NI LabVIEW Communications 1.1 to 2.0 Migration Guide

Recommended Steps to Migrate Pre-Existing Applications

Revision No. 1.0 – August 2016

NATIONAL INSTRUMENTS™

TABLE OF CONTENTS

INTRODUCTION

This guide provides information and walkthroughs for migrating existing LabVIEW Communications 1.1 applications to LabVIEW Communications 2.0. If you prefer to migrate over an existing application, this guide addresses known issues that may arise in the process. If possible, it is recommended to start with a new project in 2.0 and duplicate the original project in the newer version.

For an overview of new features and changes in LabVIEW Communications 2.0, see the New Features and Changes topic online.

HOW TO USE THIS GUIDE

This document is divided into the following main topics:

- 802.11 Application Framework
- LTE Application Framework
- USRP Projects
- FlexRIO Projects
- General Projects

Each section discusses the required steps to migrate that specific type of project. For example, if you would like to migrate a USRP streaming project, you should navigate to the USRP Projects section for guidance. If your project does not fall into the first three categories listed above, you should navigate to the General Projects section. Each section is meant to be self-contained and used independently of the others, so redundant information may be included in multiple sections.

Note: Always back up a copy of the original project before attempting any migration efforts.

802.11 APPLICATION FRAMEWORK

This section describes the necessary steps to reuse a project that was created based on the 802.11 Application Framework v1.1 (created in LabVIEW Communications 1.1) in LabVIEW Communications 2.0.

The section includes the following steps, each with their own subsection:

1. Reconfigure the Xilinx IP cores
2. Update "802.11 Internal Loopback Throttle Control.gcdl"
3. Migrate to the new USRP interface
4. Update host-side DMA FIFO's

RECONFIGURE THE XILINX IP

SCOPE

Because the FPGA compile tools uses a new Xilinx Vivado Toolchain version, all Xilinx IP cores must be updated to the new version. By configuring the IP cores again, the dependencies are regenerated ensuring that they will be compatible with the latest Vivado version to ensure successful compilations.

WHICH FILES ARE AFFECTED?

- All FPGA GCDL files which include a Xilinx IP core

The affected files can be found by completing the following steps:

1. Create a backup of your project.
2. Open your project in LabVIEW Communications 2.0.
3. Click **Yes** to convert as shown in Figure 1.

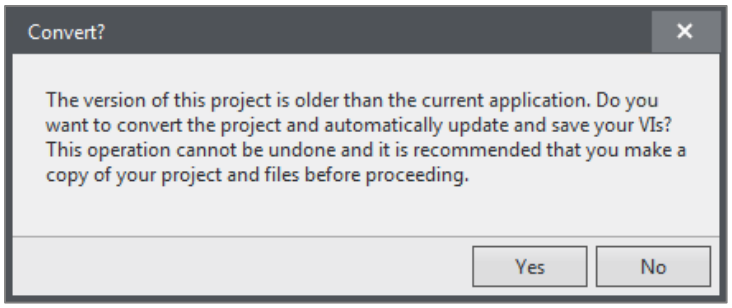

Figure 1: Convert? Dialog

4. Open the FPGA top-level VI you wish to migrate, for example, “802.11 FPGA STA.gvi“
5. Wait until type propagation has occurred.
6. If the document finished loading, you will see in the list of errors the affected files as shown in Figure 2.

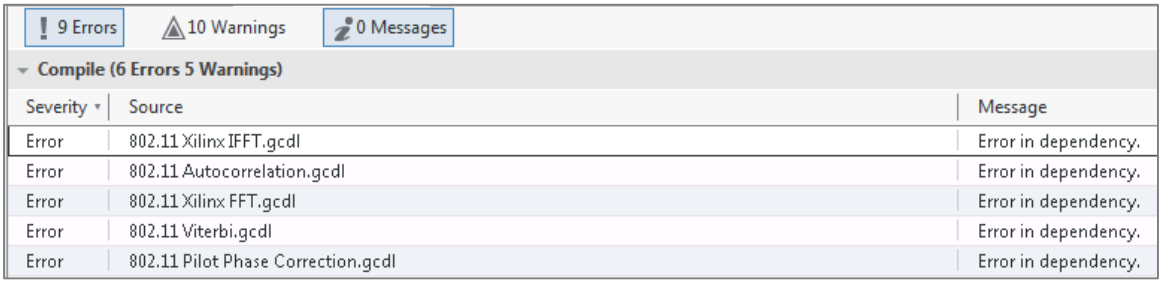

Figure 2 - Error list showing affected files for Xilinx IP Core reconfiguration

STEP-BY-STEP INSTRUCTIONS

1. If not done already, make sure to back up the original project and convert a copy for use with LabVIEW Communications 2.0. For details, refer to steps 1-3 from the previous section “Which Files Are Affected?”.
2. Once open in LabVIEW Communications 2.0, open the FPGA top-level you wish to migrate, for example, “802.11 FPGA STA.gvi“
3. Double-click on the errors, as shown in Figure 2. This will open the affected document.
4. Select the affected Xilinx IP block, and click on **Configure Xilinx IP Core** in the right-hand rail as shown in Figure 3.
5. Wait for the “Re-customize IP“ dialog to appear.
6. Inside the “Re-customize IP“ dialog, leave the pre-configured settings untouched, and click **OK** as shown in Figure 4.
7. Wait until the “Not Ready“ warning, as shown in Figure 5, has disappeared.
8. Repeat steps 3 through 6 until all errors in the FPGA top-level are gone.

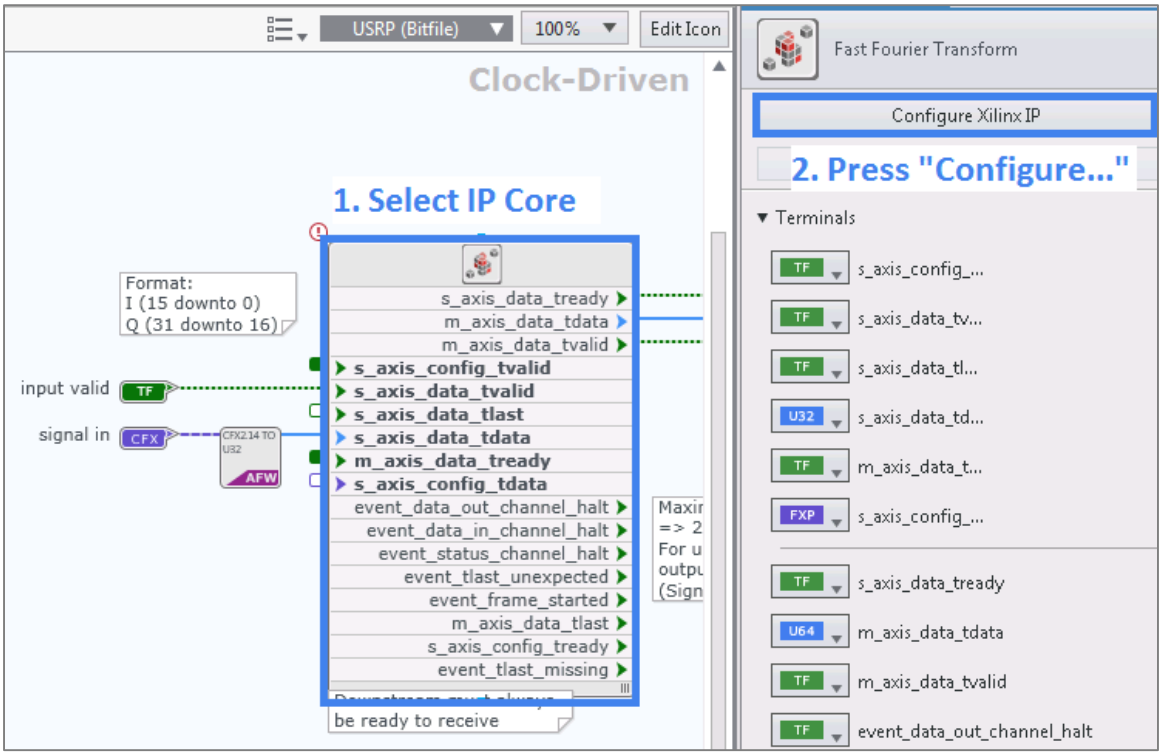

Figure 3 - How to reconfigure a Xilinx IP Core

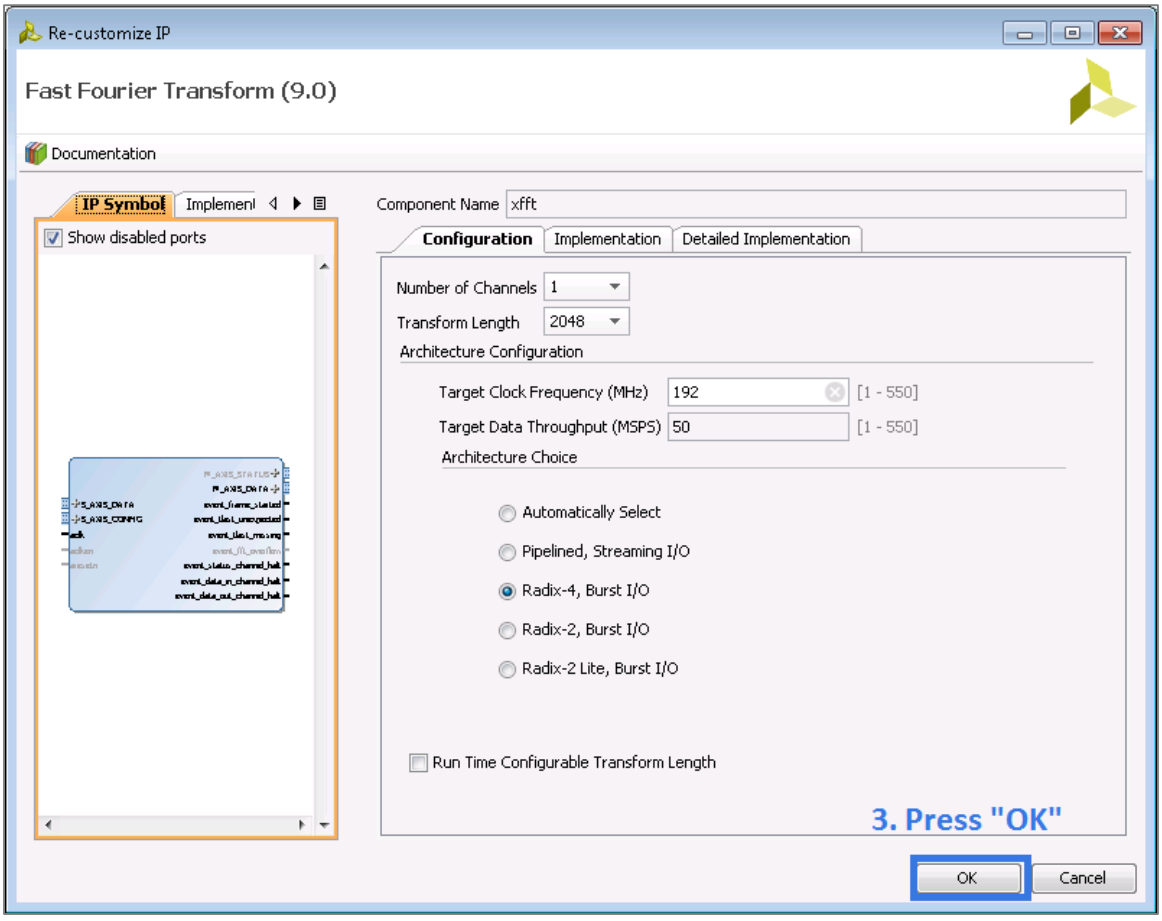

Figure 4 - How to reconfigure a Xilinx IP Core (2)

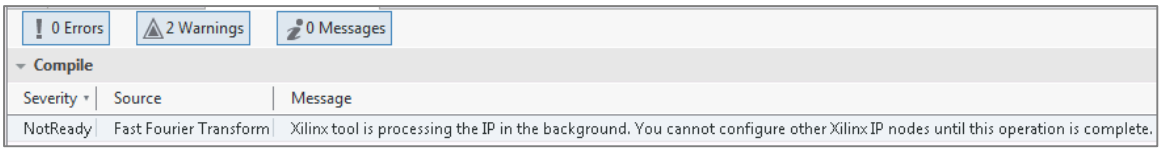

Figure 5 - "Not Ready" warning because Xilinx tool is processing the IP in the background

UPDATE "802.11 INTERNAL LOOPBACK THROTTLE CONTROL.GCDL"

SCOPE

Some changes in the Xilinx Vivado toolchain make it necessary to manually modify the "802.11 Internal Loopback Throttle Control.gcdl" to make sure the critical path is not too long. This helps to ensure that the design will meet timing and compilations will be successful.

WHICH FILES ARE AFFECTED?

- "802.11 Internal Loopback Throttle Control.gcdl"

STEP-BY-STEP INSTRUCTIONS

1. As shown in Figure 6, find "802.11 Internal Loopback Throttle Control.gcdl" by navigating to the *<project director>\802.11\FPGA\RX* folder in the Navigation Pane.

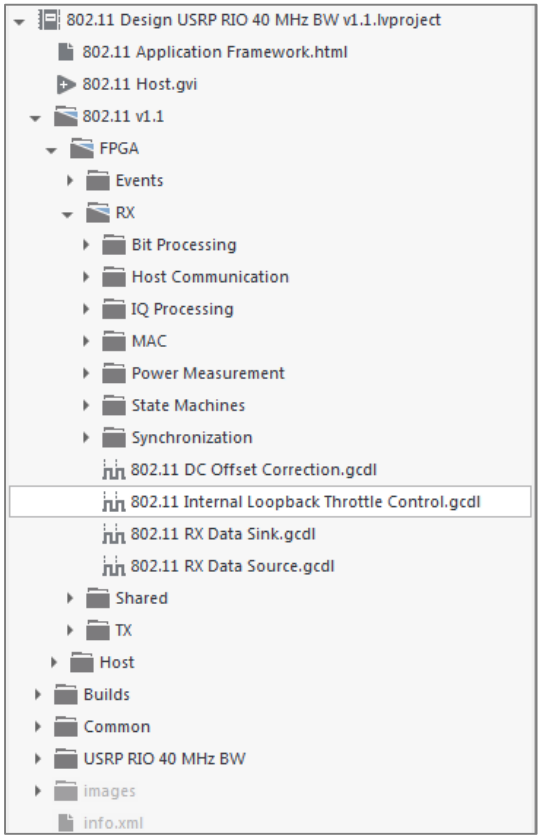

Figure 6: Navigation Pane Folder Path

2. Figure 7, shows the original, unmodified, code. Notice that the connection of the two Mod N Indexers in series together leads to a very long combinatorial path. To address this, it needs to be broken into two separate paths as shown in Figure 8.

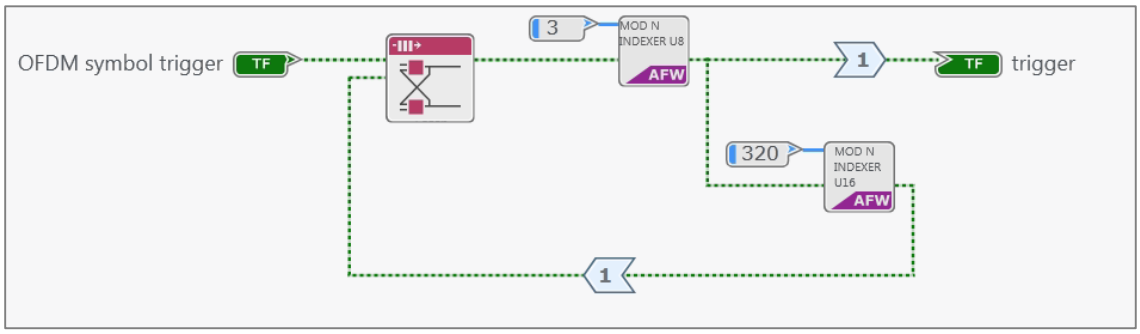

Figure 7: Unmodified "802.11 Internal Loopback Throttle Control.gcdl"

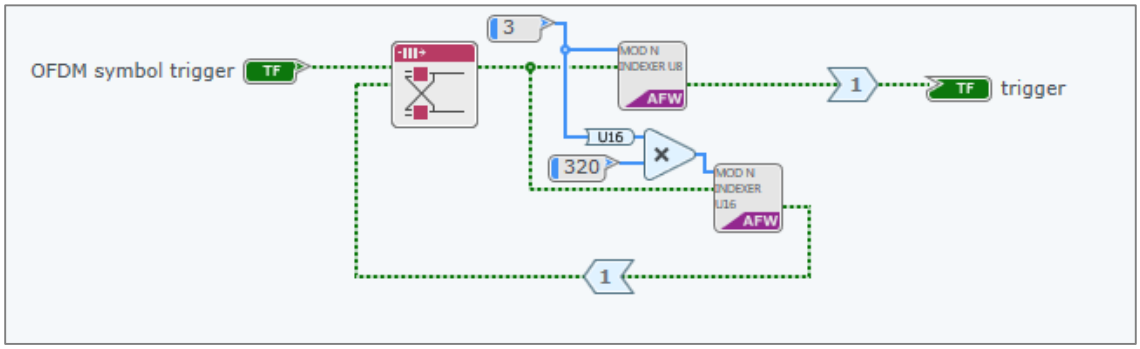

Figure 8: Adaptation of "802.11 Internal Loopback Throttle Control.gcdl"

MIGRATE TO THE NEW USRP INTERFACE

SCOPE

Because the USRP interface (40 MHz BW) is deprecated with LabVIEW Communications 2.0, which was used in the 802.11 Application Framework v1.1, a new interface must now be used on the FPGA top-level. The new interface now handles both the 40 MHz BW and 120 MHz BW USRP models with the same interface. Hence, an FPGA bitfile rebuild is required.

Without the rebuild, Error -61206 will occur when running the Host top-level "802.11 Host.gvi" as shown in Figure 9.

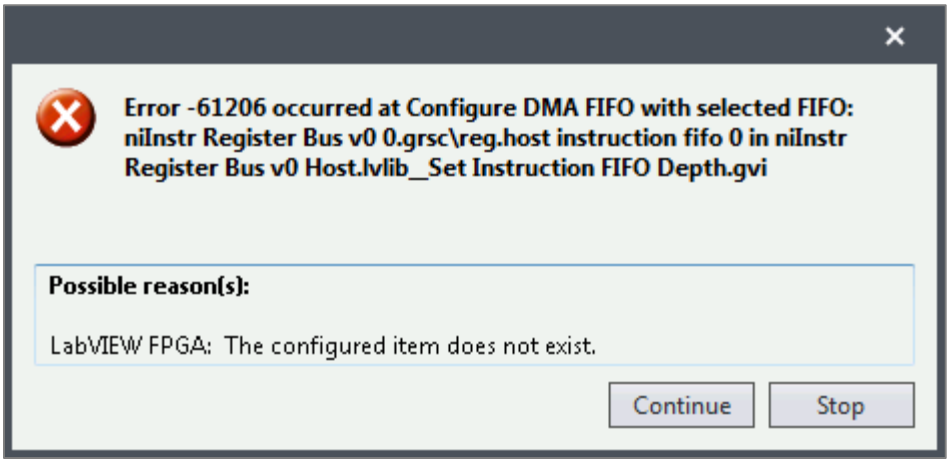

Figure 9 - Error -61206 occurs when running the Host Top-level without migrating the USRP interface

WHICH FILES ARE AFFECTED?

- The FPGA top-level "802.11 FPGA STA.gvi"

Note: Two new USRP interface files must be included in the project: "Create Resources.gvi", "Registers.gcdl"

STEP-BY-STEP INSTRUCTIONS

1. Select your USRP Target on the System Designer and change the "Model" in the right-hand rail to the non-deprecated version (ex: NI 2953R USRP (40 MHz BW) Deprecated » NI 2953R USRP (40 MHz BW)) like in Figure 10.

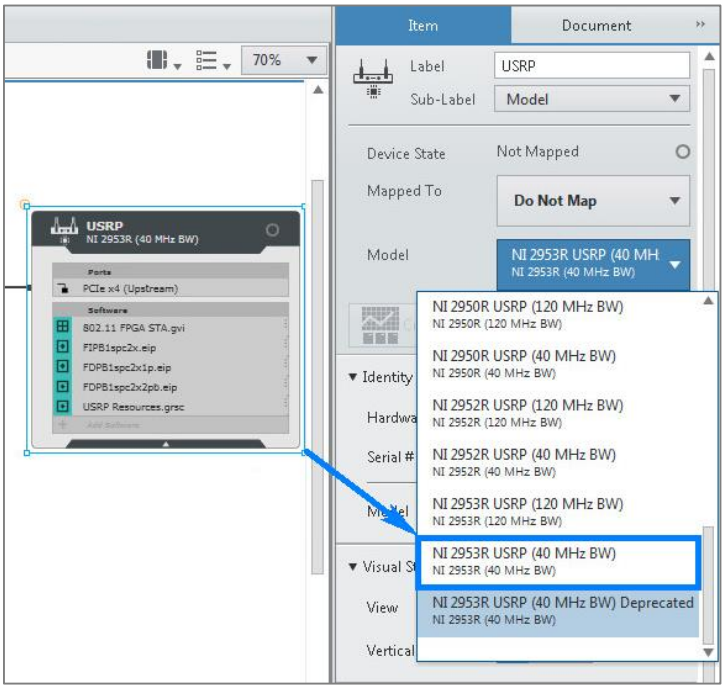

Figure 10: Change Deprecated Version

2. Generate a new 40 MHz Single-Device Streaming project in LabVIEW Communications 2.0 by navigating to New » Project Templates » USRP RIO 40 MHz BW Single-Device Streaming.
   Note: This project will only be used to copy the necessary project files, and code sections, into the previous project that is being migrated.

3. Copy two VIs from the new project into the original project. Refer to Figure 11.
   a. Find the following subVIs from the *<project directory>\FPGA subVIs* folder:
      i. Create Resources.gvi
      ii. Registers.gcdl

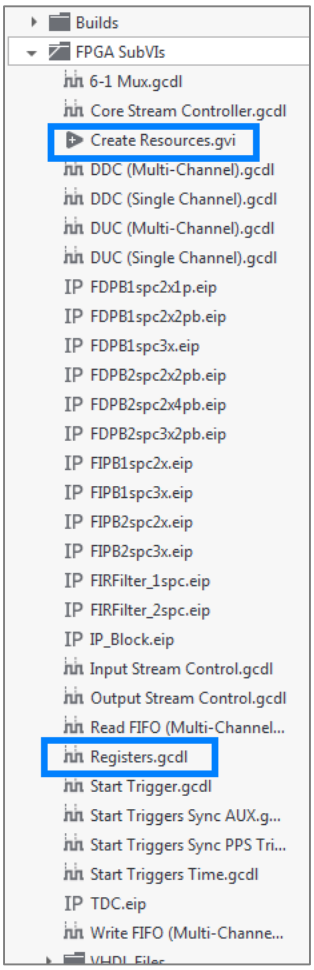

Figure 11: Snapshot of Required Files

   b. Copy these on disk and paste to the same directory in the original project.
   c. In the "Project Files" view on the Navigation Pane of the original project, the new files will appear slightly grayed out. Right-click on each and select **Include** as shown in Figure 12.
   d. Since the "Create Resources.gvi" is not a specific FPGA document (for example, CDL, MRD, and so on), there may be a prompt to select the target that the file should reside under. Select the same USRP target that your other files in this directory reside under.

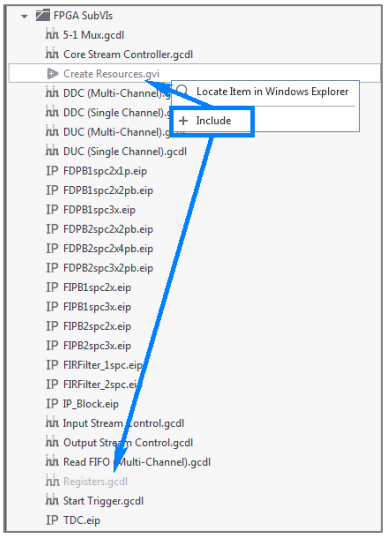

Figure 12: How to Include Additional Files in Project

4. In the original project to be migrated, open the FPGA top-level "802.11 FPGA STA.gvi"
   Note: If you finished all of the instructions of the first section, *Reconfigure the Xilinx IP*, then no error should be displayed for the FPGA top-level.
5. Find the three Clock-Driven Loops used for interfacing with the USRP and remove those loops as shown in Figure 13.

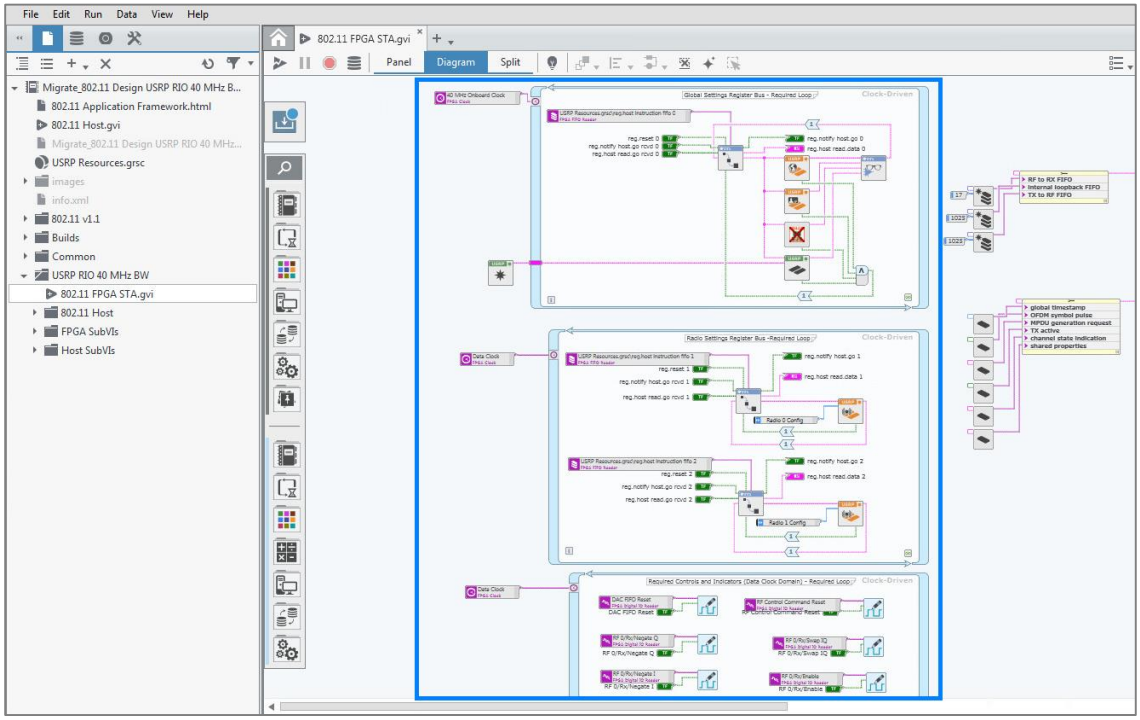

Figure 13 - Deleting the Clock-Driven Loops for interfacing with the USRP

6. Open the "Streaming Xcvr (FPGA).gvi" of the new project.
   a. As shown in Figure 14, select and copy the following from the Diagram: "Create Resources.gvi", "Process.gvi", the Clock Constant, and the Clock-Driven Loop.

