NI LabVIEW Communications 2.0 to 2.1 Migration Guide

Recommended Steps to Migrate Pre-Existing Applications

August 2018

NATIONAL INSTRUMENTS

Contents

Introduction

LabVIEW Communications 2.1 automatically converts your LabVIEW Communications 2.0 projects to equivalent versions in LabVIEW Communications 2.1. Due to inherent differences between LabVIEW Communications 2.0 and LabVIEW Communications 2.1, however, the organization and structure of your projects will be different following the conversion process. Additionally, some manual reconfiguration might be required depending on the application. The purpose of this guide is to highlight the major differences in project structure and organization after the conversion process.

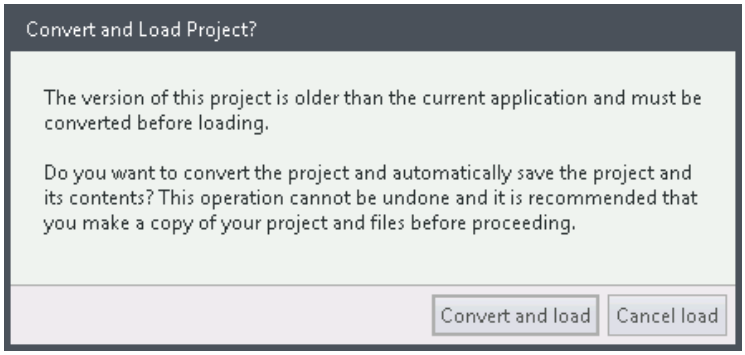
It is not possible to open projects created with LabVIEW Communications 1.0 or 1.1 in LabVIEW Communications 2.1. Older projects should first be migrated to LabVIEW Communications 2.0 using the LabVIEW Communications 1.1 to 2.0 Migration Guide or rewritten entirely in LabVIEW Communications 2.1.

Loading LabVIEW Communications 2.0 Projects in LabVIEW Communications 2.1

Caution: Before you begin, NI recommends that you create a backup copy of the 2.0 version of the project before starting the conversion process. The conversion process cannot be undone, and LabVIEW Communications 2.1 projects are not backwards compatible.

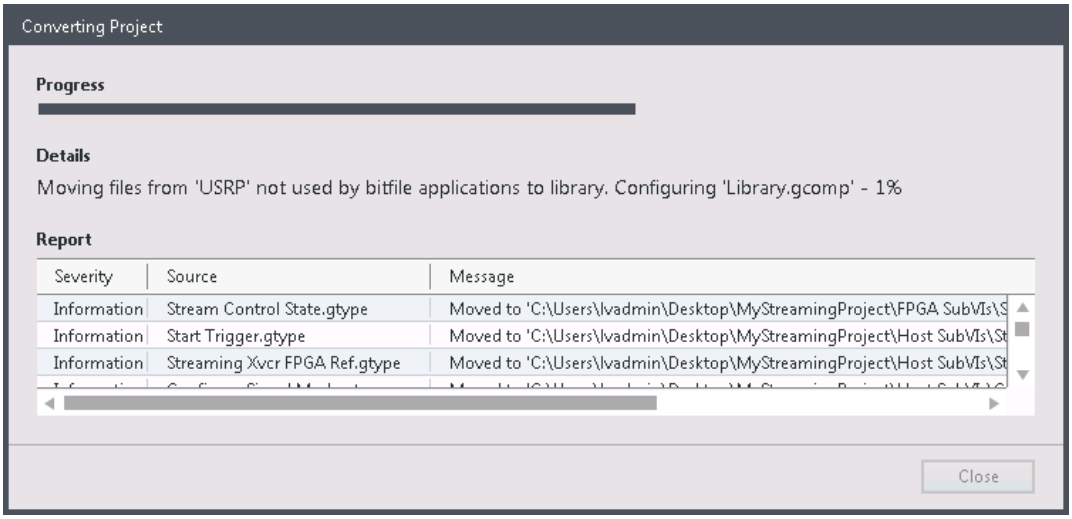
1. To start the conversion process, open a LabVIEW Communications 2.0 project in LabVIEW Communications 2.1.

