# Veristand FPGA Configuration Tool (VCE) – Getting Started

## Contents



This document explains how to use the Veristand FPGA Configuration Tool (VCE) to create/modify XML files (.fpgaconfig) required by Veristand to load FPGA devices into the system definition file.

## Prerequisites

Before reading this document you must have read the following tutorial from the Veristand Help:

Creating a Custom FPGA Bitfile

It provides an overview about how to use FPGA targets (e.g. cRIO, R-series) inside NI Veristand.

## Example: Configure a CompactRIO 908x to work with NI Veristand

As a simple example, we use NI CompactRIO 9082 (link) that has two I/O modules connected:

- NI 9215 Analog Input Module
- NI 9263 Analog Output Moudle

In order to interface these modules with the NI Veristand Engine, there are two options:

1. Using the Scan Interface through https://decibel.ni.com/content/docs/DOC-15510
2. Using the FPGA Interface

The second option gives you more flexibility and customization of the FPGA code: however, it requires to program an FPGA VI and interface it with the NI Veristand Engine by means of some constraints. The best

way to develop an FPGA VI that follows such constraints is to start from the template code provided by this tool:

NI VeriStand FPGA-Based I/O Interface Tools

To complete the example, close LabVIEW and install this component on your computer.

## Create the *NI Veristand FPGA Project* Template

Once you installed the NI VeriStand FPGA-Based I/O Interface Tools, launch the LabVIEW Getting Started Window and select ***Create Project...*** and then choose ***NI Veristand FPGA Project***:

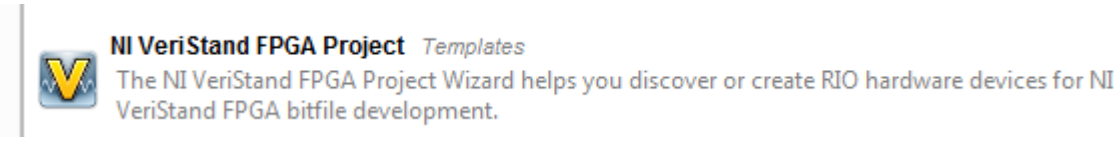


**Figure 1: Template Project Option inside LabVIEW Project Creation Wizard**

Sekect the first option, *CompactRIO Reconfigurable Embedded System*

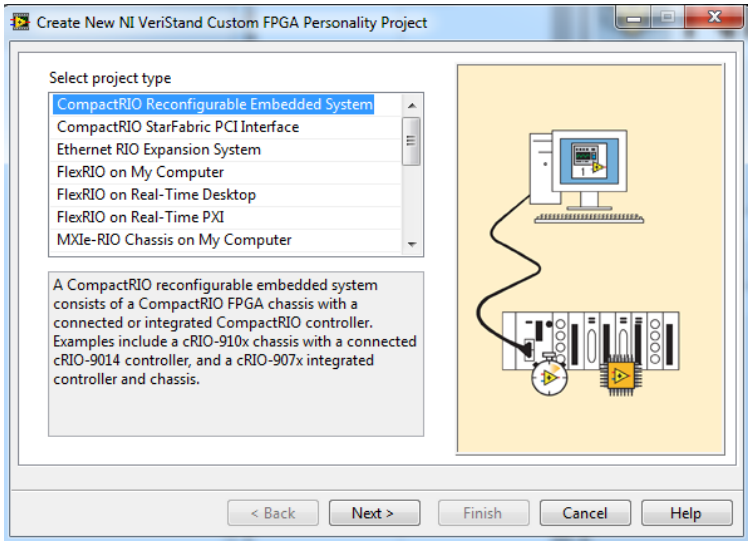


**Figure 2: Selection Prompt for the desired FPGA Hardware Type**

Then locate your existing CompactRIO or create a new system; for the example described in this document we use a **NI-9082**.