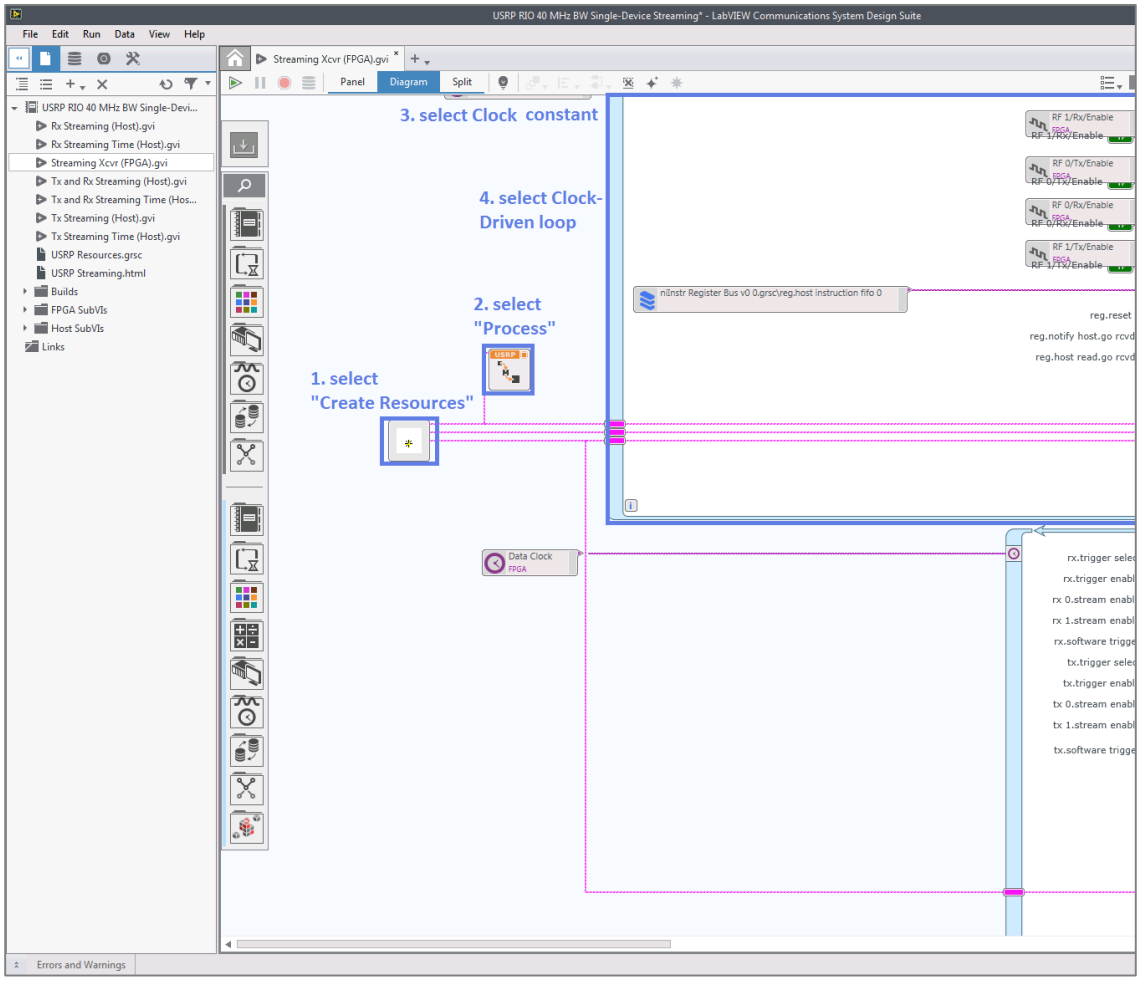

Figure 14 - USRP Interface Elements to be copied to the FPGA Top-Level

7. In the project you wish to migrate, paste the elements you copied from the new streaming project and wire as shown in Figure 15.

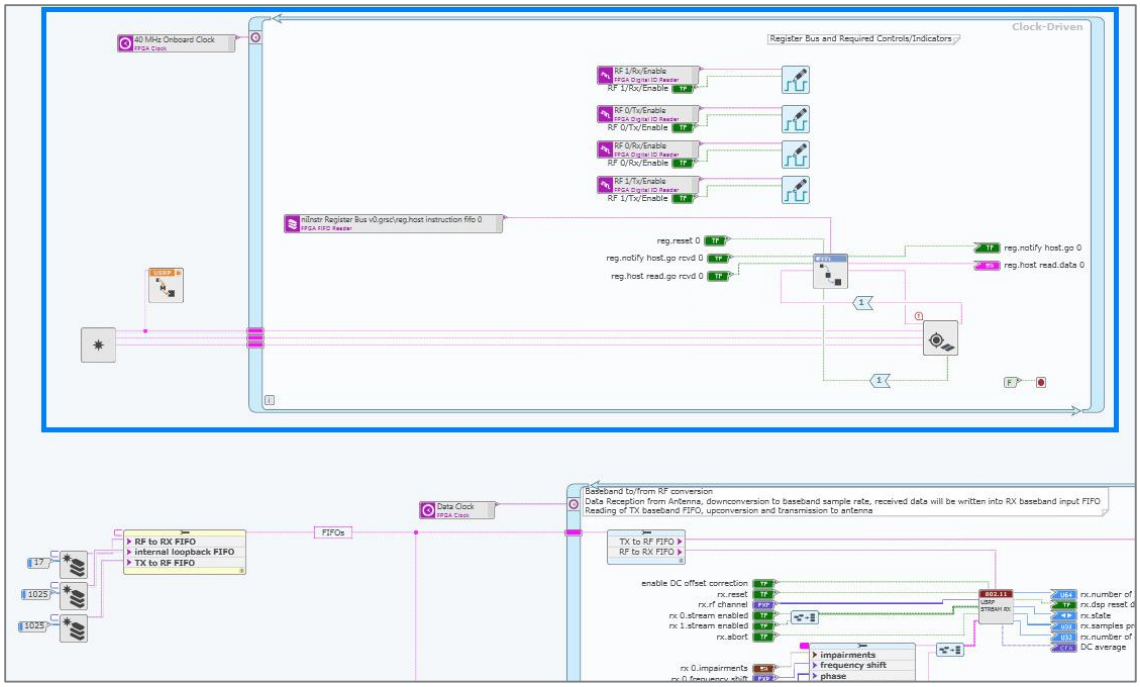

Figure 15 - Pasted USRP interface elements wired up

8. Allow all document changes to finish loading and save all files.
9. Close the new streaming project as it is no longer needed.

10. Compile the FPGA VI.
    a. Open the System Designer (in the "Project Files" view of the Navigation Pane on the left) and double-click on the top most project file.
    b. Select the FPGA top-level under the target you want to build the bitfile for.
    c. Click **Build** on the right-hand rail as shown in Figure 16.

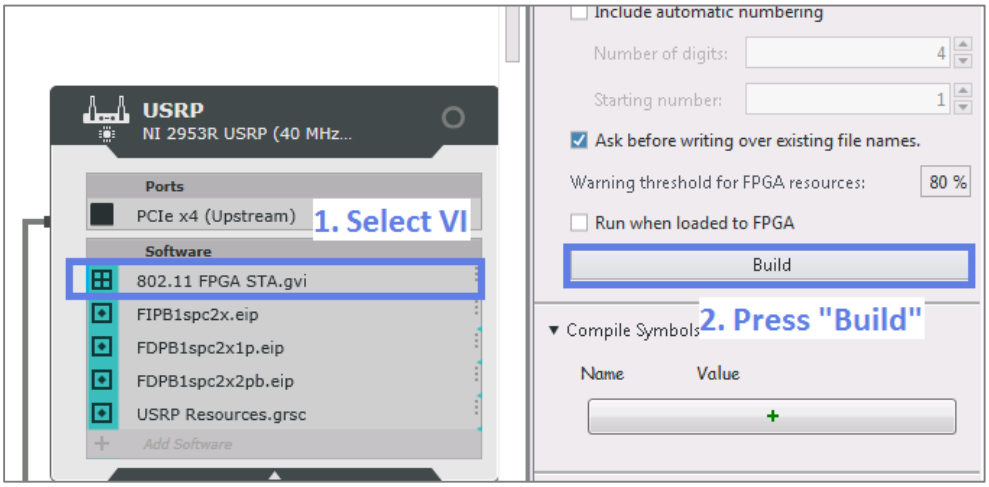

Figure 16 - Build a bitfile for the selected FPGA Top-Level VI

11. **(Recommended)** Once the bitfile is finished building, exclude it from the project as shown in Figure 17 and include it again. This will ensure that the latest changes are actually applied.

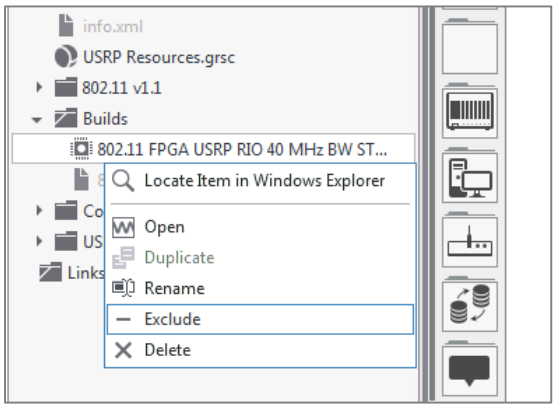

Figure 17 - Exclude and include the FPGA Top-Level again to make sure it is loaded correctly

UPDATE HOST-SIDE DMA FIFO'S

SCOPE

The FIFO resource names in the Host subVIs which interface with DMA FIFOs must be updated.

In LabVIEW Communications 1.0/1.1, the FIFO name contained only the name itself, for example, "T2H Events". In LabVIEW Communications 2.0, the FIFO resource contains additional prefixes for the FPGA resource file name along with the FIFO name itself, for example "USRP Resources.grsc\T2H Events".

Therefore, all occurrences of "Configure DMA FIFO", "Read DMA FIFO" and "Write DMA FIFO" must be updated to reference the new name.

WHICH FILES ARE AFFECTED?

- All Host subVIs which contain a FIFO node, for example, Start/Stop, Configure, or Read/Write operations

The list of VIs which require modification for "802.11 Host.gvi" is included in Table 1. If you miss one occurrence of a DMA FIFO operation, you will receive error -61206 during run-time as shown in Figure 18. In this example, the error indicates that the FIFO resource name of the node "Configure DMA FIFO" is incorrect in the file "LTE DL RX FIFO Initialization.gvi."

File	FIFO name	DMA FIFO operation
802.11 Setup DMA FIFOs.gvi	T2H Events	Configure, Start
	H2T RX Baseband	Configure, Start
	T2H RX Power Spectrum	Configure, Start
	T2H Constellation	Configure, Start
	T2H RX Bit Processing Output	Configure, Start
	T2H RX Data	Configure, Start
	H2T TX Data	Configure, Start
	T2H TX Bit Processing Output	Configure, Start
	T2H TX Baseband	Configure, Start
	T2H Channel Estimation	Configure, Start
802.11 Send TX ICP.gvi	H2T TX Data	Write
802.11 Receive RX ICP.gvi	T2H RX Data	Read
802.11 Prepare Constellation.gvi	T2H Constellation	Read
802.11 Prepare RX Power Spectrum.gvi	T2H RX Power Spectrum	Read
802.11 Prepare TX Power Spectrum.gvi	T2H TX Baseband	Read
802.11 Prepare Channel Estimation.gvi	T2H Channel Estimation	Read
802.11 Read Event FIFO.gvi	T2H Events	Read

Table 1: FIFO Operations in "802.11 Host.gvi"

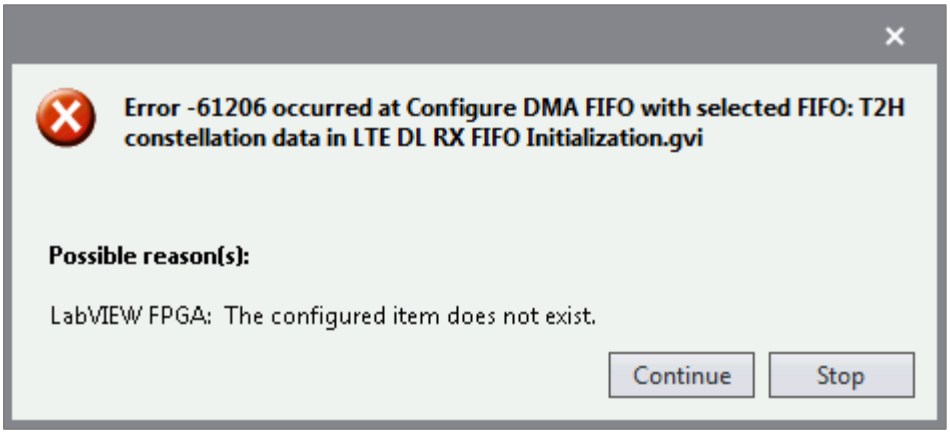

Figure 18 - Error -61206 occurred because of incorrect FIFO resource name

STEP-BY-STEP INSTRUCTIONS

1. Open the affected VI. In this example, we are demonstrating changes needed for the "802.11 Setup DMA FIFOs.gvi".
2. **(Recommended)** Add comments above the affected nodes which note the currently configured resource name as shown in Figure 19.

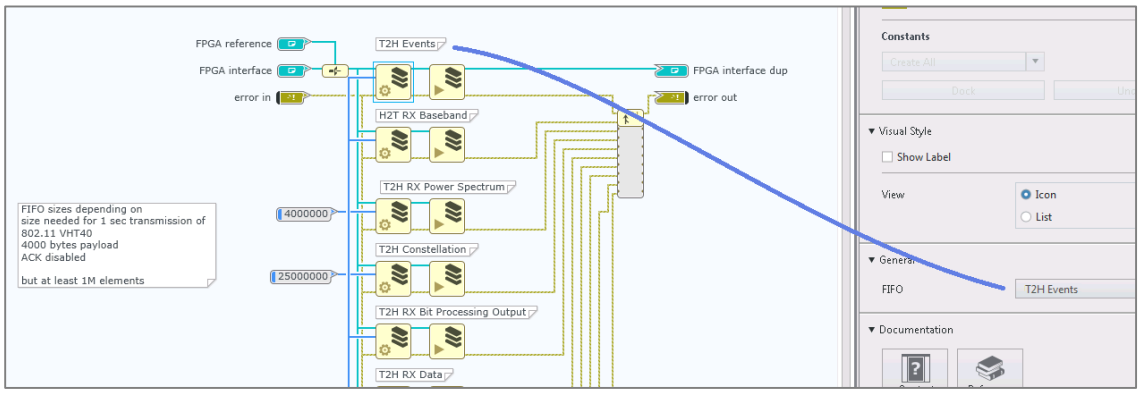

Figure 19 – "802.11 Setup DMA FIFOs.gvi" (after adding FIFO name comments)

3. Select the "FPGA reference" constant and click **Configure** on the right-hand rail as shown in Figure 20.

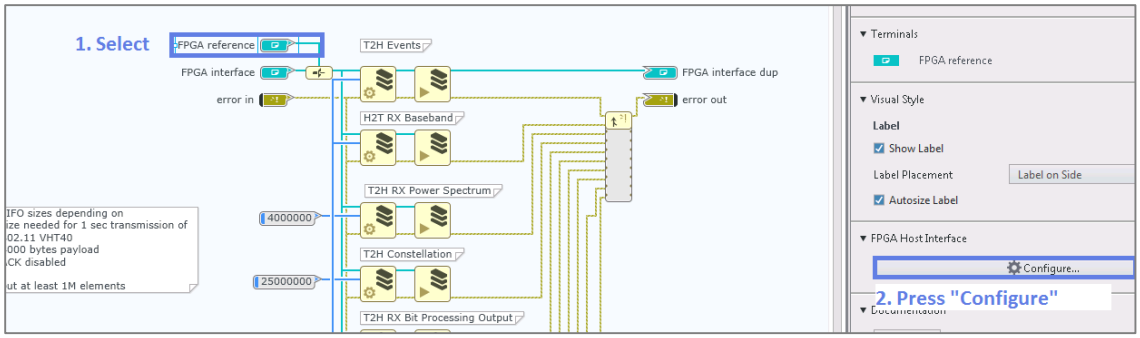

Figure 20 – "802.11 Setup DMA FIFOs.gvi" - Reconfigure FPGA type

4. In the "FPGA Interface Dynamic Refnum Configuration" dialog, click **Import bitfile** and navigate to the bitfile you just built in the last step of the third section, *Migrate to the New USRP Interface*.
5. **(Optional)** With the "FPGA Interface Dynamic Refnum Configuration" dialog still open, remove all unused resources by clicking the **X** symbol on the right side. This will not affect functionality but may make it easier to view used resources at a later time.
6. Click **OK** in the "FPGA Interface Dynamic Refnum" dialog.
7. Select the first FIFO node and adopt the FIFO name as shown in Figure 21.

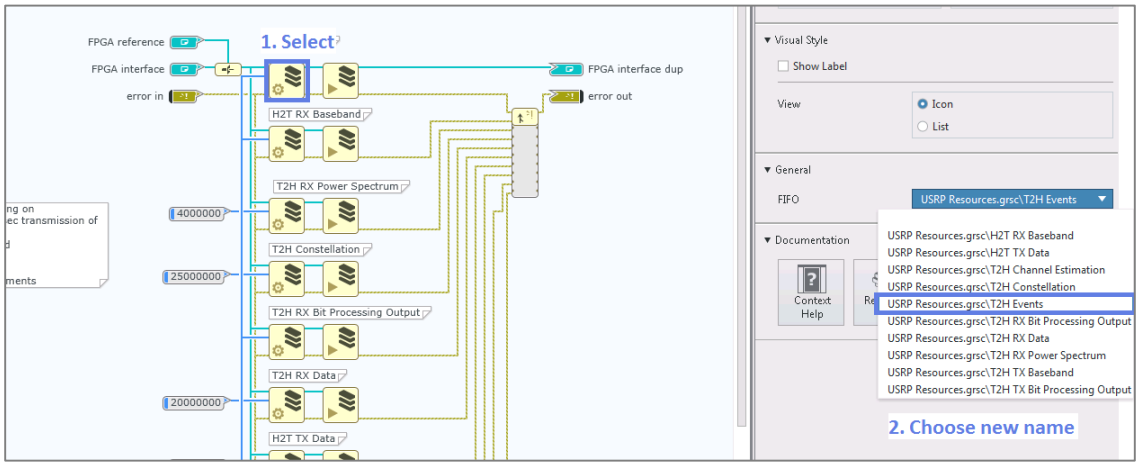

Figure 21 – "802.11 Setup DMA FIFOs.gvi" - Adopt FIFO Name

8. Repeat the previous step for all FIFO nodes in the current VI.
   Caution: Make sure to select the correct FIFO. Double-check if the name you selected matches the comment you added to the document in step 2.
9. Repeat steps 1-8 for all VIs which contain a DMA FIFO node.

LTE APPLICATION FRAMEWORK

This section describes the necessary steps if you want to reuse a project that was created based on the LTE Application Framework v1.1 (created in LabVIEW Communications 1.1) in LabVIEW Communications 2.0.

The section includes the following steps, each with their own subsection:

1. Reconfigure the Xilinx IP cores
2. Migrate to the new USRP interface
3. Update host-side DMA FIFO's

RECONFIGURE THE XILINX IP

SCOPE

Because the FPGA compile tools uses a new Xilinx Vivado Toolchain version, all Xilinx IP cores must be updated to the new version. By configuring the IP cores again, the dependencies are regenerated ensuring that they will be compatible with the latest Vivado version to ensure successful compilations.

WHICH FILES ARE AFFECTED?

- All FPGA GCDL files which include a Xilinx IP core

The affected files can be found by completing the following steps:

1. Create a backup of your project.
2. Open your project in LabVIEW Communications 2.0.
3. Click **Yes** to convert as shown in Figure 22.

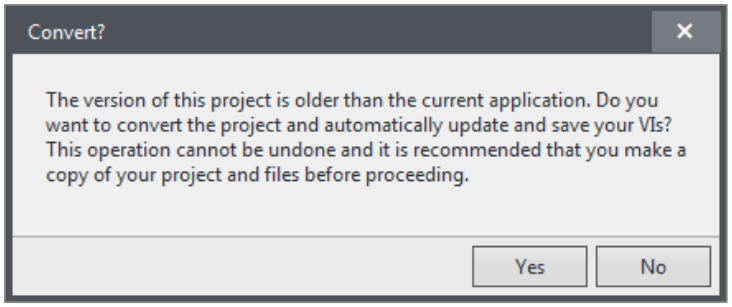

Figure 22: Convert Dialog

4. Open the FPGA top-level VI you wish to migrate, for example, "LTE FPGA USRP RIO 40 MHz BW DL.gvi."
5. Wait until type propagation has occurred.
6. If the document finished loading, you will see in the list of errors the affected files as shown in Figure 23.

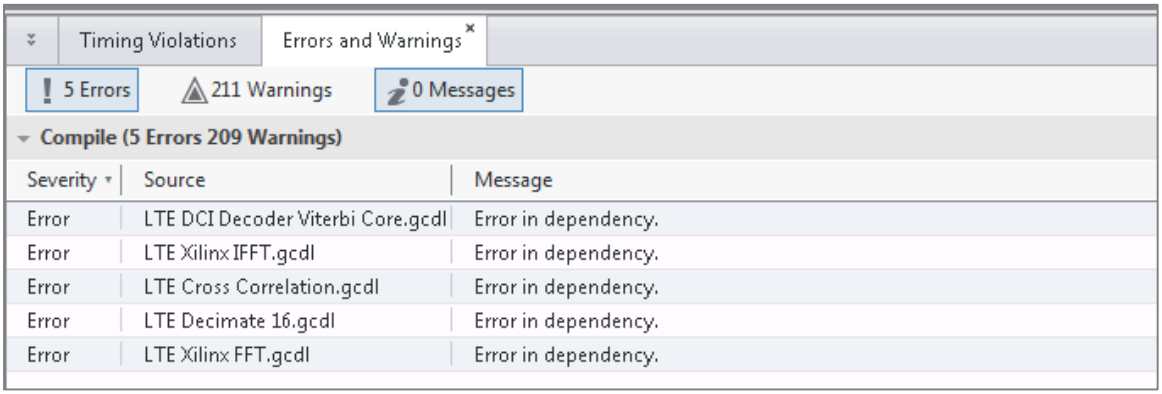

Figure 23 - Error list showing affected files for Xilinx IP Core reconfiguration

STEP-BY-STEP INSTRUCTIONS

1. If not done already, make sure to back up the original project and convert a copy for use with LabVIEW Communications 2.0. For details, refer to steps 1-3 from the previous section "Which Files Are Affected?".
2. Once open in LabVIEW Communications 2.0, open the FPGA top-level you wish to migrate, for example, "LTE FPGA USRP RIO 40 MHz BW DL.gvi".
3. Double-click on the errors as shown in Figure 23. This will open the affected document.
4. Select the affected Xilinx IP block and click **Configure Xilinx IP Core** in the right-hand rail as shown in Figure 24.
5. Wait for the "Re-customize IP" dialog to appear.
6. Inside the "Re-customize IP" dialog, leave the settings untouched and click **OK** as shown in Figure 25.
7. Wait until the warning "Not Ready", as shown in Figure 26, has disappeared.
8. Repeat steps 3 through 6 until all errors in the FPGA top-level have disappeared.

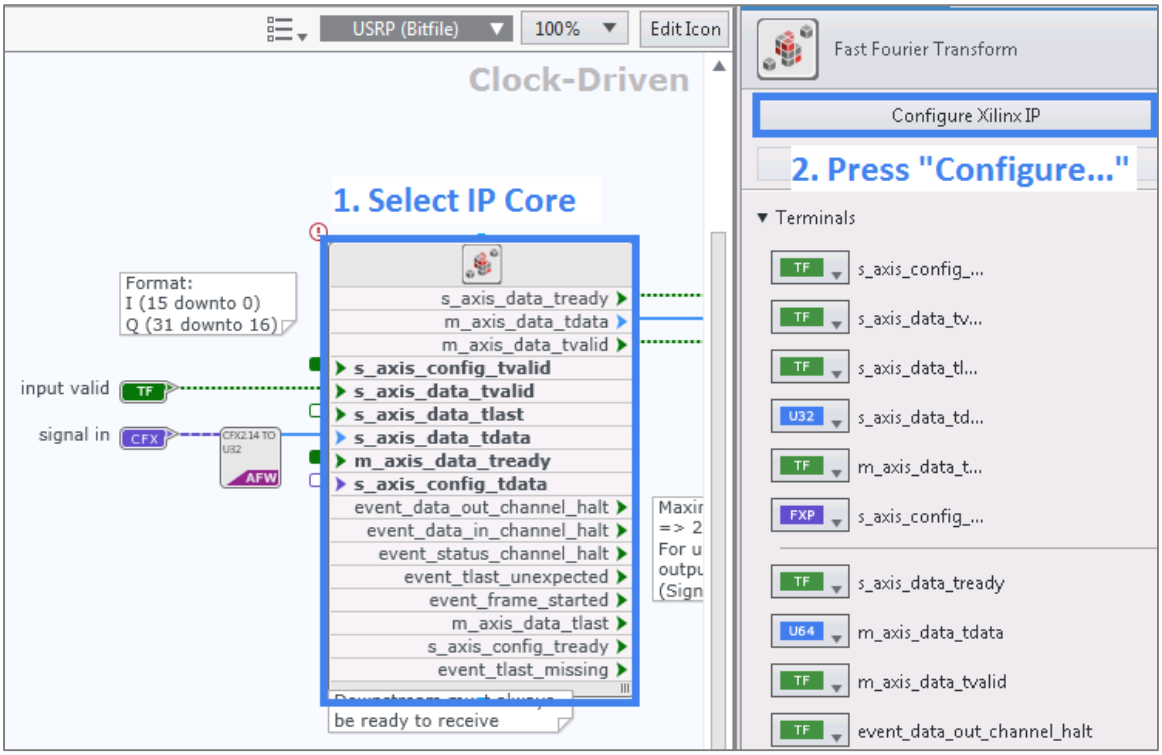

Figure 24 - How to reconfigure a Xilinx IP Core (1)

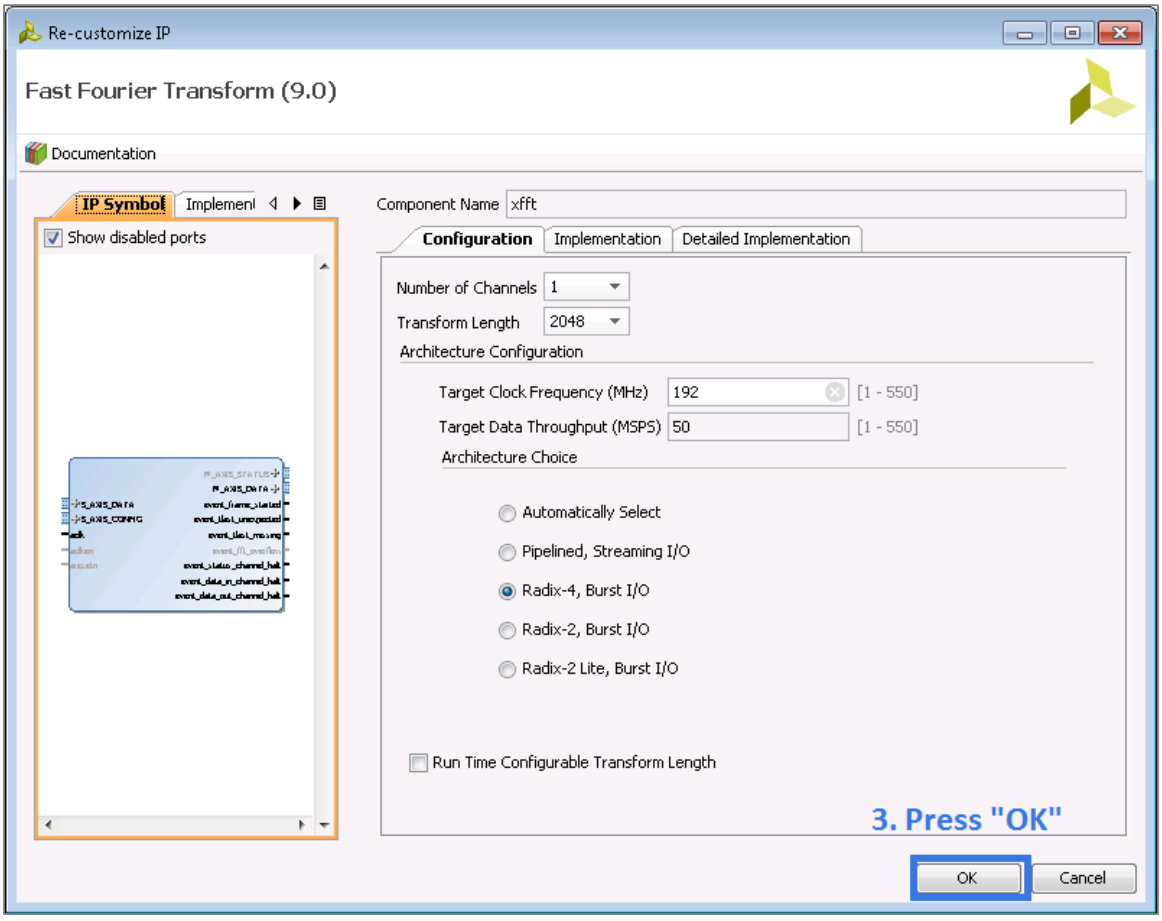

Figure 25 - How to reconfigure a Xilinx IP Core (2)

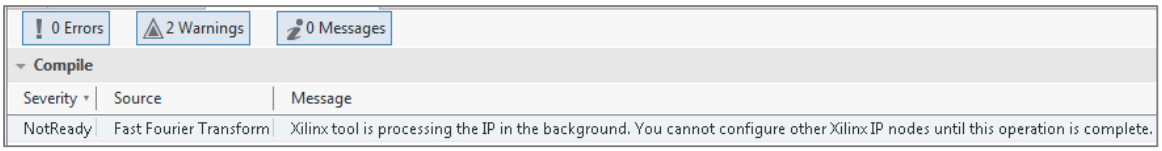

Figure 26 - "Not Ready" warning because Xilinx tool is processing the IP in the background

MIGRATE TO THE NEW USRP INTERFACE

SCOPE

Because the USRP interface (40 MHz BW) used in the LTE Application Framework v1.1 is deprecated with LabVIEW Communications 2.0, the new interface must now be used on the FPGA top-level. The new interface now handles both the 40 MHz BW and 120 MHz BW USRP models with the same interface. Hence, an FPGA bitfile rebuild is required.