   When opening a LabVIEW Communications 2.0 project in LabVIEW Communications 2.1 the following dialog box will display:

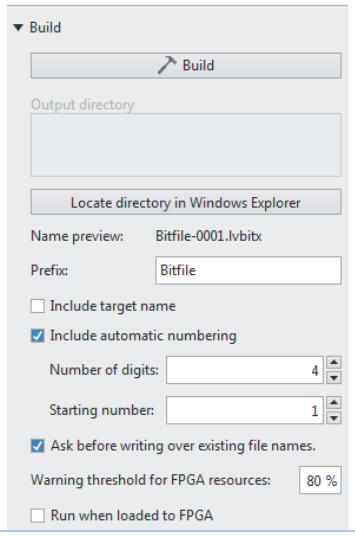


2. Select **Convert and Load** to proceed.

   A progress dialog box displays the current state of the conversion process:



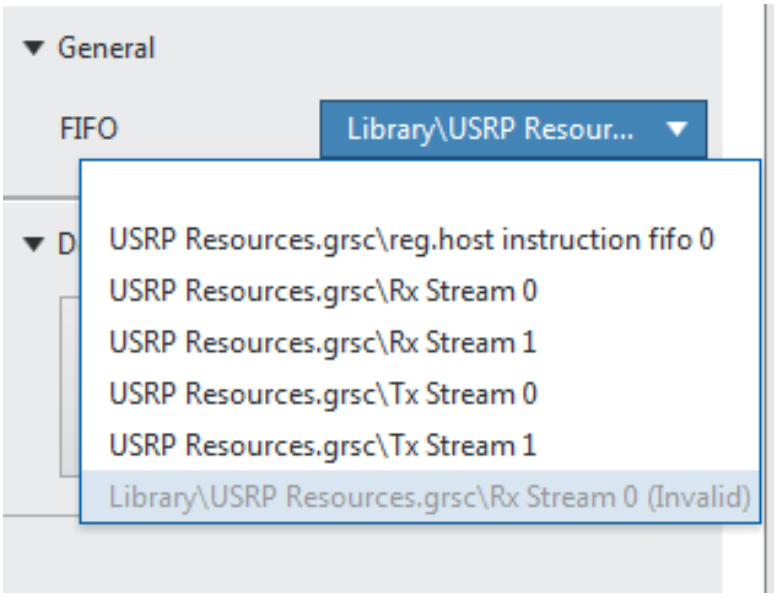
3. When the conversion completes, review the report and then select **Close** on the dialog box. The following table lists common errors that can occur during conversion:

Example Error Message	Possible Cause	Possible Resolution
`'Include automatic numbering' bitfile build setting had conflicting values: 'True' vs. 'False'. Value 'True' will be used. To use the other bitfile build setting value change the value in the component editor.`	The project has a Bitfile VI with multiple build settings and the conversion utility does not know which setting to use. Therefore, the conversion utility will automatically select one of the build setting values.	To use a different build setting value you must manually change the value in the component editor: 
`'MySymbol' compiler symbol had conflicting values: '0' vs '1'. Value '0' will be used. To use the other symbol value move the component to a different FPGA target.`	In LabVIEW Communications 2.1 the compile symbols are associated with a target as opposed to the symbols being associated with a Bitfile VI and a target as they were in LabVIEW Communications 2.0. Therefore, if two separate Bitfile VIs in a single target have the same compile symbols set for different values then the conversion utility does not know what value to set and will automatically set one of the values.	To use the other symbol value, move the desired component to a different FPGA target and set the desired compile symbol and value for the new target.
`Could not locate 'myVI.gvi' under this target.`	The project file was manually edited or was otherwise corrupted.	Open the converted project to determine if the conversion utility was able to resolve the problem automatically. If the issue is not resolved then manually remove the VI, add it to the correct library, and update all calls to the VI.
`This file might not have been converted correctly because it could not be saved. Verify write access to the location and available disk space for the file: "\Path\to\Project.lvproject"`	The project became read-only during the conversion or there is insufficient disk space on the machine. One possible cause of a file permission change could be interaction from source code control software.	After this error is thrown the project should be considered corrupt because the project was not able to save information needed for persistence. To resolve the issue revert to a known good LabVIEW Communications 2.0 version of the project, verify sufficient disk space and attempt the conversion again.

Host Application Conversion Behaviors

Use the following conversion behavior explanations to better understand the new format of your host applications and libraries in LabVIEW Communications 2.1 and make any necessary changes to your converted projects to restore their original functionality.