Once the configuration steps are completed, we should have a LabVIEW project similar to the one illustrated below:

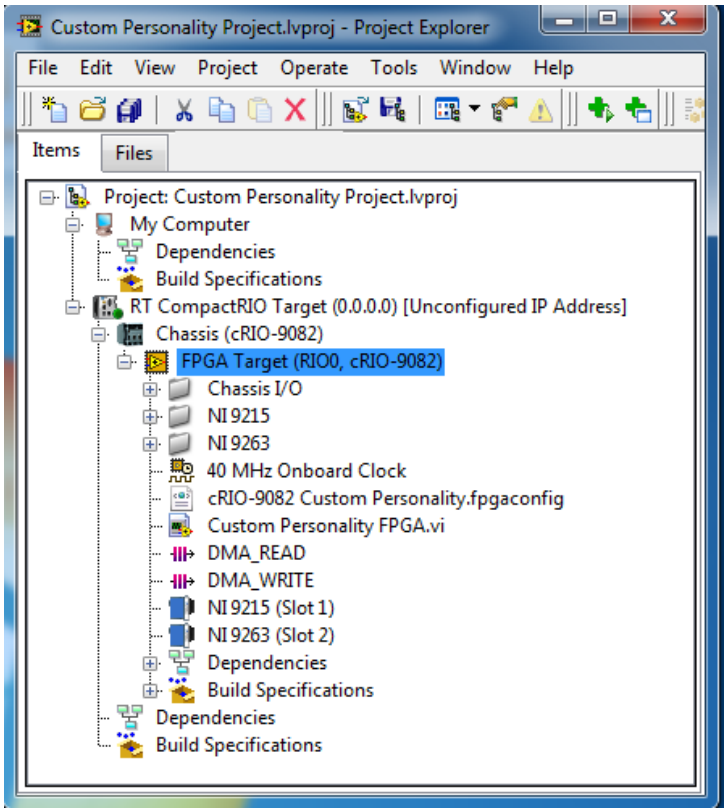


**Figure 3: Snippet of the LabVIEW Project generated by the Veristand FPGA Project Template Wizard. C-series module list can be different depending on the hardware you are using. In this case we're using an analog input (9215) and an analog output (9263) module.**

## Modifying the FPGA Code

Open the *<Your Project Name> FPGA.vi* and remove the code that is not necessary for the hardware we have in our example. The final aspect of the Block diagram should be similar to the one shown in fig. 4

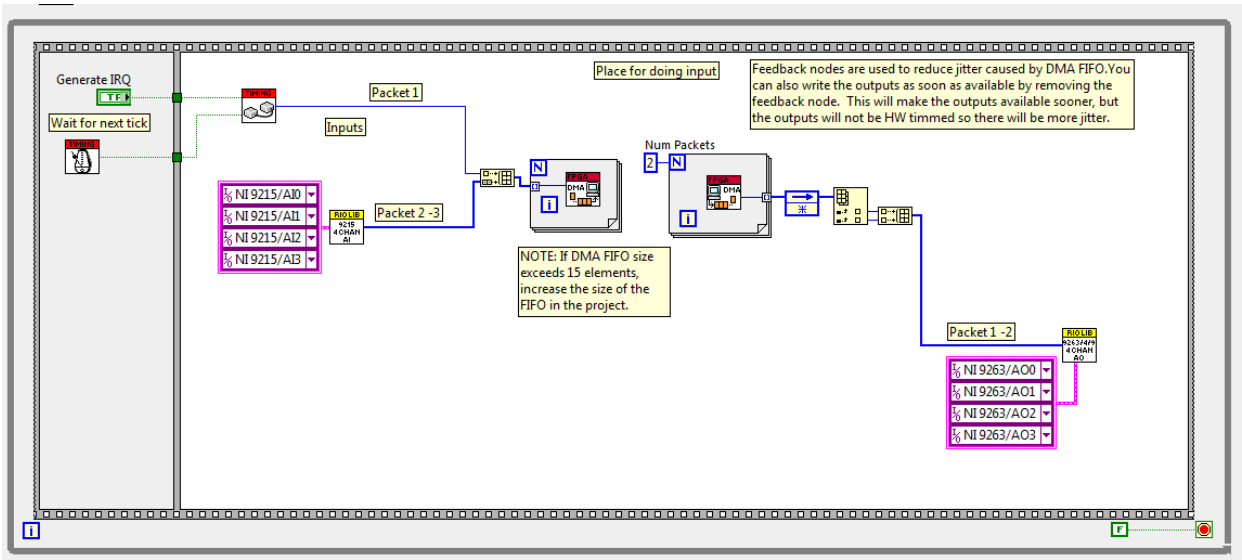


**Figure 4: Modified version of the FPGA VI: in the example we need only to access 4 AI channels from a 9215 and 4 AO from the 9263.**

# Working with FPGA Configuration files (.fpgaconfig)

FPGA Configuration file for NI Veristand are XML-based files with *.fpgaconfig* extension that are used to import an FPGA Device (e.g. CompactRIO chassis, PXI R-Series) into the System Explorer. Such files contain the following information:

1. Specify the content of the DMA FIFOs used to exchange data between the FPGA device and the Veristand Engine
2. Determine how the FPGA target appears in the System Explorer

Refer to Creating a Custom FPGA Configuration File for more information about the structure and the contents of .fpgaconfig files. The Veristand FPGA Configuration Editor provides a GUI for easily creating and modifying such files without being forced to edit them manually using, for example, a text editor.

## Manage Categories

Categories are used to create the section/channel hierarchy that you see inside the System Definition Explorer. The fpgaconfig file we loaded in the example contains the following categories:

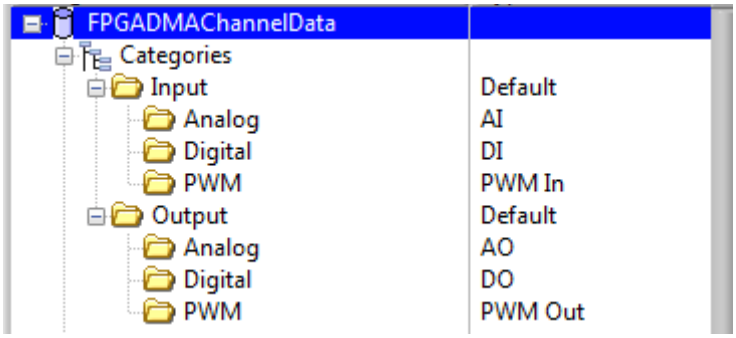


Figure 3: Category hierarchy represented inside the vCE.

The first column is the name of the category while the second column represents the type. For the list supported category types, refer to Creating a Custom FPGA Bitfile topic on the Veristand Online help.

Categories will appear as sections and sub-sections inside the System Explorer:

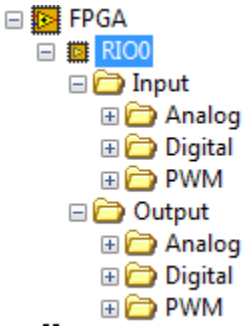


Figure 4: The same category hierarchy shown in figure 10 as it appears inside the System Explorer. Channels like Analog and PWM out are also represented.

## Add/Remove/Duplicate Categories

By right clicking on one *Categories* item or one of the existing *Category* item, you can do one of the operations shown below:

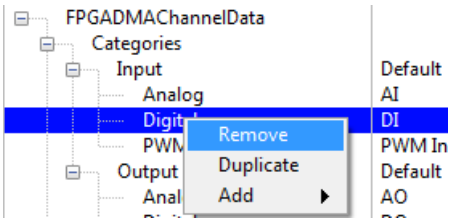


Figure 5: Removing a Category item from the vCE hierarchy

- ***Add*** a new child category
- ***Remove*** the selected category and all its children
- ***Duplicate*** the selected category and all its children

## Manage Packets and Channels

Packets and Channels are the items that tell Veristand about the data provided by FPGA and vice versa.

- **Packets** are the elements of the DMA channels. They're U64 data on which the FPGA VI extracts or inserts values to be exchanged with the Veristand Engine
- **Channels** are the basic data values and correspond to the channels that you can create inside the System Explorer

The following picture shows the representation of packets and channels in the FPGA and in the vCE window:

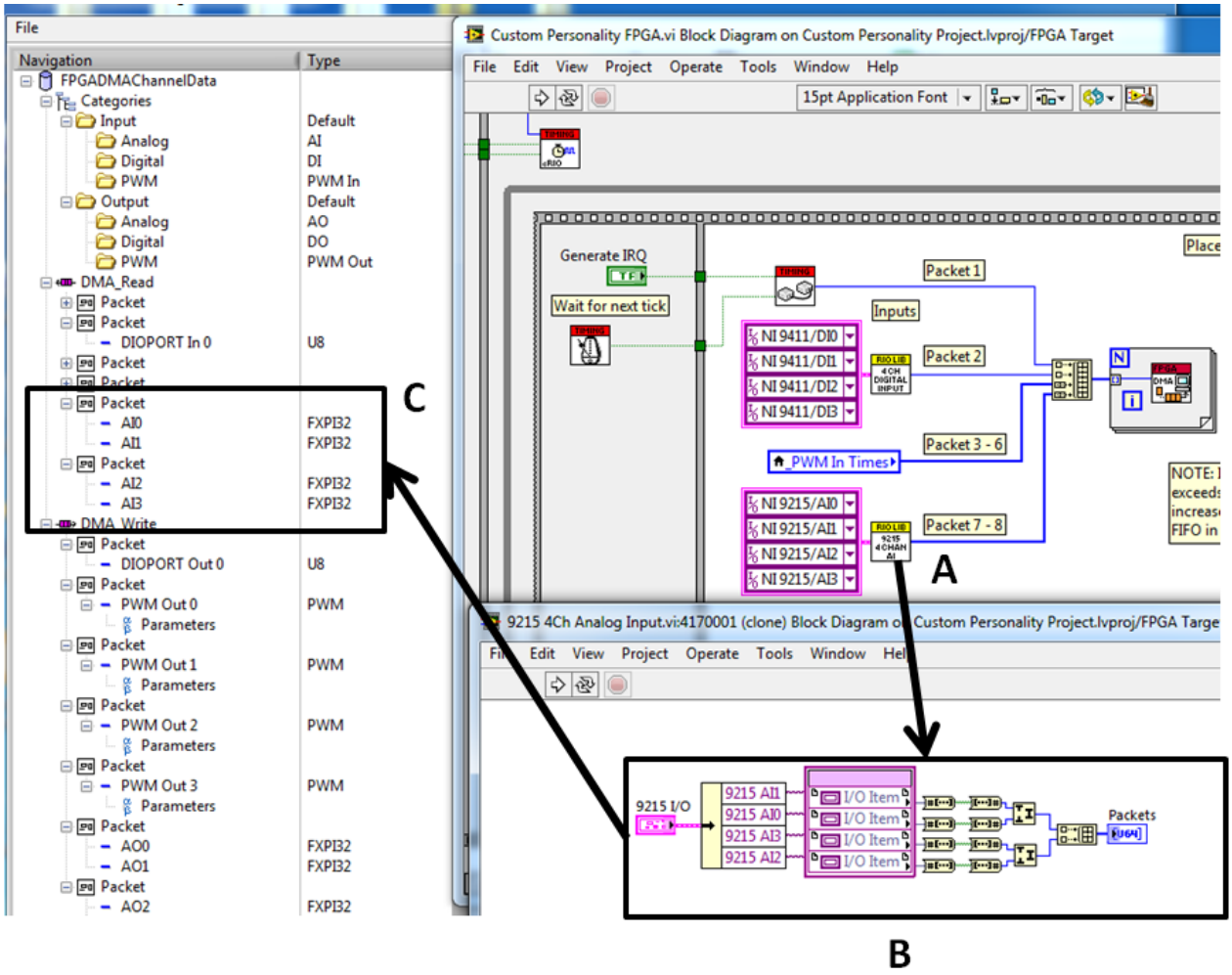


**Figure 6: Representation of packets and channels. A: The FPGA VI acquires 4 AI channels from the 9215; each channel is a 32-bit wide fixed point. B: Channels are packed into two U64 packets. C: packets are shown as items in the fpgaconfig file, respectively packet 7 and 8.**

## Creating new packets and channels

For creating a new packet, right-click on *DMA_Read* or *DMA_Write* items and select *Add -> Packet.* You can also duplicate existing packets and all their sub-items by right clicking the *Packet* item and selecting *Duplicate*

## Move Packets and Channels

When creating a new packet or channel, it will be placed as the first children on the list of the DMA_Read or DMA_Write parent item:

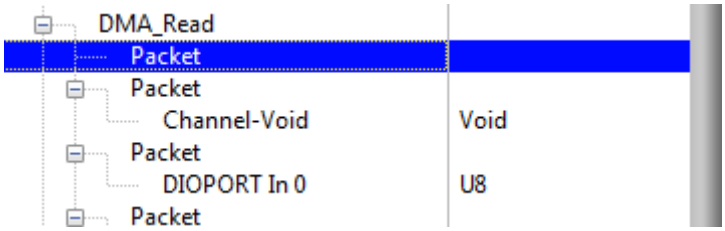


**Figure 7: Highlight of a newly created packet inside cVE.**

If you want to move the packet up and down, select the packet and then click **Ctrl + Up** or **Down** arrow to move the item up or down respectively:

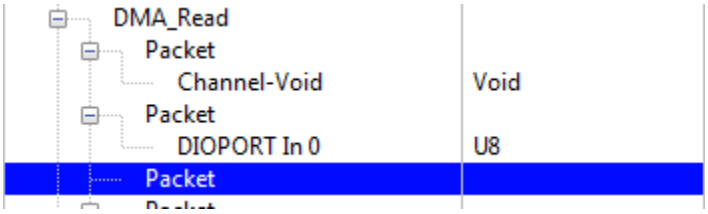


Figure 8: New packet being moved down by pressing Ctrl+Down Arrow

This operation can be done with channels and categories as well.

## Example: Modify the .fpgaconfig file provided by the Veristand Project Template

The LabVIEW Project generated before should include an FPGA configuration file called *<cRIO Model (e.g cRIO-9082)><Project Name>.fpgaconfig*. We're going to use the VCE to modify it in order to include only the necessary packets.

1. Open the Veristand FPGA Configuration Editor by selecting **Tools -> Veristand FPGA Configuration Editor** from the LabVIEW Project menu. Then load the template fpgaconfig file.

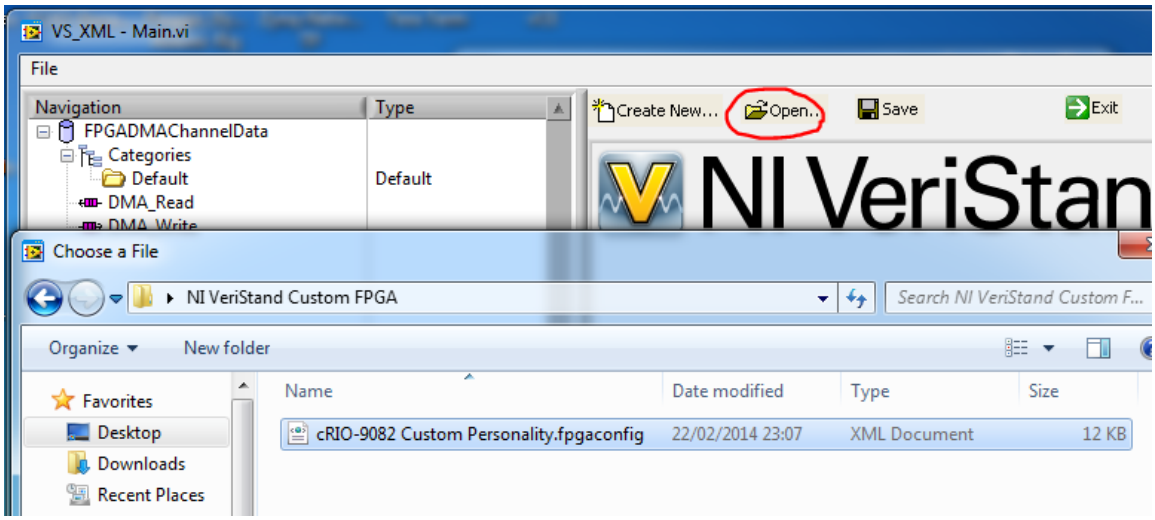


Figure 9: .fpgaconfig file selection window.

The VCE window should look like this:

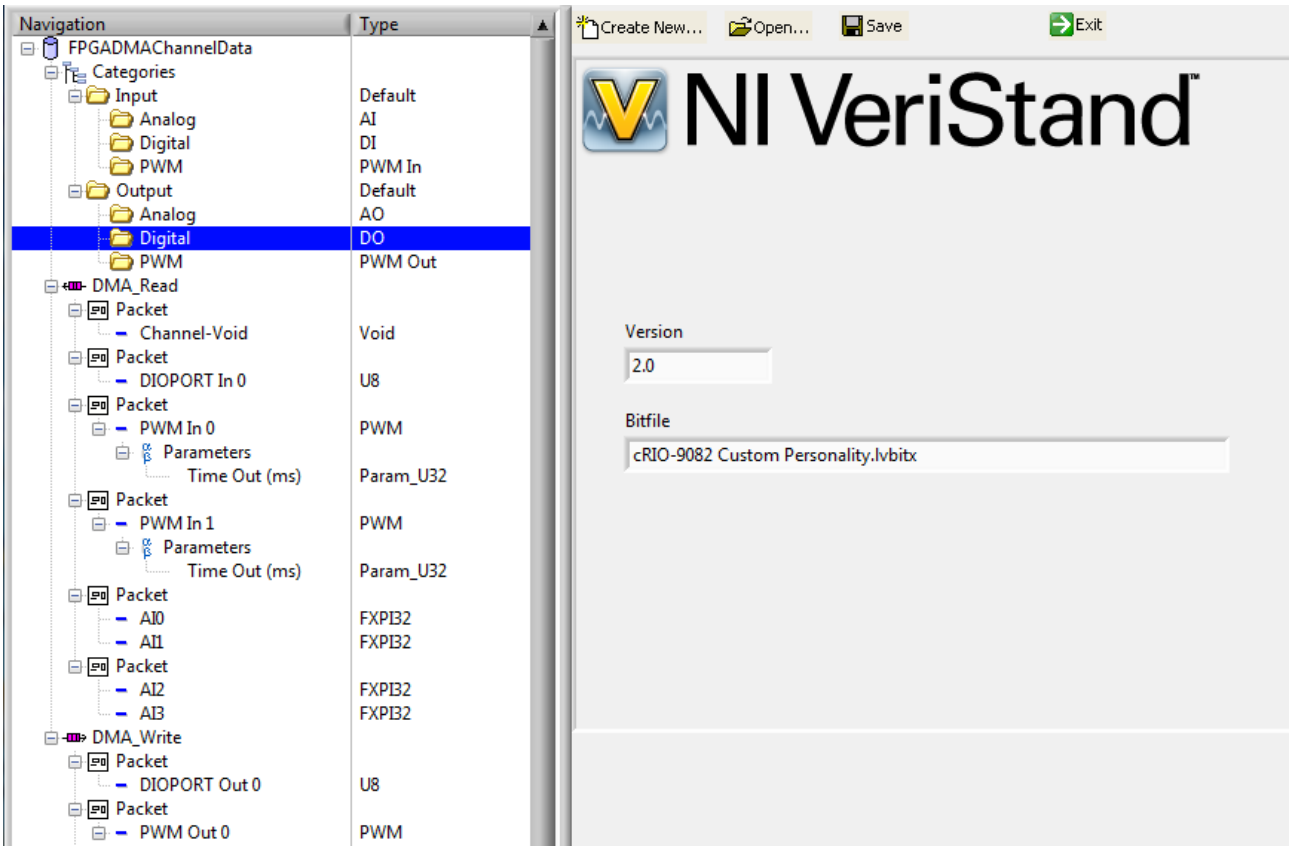


**Figure 10: Illustration of the vCE main window with the template .fpgaconfig file. On the left there are the items (categories, packets, channels) loaded from the fpgaconfig file. On the right there are the specific information associated with the highlighted item.**

2. Remove packets and categories until the item tree inside vCE looks like the one shown below:

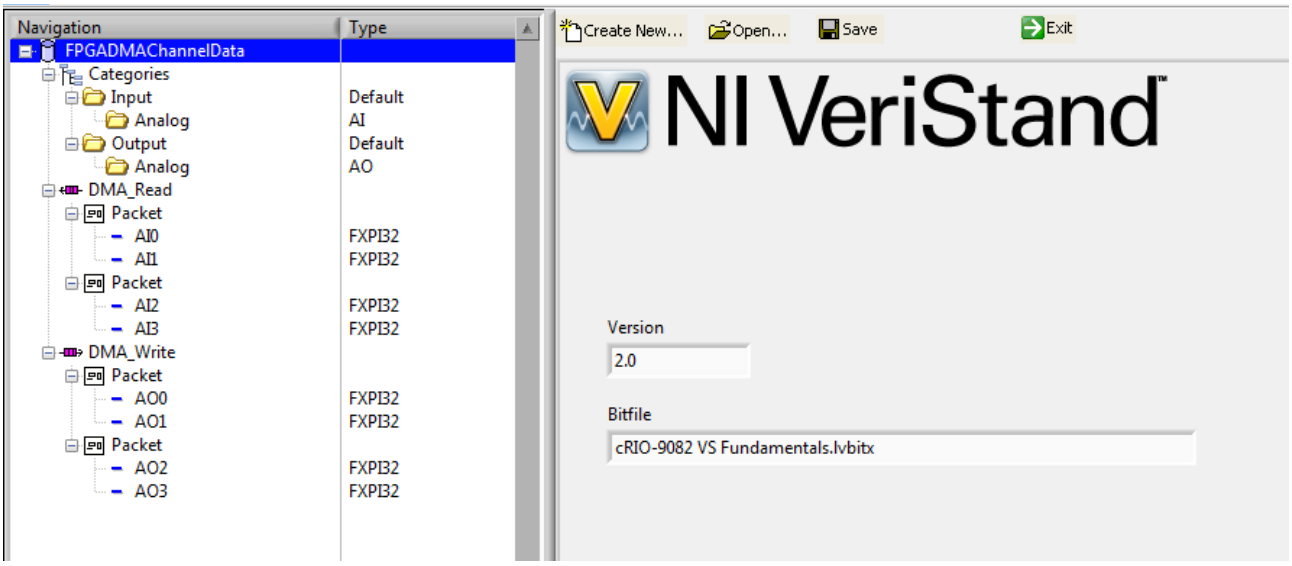


**Figure 11: Content of the .fpgaconfig file required by the example. Only packets related to 9215 (analog input) and 9263 (analog output) are included.**

3. Before saving and loading the fpgaconfig file into the System Explorer, take care of some important aspects:
   A. Compile the FPGA bitfile for the FPGA VI. Without the bitfile, Veristand will not load the fpgaconfig file in the System Explorer and return an error

B. **Copy** the bitfile in the **same folder** as the .fpgaconfig, as the bitfile is referenced by its relative path. All fpgaconfig file and the associated bitfiles are usually located in the *<Public Documents>\National Instruments\NI Veristand <version>\FPGA* folder, but you can place them wherever you want.

C. If not present, add the bitfile name in the *Bitfile* control shown below:

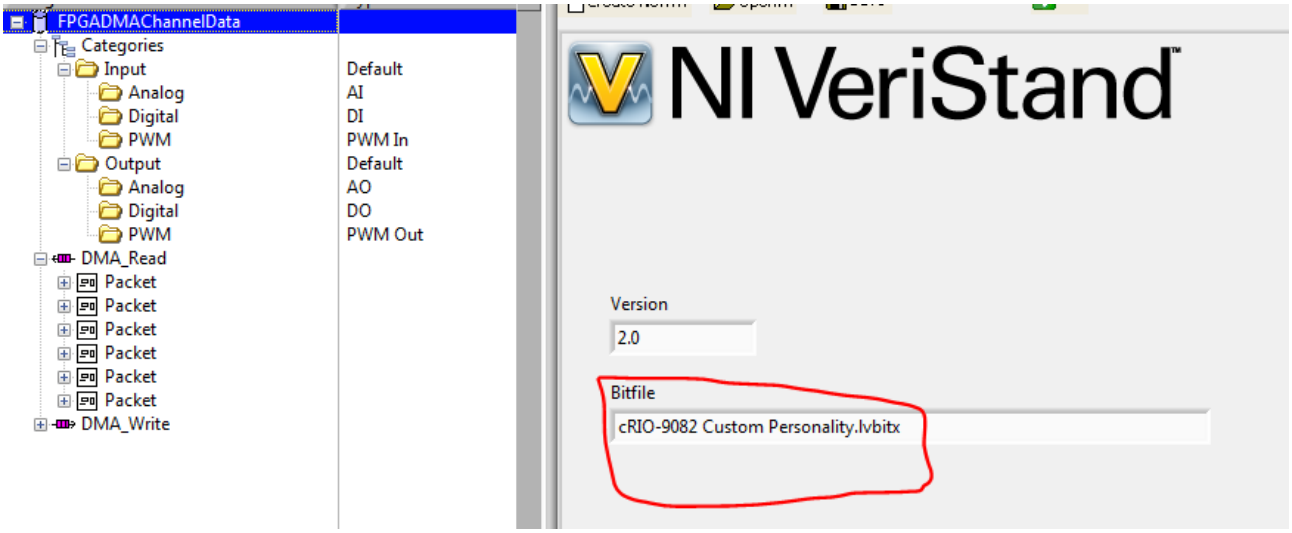


**Figure 12: Bitfile String Control where to insert the bitfile name associated with the fpcaconfig file.**

Once you checked all these steps, you can upload the FPGA target inside Veristand. For additional information about how to load FPGA devices into the System Explorer, please refer to this topic on the Veristand Online help:

http://zone.ni.com/reference/en-XX/help/372846G-01/veristand/fpga_se/

## Licensing

Any attached Code is provided As Is. It has not been tested or validated as a product, for use in a deployed application or system, or for use in hazardous environments. You assume all risks for use of the Code and use of the Code is subject to the Sample Code License Terms which can be found at: http://ni.com/samplecodelicense