Without the rebuild, Error -61206 will occur when running the Host top-level, for example, "LTE Host DL.gvi" as shown in Figure 27.

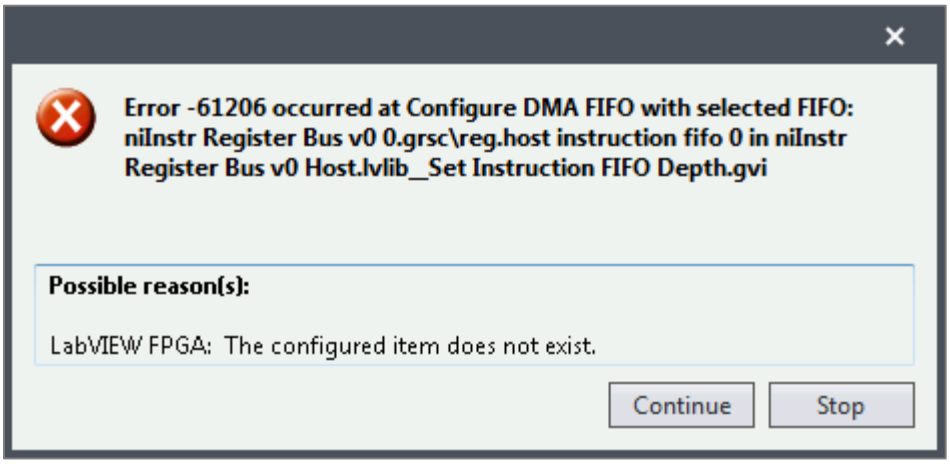

Figure 27 - Error -61206 occurs when running the Host Top-Level without migrating the USRP interface

WHICH FILES ARE AFFECTED?

- The FPGA top-level, for example, "LTE FPGA USRP RIO 40 MHz BW DL.gvi"

Note: Two new USRP interface files must be included in the project: "Create Resources.gvi", "Registers.gcdl"

STEP-BY-STEP INSTRUCTIONS

1. Select your USRP Target on the System Designer and change the "Model" in the right-hand rail to the non-deprecated version (ex: NI 2953R USRP (40 MHz BW) Deprecated » NI 2953R USRP (40 MHz BW)).

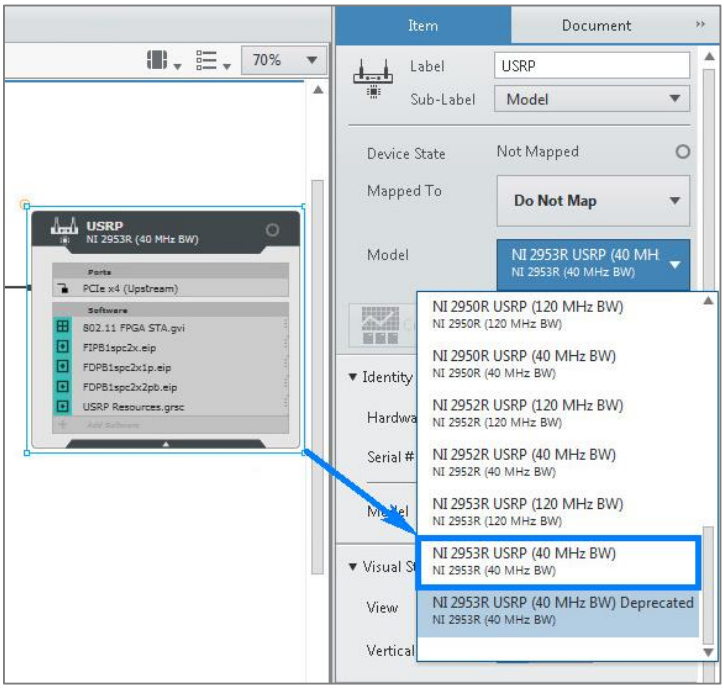

Figure 28: Change Deprecated Version

2. Generate a new 40 MHz Single-Device Streaming project in LabVIEW Communications 2.0 by navigating to New » Project Templates » USRP RIO 40 MHz BW Single-Device Streaming.
Note: This project will only be used to copy the necessary project files, and code sections, into the previous project that is being migrated.

3. Copy two VIs from the new project must be copied into the original project as shown in Figure 29.
   a. Find the following subVIs from the *<project directory>\FPGA subVIs* folder:
      i. Create Resources.gvi
      ii. Registers.gcdl

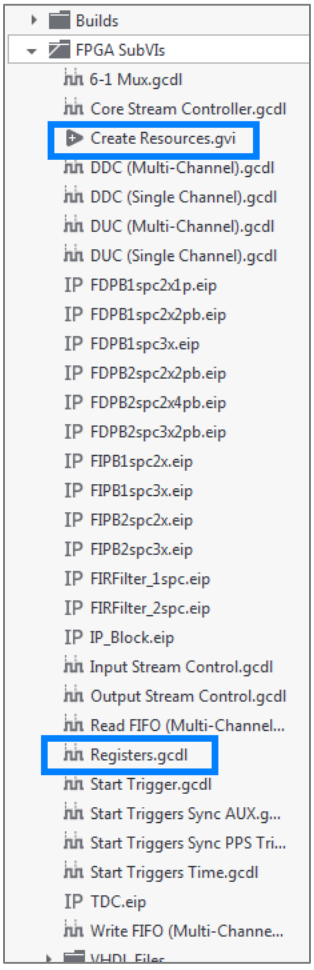

Figure 29: Files to Copy

   b. Copy these on disk and paste to the *<project directory>\USRP RIO 40 MHz BW\FPGA* directory in the original project.
   c. In the "Project Files" view on the Navigation Pane of the original project, the new files will appear slightly grayed out. Right-click on each and select **Include**, as shown in Figure 30.
   d. Since the "Create Resources.gvi" is not a specific FPGA document (for example, CDL, MRD, and so on), there may be a prompt to select the target that the file should reside under. Select the same USRP target that your other files in this directory reside under.

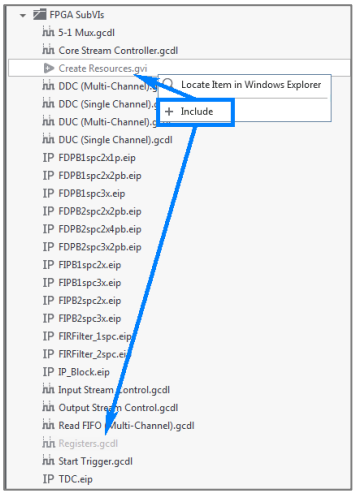

Figure 30: Include Copied Files

4. In the original project to be migrated, open the FPGA top-level "LTE FPGA USRP RIO 40 MHz BW DL.gvi".
   Note: If you finished all of the instructions of the first section, *Reconfigure the Xilinx IP*, no error should be displayed for the FPGA top-level.
5. Find the three Clock-Driven Loops which are used for interfacing with the USRP as shown in the following Figure 31.

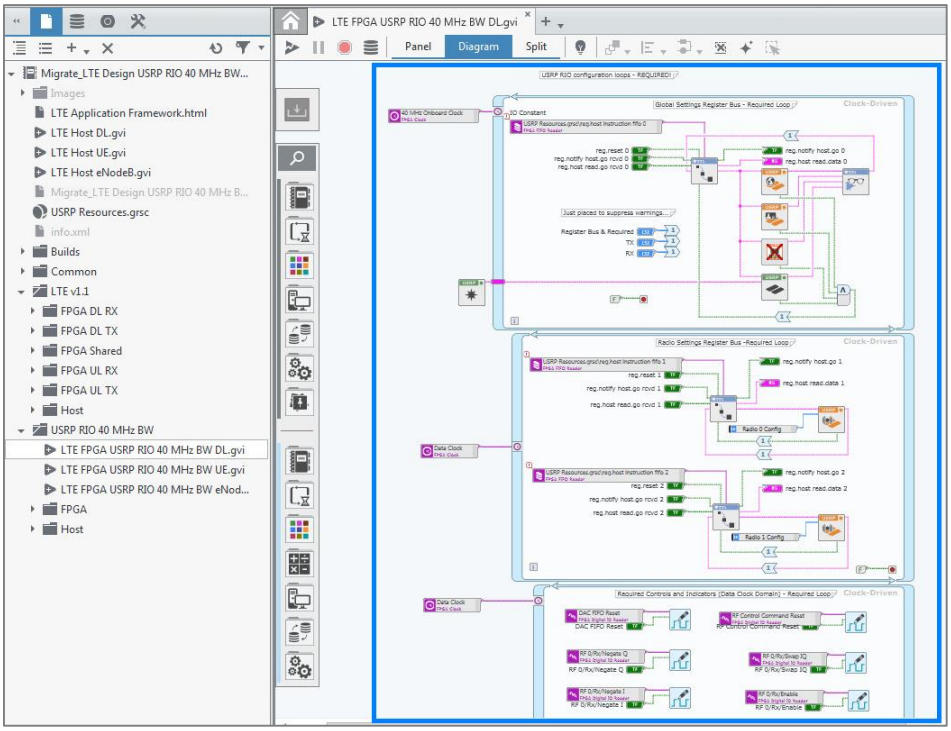

Figure 31: CDLs noted in Step 7

6. Move the controls “Register Bus & Required“, “TX“ and “RX“ outside the first USRP Clock-Driven Loop as shown in Figure 32.

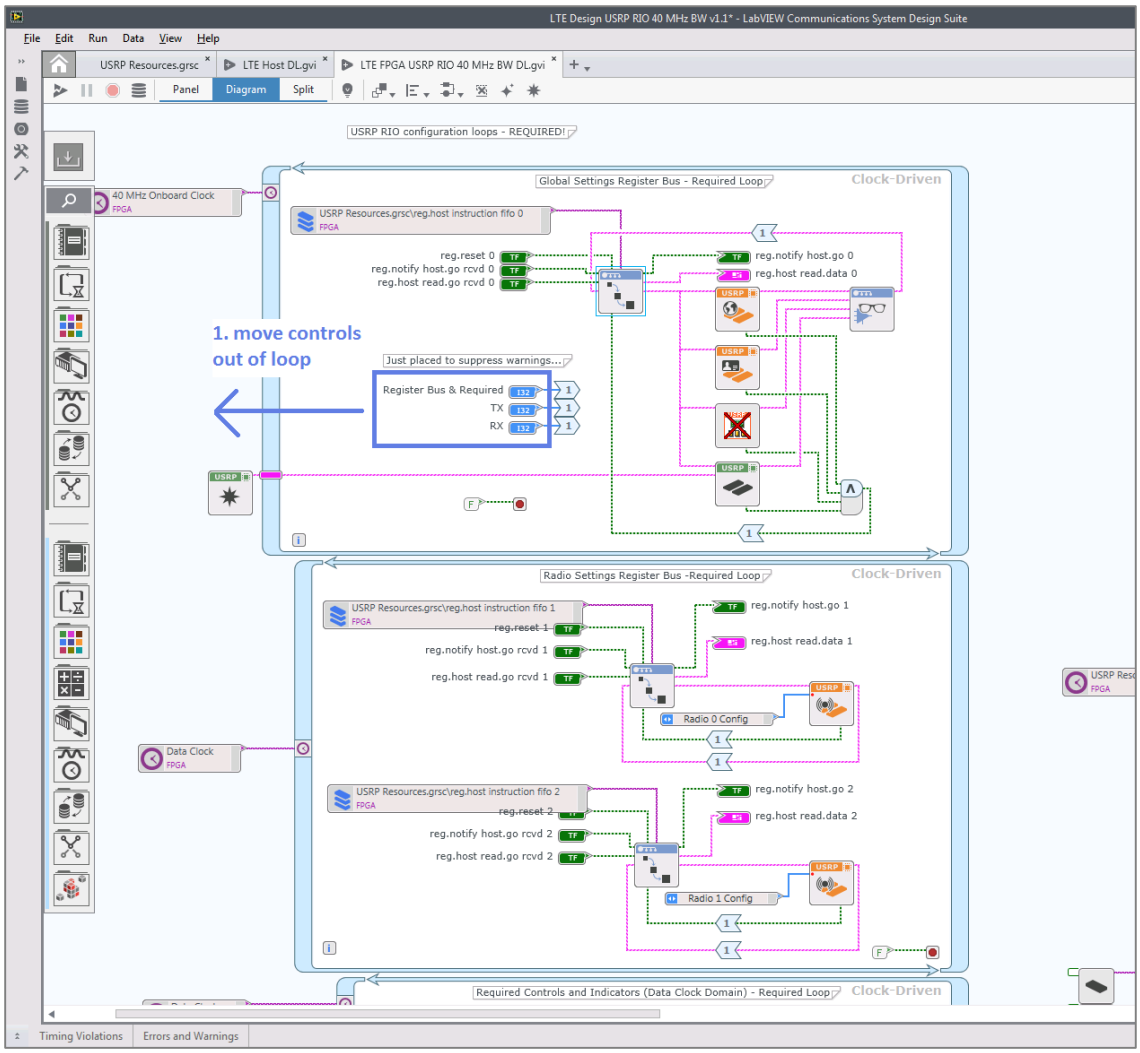

Figure 32 - Clock-Driven Loops for interfacing with the USRP on the FPGA Top-Level

7. Delete the three Clock-Driven Loops used for interfacing with the USRP as shown in Figure 33.

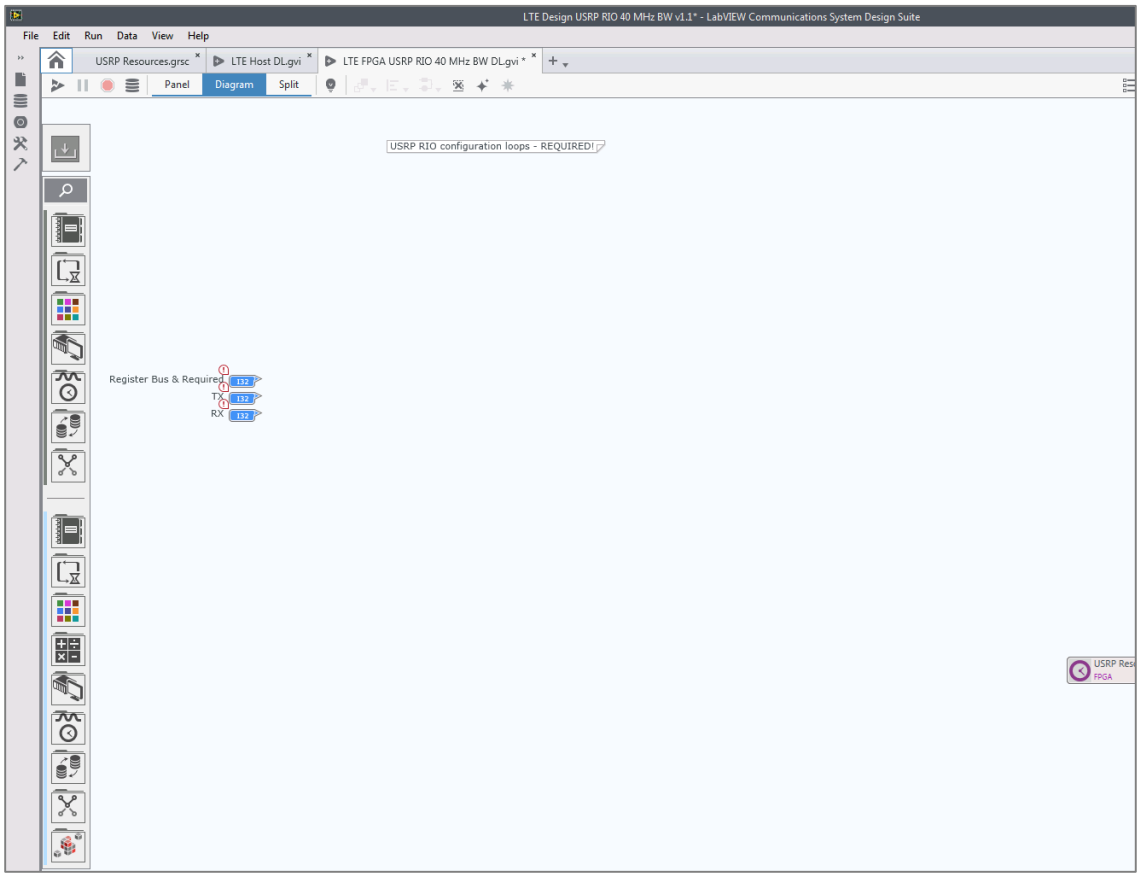

Figure 33 - Deleting the Clock-Driven Loops for interfacing with the USRP

8. Open the “Streaming Xcvr (FPGA).gvi” of the new project.
   As shown in Figure 34, select and copy the following from the Diagram: “Create Resources.gvi”, “Process.gvi”, the Clock Constant, and the Clock-Driven Loop.

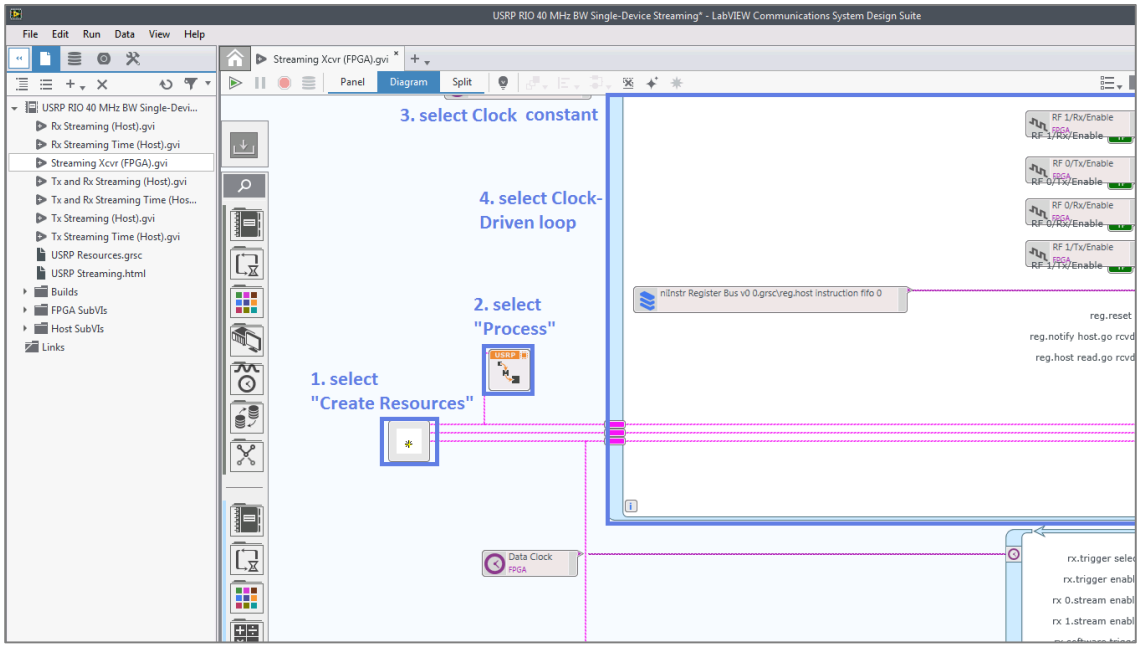

Figure 34 - USRP Interface Elements to be copied to the FPGA Top-Level

9. In the project you wish to migrate, paste the elements you copied from the new streaming project and wire as shown in Figure 35.

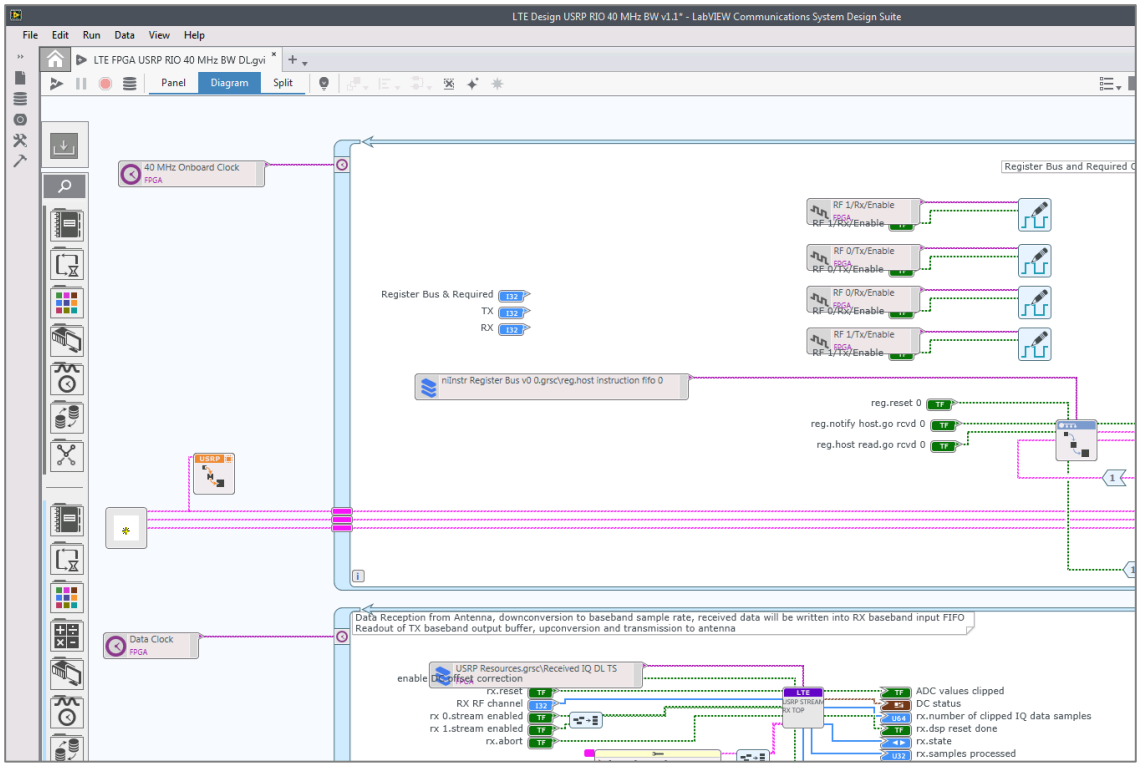

Figure 35 - Pasted USRP interface elements wired up

10. Allow all document changes to finish loading and save all files.
11. Close the new streaming project as it is no longer needed.

12. Compile the FPGA VI.
    a. Open the System Designer (in the "Project Files" view of the Navigation Pane on the left) and double-click on the top most project file.
    b. Select the FPGA top-level under the target you want to build the bitfile for.
    c. Click **Build** on the right-hand rail as shown in Figure 36.

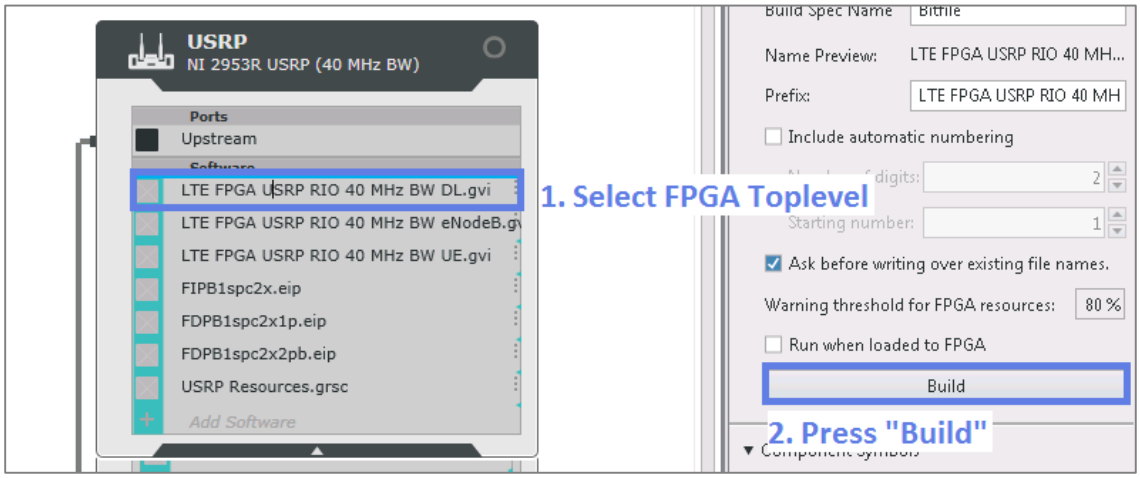

Figure 36 - Build a bitfile for the selected FPGA Top-Level VI

13. **(Recommended)** Once the bitfile is finished building, exclude it from the project as shown in Figure 37 and include it again. This will ensure that the latest changes are actually applied.

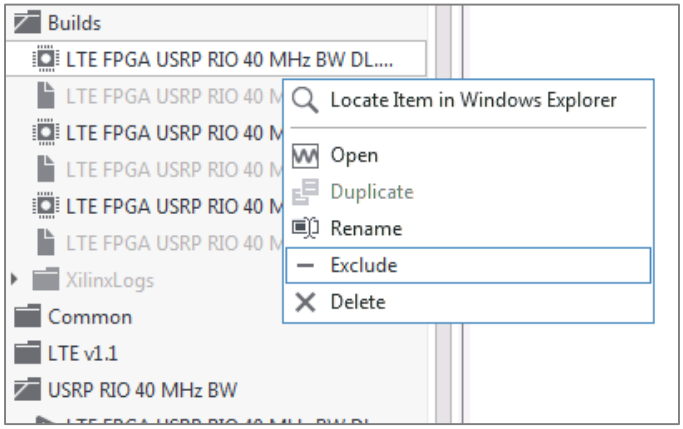

Figure 37 - Exclude and include the FPGA Top-Level again to make sure it is loaded correctly

UPDATE HOST-SIDE DMA FIFO'S

SCOPE

The FIFO resource names in the Host subVIs which interface with DMA FIFOs must be updated.

In LabVIEW Communications 1.0/1.1, the FIFO name contained only the name itself, for example, "PDSCH Decoding Status". In LabVIEW Communications 2.0, the FIFO resource contains of the prefixes the FPGA resource file name and the FIFO name itself, for example, "USRP Resources.grsc\PDSCH Decoding Status".

Therefore, all occurrences of "Configure DMA FIFO", "Read DMA FIFO" and "Write DMA FIFO" must be updated to reference the new name.