- The conversion process moves top-level VIs, type definitions (now G Types), and subVIs into Library documents.
    a) The names of the libraries are **Library.gcomp** or **Library_*X*.gcomp** where *X* is a unique number identifying the library.
    b) The conversion process creates namespaces in these libraries that correspond to the original on-disk folders containing the various project documents.
- FPGA Host Interface nodes have broken references after conversion. To resolve the issue, update each affected FPGA Host Interface node to point to the correct resource in the existing bitfile. For example, to update the **Read DMA FIFO** node, select the node on the diagram, select the **Item** tab, and in the **General** section select the **FIFO** drop-down to select the correct FIFO name:



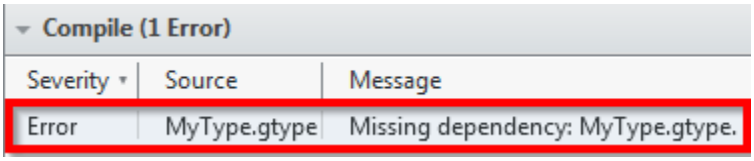
- The MathScript Node is not supported in LabVIEW Communications 2.1. Use an Interface for MATLAB® to call MATLAB functions or scripts. Refer to the *Migrating from MathScript Node to Interface for MATLAB®* help topic for more information.

  Note: There is a known issue tracked by Bug ID 707042 that any project containing a MathScript Node that had the **Show Label** checkbox selected will fail to migrate properly. The failure will occur even if the **Show Label** checkbox was later deselected. The Conversion dialog may report "`Converting source files to components failed with unknown error`" and when you attempt to open the VI that contained the MathScript Node an error is thrown stating that "`The source file format is invalid`." The issue is captured on the LabVIEW Communications System Design Suite 2.1 Known Issues List. To workaround the issue:

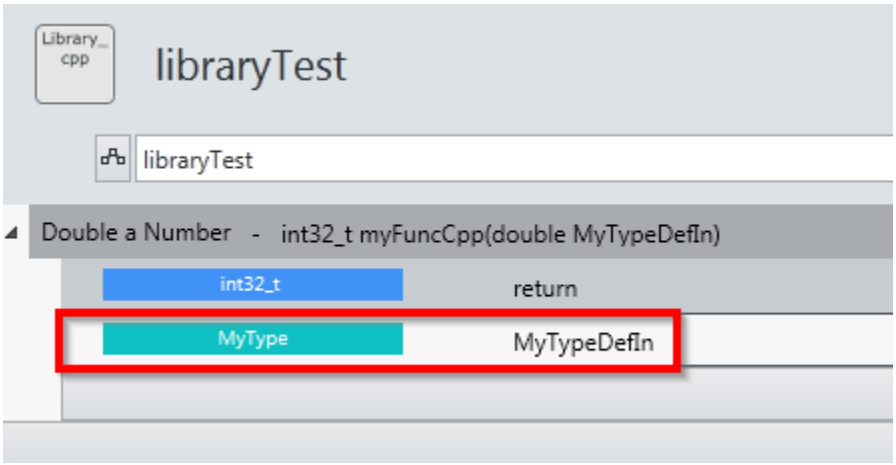
  a) Open a backup copy of the project in LabVIEW Communications 2.0.
  b) Open the VI containing the MathScript Node.

c) Drop a new MathScript Node onto the diagram.
d) Copy the text from the old node into the new node.
e) Delete the old node from the diagram.
f) Rewire as necessary.
g) Save the VI and attempt the conversion process again.

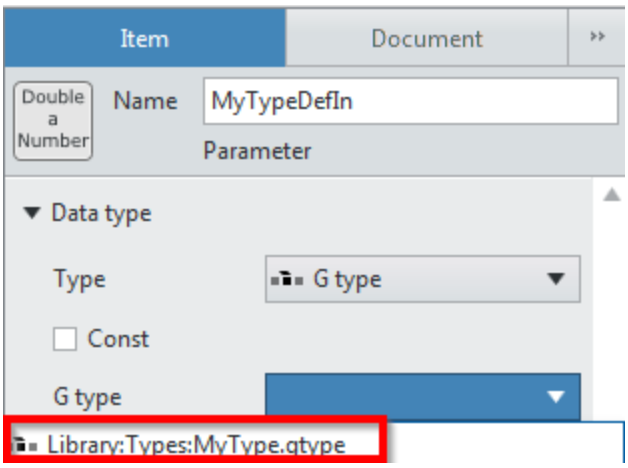
- Shared Library Interface documents that use Type Definitions as the Type for function parameters will be broken following conversion. This is a known issue tracked by Bug ID 706310 and is captured on the LabVIEW Communications System Design Suite 2.1 Known Issues List. To resolve the issue:
  a) Open each affected SLI document
  b) Review the **Errors and Warnings** tab for the SLI document and note the missing GType names:



  c) Locate the function parameter(s) associated with each affected GType:



  d) Click on each affected function parameter and on the **Item** tab click the **GType** dropdown and select the correct GType from the list:

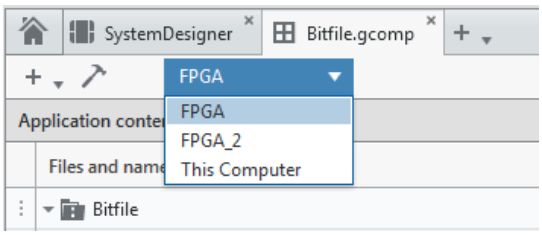


e) Repeat the process until there are no more errors on the **Errors and Warnings** tab of the SLI document

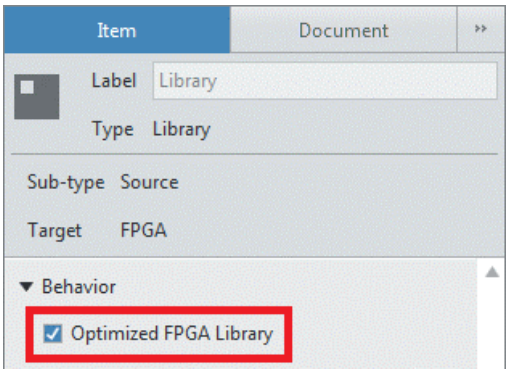
FPGA Application Conversion Behaviors

Use the following conversion behavior explanations to better understand the new format of your FPGA applications and libraries in LabVIEW Communications 2.1 and make any necessary changes to your converted projects to restore their original functionality.

- The conversion process places any FPGA VI with a Build Output set to **Bitfile** in its own Application document, inheriting the **Build Name** from the original VI's **Build Options**. The root namespace in the Application document is also the **Build Name**.
- If a single VI was targeted to multiple FPGA targets (and each had the Build Output set to **Bitfile**), the conversion process creates a single Application document, but you can select from multiple FPGA targets when viewing the Application document contents.

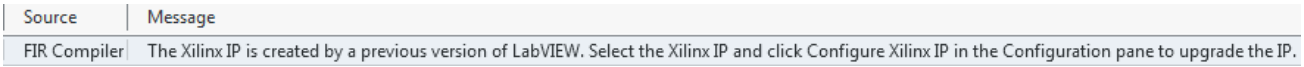


- The conversion process moves any FPGA VI with a Build Output set to **None** into a Library document. To compile a top-level FPGA VI after conversion you must move it into an Application document.
- The conversion process generally groups other documents (.gcdl, .gmrd, .gvi, .eip, etc.) used in a single Bitfile Application into a single Library document.
    - The names of the libraries are **Library.gcomp** or **Library_*X*.gcomp** where is *X* is a unique number identifying the library. The root namespace will match the library document name.
    - The conversion process creates namespaces in these libraries that correspond to the original on-disk folders containing the various project documents.
- The conversion process moves FPGA VIs with a Build Output Setting of **Optimized FPGA VI** into Library documents, but note that the **Optimized FPGA Library** checkbox is not selected. Manually select this checkbox from the **Item** tab of the **SystemDesigner** view of the Library to treat those VIs as Optimized FPGA VIs after the conversion process. In some cases an Optimized FPGA VI can be moved into a Library containing other document types during conversion. In order to treat it as an Optimized FPGA VI, you must move it into a library separate from the other document types.



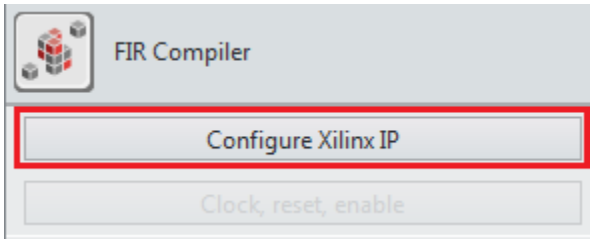
- If an Optimized FPGA VI is not part of a Bitfile Application, the conversion process moves it into its own Library document. If the same Bitfile Application uses multiple Optimized FPGA VIs, the conversion process moves the Optimized FPGA VIs into the same Library document. If multiple Bitfile Applications use an Optimized FPGA VI, the conversion process moves the Optimized FPGA VI into its own Library document. You must replace call sites of the Optimized FPGA VIs within other documents by navigating to each Optimized FPGA VI from the Project Items palette category and adding them to the diagram.

