** be prefixed with the **Device Prefix**.

[FPGA 1.3.2] FPGA Resource controls intended to be chosen by the **Integrator** **SHOULD NOT** contain the **Device Prefix**.

[FPGA 1.3.3] FPGA Resource controls intended to be chosen by the **User** **SHOULD** be prefixed by the **Device Prefix**.

* NOTE: This name is not the name used by the resource manager, but is provided as a suggestion to the **Integrator**.

[FPGA 1.3.4] If a control is in a cluster, the **Device Prefix** **MAY** be omitted.

## [FPGA 2] FPGA VI Block Diagram Requirements

### [FPGA 2.1] Resource Usage

[FPGA 2.1.1] Any FPGA Resources chosen by the user SHALL NOT be implicitly linked.

[FPGA 2.1.2] The FPGA VI **SHALL NOT** contain a top-level loop

[FPGA 2.1.3] The FPGA VI **SHALL NOT** use any shared DMA FIFOs

## [FPGA 3] FPGA VI Documentation Requirements

### [FPGA 3.1] VI Name

[FPGA 3.1.1] The VI **SHALL** be named “{**Device Prefix**} main.vi”, e.g. “NI.myDevice main.vi”.

### [FPGA 3.2] VI Description

[FPGA 3.2.1] The VI Description **SHALL** contain a description of the functionality

[FPGA 3.2.2] The VI Description **SHALL** contain the FPGA Developer’s company name

[FPGA 3.2.3] The VI Description **MAY** contain the contact information for support

[FPGA 3.2.4] The VI Description **SHALL** contain the associated Custom Device name

[FPGA 3.2.5] The VI Description **SHOULD** use the template provided in “Documentation Templates”

### [FPGA 3.3} Control Descriptions

[FPGA 3.3.1] Controls SHALL include a description of the Control’s functionality

[FPGA 3.3.2] Controls SHALL include a description of the Control’s Resource Manager type (User Control, User Resource, Developer Resource).

[FPGA 3.3.3] The Control Descriptions **SHOULD** use the template provided in “Documentation Templates”

# Distributing Resource Manager-Compliant FPGA Code

## Overview

To make distributing and maintaining code easier, the FPGA VI should be placed in another VI, which we’ll call the Placer VI. This makes it possible for the **Integrator** to select a single item from the palette and get the engine VI and all the controls and constants associated with it, as well as any Single-cycle timed loops, etc. that the FPGA VI may rely on for functionality. This will dramatically reduce the amount of effort to integrate several FPGA VIs together, and should have the effect of reducing the support burden on the FPGA developer.

## Example

An example of this is provided [here](Builds/).

## [FPGA 4] Placer VI Front Panel Requirements

### [FPGA 4.1] User Interface

[FPGA 4.1.1] Items intended be exposed to **User** **SHALL** be placed as Controls.

[FPGA 4.1.2] Controls intended for the **User** **SHALL** be grouped together.

## [FPGA 4.2] Placer VI Integrator Interface

[FPGA 4.2.1] Items intended to be selected by the **Integrator SHALL** be placed as Constants.

## [FPGA 5] Distributing The FPGA VI

To ensure a uniform feel, all the VI packages should follow these guidelines. An example package is provided [here](FPGA%20Resource%20Manager%20Example%20Engine.vipb).

### [FPGA 5.1] VI Package Manager

[FPGA 5.1.1] The FPGA VI should be distributed using VI Package Manager 2017 or later.

[FPGA 5.1.2] The FPGA VI SHALL NOT directly appear on the palette

[FPGA 5.1.3] The Placer VI SHALL appear on the palette

[FPGA 5.1.4] The Placer VI SHALL install to ***Functions (FPGA)>>VeriStand Custom Device Engines>>{Company Name}***

[FPGA 5.1.5] The Placer VI SHALL be set to “Place VI Contents”

# 

# Custom Device Developer Requirements

## [CD 1] System Definition

### [CD 1.1] Adding the FPGA Resource Manager Control to the Custom Device

### [CD 1.2] Setting Filters for the Custom Device

## [CD 2] Engine

### [CD 2.1] Getting the FPGA Reference

### [CD 2.2] Using the FPGA Reference

# Integrator Requirements

## [UI 1] Making a new FPGA VI

To make a new FPGA VI for deployment, an integrator will be required to combine FPGA VIs from FPGA Developers in to various loops and assign hw resources to them. They should avoid changing the naming scheme of controls on the FP, as these are what the resource manager uses for filtering. If two instances of the same FPGA VI are placed, the Integrator should remove the auto-generated trailing number and edit each instance to include an instance number at the resource, for example:  
  
NI.myDevice.Serial Interface 0.TX

NI.myDevice. Serial Interface 1.TX

rather than NI.myDevice.Serial Interface.TX and NI.myDevice.Serial Interface.TX 1

## [UI 2] Using the FPGA Resource Custom Device

Each RIO device will have a CD associated with it. All configuration at the device level will be handled here.

## [UI 3] Using the Resource Manager in a Custom Device

There will be a common selection method available across Custom Devices that use the Resource Manager that will provide filtering, and only show controls and resources that are compatible with the Custom Device in question and not in use.

In the engine, there will be two primary VIs:  
  
Get Reference: Gets a reference to the RIO Device that is assigned to this CD instance.

Report Done: Tells the FPGA CD that it is okay to close the reference. The FPGA CD won’t close the reference until all CDs that are using the reference are done.