WHICH FILES ARE AFFECTED

- All Host subVIs which contain a FIFO node (for example, Start/Stop, Configure, or Read/Write operations)

The list of VIs which require modification for "LTE Host DL.gvi" is included in Table 2. If you miss one occurrence of a DMA FIFO operation, you will receive error -61206 during run-time as shown in Figure 38 in which the error indicates that the FIFO resource name of the node "Configure DMA FIFO" is incorrect in the file "LTE DL RX FIFO Initialization.gvi."

File	FIFO name	DMA FIFO operation
LTE DL RX FIFO Initialization.gvi	T2H constellation data	Configure, Start
	T2H PDCCH	Configure, Start
	PDSCH Decoded Data T2H	Configure, Start
	PDSCH Decoding Status T2H	Configure, Start
	T2H Rx Stream	Configure, Start
	T2H Channel Estimates	Configure, Start
LTE DL TX FIFO Initialization.gvi	TX BB T2H	Configure, Start
	H2T PDSCH	Configure, Start
LTE Read PDSCH Decoder Status.gvi	PDSCH Decoding Status T2H	Read
LTE Read TX Data.gvi	TX BB T2H	Read
LTE Read RX Data.gvi	T2H Rx Stream	Read
LTE Read PDCCH Data.gvi	T2H PDCCH	Read
LTE Read Constellation Data.gvi	T2H constellation data	Read
LTE Read Channel Estimates.gvi	T2H Channel Estimates	Read
LTE Send UDP Data.gvi	PDSCH Decoded Data T2H	Read
LTE Receive UDP Data.gvi	H2T PDSCH	Write

Table 2: FIFO "Operations in LTE Host DL.gvi"

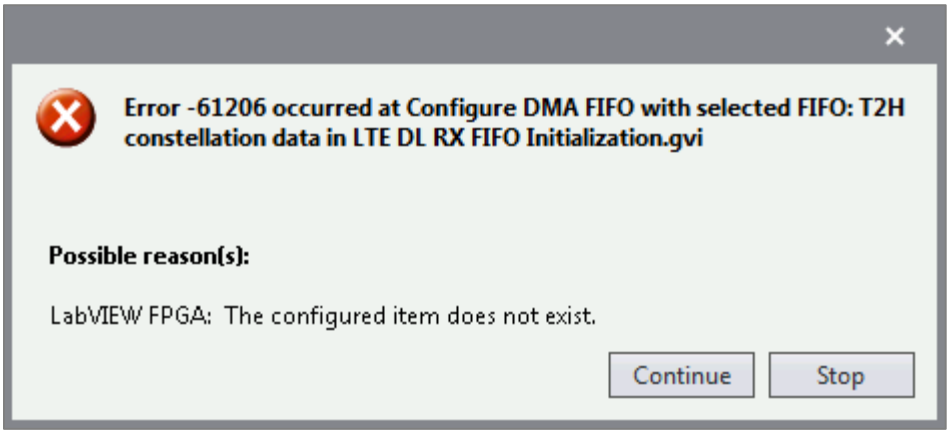

Figure 38 - Error -61206 occurred because of incorrect FIFO resource name

STEP-BY-STEP INSTRUCTIONS

1. Open the affected VI. In this example, we are demonstrating changes needed for the "LTE DL RX FIFO Initialization.gvi" as shown in Figure 39.

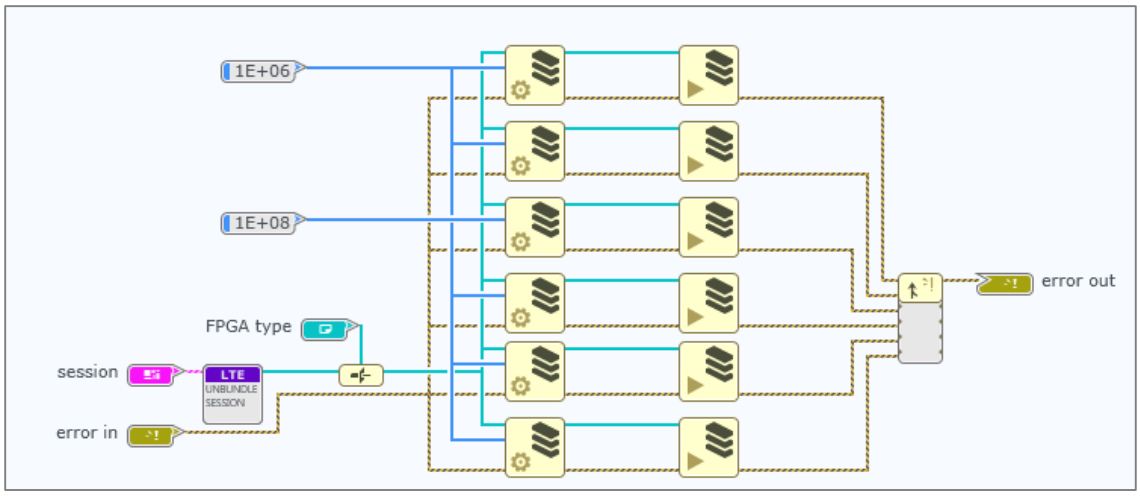

Figure 39 – "LTE DL RX FIFO Initialization.gvi" (original file)

2. **(Recommended)** Add comments above the affected nodes that note the currently configured resource name as shown in Figure 40.

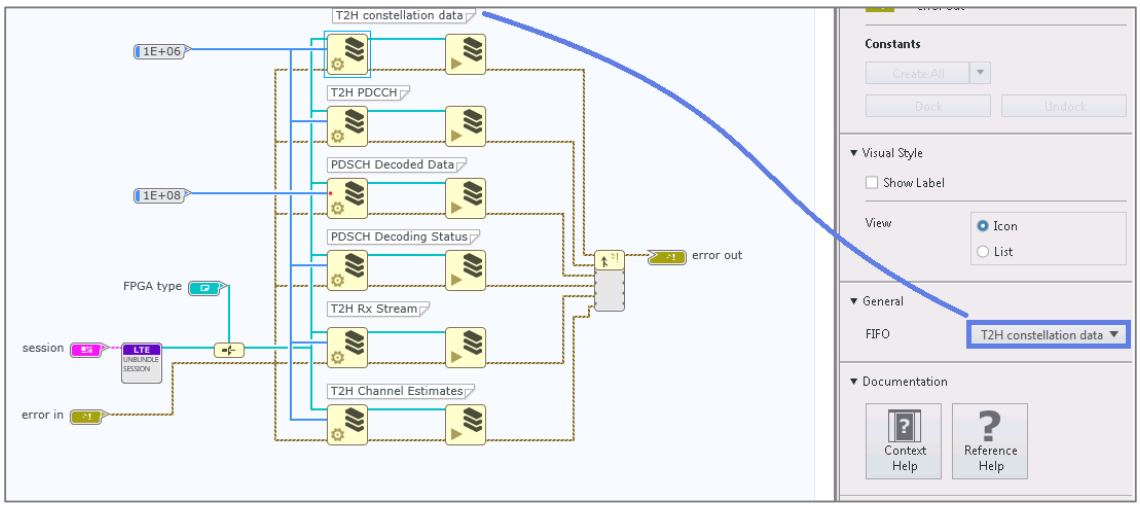

Figure 40 – "LTE DL RX FIFO Initialization.gvi" (after adding FIFO name comments)

3. Select the "FPGA type" constant and click **Configure** in the right-hand rail as shown in Figure 41.

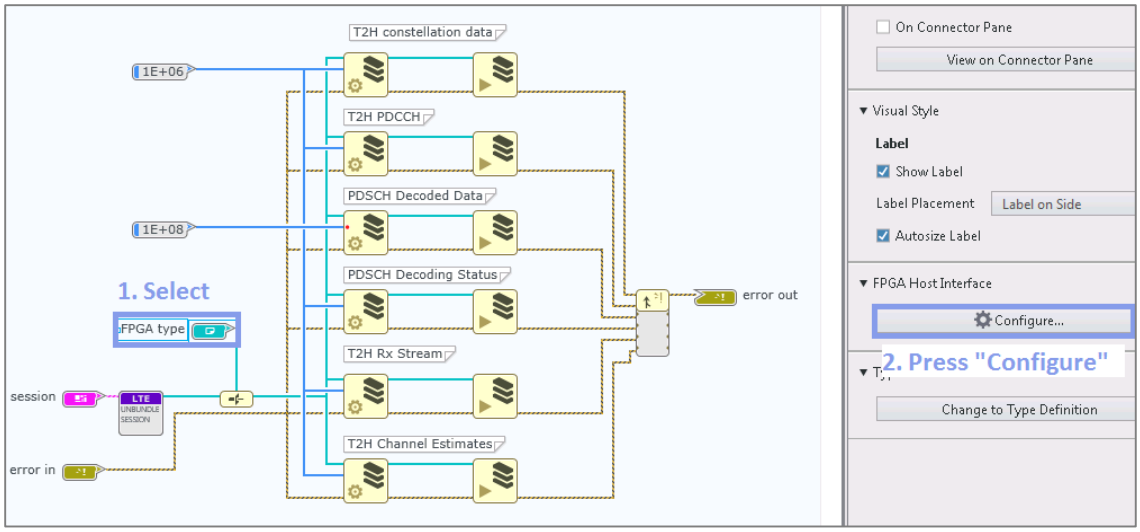

Figure 41 – "LTE DL RX FIFO Initialization.gvi" - Reconfigure FPGA Type

4. In the "FPGA Interface Dynamic Refnum Configuration" dialog, click **Import bitfile** and navigate to the bitfile you just built in the last step of the second section, *Migrate to the New USRP Interface*.
5. **(Optional)** With the "FPGA Interface Dynamic Refnum Configuration" dialog still open, remove all unused resources by clicking the **X** symbol on the right side. This will not affect functionality but may make it easier to view used resources at a later time.
6. Click **OK** in the "FPGA Interface Dynamic Refnum" dialog.
7. Select the first FIFO node and adopt the FIFO name as shown in Figure 42.

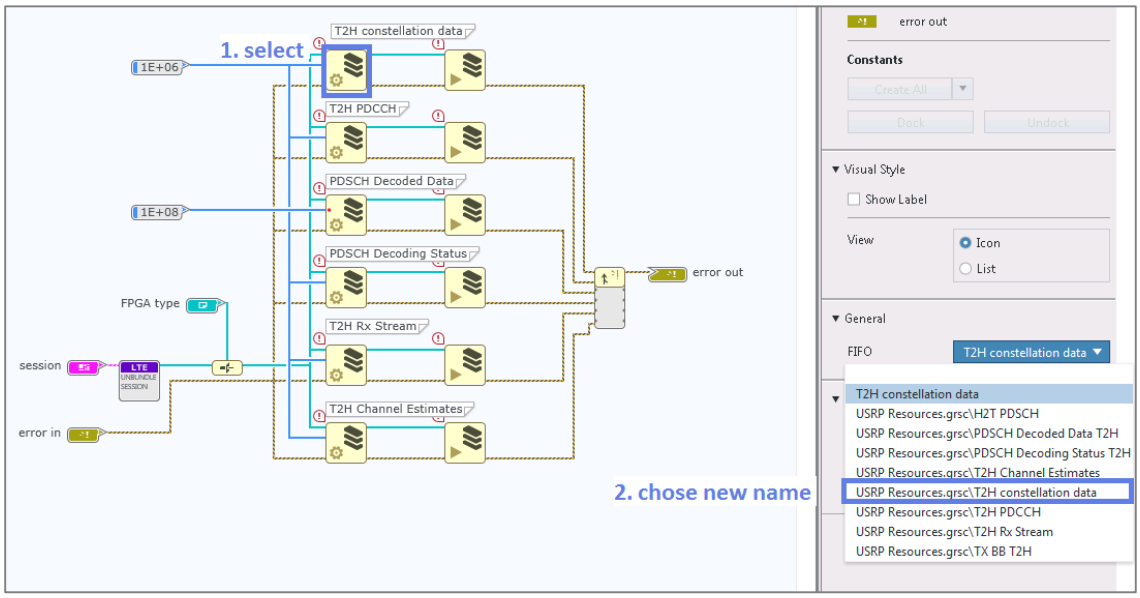

Figure 42 – "LTE DL RX FIFO Initialization.gvi" - Adopt FIFO name

8. Repeat the previous step for all FIFO nodes in the current VI.
   Caution: Make sure to select the correct FIFO. Double-check if the name you selected matches the comment you added to the document in step 2.
9. Repeat steps 1 through 8 for all VIs which contain a DMA FIFO node.

USRP PROJECTS

This section describes the steps necessary for reusing a project that was initially based on the USRP API and example projects in LabVIEW Communications 2.0.

This section includes the following subsections. Use the one that is most appropriate for the type of USRP project you are migrating.

- Host API
- PC/PXIe 40 MHz USRP RIO Single and Multi-Device
- PC/PXIe 120 MHz USRP RIO Single and Multi-Device
- SISO TDD OFDM on NI USRP RIO

NIUSRP API

HOST

The USRP host-side API has several changes. When migrating code, this may cause VIs to be broken of functionality to be lost. Review the following items and make updates to the migrated code as needed.

- "niUSRP Open Tx/Rx Session" and "niUSRP Close Session" had their error terminal placements moved down to be match the other API calls.

    1. Rewire any broken terminals as necessary.

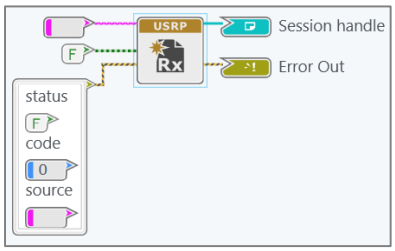

Figure 43: "niUSRP Open Rx Session.gvi" (1.1)

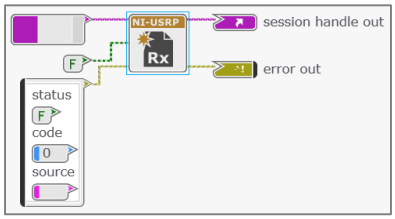

Figure 44: "niUSRP Open Rx Session.gvi" (2.0)

- Elements in the niUSRP property node will mutate to the correct new mappings during the conversion process. However, some data types have changed (see step "Enabled Channels" property below). Property nodes using the Active Channel filter must be rewired.

1. The property node filters in LabVIEW Communications 2.0 are located beneath the property node title as a dropdown menu ("All Active Channel") as shown in Figure 45.

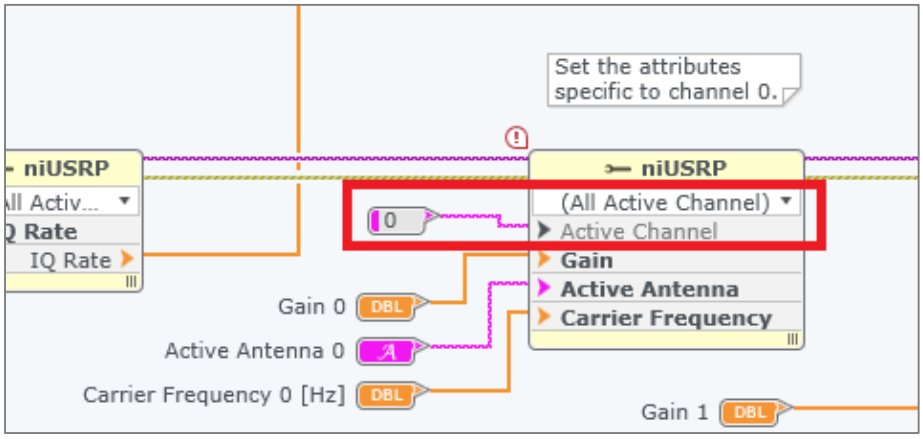

Figure 45: Active Channel Filter

2. To specify a channel, select **Wire on Diagram** from the property node dropdown menu and wire in the string. Delete the Active Channel element.

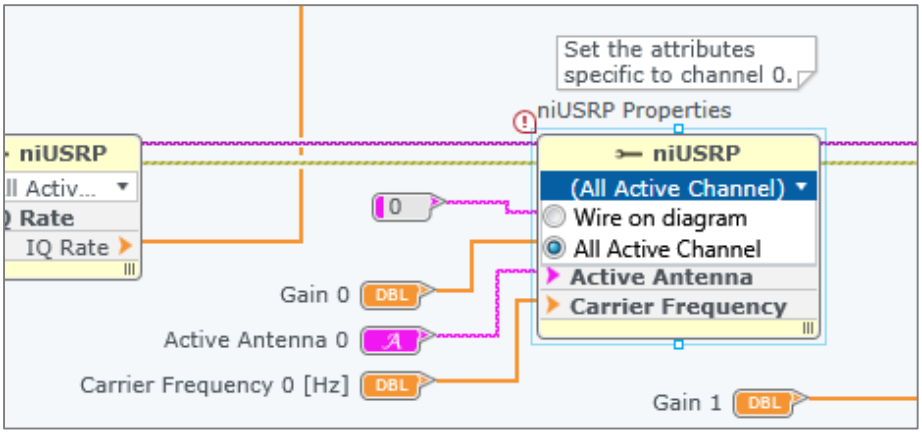

Figure 46: Drop Down Menu from Active Channel Filter

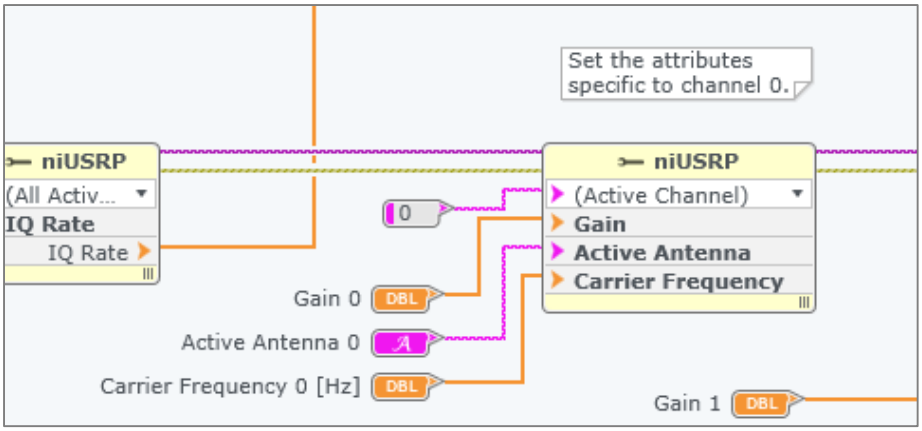

Figure 47: Corrected Active Channel Filter (No Longer Shows Exclamation or Error)

- Any subVI or cluster containing the previous niUSRP host session handles will be broken. The data type for the session handle was changed completely. Any instance of the session that exists between API calls or property nodes will load correctly. For reference, see Figure 44.

  1. To fix subVIs and clusters with the old session handle, the controls/indicators must be updated to use the new instance, and any VI terminals must be updated.

- The "Enabled Channels" property was changed from an enum to a string.

    1. The data type of any controls/constants wired to this property must be updated to a string.

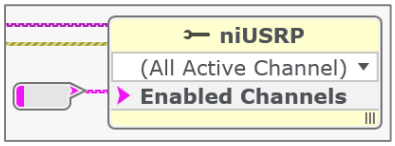

Figure 48: "Enabled Channels" Property (2.0)

- "niUSRP Fetch Rx Data (CDB WDT)" was updated to use the waveform data type rather than the previous replica type.

    1. Any files using a property node wired to the output may need to be updated.

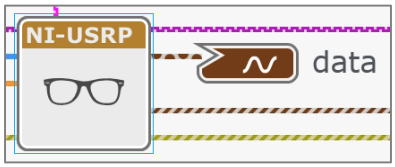

Figure 49: "niUSRP Fetch Rx Data (CDB WDT).gvi" (2.0)

FPGA AND SYSTEM RESOURCES

Because the USRP interface (40 MHz BW) is deprecated with LabVIEW Communications 2.0, the new interface must now be used on the FPGA top-level. The new interface now handles both the 40 MHz BW and 120 MHz BW USRP models with the same interface. Hence, an FPGA bitfile rebuild is required.

1. Create a backup of your project.
2. Open your project in LabVIEW Communications 2.0.
3. Click **Yes** to convert as shown in Figure 50.

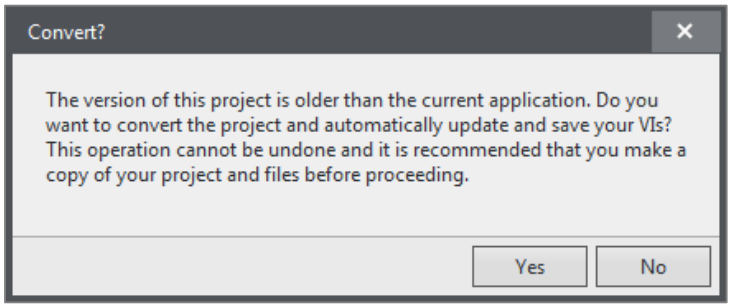

Figure 50: Convert Dialog

4. Delete the following items from "USRP Resources.grsc":
   a. radio.responses 0
   b. radio.responses 1
   c. reg.host instruction fifo 0
   d. reg.host instruction fifo 1
   e. reg.host instruction fifo 2

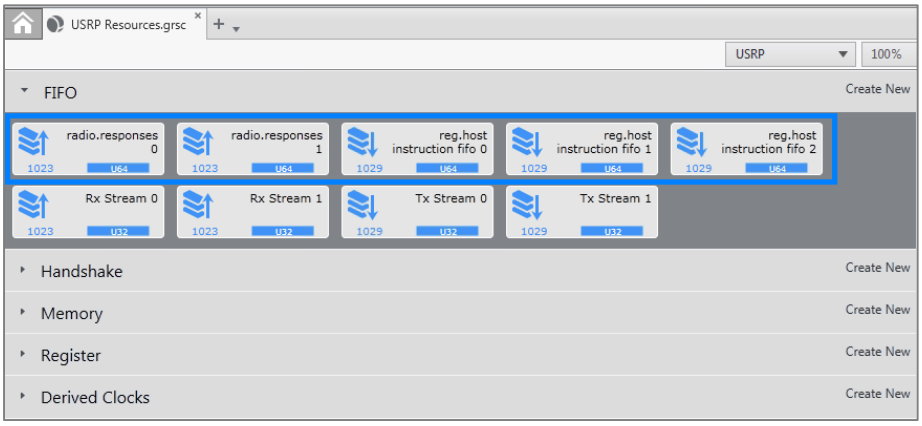

Figure 51: Files to Delete from "USRP Resources.grsc"

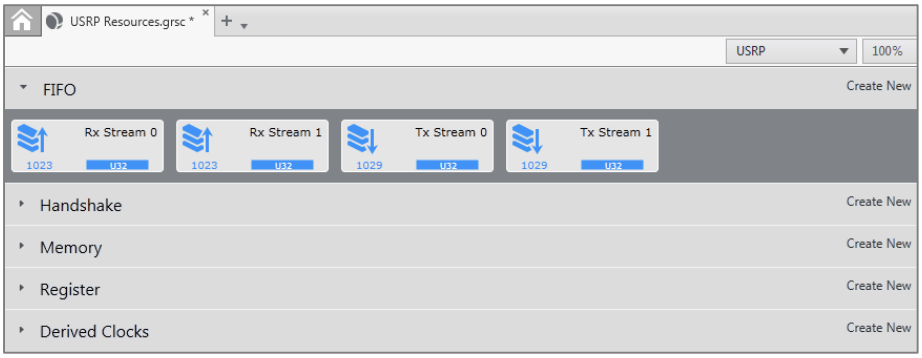

Figure 52: Final "URSP Resources.grsc"

5. Select your USRP Target on the System Designer and change the "Model" in the right-hand rail to the non-deprecated version (ex: NI 2953R USRP (40 MHz BW) Deprecated » NI 2953R USRP (40 MHz BW)) as shown in Figure 53.

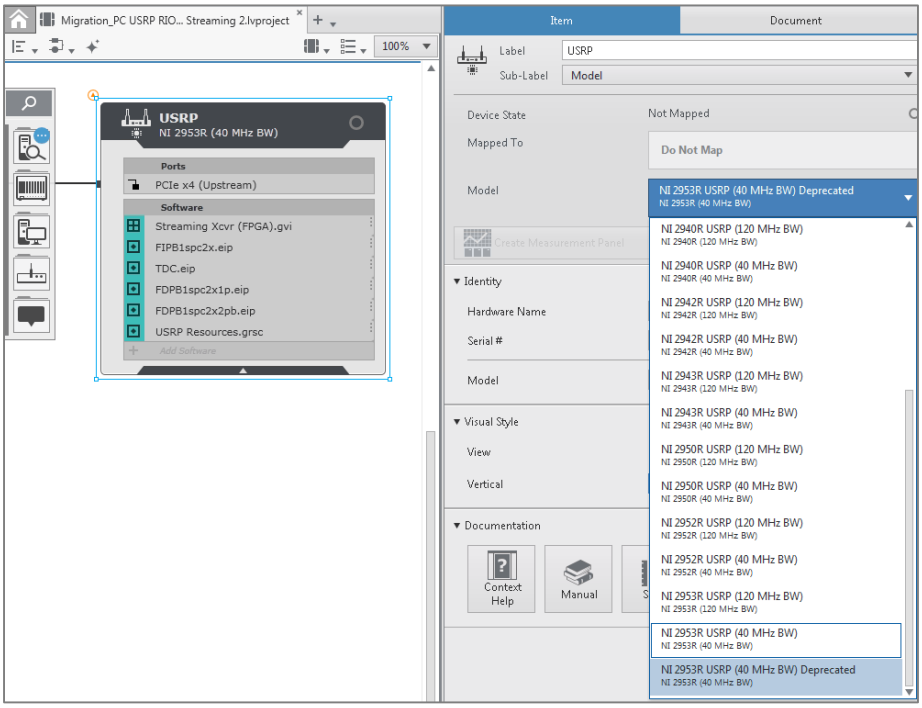

Figure 53: Select USRP Target from Right-Hand Rail

6. Generate a new 40 MHz Single-Device Streaming project in LabVIEW Communications 2.0 by navigating to New » Project Templates » USRP RIO 40 MHz BW Single-Device Streaming.
Note: This project will only be used to copy the necessary project files and code sections into the previous project that is being migrated.

7. Copy two VIs from the new project into the original project.
   a. As shown in Figure 54, find the following subVIs from the *<project directory>\FPGA SubVIs* folder:
      i. Create Resources.gvi
      ii. Registers.gcdl

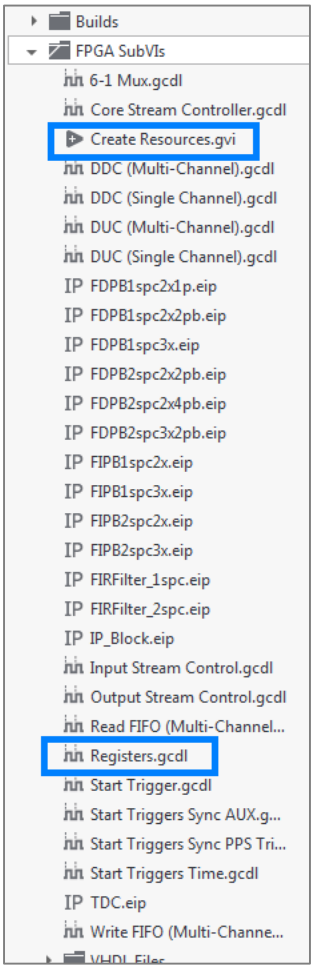

Figure 54: Files to Copy

   b. Copy these on disk and paste to the same directory in the original project.

c. In the "Project Files" view on the Navigation Pane of the original project, the new files will appear slightly greyed out. Right-click on each and select **Include** as shown in Figure 55.
Note: Since the "Create Resources.gvi" is not a specific FPGA document (for example, CDL, MRD, and so on), there may be a prompt to select the target that the file should reside under. Select the same USRP target that your other files in this directory reside under.