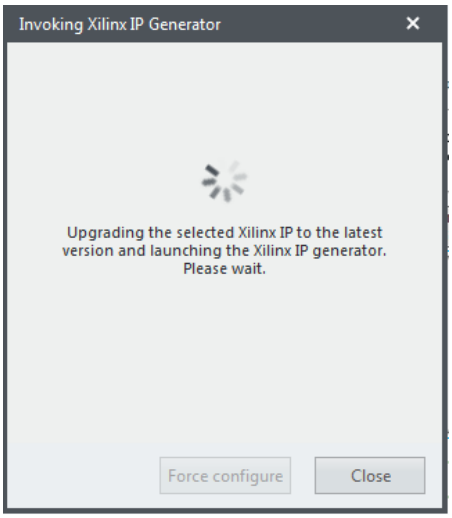
- The conversion process moves Resource Collections into Library documents, and the names of the resources contained in the **.grsc** documents change to include the namespace in the library. As a result, the names of the resources in the project are different from the names of the resources in bitfiles compiled before the conversion.
- Xilinx CORE Generator IP Nodes are broken after conversion.



To resolve the issue:

a) Select the Xilinx IP Node and click **Configure Xilinx IP** to open a dialog that automatically upgrades the IP.





b) Confirm the correct configuration options in the Re-customize IP dialog and click **OK**. The **Errors and Warnings** pane displays an error message in the VI that contains the Xilinx COREGEN IP Node while the Xilinx tool processes the changes:

Source	Message
FIR Compiler	Xilinx tool is processing the IP in the background. You cannot configure other Xilinx IP nodes until this operation is complete.

- External IP (EIP) Documents and External IP Nodes are broken following conversion. The reason for this is that the conversion process moves EIP Documents into components, but it does not move the VHD source files they reference. Therefore, the VHD source files are in a different location than the EIP Document expects.

  Example error message in a VI containing an EIP Node:

SimpleAdder.eip	Error in dependency.
SimpleAdder	The dependency files for the External FPGA IP are not found. Double click the IP to get more detailed information. Missing dependencies: SimpleAdd.vhd

  Example error message in an EIP Document:

Source	Message
SimpleAdder.eip	The following external FPGA IP file dependencies are not found. Correct the paths to proceed.

Details: : Missing dependencies: C:\Users\pse\Desktop\External HDL IP\Library.gcomp\HDL Source\SimpleAdder\SimpleAdd.vhd

You can use either of the following methods to resolve the issue:

a) Move the VHD source file to the expected location.
   i. Close the project.
   ii. Copy the VHD file into the component folder that contains the EIP Document using the same folder structure outlined in the error message from the EIP Document. In the example above the `HDL Source` folder and its contents would be copied into the `Library.gcomp` folder on disk.
   iii. Open the project and confirm that the EIP Document is no longer broken.

b) Point the EIP Document to the VHD source at the correct location and click **Parse and Verify**. Note that all Signal Configuration, Generic, and Clock settings in the EIP Document will revert to default values. If you were using non-default values, then you must manually set those values.

RT Application Conversion Behaviors

Use the following conversion behavior explanations to better understand the new format of your RT applications and libraries in LabVIEW Communications 2.1 and make any necessary changes to your converted projects to restore their original functionality.

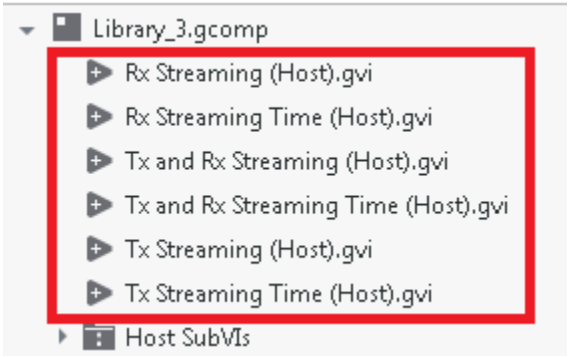
- Generally, the same rules apply to RT target conversion as Host conversion.
- A real-time controller with the LabVIEW Communications 2.0 image is not compatible with LabVIEW Communcations 2.1.
- Installing LabVIEW Communications 2.1 on a machine with an existing installation of LabVIEW Communications 2.0 prevents the older version of the software from communicating with real-time targets. This is a result of LabVIEW Communications 2.1 upgrading software components used for system discovery and communication, rendering these components incompatible with LabVIEW Communications 2.0. This a known issue tracked by Bug ID 688821 and is captured on the LabVIEW Communications System Design Suite 2.1 Known Issues List.
- The version of the real-time image applied on the RT Controller must match the version of LabVIEW Communcations used on the Windows Host PC. Therefore, the LabVIEW Communications 2.1 real-time image must be applied to any RT controller being used with LabVIEW Communications 2.1. Refer to the Provisioning a Real-time Controller for LabVIEW Communications System Design Suite tutorial for instructions on applying the LabVIEW Communications 2.1 image.