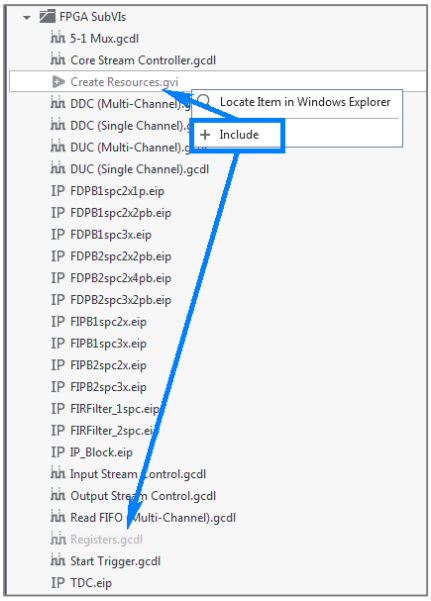

Figure 55: Files to Include

8. In your original project, open the "Streaming Xcvr (FPGA).gvi".

9. Find the three Clock-Driven Loops used for interfacing with the USRP and remove those loops as shown in Figure 56.

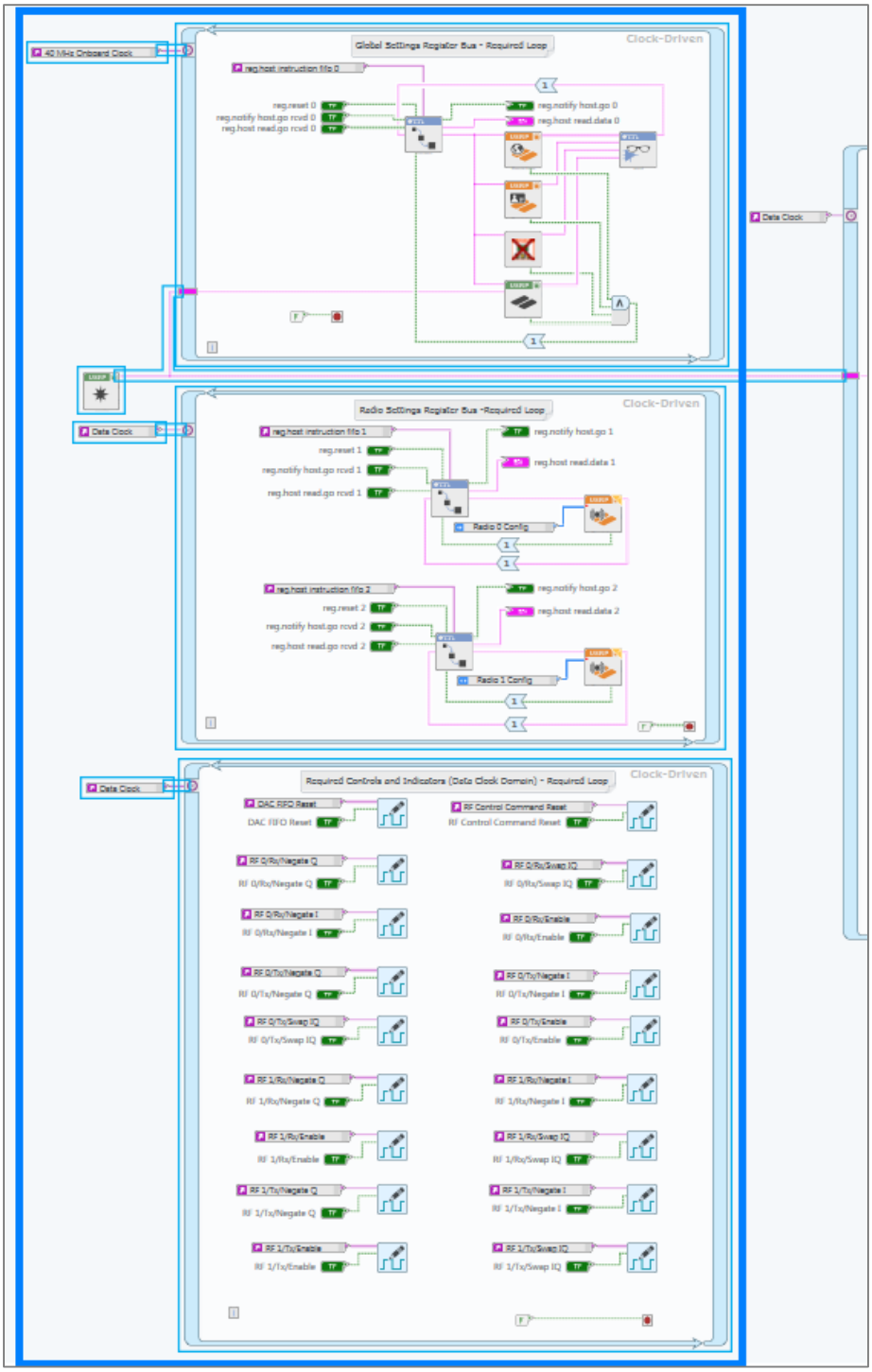

Figure 56: Clock Driven Loops

10. Open the “Streaming Xcvr (FPGA).gvi” of the new project.
    a. As shown in Figure 57, select and copy the following from the Diagram: “Create Resources.gvi”, “Process.gvi”, the Clock Constant, and the Clock-Driven Loop.

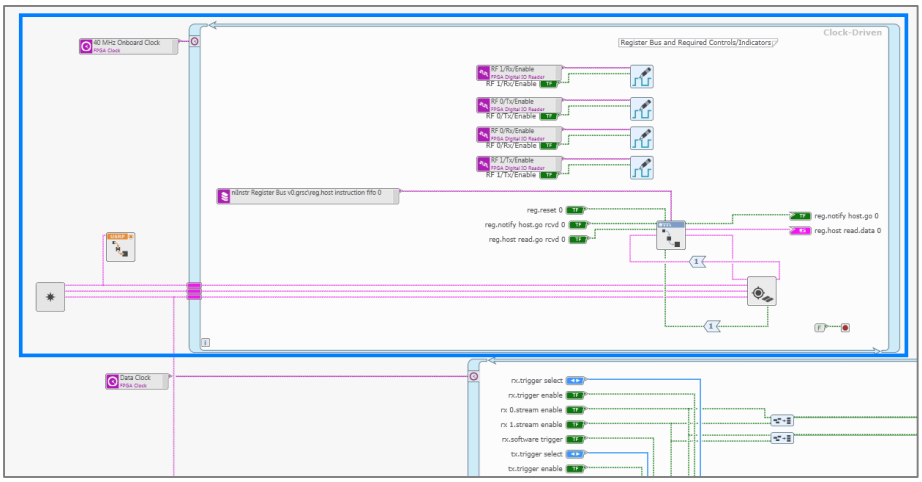

Figure 57: Clock-Driven Loop to Copy From

11. In the project you wish to migrate, paste the elements you copied from the new streaming project.
    a. Rewire the “sync.resources” cluster wire from the “Create Recources.gvi” to the “Start Trigger.gcdl” as shown in Figure 58.

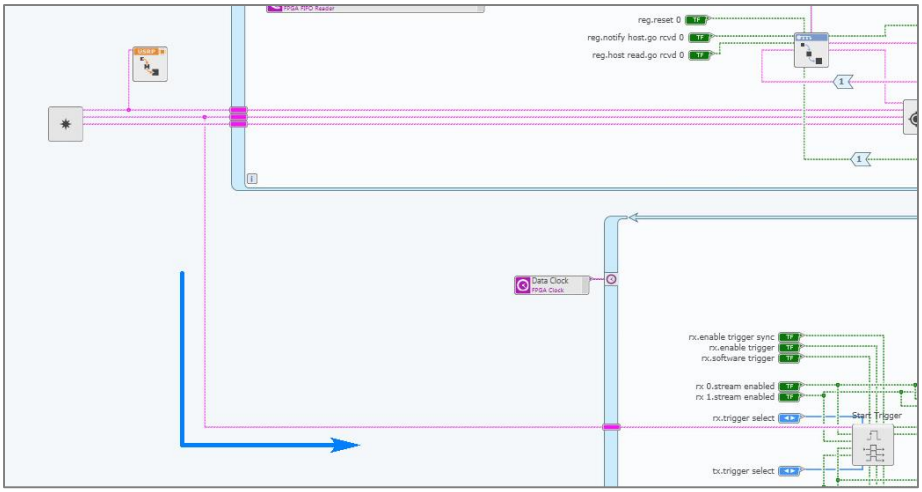

Figure 58: Rewire Cluster Wire to Start Trigger

12. Allow all document changes to finish loading and save all files.
13. Close the new streaming project as it is not needed anymore.

14. Compile the FPGA VI.
    a. Open the System Designer (in the "Project Files" view of the Navigation Pane on the left) by double-clicking on the top project file.
    b. Select the FPGA top-level under the target you want to build the bitfile for.
    c. Click **Build** on the right pane as shown in Figure 59.

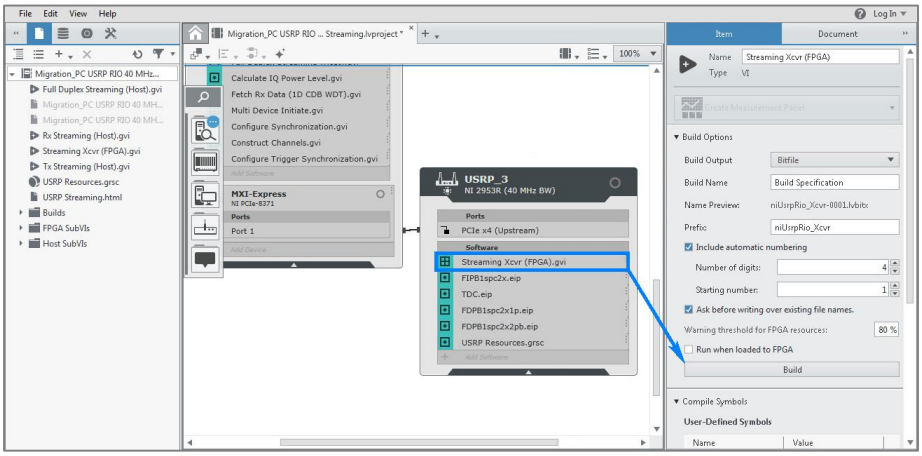

Figure 59: Compile the Project

15. **(Recommended)** Once the bitfile is finished building, exclude it from the project as shown in Figure 60 and include it again. This will ensure that the latest changes are actually applied.

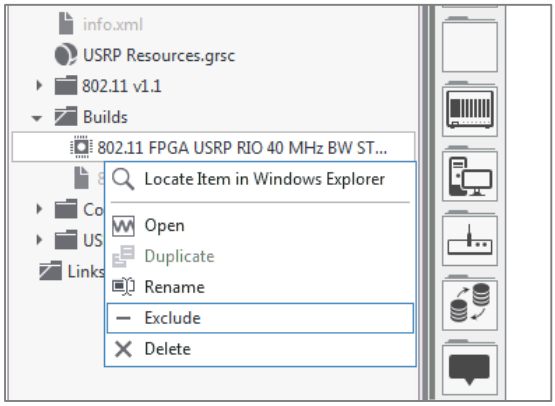

Figure 60: Option to Exclude Bitfile

HOST-SIDE

The FIFO resource names in the Host subVIs that interface with DMA FIFOs must be updated. This includes VIs which aren't in VI.lib (Sample Project VIs and User VIs). In LabVIEW Communications 2.0, the FIFO resource contains additional prefixes for the FPGA resource file name along with the FIFO name itself. For example, "Rx Stream 0" becomes "USRP Resources.grsc\Rx Stream 0." All FIFO node instances (Start/Stop, Configure, and Read/Write), along with any FPGA reference terminals, must be updated.

Use your newly compiled bitfile from step 14 above to get access to the DMA FIFO references required.

1. Open the "Configure Stream.gvi" in the *<project directory>\Host SubVIs* folder of your project.
    a. Select the "ConstantData4" terminal and in the right-hand rail click **Configure**.

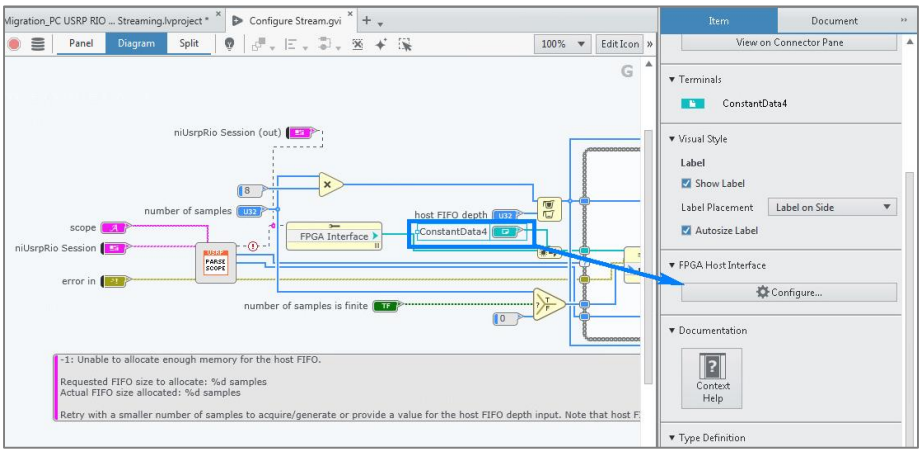

Figure 61: Right-Hand Rail Configure Option

i. In the "FPGA Interface Dynamic Refnum Configuration" dialog, click **Import bitfile** and browse to your recently compiled bitfile.

   Note: The default location will be in the *<project directory>\Builds* folder but this, along with the file name, will vary based on the configuration of the build spec used during compile.

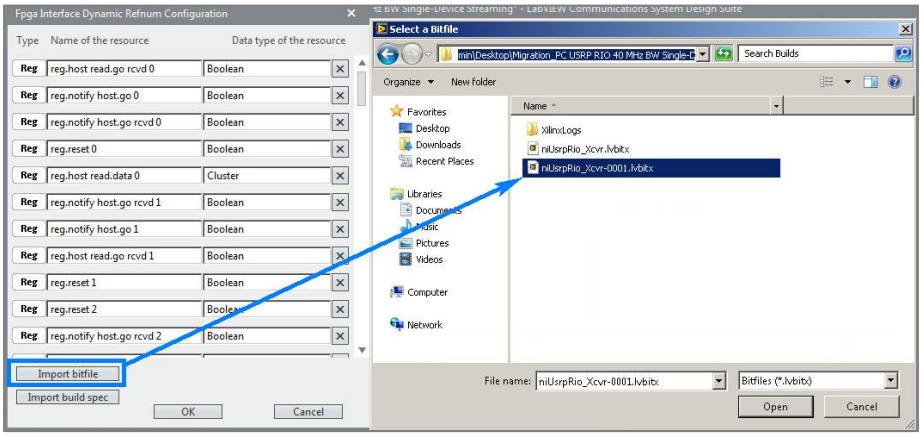

Figure 62: Import Bitfile

ii. **(Optional)** With the "FPGA Interface Dynamic Refnum Configuration" dialog still open, remove all unused resources by clicking the **X** symbol on the right side.
   Note: This will not affect functionality but may make it easier to view used resources at a later time.

iii. Click **OK** in the "FPGA Interface Dynamic Refnum" dialog.

b. Update to the new DMA FIFO's in each Case Structure for each “rx” and “tx” case that contains a nested Case Structure for Stream, 0 and 1. Each case will contain one, or more, FIFO nodes of a Start DMA FIFO, Stop DMA FIFO, and Configure DMA FIFO. Using the right-hand rail configuration, update each FIFO node to the following:
    i. Rx Stream 0 » USRP Resources.grsc\Rx Stream 0
    ii. Rx Stream 1 » USRP Resources.grsc\Rx Stream 1
    iii. Tx Stream 0 » USRP Resources.grsc\Tx Stream 0
    iv. Tx Stream 1 » USRP Resources.grsc\Tx Stream 1

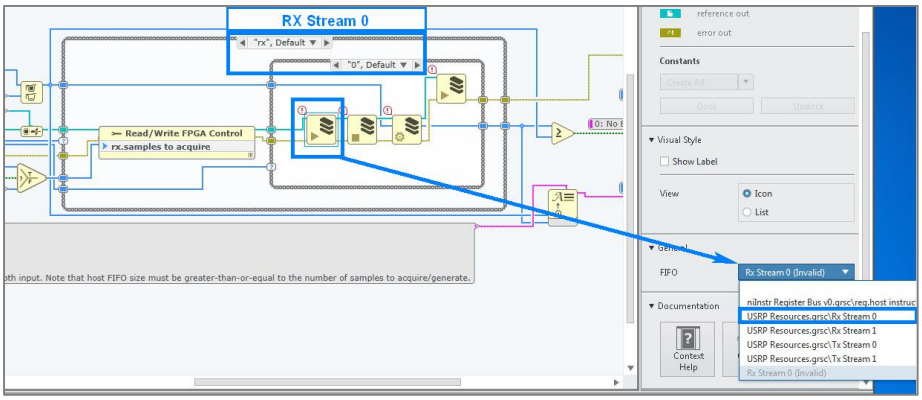

Figure 63: Update DMA FIFOs

2. Follow the same general steps from step 1 for the "Fetch Rx Data (U32).gvi" in the *<project directory>\Host SubVIs* folder of your project
    a. Select "ConstantData3" and click **Configure** in the right-hand rail as shown in Figure 64.

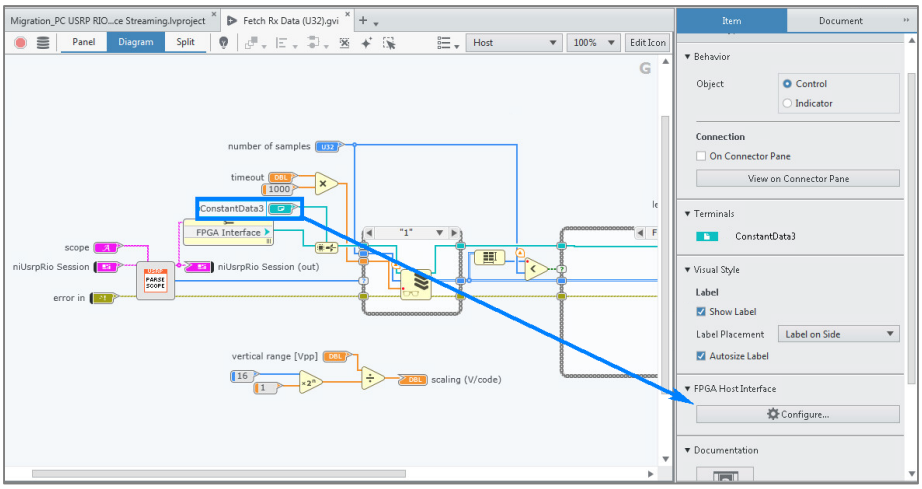

Figure 64: Configure Dialog in Right-Hand Rail

b. Click **Import bitfile** and browse to your recently compiled bitfile. Click **OK**.
c. Update the left-most Case Structure's Read DMA FIFO to point at the new DMA FIFO's for both cases. Change this by going to the right-hand rail configuration. Update the following:
    i. Rx Stream 0 » USRP Resources.grsc\Rx Stream 0
    ii. Rx Stream 1 » USRP Resources.grsc\Rx Stream 1

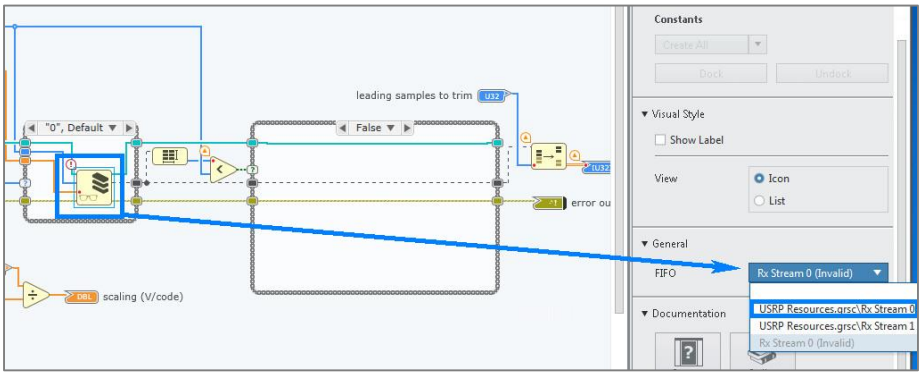

Figure 65: Update Read DMA FIFO

3. Follow the same general steps from step 1 and step 2 for the "Write Tx Data (U32).gvi" in the *<project directory>\Host SubVIs* folder of your project.
    a. Select "ConstantData2" and click **Configure** in the right-hand rail.

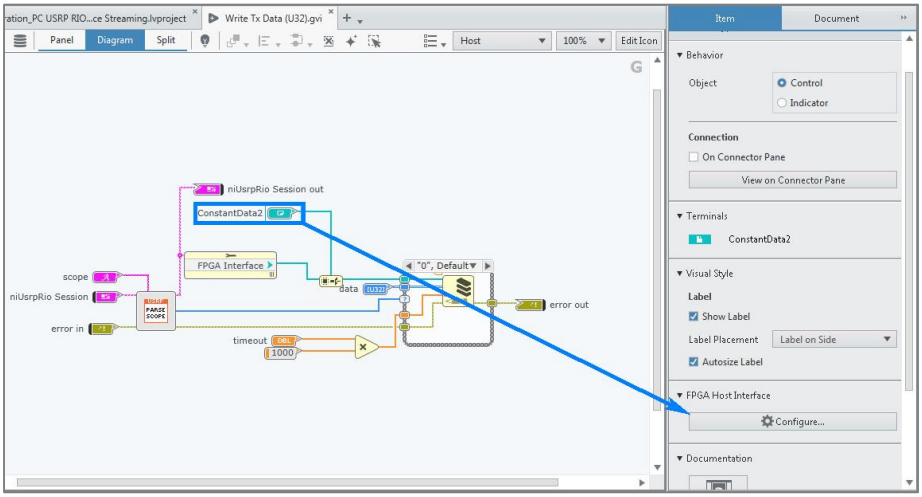

Figure 66: Configure Dialog in Right-Hand Rail

b. Click **Import bitfile** and browse to your recently compiled bitfile. Click **OK**.

c. Update the Case Structure's Write DMA FIFO to point at the new DMA FIFO's for each case. Change this by going to the right-hand rail configuration. Update the following:
    i. Tx Stream 0 » USRP Resources.grsc\Tx Stream 0
    ii. Tx Stream 1 » USRP Resources.grsc\Tx Stream 1

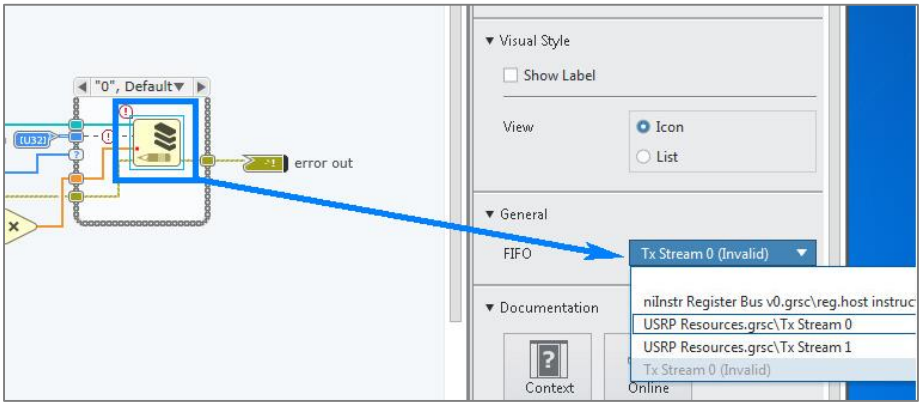

Figure 67: Update Write DMA FIFO

Note: Similar changes must be made if any DMA FIFO's were added to the project in previous versions of LabVIEW Communications.

FPGA AND SYSTEM RESOURCES

1. Create a backup of your project.
2. Open your project in LabVIEW Communications 2.0.
3. Click **Yes** to convert as shown in Figure 68.

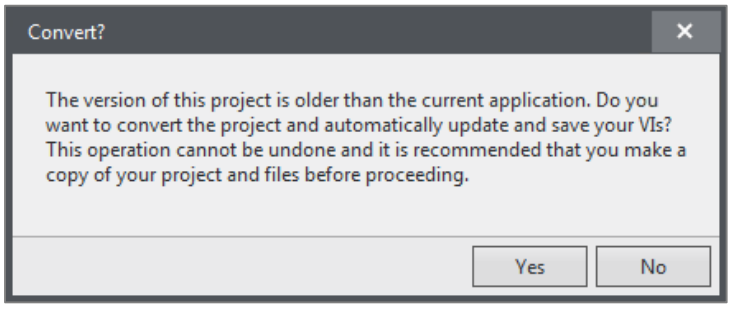

Figure 68: Convert Dialog

4. Delete the following items from "USRP Resources.grsc":
    a. radio.responses 0
    b. radio.responses 1
    c. reg.host instruction fifo 0
    d. reg.host instruction fifo 1
    e. reg.host instruction fifo 2

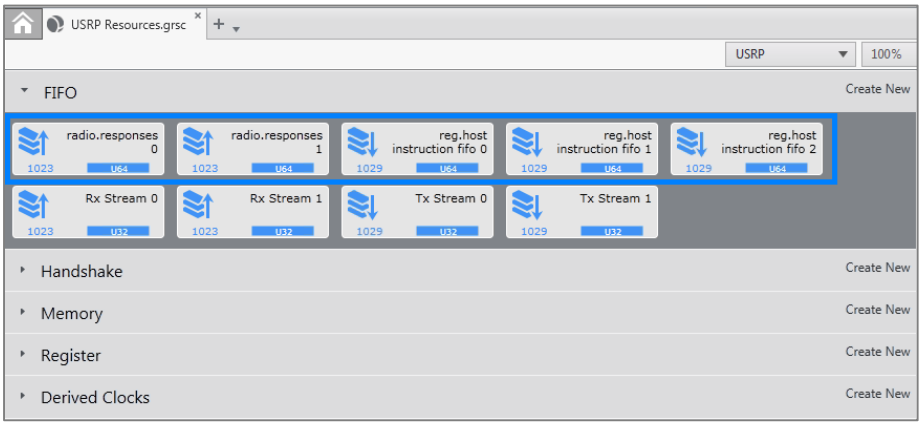

Figure 69: Items to Delete from "USRP Resources.grsc"

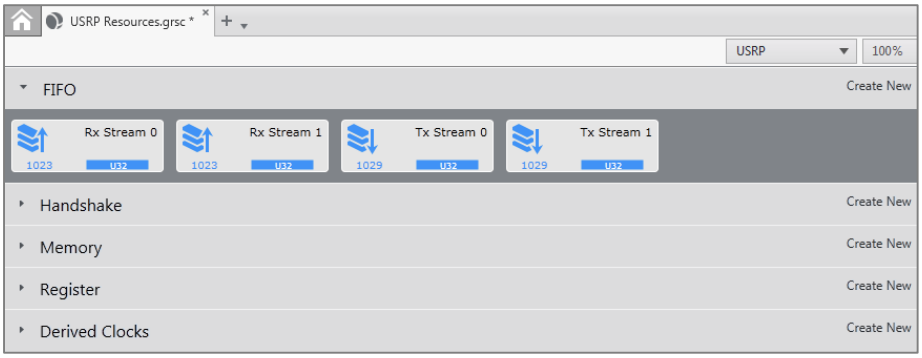

Figure 70: Corrected "USRP Resources.grsc"

5. Open the "Streaming Xcvr (FPGA).gvi" of the new project.
6. Change the DMA FIFO reference going into the "Process.gcdl" by clicking the IO Constant drop-down. From that drop-down menu, select "niInstr Register Bus v0.grsc\reg.host instruction fifo 0".

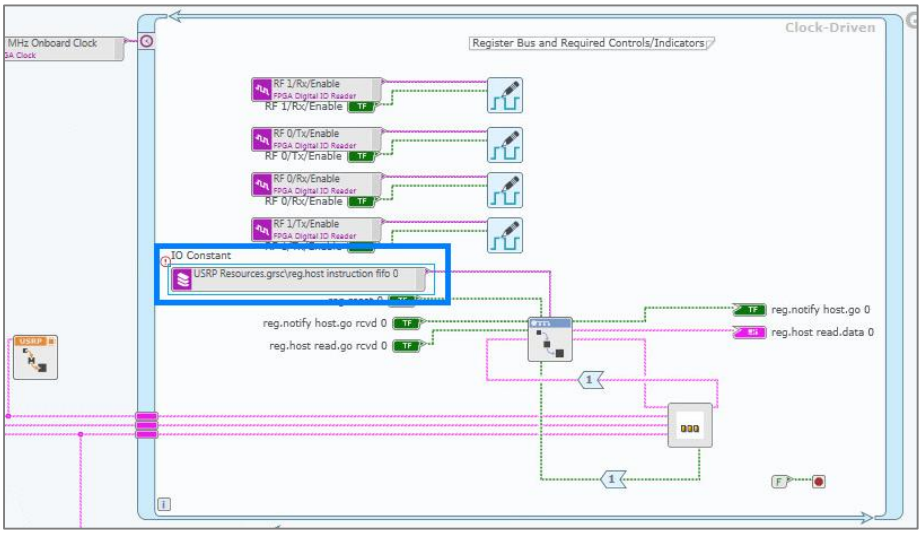

Figure 71: Change DMA FIFO Reference

7. Compile the FPGA VI.
    a. Open the System Designer (in the "Project Files" view of the Navigation Pane on the left) and double-click on the top-most project file.
    b. Select the FPGA top-level under the target you want to build the bitfile for.
    c. Click **Build** on the right-hand rail as shown in Figure 72.

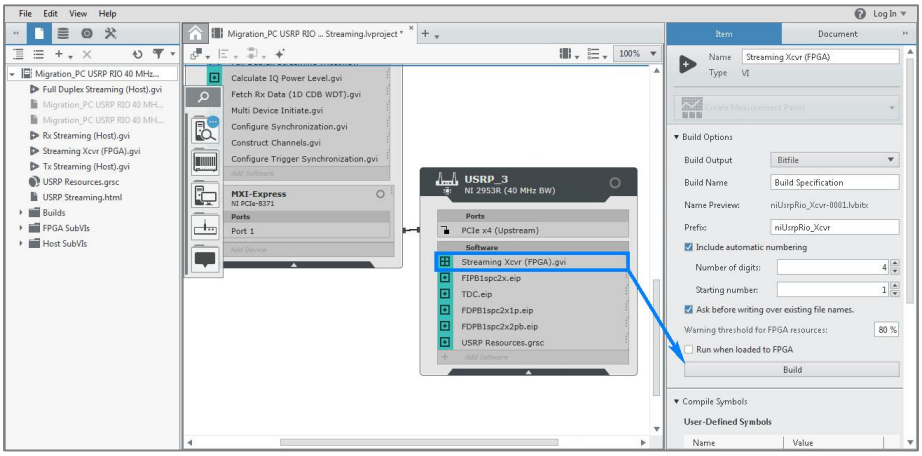

Figure 72: Compile FPGA VI by Clicking Build

8. **(Recommended)** Once the bitfile is finished building, exclude it from the project as shown in Figure 73 and include it again. This will ensure that the latest changes are actually applied.

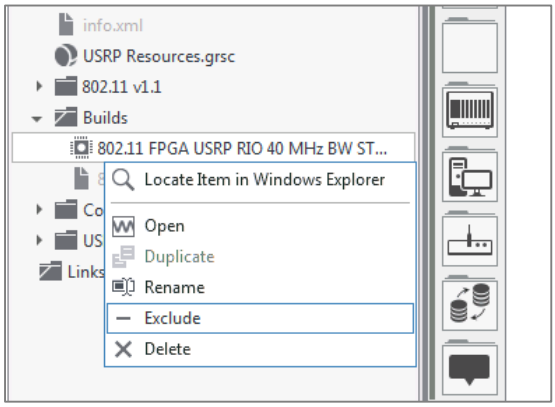

Figure 73: Exclude File

HOST-SIDE

Follow the same steps listed in the PC/PXIe 40 MHz USRP RIO Single and Multi-Device Host-Side section.

FPGA AND SYSTEM RESOURCES

Because the USRP interface (40 MHz BW) is deprecated with LabVIEW Communications 2.0, the new interface must now be used on the FPGA top-level. The new interface now handles both the 40 MHz BW and 120 MHz BW USRP models with the same interface. Hence, an FPGA bitfile rebuild is required.

1. Create a backup of your project.
2. Open your project in LabVIEW Communications 2.0.
3. Click **Yes** to convert as shown in Figure 74.

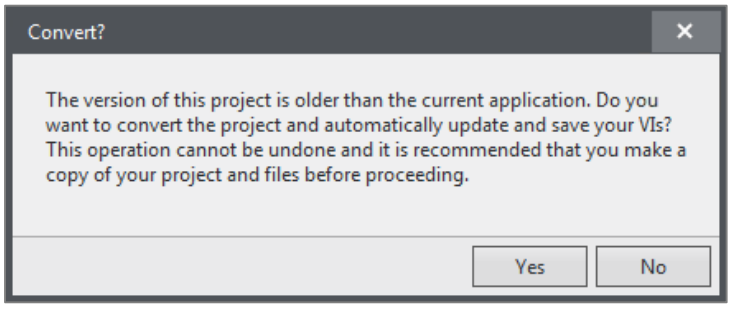

Figure 74: Convert Dialog

4. Delete the following items from "USRP Resources.grsc":
   a. radio.responses 0
   b. radio.responses 1
   c. reg.host instruction fifo 0
   d. reg.host instruction fifo 1
   e. reg.host instruction fifo 2

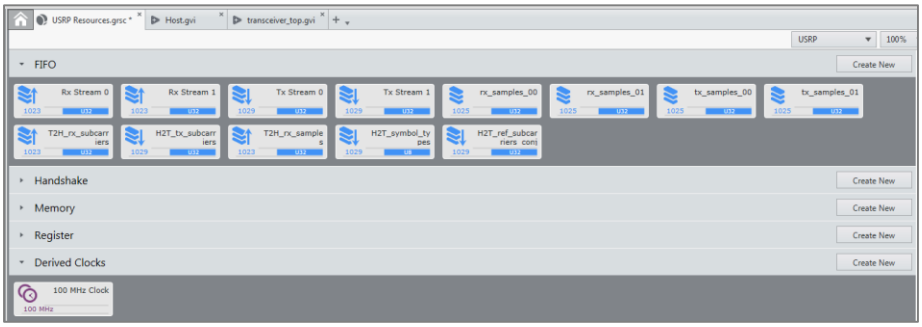

Figure 75: Final "URSP Resources.grsc"

5. Select your USRP Target on the System Designer and change the "Model" in the right-hand rail to the non-deprecated version (ex: NI 2953R USRP (40 MHz BW) Deprecated » NI 2953R USRP (40 MHz BW)) as shown in Figure 76.

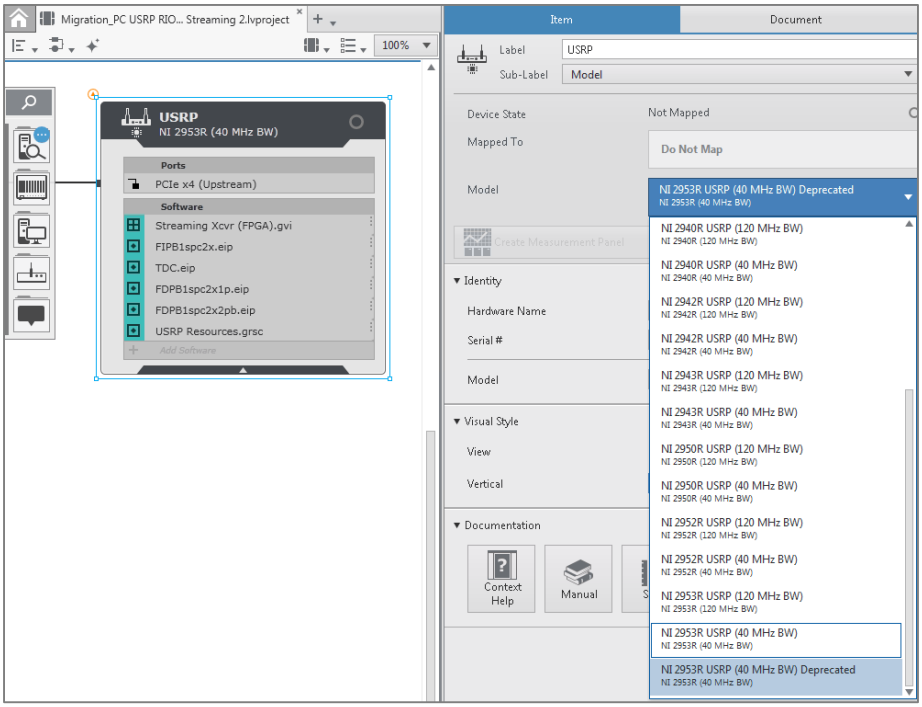

Figure 76: Select USRP Target from Right-Hand Rail

6. Generate a new SISO TDD OFDM project by navigate to Learning » Hardware Input and Output » NI USRP RIO » SISO TDD OFDM on NI USRP RIO.
   Note: This project will only be used to copy the necessary project files and code sections into the previous project that is being migrated.

7. Copy two VIs from the new project into the original project.
   a. As shown in Figure 77, find the following subVIs from the *<project directory>\PXIe USRP RIO Streaming \FPGA SubVIs* folder:
      i. Create Resources.gvi
      ii. Registers.gcdl

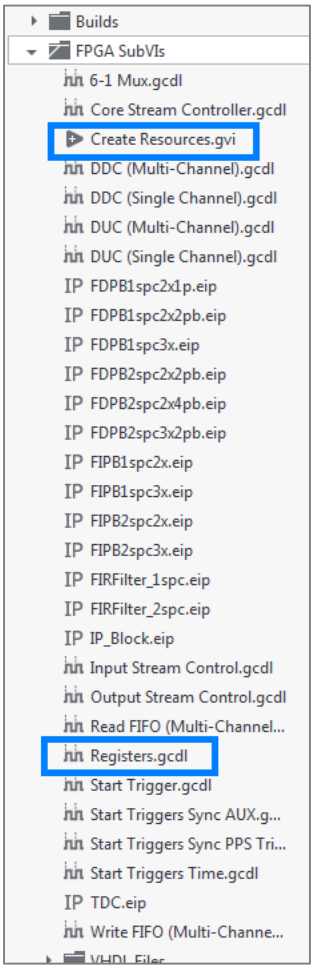

Figure 77: Files to Copy

   b. Copy these on disk and paste to the directory in the original project: *<project directory>\PXIe USRP RIO Streaming\FPGA SubVIs*

c. In the "Project Files" view on the Navigation Pane of the original project, the new files will appear slightly greyed out. Right-click on each and select **Include** as shown in Figure 78.
Note: Since the "Create Resources.gvi" is not a specific FPGA document (for example, CDL, MRD, and so on), there may be a prompt to select the target that the file should reside under. Select the same USRP target that your other files in this directory reside under.

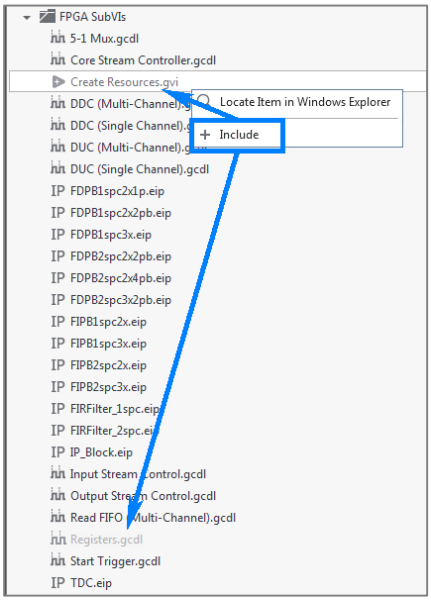

Figure 78: Files to Include

8. In the original project, open the "transceiver_top.gvi".

9. Find the three Clock-Driven Loops used for interfacing with the USRP and remove those loops as shown in Figure 79.

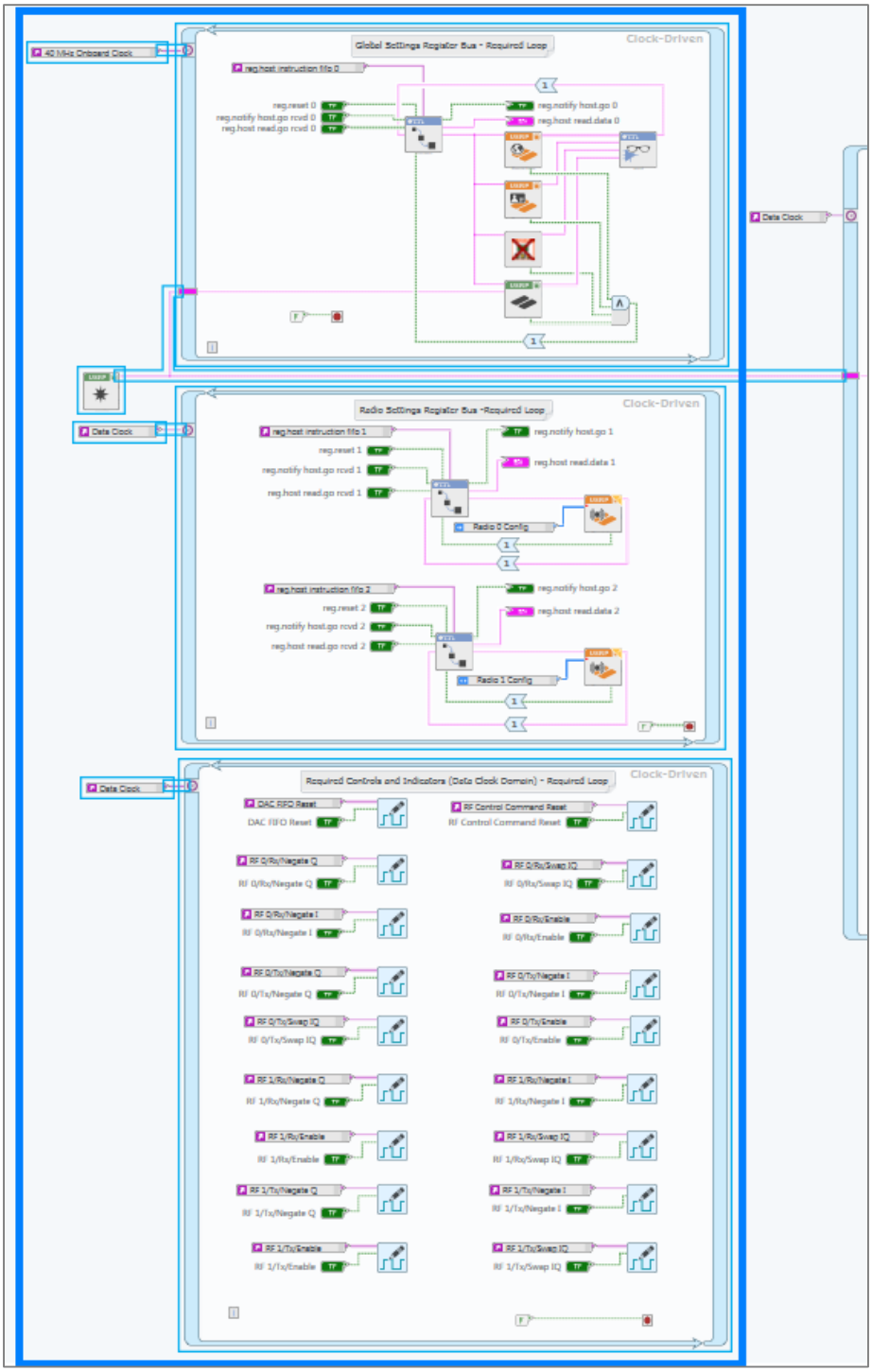

Figure 79: Clock Driven Loops

10. Open the "Streaming Xcvr (FPGA).gvi" of the new project.
    a. As shown in Figure 80, select and copy the following from the Diagram: "Create Resources.gvi", "Process.gvi", the Clock Constant, and the Clock-Driven Loop.

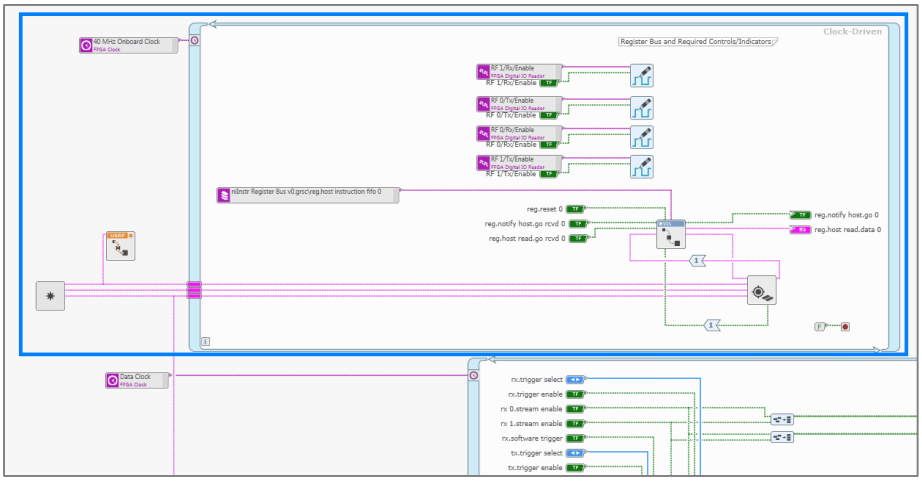

Figure 80: Clock-Driven Loop to Copy From

11. In the project you wish to migrate, paste the elements you copied from the new sample streaming project.
    a. Rewire the "sync.resources" cluster wire from the "Create Recources.gvi" to the "Start Trigger.gcdl" as shown in Figure 81.

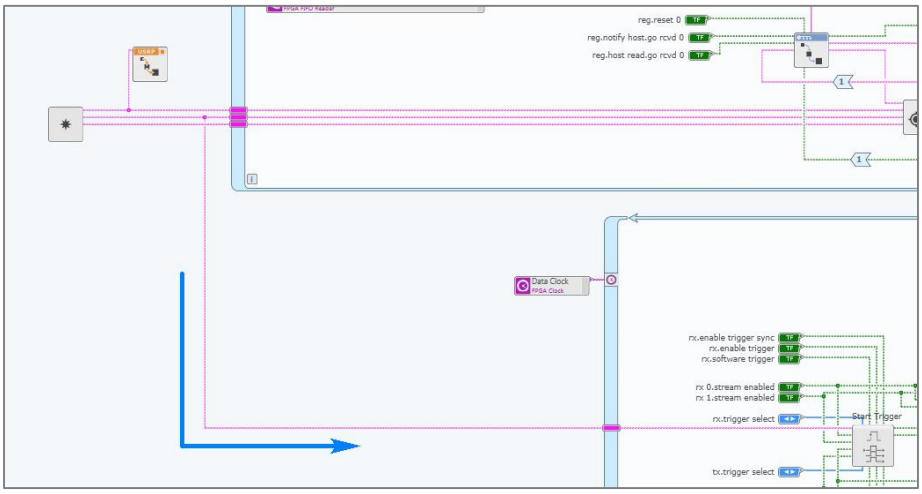

Figure 81: Rewire Cluster Wire to Start Trigger

12. Allow all document changes to finish loading and save all files.

13. Close the new project as it is not needed anymore.
14. Compile the FPGA VI.
    a. Open the System Designer (in the "Project Files" view of the Navigation Pane on the left) by double-clicking on the top project file.
    b. Select the FPGA top-level under the target you want to build the bitfile for.
    c. Click **Build** in the right-hand rail as shown in Figure 82.

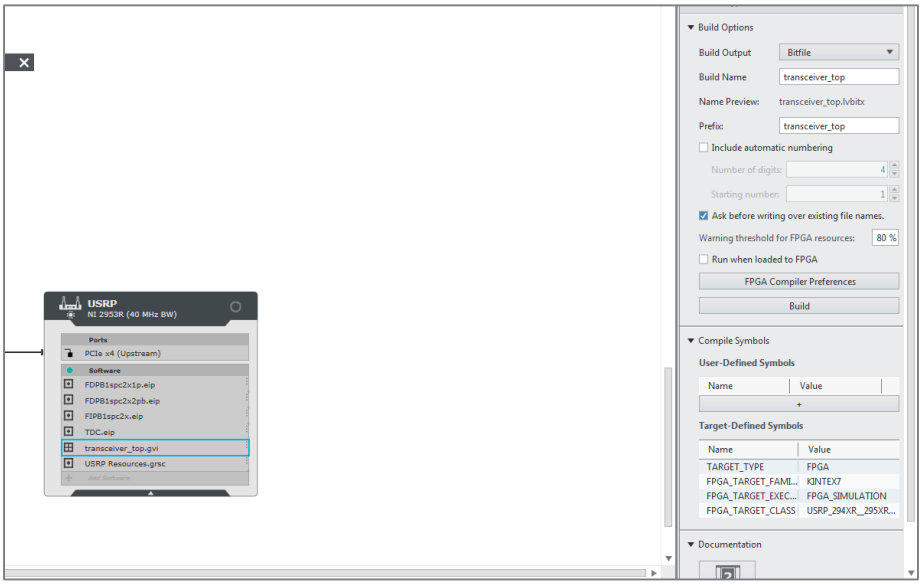

Figure 82: Compile the Project

15. **(Recommended)** Once the bitfile is finished building, exclude it from the project as shown in Figure 83 and include it again. This will ensure that the latest changes are actually applied.

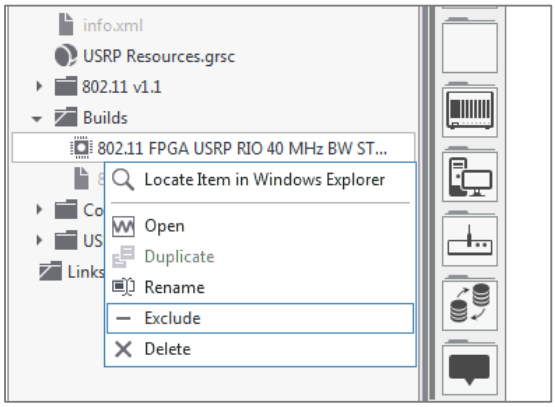

Figure 83: Option to Exclude Bitfile

HOST-SIDE

The FIFO resource names in the Host subVIs that interface with DMA FIFOs must be updated. This includes VIs which aren't in VI.lib (Sample Project VIs and User VIs). In LabVIEW Communications 2.0, the FIFO resource contains additional prefixes for the FPGA resource file name along with the FIFO name itself. For example, "Rx Stream 0" becomes "USRP Resources.grsc\Rx Stream 0." All FIFO node instances (Start/Stop, Configure, and Read/Write), along with any FPGA reference terminals, must be updated.

Use your newly compiled bitfile from step 14 above to get access to the DMA FIFO references required.

1. Open the "Configure Stream.gvi" in the *<project directory>\Host SubVIs* folder of your project.
    a. Select the "ConstantData4" terminal and in the right-hand rail click **Configure**.

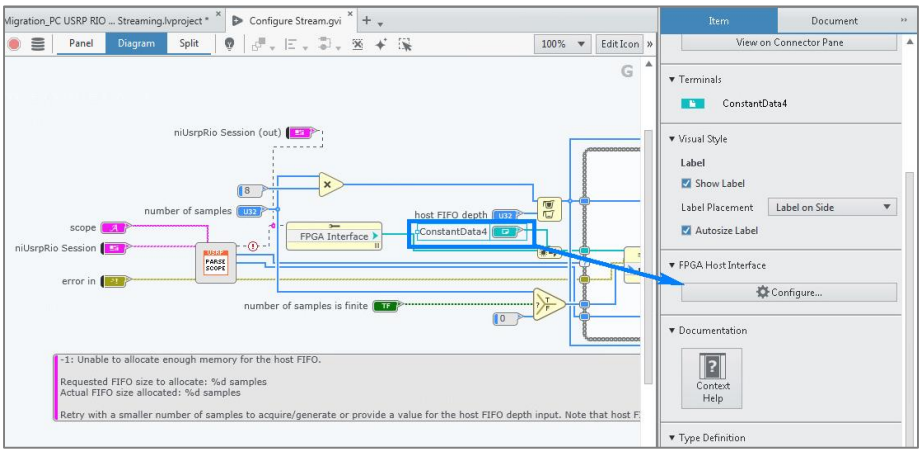

Figure 84: Right-Hand Rail Configure Option

i. In the "FPGA Interface Dynamic Refnum Configuration" dialog, click **Import bitfile** and browse to your recently compiled bitfile.

   Note: The default location will be in the *<project directory>\Builds* folder but this, along with the file name, will vary based on the configuration of the build spec used during compile.

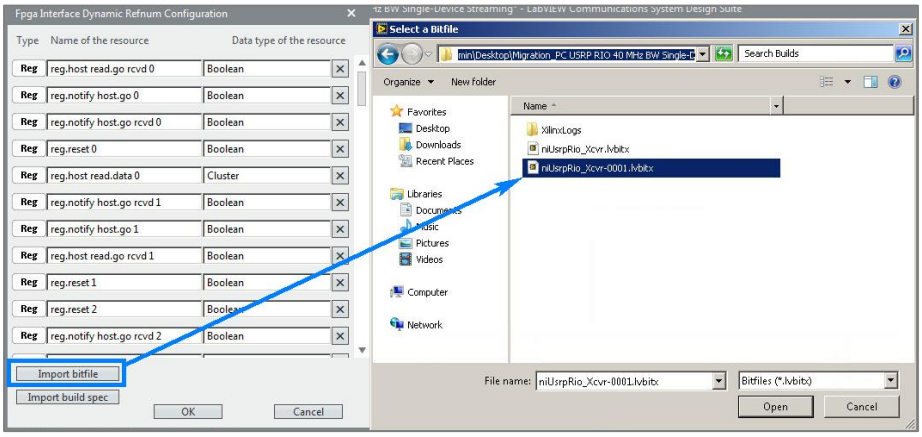

Figure 85: Import Bitfile

ii. **(Optional)** With the "FPGA Interface Dynamic Refnum Configuration" dialog still open, remove all unused resources by clicking the **X** symbol on the right side.
   Note: This will not affect functionality but may make it easier to view used resources at a later time.

iii. Click **OK** in the "FPGA Interface Dynamic Refnum" dialog.

b. Update to the new DMA FIFO's in each Case Structure for each “rx” and “tx” case that contains a nested Case Structure for Stream, 0 and 1. Each case will contain one, or more, FIFO nodes of a Start DMA FIFO, Stop DMA FIFO, and Configure DMA FIFO. Using the right-hand rail configuration, update each FIFO node to the following:
    i. Rx Stream 0 » USRP Resources.grsc\Rx Stream 0
    ii. Rx Stream 1 » USRP Resources.grsc\Rx Stream 1
    iii. Tx Stream 0 » USRP Resources.grsc\Tx Stream 0
    iv. Tx Stream 1 » USRP Resources.grsc\Tx Stream 1

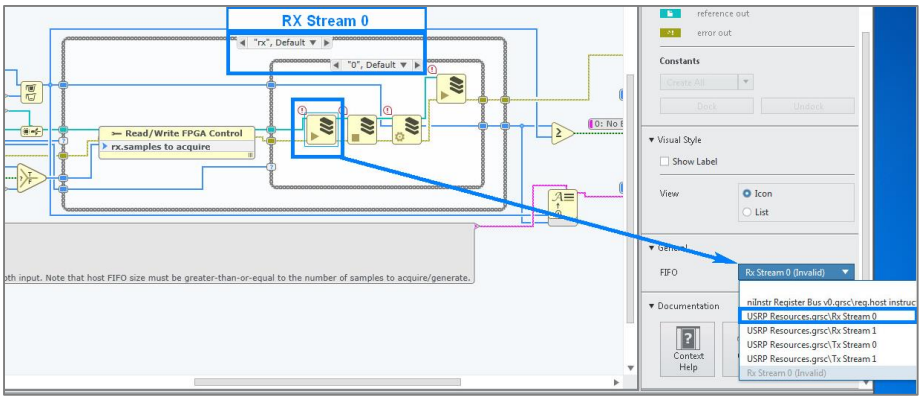

Figure 86: Update DMA FIFOs

2. Follow the same general steps from step 1 for the "Fetch Rx Data (U32).gvi" in the *<project directory>\Host SubVIs* folder of your project
    a. Select "ConstantData3" and click **Configure** in the right-hand rail as shown in Figure 87.