USRP Streaming Project Conversion Behaviors

Use the following conversion behavior explanations to better understand the new format of your USRP streaming applications and libraries in LabVIEW Communications 2.1 and make any necessary changes to your converted projects to restore their original functionality.

Host

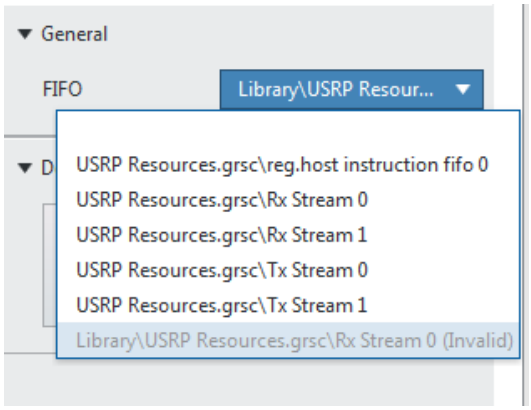
- The conversion process groups the top-level host VIs into a single Library document:



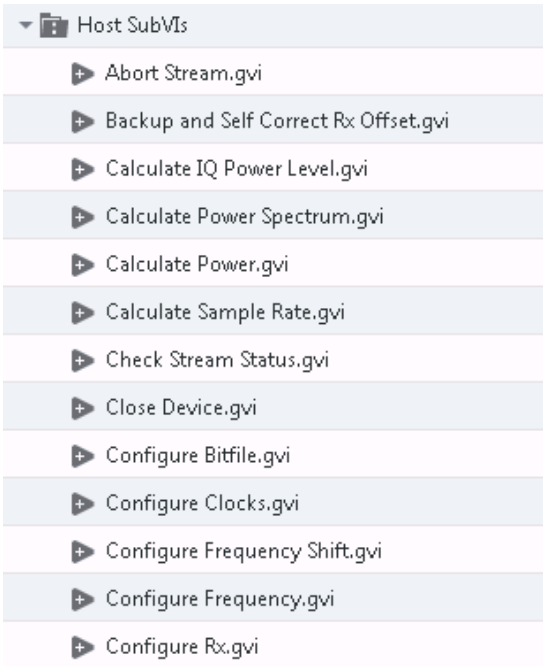
- When opened, the top-level Host VIs list errors in dependencies that are related to a mismatch between the resource names in the project and those configured on FPGA Host Interface Nodes:

Severity	Source	Message
Error	Rx Stream 0	The FIFO name does not exist. FIFO name: Library\USRP Resources.grsc\Rx Stream 0
Error	Tx Stream 0	The FIFO name does not exist. FIFO name: Library\USRP Resources.grsc\Tx Stream 0

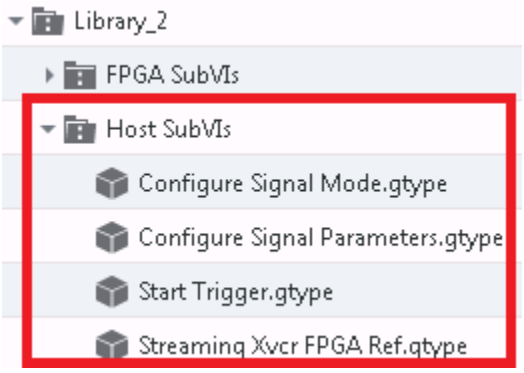
- Due to the fact that the **USRP Resources.grsc** Resource Collection is now located in a Library document (see FPGA section below), the resource names in the project no longer match those compiled into the bitfile. To resolve the issue, update each affected FPGA Host Interface node to point to the correct resource name in the existing bitfile. For example, to update the **Read DMA FIFO** node, select the node on the diagram, select the **Item** tab, and in the **General** section, select the **FIFO** drop-down to select the correct FIFO name:



- Within the library that also contains the top-level Host VIs, the conversion process creates a namespace for the **Host SubVIs** folder that contains all .gvi documents.