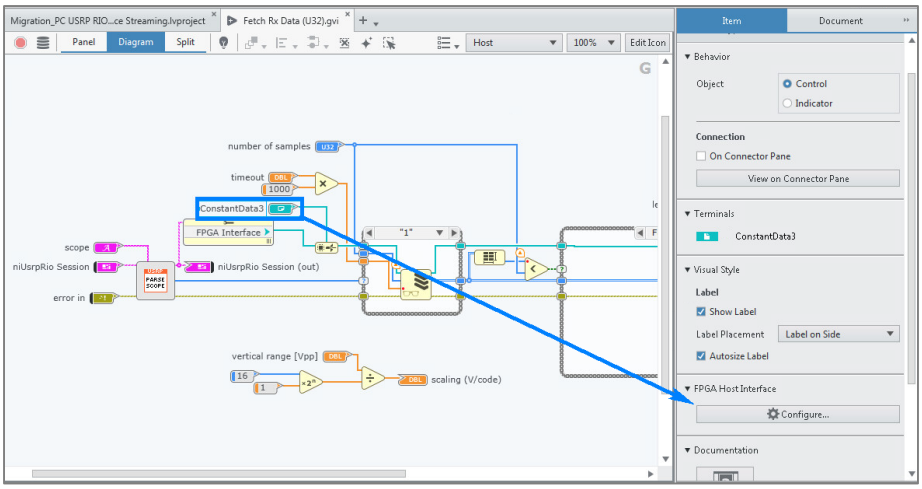

Figure 87: Configure Dialog in Right-Hand Rail

b. Click **Import bitfile** and browse to your recently compiled bitfile. Click **OK**.
c. Update the left-most Case Structure's Read DMA FIFO to point at the new DMA FIFO's for both cases. Change this by going to the right-hand rail configuration. Update the following:
    i. Rx Stream 0 » USRP Resources.grsc\Rx Stream 0
    ii. Rx Stream 1 » USRP Resources.grsc\Rx Stream 1

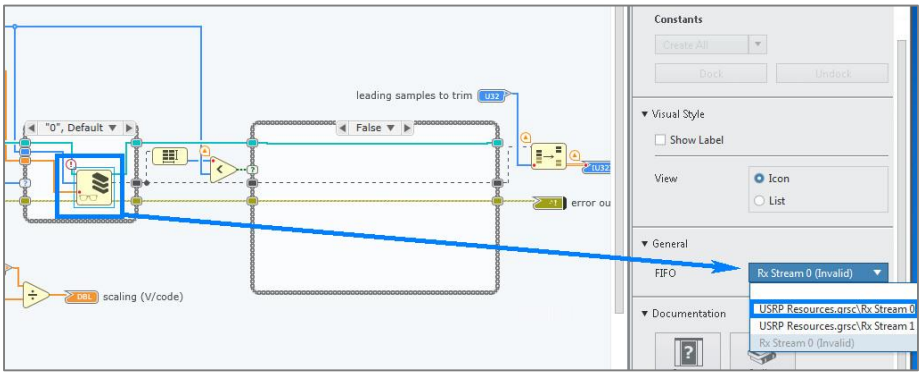

Figure 88: Update Read DMA FIFO

3. Follow the same general steps from step 1 and step 2 for the "Write Tx Data (U32).gvi" in the *<project directory>\Host SubVIs* folder of your project.
    a. Select "ConstantData2" and click **Configure** in the right-hand rail.

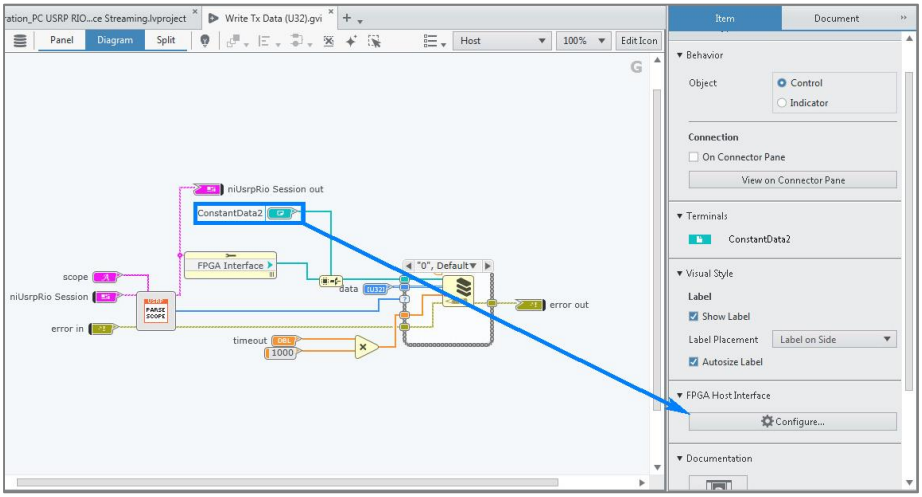

Figure 89: Configure Dialog in Right-Hand Rail

b. Click **Import bitfile** and browse to your recently compiled bitfile. Click **OK**.

c. Update the Case Structure's Write DMA FIFO to point at the new DMA FIFO's for each case. Change this by going to the right-hand rail configuration. Update the following:
    i. Tx Stream 0 » USRP Resources.grsc\Tx Stream 0
    ii. Tx Stream 1 » USRP Resources.grsc\Tx Stream 1

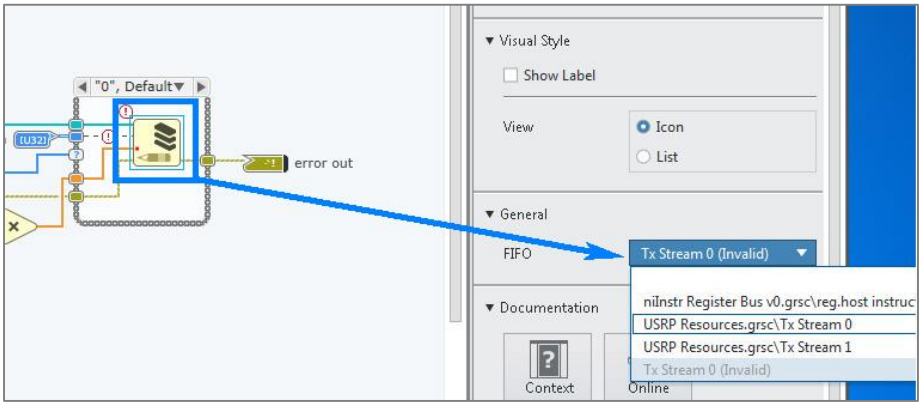

Figure 90: Update Write DMA FIFO

4. Open the "FPGA Reference Cast.gvi": *<project directory>\Host\FPGA Reference Cast.gvi*
    a. Update the "transceiver_top" to use the newly compiled bitfile by going to configure and pointing it at the bitfile.

5. In the top-level VI, update all DMA FIFO References to use the new G Resource DMA FIFO Name. They are highlighted in yellow in Figure 91.

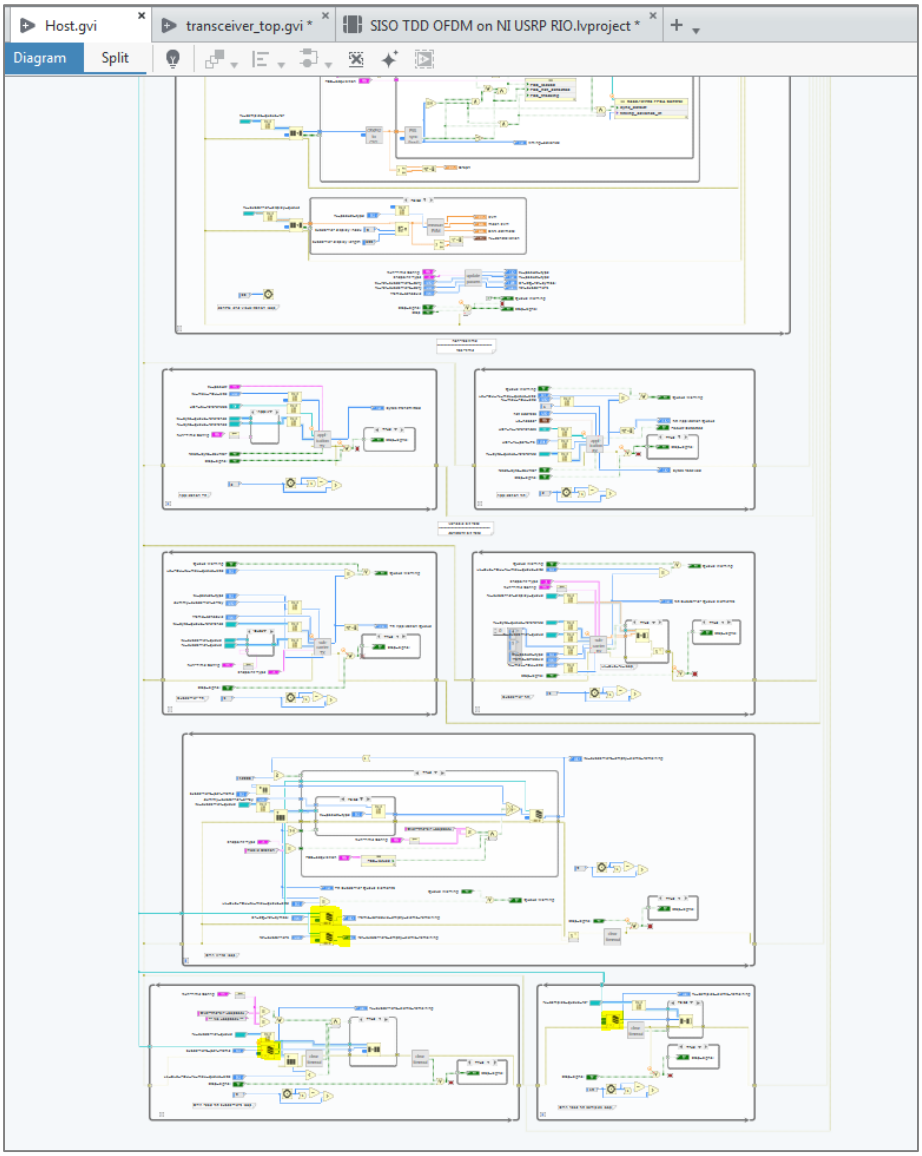

Figure 91: DMA FIFO References

Note: Similar changes must be made if any DMA FIFO's were added to the project in previous versions of LabVIEW Communications.

FLEXRIO PROJECTS

This section describes the steps necessary for reusing a project that was initially based on the FlexRIO API and example projects in LabVIEW Communications 2.0.

This section includes the following subsections. Use the one that is most appropriate for the type of FlexRIO project you are migrating.

- PXIe NI-579xR Streaming

PXIE NI-579XR STREAMING

FPGA AND SYSTEM RESOURCES

Projects using 579x targets will have broken FPGA VIs that reference the "Sample Clock". This clock has been renamed to the "Data Clock" and must be reconfigured. Additionally, the register bus FIFO has changed. Hence, an FPGA bitfile rebuild is required.

The following steps demonstrate the changes for the basic NI 5971 Rx and Tx portions of the PXIe NI-579x Streaming project. However, these same changes can be duplicated for the other NI 579x and MIMO portions.

1. Create a backup of your project.
2. Open your project in LabVIEW Communications 2.0.
3. Click **Yes** to convert as shown in Figure 92.

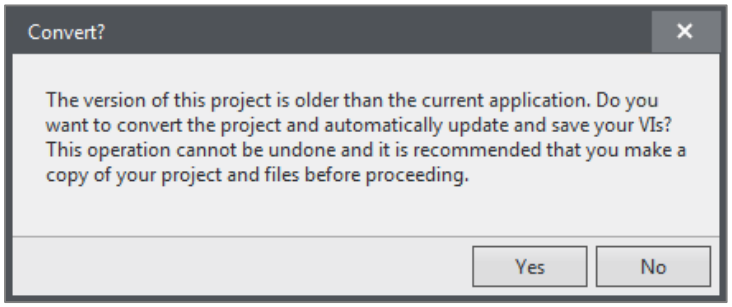

Figure 92: Convert Dialog

4. Ensure that newly created resource document matches the correct target name. For example, "5791 Resources.grsc".
5. Delete the following items from "5791Resources.grsc":
   a. reg.host instruction fifo 0

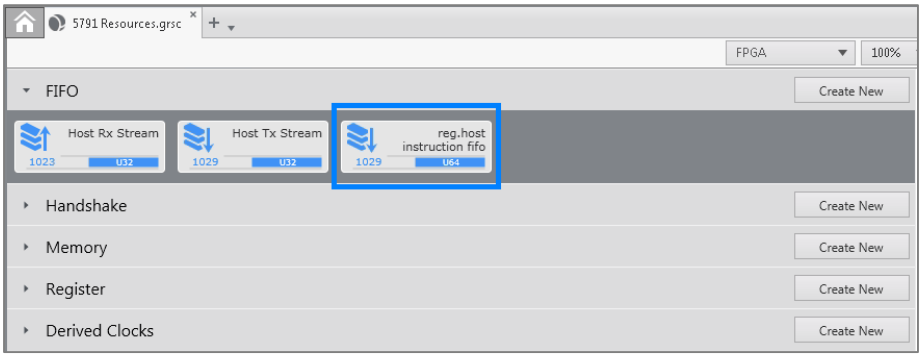

Figure 93: Files to Delete from "5791 Resources.grsc"

6. Open the "Streaming Xcvr (NI 5791) (FPGA).gvi".
7. Change references from "Sample Clock" and "Sample Clock x2" to "Data Clock" and "Data Clock x2" respectively.

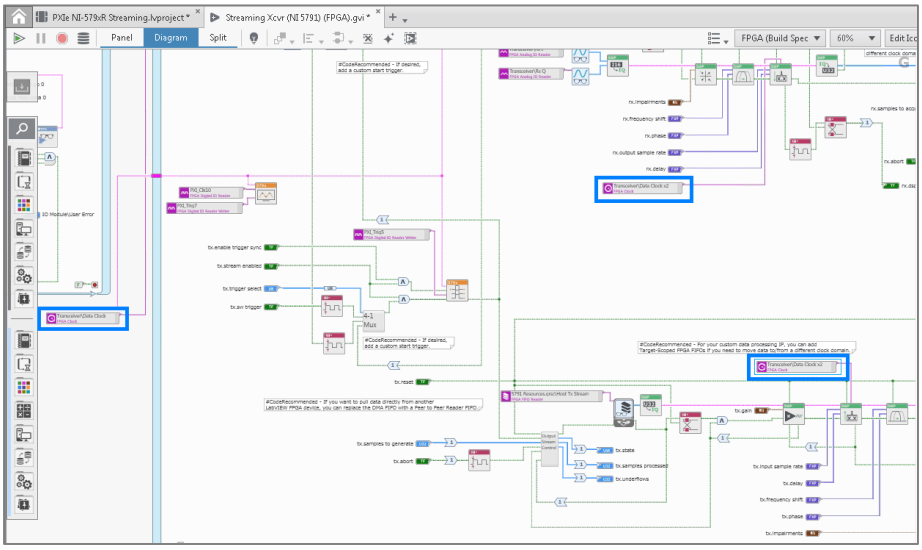

Figure 94: Data Clock References in "Streaming Xcvr (NI 5791) (FPGA).gvi"

8. Change the reference input on “Process.gcdl” to the new register bus FIFO “niInstr Register Bus v0.grsc\reg.host instruction fifo 0” as shown in Figure 95.

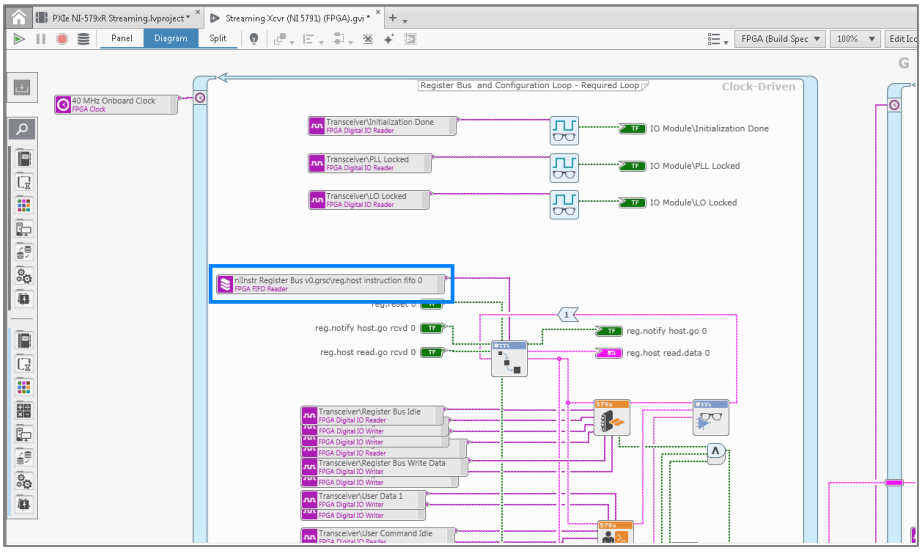

Figure 95: Updated Register Bus in “Streaming Xcvr (NI5791) (FPGA).gvi”

9. Allow all document changes to finish loading and save all files.
10. Compile the FPGA VI.
    a. Open the System Designer (in the “Project Files” view of the Navigation Pane on the left) by double-clicking on the top project file.
    b. Select the FPGA top-level under the target you want to build the bitfile for.
    c. Click **Build** on the right pane as shown in Figure 96.

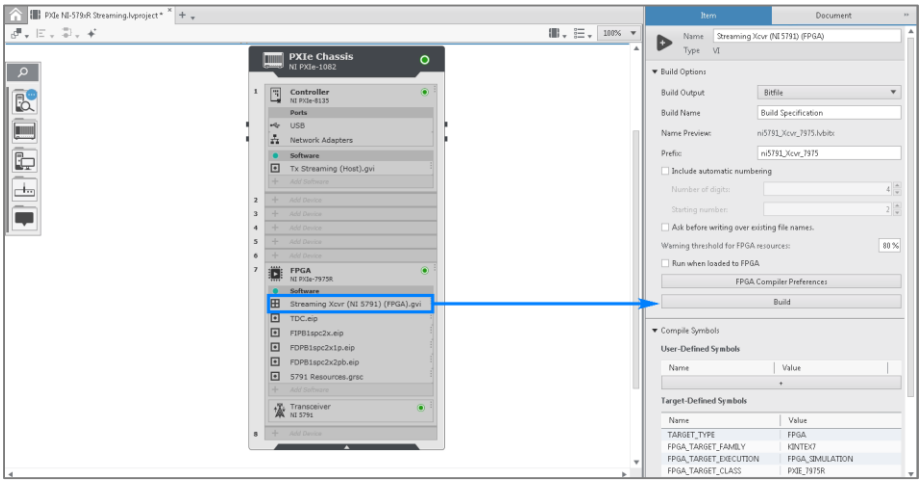

Figure 96: Compile the Project

11. **(Recommended)** Once the bitfile is finished building, exclude it from the project as shown in Figure 97 and include it again. This will ensure that the latest changes are actually applied.

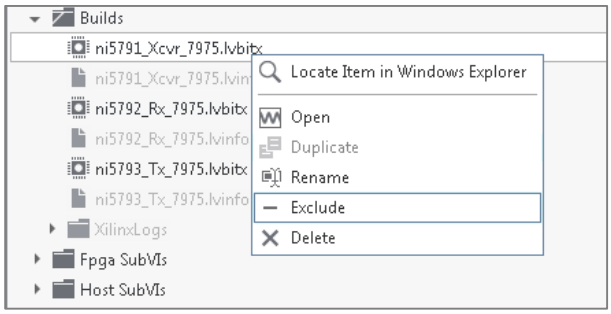

Figure 97: Option to Exclude Bitfile

HOST-SIDE

The FIFO resource names in the Host subVIs that interface with DMA FIFOs must be updated. This includes VIs which aren't in VI.lib (Sample Project VIs and User VIs). In LabVIEW Communications 2.0, the FIFO resource contains additional prefixes for the FPGA resource file name along with the FIFO name itself. For example, "Host Tx Stream" becomes "5791 Resources.grsc\Host Tx Stream." All FIFO node instances (Start/Stop, Configure, and Read/Write), along with any FPGA reference terminals, must be updated.

Use your newly compiled bitfile from step 10 above to get access to the DMA FIFO references required.

1. Open the "Configure Stream.gvi" in the *<project directory>\Host SubVIs* folder of your project.
    a. Navigate to the "tx" case.
    b. Select the "ConstantData1" terminal and in the right-hand rail click **Configure**.

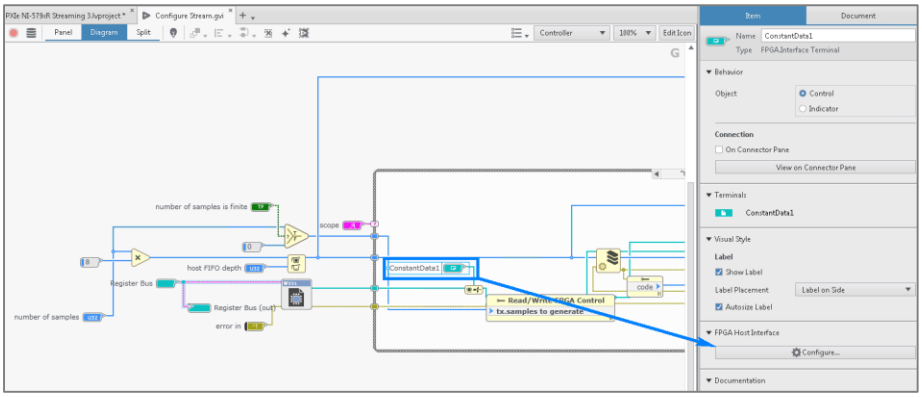

Figure 98: Right-Hand Rail Configure Option for "ConstantData1"

i. In the "FPGA Interface Dynamic Refnum Configuration" dialog, click **Import bitfile** and browse to your recently compiled bitfile.

   Note: The default location will be in the *<project directory>\Builds* folder but this, along with the file name, will vary based on the configuration of the build spec used during compile.

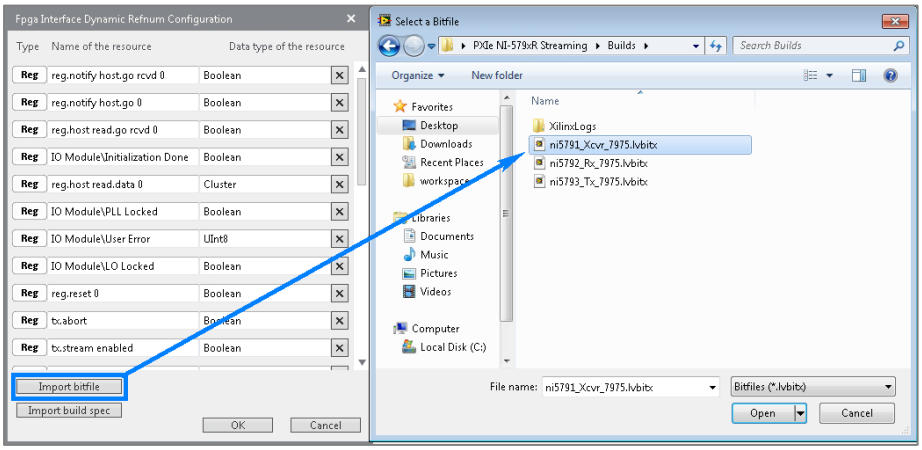

Figure 99: Import Bitfile

ii. **(Optional)** With the "FPGA Interface Dynamic Refnum Configuration" dialog still open, remove all unused resources by clicking the **X** symbol on the right side.

   Note: This will not affect functionality but may make it easier to view used resources at a later time.

iii. Click **OK** in the "FPGA Interface Dynamic Refnum" dialog.

c. Repeat this for “ConstantData1_2”.

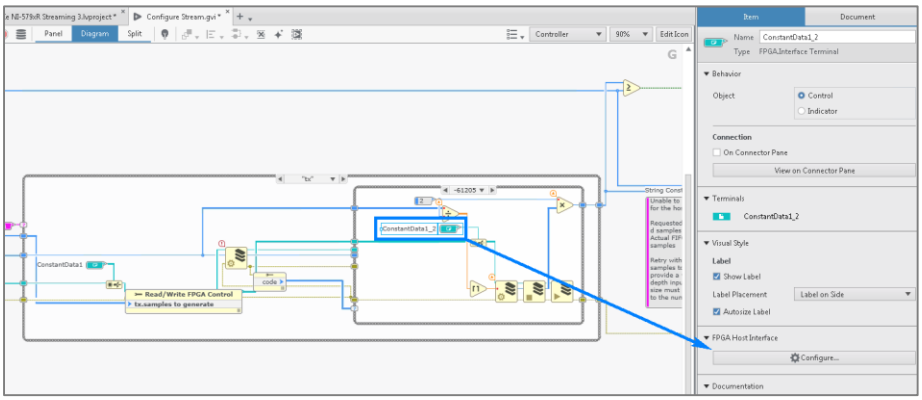

Figure 100: Right-Hand Rail Configure Option for “ConstantData1_2”

d. Navigate to the “rx” case to repeat this process for “ConstantData2” and “ConstantData2_2”

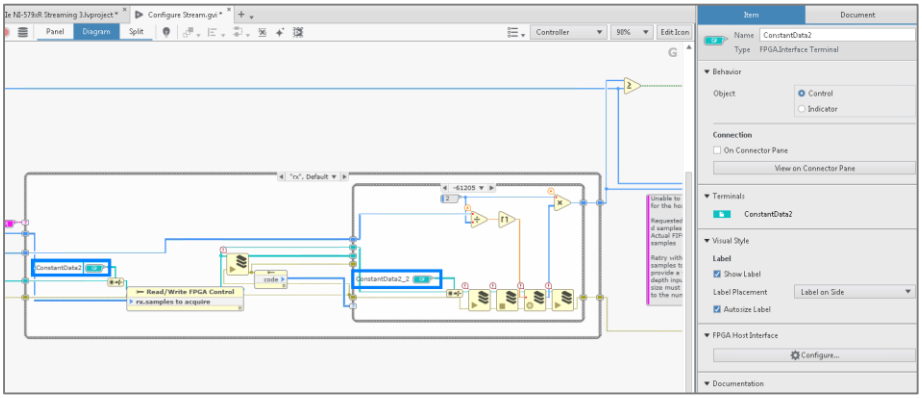

Figure 101: Update References in “rx” Case

e. Update to the new DMA FIFO's in each Case Structure for each “rx” and “tx” case as well as any nested Case Structures. Each case will contain one, or more, FIFO nodes of a Start DMA FIFO, Stop DMA FIFO, and Configure DMA FIFO. Using the right-hand rail configuration, update each FIFO node to the following:
    i. Host Tx Stream » 5791 Resources.grsc\Host Tx Stream
    ii. Host Rx Stream » 5791 Resources.grsc\Host Rx Stream

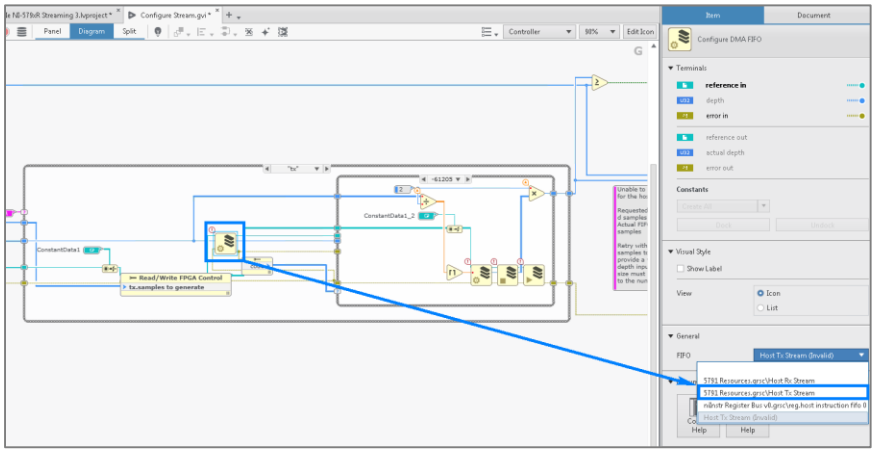

Figure 102: Update DMA FIFOs

2. Follow the same general steps from step 1 for the "Fetch Rx Data (CDB WDT).gvi" in the *<project directory>\Host SubVIs* folder of your project
    a. Update the following Read DMA FIFOs as shown in Figure 103.
        i. Host Rx Stream » 5791 Resources.grsc\Host Rx Stream
        ii. Host Rx Stream » 5791 Resources.grsc\Host Rx Stream

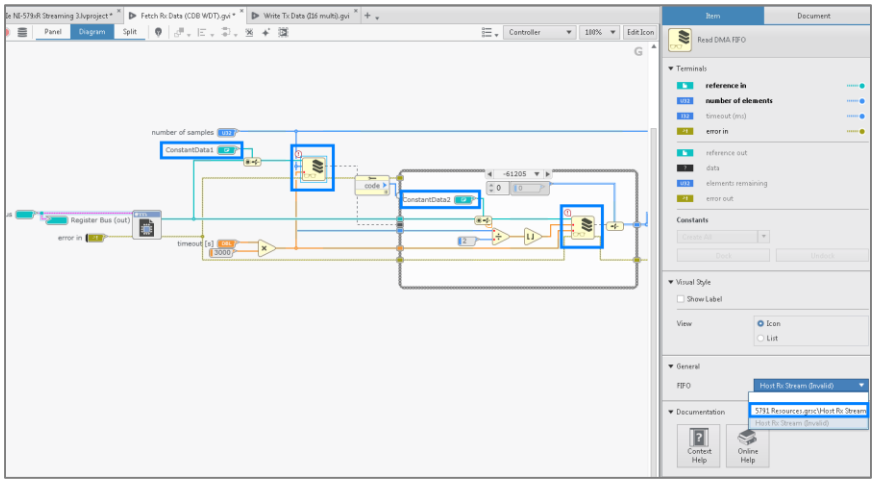

Figure 103: Update FIFOs and References for “Fetch Rx Data (CDB WDT).gvi”

3. Follow the same general steps from step 1 and step 2 for the "Write Tx Data (I16 multi).gvi" in the *<project directory>\Host SubVIs* folder of your project.
    a. Update the following Read DMA FIFOs as shown in Figure 104.
        i. Host Tx Stream » 5791 Resources.grsc\Host Tx Stream
        ii. Host Tx Stream » 5791 Resources.grsc\Host Tx Stream

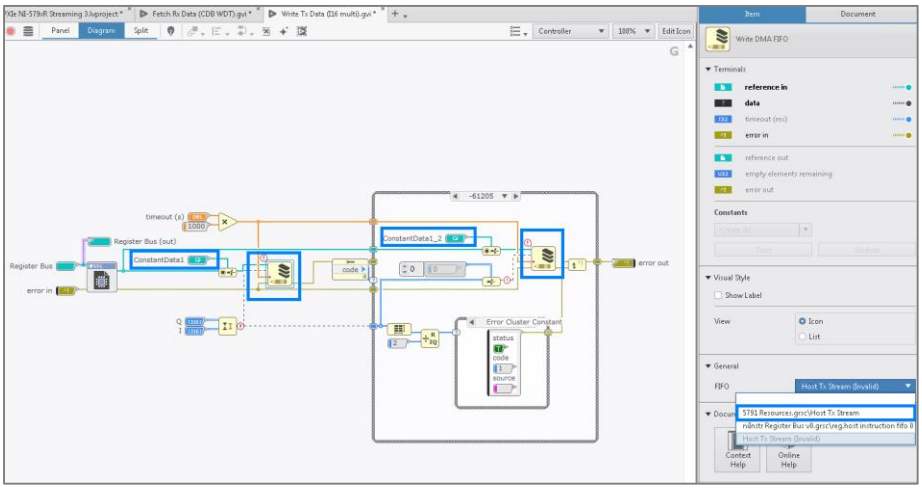

Figure 104: Update FIFOs and References for "Write Tx Data (I16 multi).gvi"

4. Configure the Open FPGA VI Reference to select the newly compiled bitfile generated from the previous section. Do this for "Tx Streaming (Host).gvi", "Rx Streaming (Host).gvi", or any top-level VI in the project that will be referencing the FPGA.

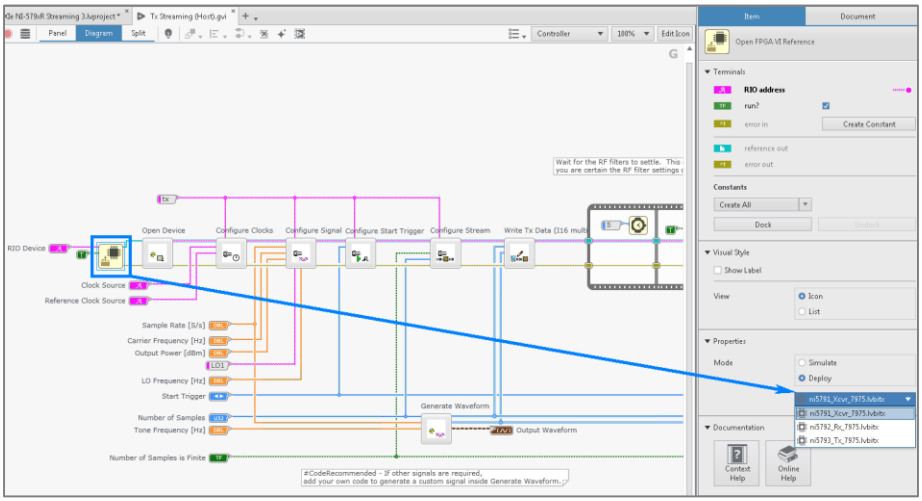

Figure 105: Update the Bitfile Reference on "Tx Streaming (Host).gvi"

5. **(Optional)** Configure the Amplitude input on the "Generate Waveform.gvi". The default setting is **1**, but this may cause DSP overflow warnings.

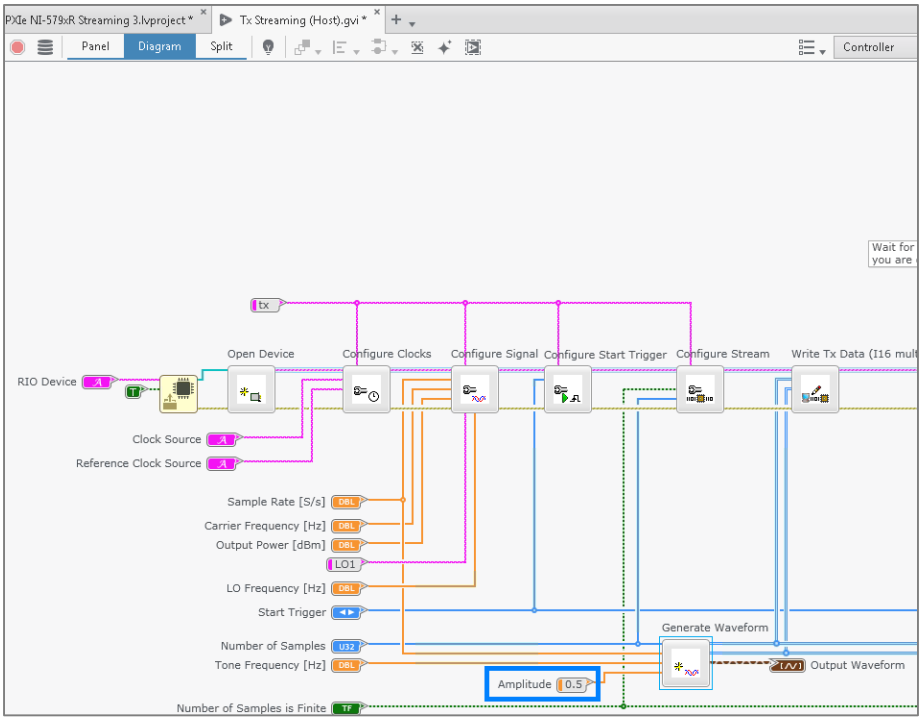

Figure 106: Lowering the Amplitude Setting for the "Generate Waveform.gvi"

Note: Similar changes must be made if any DMA FIFO's were added to the project in previous versions of LabVIEW Communications.

GENERAL PROJECTS

This section describes the necessary changes needed if you want to reuse a general project that was created based on projects in LabVIEW Communications 2.0. For details specific to USRP or the Application Frameworks, refer to those sections.

This section includes the following parts:

- FPGA
- Host
- Hardware

FPGA

PROJECTS CONTAINING MULTIPLE FPGA TARGETS

It is generally recommended that project migration should be limited to a single target at a time to reduce complexity and to make migration efforts easier. If a project does contain multiple targets, there may be issues if the same resource name is shared between these targets. For example, top-level VIs, subVIs, or common resource names that are shared between targets. To address this, remove all but one of the targets, migrate, and, if necessary, add the additional targets back into the newer project. Additionally, any broken VIs must be fixed by manually selecting the correct FPGA resources.

579X SAMPLE CLOCK

Projects using 579x targets will have broken VIs that reference the "Sample Clock". This clock has been renamed to the "Data Clock" and must be reconfigured.

For additional details, or information on migrating one of the NI-579xR streaming projects, see the FlexRIO Projects topic.

RECONFIGURE XILINX IP

Xilinx IP needs to be reconfigured due to the new Vivado version used in LabVIEW Communications 2.0.

To find the affected files and make the necessary updates, follow these steps:

1. Open the FPGA top-level you wish to migrate.
2. Wait until type propagation has occurred.
3. Once the document finishes loading, you will see in the list of errors the affected files, as shown in Figure 107.

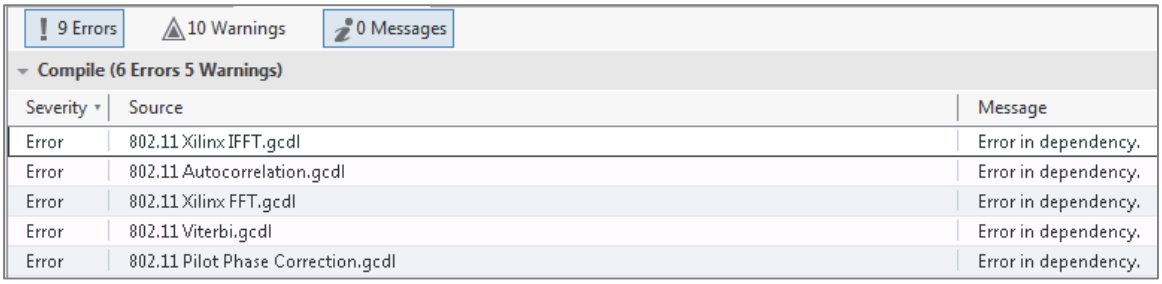

Figure 107 – Example error list showing affected files for Xilinx IP Core reconfiguration

4. Double-click on the errors as shown in Figure 107. This will open the affected document
5. Select the affected Xilinx IP block, and click **Configure Xilinx IP Core** in the right-hand rail as shown in Figure 108.
6. Wait for the "Re-customize IP" dialog to appear.
7. Inside the "Re-customize IP" dialog, leave the settings untouched and click **OK** as shown in Figure 109.
8. Wait until the warning "Not Ready" as shown in Figure 110, has disappeared.
9. Repeat steps 2 through 6 until all Errors in the FPGA top-level have disappeared.

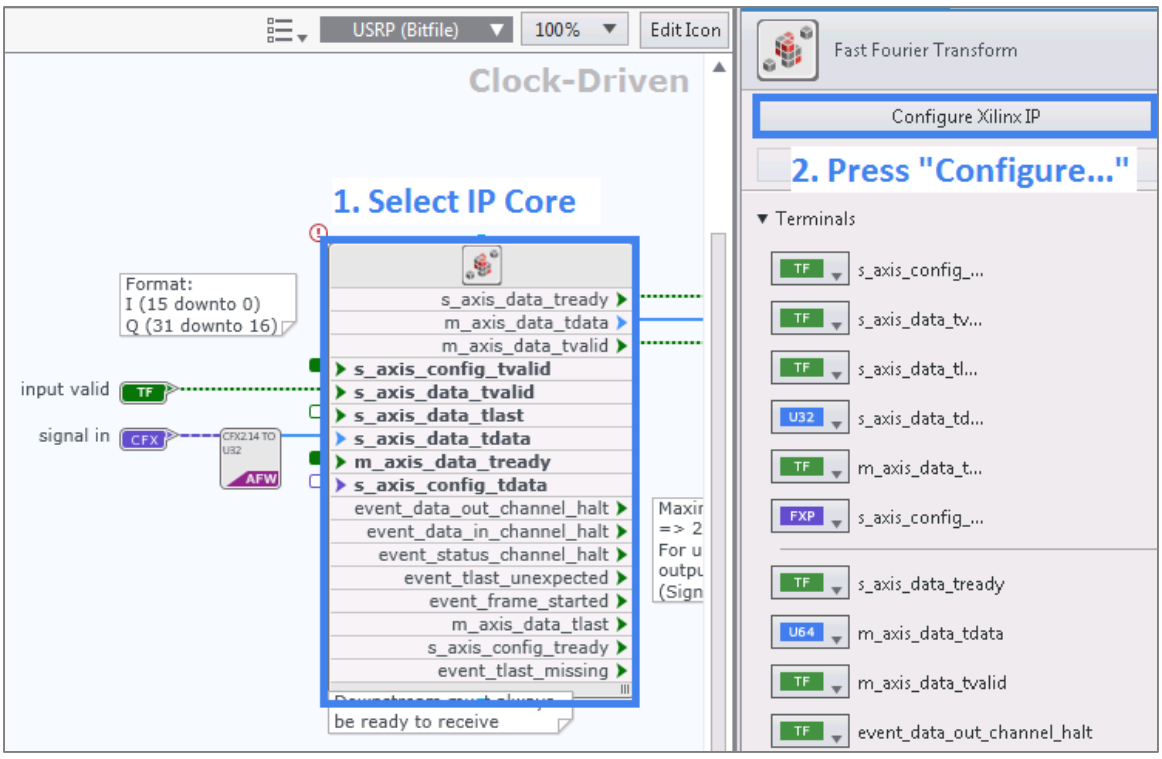

Figure 108 - How to reconfigure a Xilinx IP Core (1)

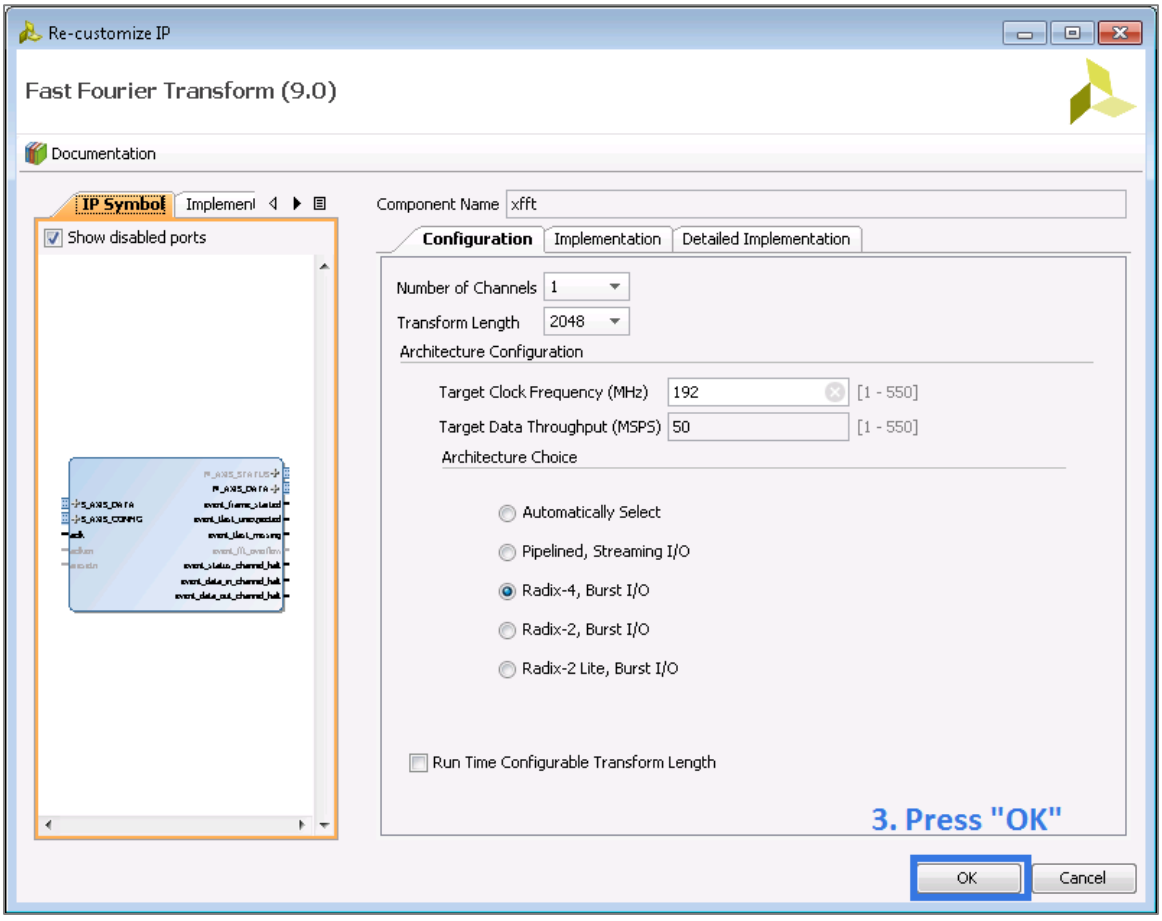

Figure 109 - How to reconfigure a Xilinx IP Core (2)

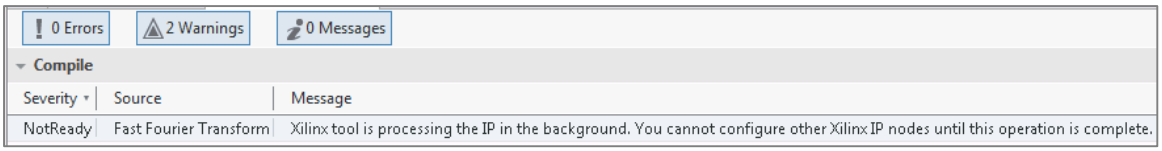

Figure 110 - "Not Ready" warning because Xilinx tool is processing the IP in the background

FPGA RESOURCES

The FIFO resource names in the Host VIs which interface with DMA FIFOs must be updated. In LabVIEW Communications 2.0, the FIFO resource contains additional prefixes for the FPGA resource file name along with the FIFO name itself. For example, in the USRP Streaming projects, "Rx Stream 0" becomes "USRP Resources.grsc\Rx Stream 0." Because of this, all FIFO node instances (Start/Stop, Configure, and Read/Write), along with any FPGA reference terminals, must be updated.

If the host is not updated correctly, runtime errors can occur after migrating over a previous project. These errors may persist even after a recompile in the newer version. A typical error is shown in Figure 111.

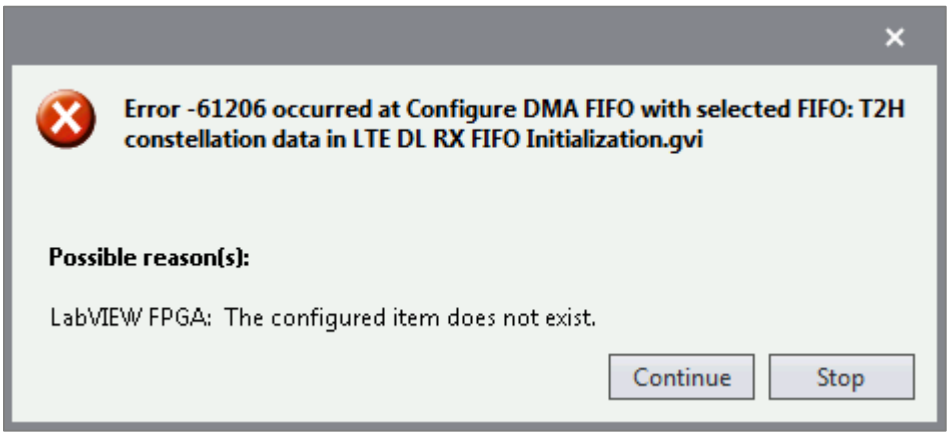

Figure 111: Typical Error if Host Not Updated

Note: The bitfile may need to be recompiled in version 2.0 to make use of any changes in resource files that may have been added during migration of the project. This must be done prior to any host-side changes that reference that bitfile.

To update FPGA refnums, complete the following steps:

Note: If the application uses FPGA terminals, or constants, then they may need to be updated to the latest bitfile.

1. Select the FPGA reference and click **Configure** in the right-hand rail.

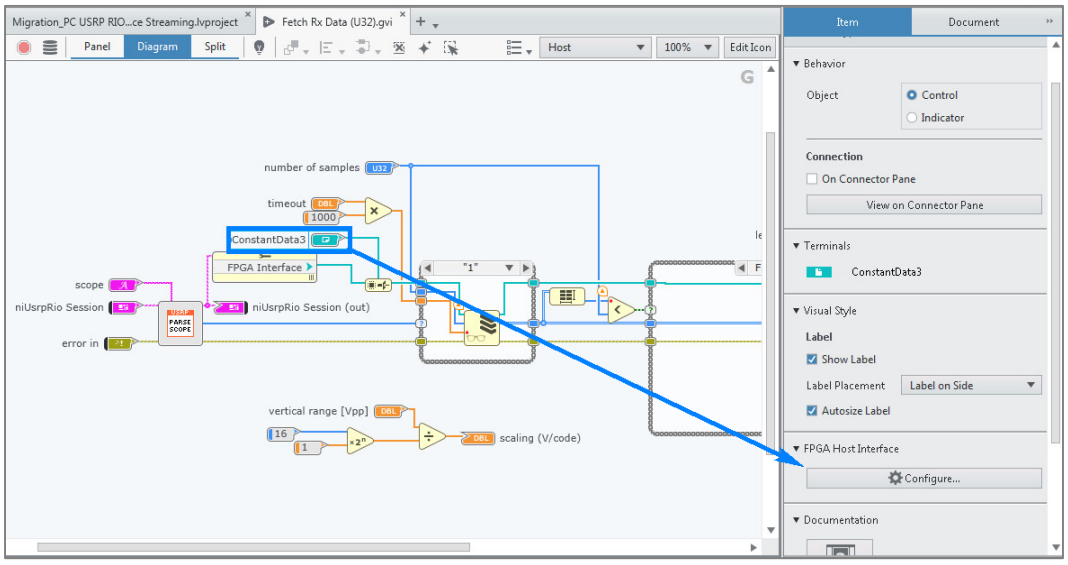

Figure 112: Configure Dialog in Right-Hand Rail

2. In the "FPGA Interface Dynamic Refnum Configuration" dialog, click **Import bitfile** and browse to your recently compiled bitfile.
   Note: The default location will be in the *<project directory>\Builds* folder. However, this, along with the file name, will vary based on the configuration of the build spec used during compile

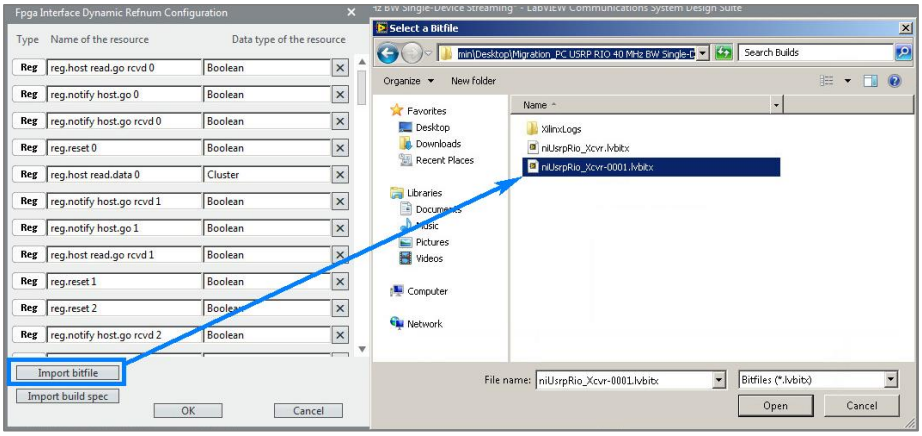

Figure 113: Import Bitfile

3. **(Optional)** With the "FPGA Interface Dynamic Refnum Configuration" dialog still open, remove all unused resources by clicking the **X** symbol on the right side. This will not affect functionality but may make it easier to view used resources at a later time.
4. Click **OK** in the "FPGA Interface Dynamic Refnum" dialog.

To update DMA FIFO nodes, complete the following steps:

1. Select the affected DMA FIFO node.
2. In the right-hand rail is a drop-down selection for available FIFO names referenced by the bitfile being used. Select the appropriate resource as shown in Figure 114.

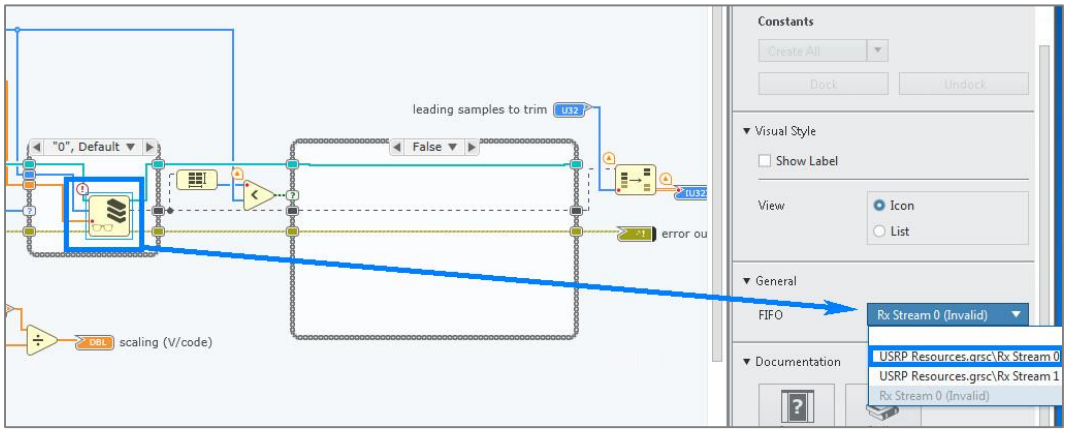

Figure 114: Select Appropriate Resource

USRP API

Several changes have been made to the USRP API. For a list of these changes, and needed adjustments, see the Host API section under the USRP Projects topic.

HARDWARE

USRP 40 MHZ

The USRP RIO 40 MHz target will be deprecated in future versions of LabVIEW Communications. Previous projects built for these targets will still be usable in version 2.0 but you should consider migration efforts in future designs. Please see the USRP Projects topic for instructions on how to update to the latest USRP targets.

NEXT STEPS

The LabVIEW Communications product documentation provides detailed information in addition to the tasks discussed in this guide. Hardware manuals also contain valuable information about the features and performance characteristics of NI RIO devices.

The main NI support page, ni.com/support, provides quick access to manuals, KnowledgeBase documents, tutorials, example code, community forums, technical support, and customer service.

FORMAL TRAINING

In addition to online content, in-product Guided Help and examples, you can also obtain LabVIEW Communications instructor-led training from NI. For details, visit ni.com/training/communications or consult with your NI sales representative.

NI ALLIANCE PARTNERS AND SERVICES

NI partners can help you with the design, integration, and deployment of your application. The NI Alliance Partner Network is a program of more than 700 independent, third-party companies worldwide that provide engineers with complete solutions and high-quality products based on graphical system design.

Finally, you can count on NI field engineers to discuss how to solve your particular application using the NI platform.

REVISIONS AND FEEDBACK

NI strives to provide high-quality content and welcomes your comments. Please send feedback for future revisions of the guide to your local NI support team.

If there are areas you believe need clarification, submit your question to the LabVIEW Communications System Design Suite Discussion Forum, where the applications, support, and R&D engineers can answer it for the benefit of all readers.

Revision	Date	Change Summary
1.0	8/1/2016	Initial release alongside LabVIEW Communications 2.0