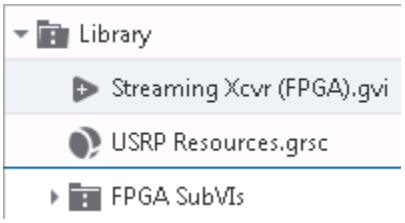
- The conversion process moves existing type definitions from the **Host SubVIs** folder to a separate library that has a **Host SubVIs** namespace:



Note: Type definitions are referred to as **G Types** in LabVIEW Communications 2.1

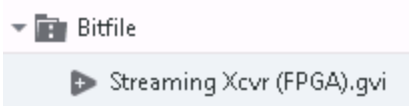
FPGA

- If the build output of the top-level FPGA VI was set to **None** prior to conversion, the conversion process moves it into a Library document:



Note: In order to compile a bitfile using this top-level VI, you must move it into an Application document. This can be accomplished by dragging and dropping the top-level VI from the Library document into an Application document (see help) targeted to the USRP.

- If the build output of the top-level FPGA VI was set to **Bitfile** prior to conversion, the conversion process moves it into an Application document named **Bitfile.gcomp**.



Note: The Application documents inherit the Build Names that were set in the VI's Build Options.

- The conversion process moves the USRP Resources.grsc Resource Collection into a Library document. Due to this change, when recompiling the bitfile, you need to update host-side VIs that reference the resources in this Resource Collection.
- The conversion process moves all documents from the FPGA SubVIs folder into FPGA SubVIs namespaces in Library documents.
- Any compiled bitfiles stored in the Builds folder still exist at the same location.

Special Considerations for the USRP-2974 Streaming Project

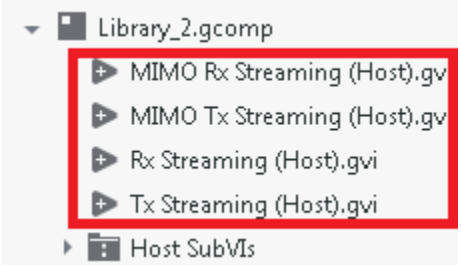
- The USRP-2974 is not supported in LabVIEW Communications 2.1 at the time of release. For more information navigate to ni.com/support.
- When support is added the behavior described in the Host section above will apply to the VIs on the Real-time Controller. Additionally, it will be necessary to provision the USRP-2974 with the LabVIEW Communications 2.1 image using the Provisioning a Real-time Controller for LabVIEW Communications System Design Suite tutorial.

FlexRIO (NI-579x Streaming) Project Conversion Behaviors

Use the following conversion behavior explanations to better understand the new format of your FlexRIO streaming applications and libraries in LabVIEW Communications 2.1 and make any necessary changes to your converted projects to restore their original functionality.

Host

- The conversion process moves all top-level host VIs into a single Library document:

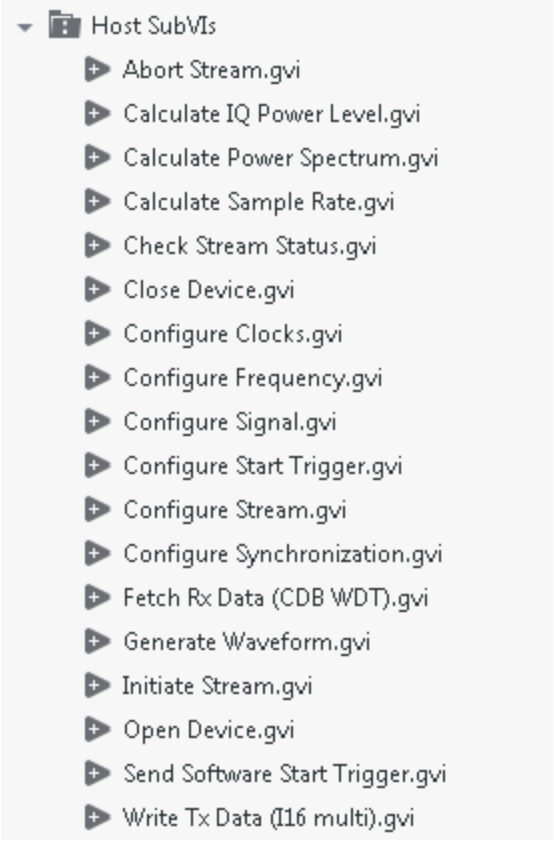


- Due to a change in the data types used in the 579x API, you must modify the Register Bus terminal and Register Bus (out) terminals to match the new data types. Alternatively, replace the affected subVIs using the VIs attached to the parent page that hosts this document. In order to manually make the changes:
    a. From the top-level VI open each broken subVI.
    b. Right-click the terminals that the broken wires are connected to and select **Create control** or **Create indicator**, as necessary.
    c. Connect the new terminals to the appropriate subVI terminals.
    d. Edit the subVI icon to include the new terminals.
    e. Delete Register Bus wires in the top-level VI and rewire as necessary.
- The top-level host VIs reveal errors in dependencies that are related to a mismatch between the resource names in the project and those configured on FPGA Host Interface Nodes:

Error	Host Tx Stream	The FIFO name does not exist or is not writable. FIFO name: Library\5793 Resources.grsc\Host Tx Stream
Error	Host Tx Stream: data	This terminal's type is unsupported on the current compilation target. Type: Void
Error	Host Tx Stream: data	Type conflict. You have connected a scalar (non-array) data type and an array type. Wired type: 1D Array of U64, Terminal type: Void
Error	Host Tx Stream	The FIFO name does not exist or is not writable. FIFO name: Library\5791 Resources.grsc\Host Tx Stream
Error	Host Tx Stream: data	This terminal's type is unsupported on the current compilation target. Type: Void
Error	Host Tx Stream: data	Type conflict. You have connected a scalar (non-array) data type and an array type. Wired type: 1D Array of U32, Terminal type: Void

Note: Due to the fact that the conversion process moved the **579x Resources.grsc** Resource Collections into a Library document (see FPGA section below), the resource names in the project no longer match those compiled into the bitfile. To resolve the issue, update each affected Host Interface node to point to the correct resource name in the existing bitfile.

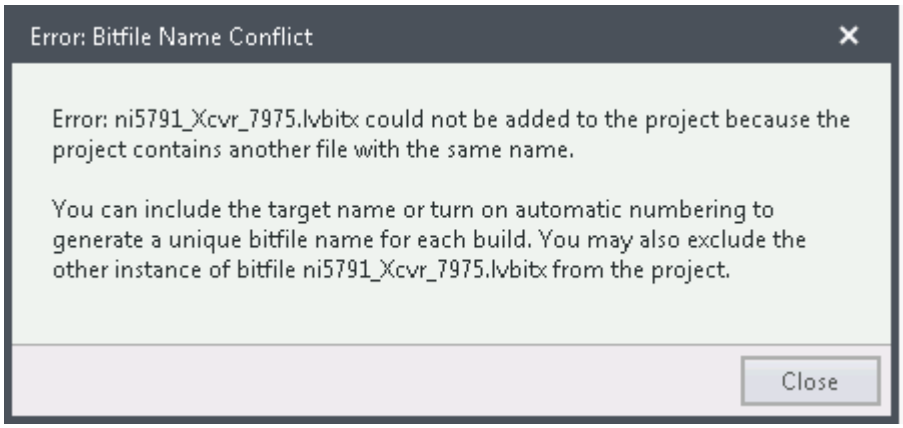
- The conversion process moves all subVIs into a **Host SubVIs** namespace in the same Library document as the top-level host VIs:

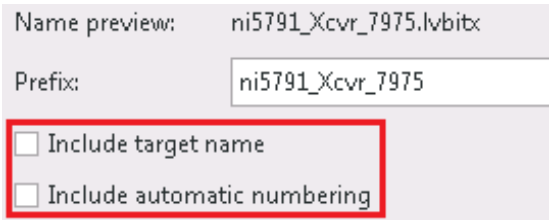


FPGA

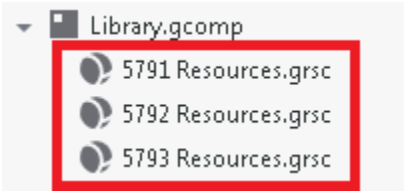
- The conversion process moves the top-level FPGA VI for each PXIe-7975R target into a **Bitfile*X*.gcomp** Application document, where *X* is a unique number identifying the build name from the original project.
- To recompile any of the bitfiles, click on the appropriate Application document and click the **Build** button.

  Note: After pressing the build button, an error dialog may appear regarding a bitfile name conflict. To resolve this issue, open the affected Application document and select the **Include target name** or **Include automatic numbering** checkboxes.





- The conversion process moves all Resource Collections into a single Library document that is reused by each target:



- The conversion process moves all External IP and Clock-Driven Logic documents into a single Library document within an FPGA SubVIs namespace: