NXP Semiconductors
User's Guide

Document Number: QCVS_FDW_User_Guide
Rev. 4.x, 02/2017

QCVS Frame Distributor Wizard User Guide

Contents

Chapter 1
Frame Distributor Wizard

This document describes how to configure the packet processing functionality of a QorIQ device using a tool, called *Frame Distributor Wizard (FDW)*.

This tool is part of the QorIQ Configuration and Validation Suite (QCVS) product.

NOTE

The Frame Distributor Wizard component of QCVS is only available for projects created with B, P, or T family of processors.

This chapter is divided into the following sections:

1.1 Introduction

FDW helps you quickly configure the parse-classify-distribute (PCD) functionality of the Data Path Acceleration Architecture (DPAA) hardware block of your QorIQ device.

The key idea is to reduce the time spent in doing configuration work so that more time can be spent in writing the application that uses the configuration.

FDW provides you with a series of pages to configure the packet processing functionality of your QorIQ device. The focus is more on how to process the packets rather than how the DPAA block internally works. FDW pages not only guide you to make the configuration but also prevent you from making incorrect settings.

The output of FDW is XML code, which is further used as frame manager driver (FMD) API for configuring the PCD functionality of the DPAA block. Therefore, FDW represents the highest abstraction level in the software toolchain responsible for PCD configuration.

FDW is available for all QorIQ devices that feature DPAA IP block. See QCVS Release Notes for details on which QorIQ SoCs are supported by FDW.

This section contains the following subsections:

1.1.1 Acronyms

This section provides a list of all acronyms used in the current document.

Table 1. Acronyms

Acronym	Meaning
DPAA	Data path acceleration architecture. It is a Freescale proprietary packet acceleration block.
FDW	Frame Distributor Wizard
FMan	Frame manager. It is a component of the DPAA IP block that incorporates the network interfaces and is responsible for PCD operations.
FMC	Frame manager configuration. It is a tool that configures the PCD DPAA starting from an XML abstraction of the block functionalities.
FMD	Frame manager driver. It is a driver provided in the QorIQ Linux SDK. It configures the PCD of the DPAA hardware block.
FQ	Frame queue. It is a zone of memory where a frame is sent after PCD operation, and from where it can be dequeued to a destination.
GPP	General purpose processor. It is a core responsible for running a general purpose application, such as an OS or bare-metal application.
Net PDL	Net protocol description language. An open standard language that formalized the definitions of networking concepts (for example, a protocol header definition).
OSI model	Open systems interconnection model. A communication standard between computing systems irrespective of their underlying structure or technology.
PCD	Parse-classify-distribute. It is a feature of the DPAA hardware block that is responsible fot packet processing activities.
QCVS	QorIQ Configuration and Validation Suite
QMan	Queue manager
Rx	Receive/receiver
SDK	Software development kit
SoC	System-on-chip. A collection of cores, buses, memories, and peripherals located on the same silicon die.
Tx	Transmit/transmitter
UDP	User datagram protocol

1.1.2 DPAA hardware block

The DPAA hardware block (also known as DPAA block) is a Freescale proprietary IP block responsible for packet acceleration in QorIQ-based SoCs.

To a great extent, it offloads the packet processing responsibility from the GPP core.

One of the key functions of DPAA is PCD, which defines how the incoming traffic will be analyzed, and if it will then be dispatched to next processing block (for example, GPP core) or other accelerator (for example, security block), or will be dropped.

As the traffic type can vary a lot, and considering the high throughput required now-a-days, the challenge is to program the PCD quickly and effectively, while taking advantage of all its features. A robust PCD programming is required before addressing the higher-level software that consumes the traffic (for example, uses control packets to determine if the network is operational) at the application level.

For more details on the DPAA architecture, see the DPAA reference manual for your QorIQ SoC.

1.1.3 Creating a new FDW project

The first step in configuring PCD is to create a QCVS project with an FDW component.

To create a QCVS project, follow these steps:

1. Start CodeWarrior Development Studio.
2. Select **File > New > QorIQ Configuration Project**. The **New QorIQ Configuration Project** wizard starts, displaying the **Create a QorIQ Configuration Project** page.
3. Specify the project name in the **Project name** text box, and click **Next**.
4. On the **Devices** page, choose a device to work with. See QCVS Release Notes for details on which QorIQ devices are supported by FDW.
5. Now, choose a device revision. If only one revision is available, it is automatically chosen. Click **Next** to continue.
6. On the **Toolset selection** page, select **Frame Distributor Wizard** (see Figure 1. Choose FDW from Toolset selection page on page 5), and click **Next** to continue.

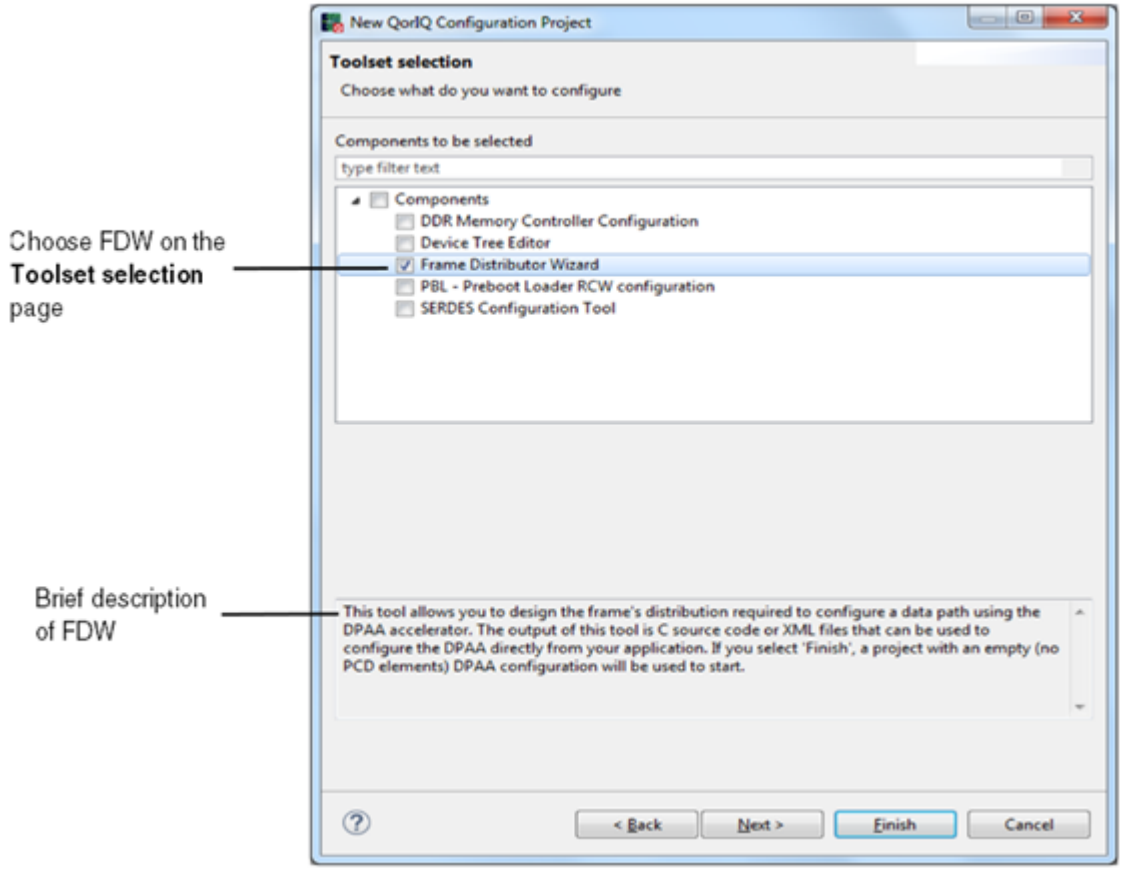

Figure 1. Choose FDW from Toolset selection page

7. Create an FDW component. Use one of the following options to start creating an FDW component:

- Empty configuration: Creates a configuration without any PCD processing elements. PCD configuration can be done later using the wizard.
- Load an example PCD configuration: Allows you to choose a PCD configuration from a rich set of example PCD configurations provided by FDW (see Figure 2. Load an example configuration on page 6), and then refine the configuration to meet your requirements.

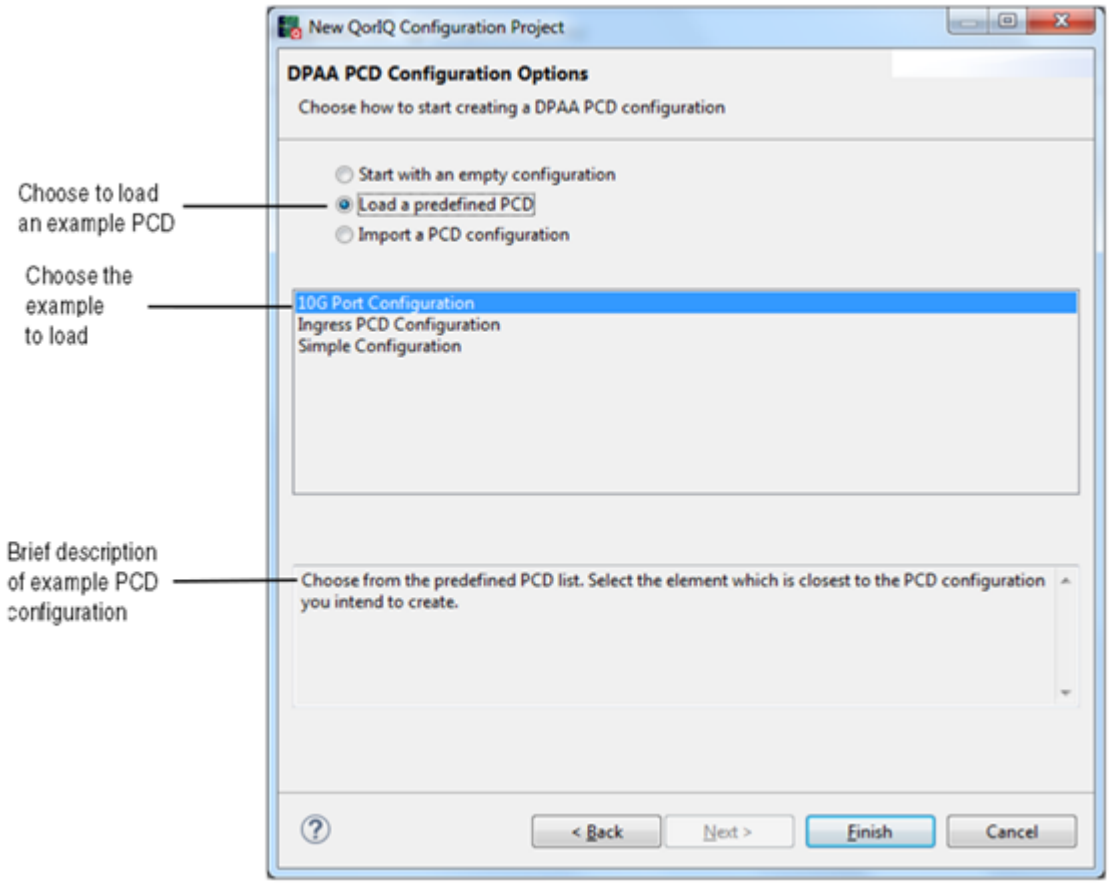

Figure 2. Load an example configuration

- Import an existing PCD configuration: Allows you to import into the tool any existing XML configuration of PCD (see Figure 3. Import from an existing PCD configuration on page 7), and then refine the configuration further. This is particularly useful when exchanging the PCD configurations between different users, or adopting the FDW tool while preserving any previously created PCD configurations.

NOTE

The import option requires two XML files: one defining PCD and the other defining how various flows are connected to the frame manager ports.

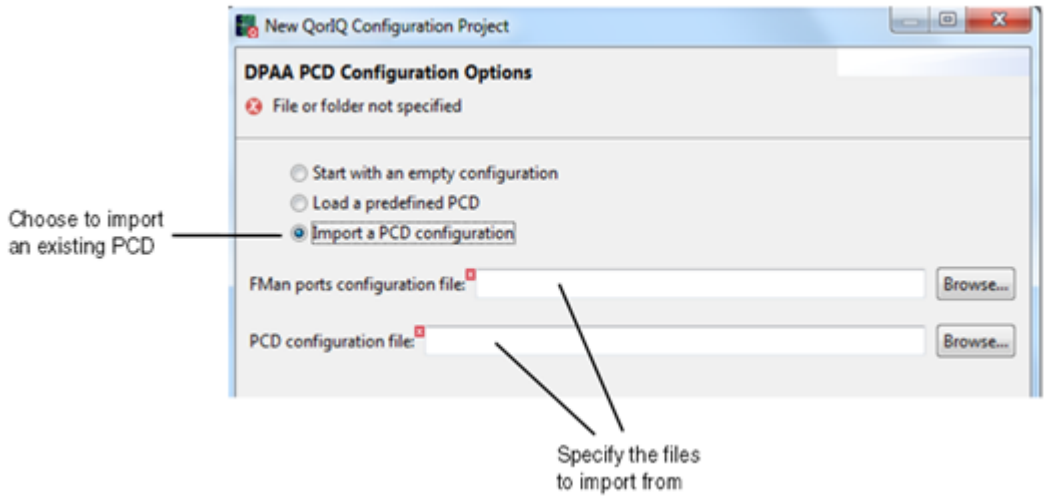

Figure 3. Import from an existing PCD configuration

8. Click **Finish** to create the project.

When you create a new QCVS project, a DPAA component is added to the project. The DPAA component contains a PCD configuration, which you can use in your project and customize as per your requirements. To open the DPAA component, right-click the component in the **Components** view and choose **Inspector** from the shortcut menu. The DPAA component opens in the **Component Inspector** view, displaying the **Frame Distributor Wizard** page (see Figure 4. Open DPAA component on page 7).

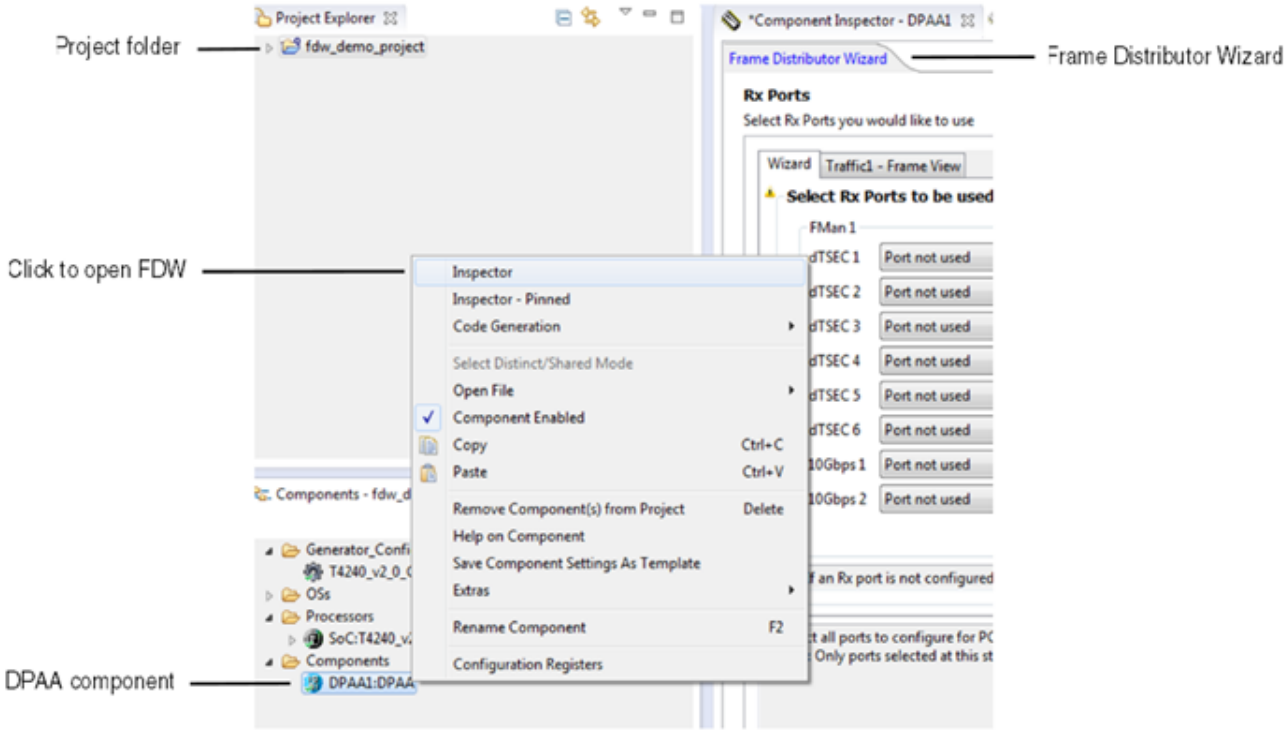

Figure 4. Open DPAA component

1.1.4 Frame Distributor Wizard concept

This section describes the concept behind FDW.

PCD is responsible for analyzing and dispatching traffic that is organized in frames. There can be various traffic types; most of them are organized in standard frames, such as Ethernet, whereas others can be custom types.

FDW allows you to define different types of traffic and then define PCD processing rules to be applied to that traffic, based on what the acceleration hardware (for example, DPAA) is capable of. In short, FDW enables you to define PCD rules for different types of traffic.

Any traffic received from the frame manager ports is sent through PCD and then dispatched to a destination, such as another frame manager port, a GPP core, another PCD, or an accelerator IP block (for example, SEC). FDW follows the potential paths of the traffic using a wizard that guides you through the configuration of each path element. The figure below shows the **Traffic Type** page of FDW. The navigation buttons (**Back** and **Next**) shown at bottom right corner of the figure allows you to navigate between the different pages of the wizard.

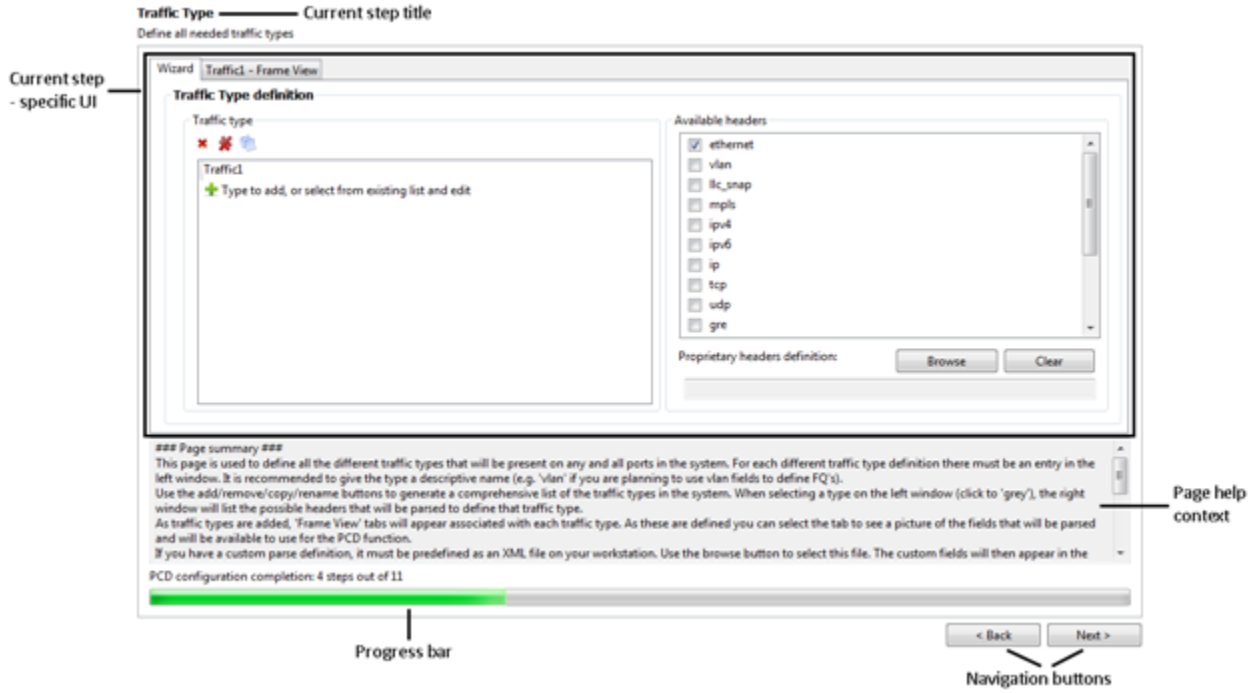

Figure 5. Traffic Type page of FDW

1.1.5 PCD design considerations

The concept behind PCD is that the traffic received through a frame manager port is first analyzed, and then either dispatched to a destination or simply ignored.

The possible destinations are:

- A frame manager port
- An acceleration block (for example, SEC)
- A GPP core
- Another PCD

The QorIQ SoC DPAA block features several frame manager ports operating at 1 Gbit/s and 10 Gbit/s speeds, allowing you to design a complex PCD architecture to handle various traffic types. PCD design starts with designing a system that answers the following questions:

- What kind of traffic do you expect to receive?

- What traffic type are you interested in?
- How can you optimally use the acceleration resources to process the traffic you need?

PCD design requires you to define a flow, which represents a series of chained processing elements (parsing, classification, and distribution) that connect to an FMan port. Each flow is usually responsible for a specific type of traffic (for example, the control frames received over UDP or the Ethernet traffic that needs further processing by the SEC engine). You can define a number of flows depending on:

- The nature of the traffic expected to be received by the FMan port
- The type of processing that is expected to be applied to a traffic type
- The number of available FMan ports on the SoC

An FMan port can have multiple flows associated with it. This is usually done when several types of traffic requiring different processing is expected to be received through the FMan port.

FDW allows you to design a PCD configuration from the following starting points:

- Start with a blank configuration.

  Step-by-step instructions allow you to create a PCD configuration based on traffic type and how that traffic is being analyzed. The options and constraints that FDW provides for PCD configuration facilitate the creation of truly optimized PCDs.

- Load example configuration.

  The provided examples represent standard PCD use cases, suitable for most PCD design paradigms. You can refine a configuration according to your system design. This saves your time that you would otherwise spend in creating the PCD configuration from scratch.

- Import an existing PCD.

  If you are an existing DPAA user and you previously created a PCD configuration using a standard text editor, then you can easily transition to FDW, by importing your existing PCD XML file to FDW. After importing the PCD file, you can continue to modify your configuration using the features provided by FDW.

1.2 PCD configuration using FDW

This section explains how to configure PCD using FDW, in a step-by-step manner.

Traffic is packaged into frames that contain elements from various OSI model layers (for example, data link, network, or transport). In FDW, each PCD operation applied to the traffic frames is captured in *frame view*. Frame view represents the network protocol fields that are impacted by a PCD operation in the context of their corresponding OSI model layer. For more details, see Frame View page on page 19.

The following major steps are followed to configure PCD using FDW:

1. Choose FMan ports on page 10
2. Define traffic types on page 11
3. Map traffic types to FMan ports on page 12
4. Define FMan port traffic processing elements on page 13
5. Define flows on page 14
6. Configure flow elements on page 15
7. Generate PCD configuration files on page 18
8. Use PCD configuration files on page 19

1.2.1 Choose FMan ports

The first page of FDW is the **Rx Ports** page, which allows you to choose the FMan ports to be configured for PCD.

Each SoC containing the DPAA IP block has a different number of FMan IP blocks, each supporting a different number of FMan ports and/or port speed. For each FMan port supported by the SoC, you can choose the following options:

- **Port not used**: Indicates that no PCD will take place over the traffic received at this port (default option)
- **Use FMan port**: Indicates that the traffic received at this port is a candidate for PCD operations
- **Use FMan port with default Linux PCD information**: Indicates that the port will be preconfigured with the PCD rules in sync with the QorIQ Linux SDK default setup

Figure 6. Rx Ports page on page 10 shows the **Rx Ports** page.

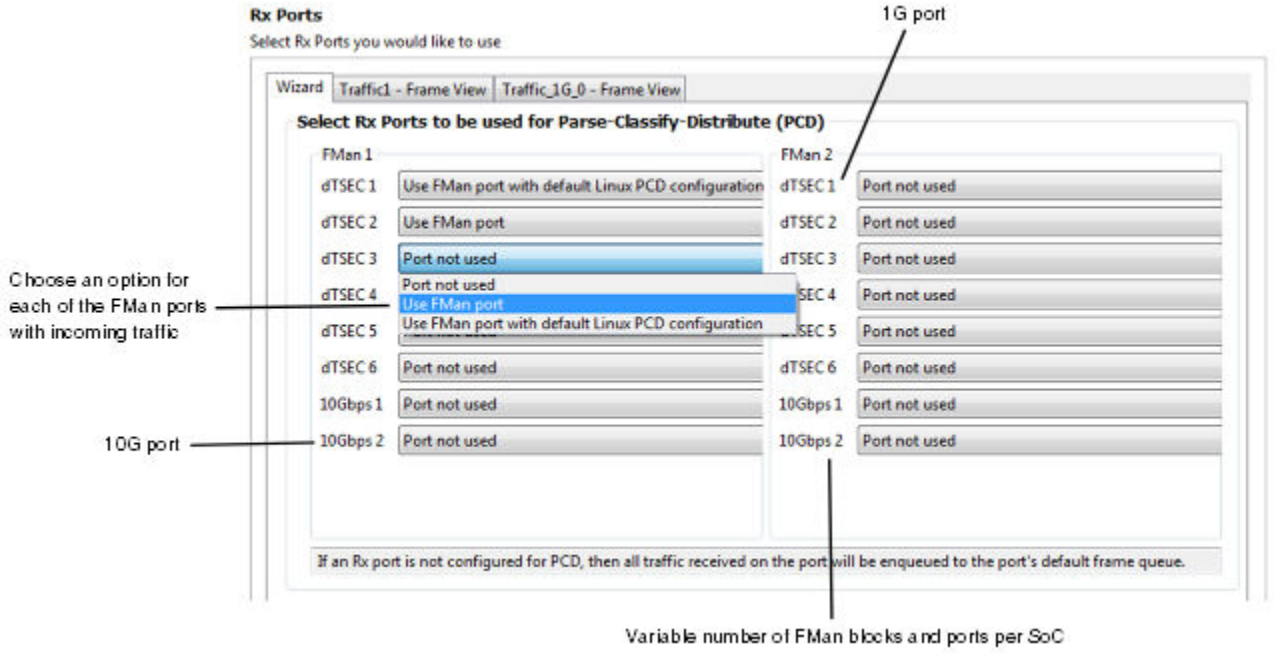

Figure 6. Rx Ports page

NOTE

The "default Linux PCD configuration" option for a port implies that the port will be configured with the PCD settings provided in the Linux QorIQ SDK package. If this option is not available, it means that no such default PCD configuration is provided in the QorIQ SDK.

The next page of FDW is the **Offline Ports** page, where you can choose the FMan offline ports you want to use. An FMan offline port is a mechanism by which you can send traffic from one PCD to another PCD, creating a chain of PCDs. The figure below shows the **Offline Ports** page. Select **No** if you do not plan to use any offline port.

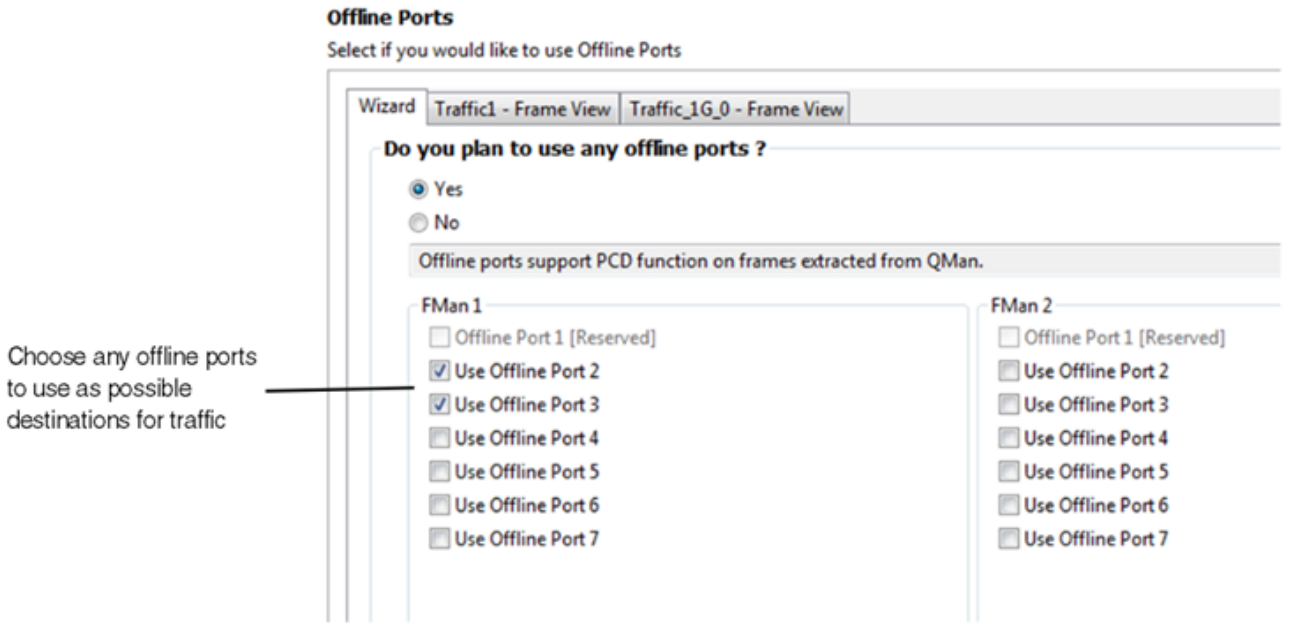

Figure 7. Offline Ports page

The next page provides a summary of the FMan ports (Rx and offline) that were chosen to be configured for PCD.

1.2.2 Define traffic types

The **Traffic Type** page of FDW allows you to define the traffic expected to be received through the assigned FMan ports.

Traffic type definition indicates the type of headers the packets will have. The header identifies the packet type, whereas the payload contains the effective data.

Define traffic types by specifying a name and the expected packet headers for each traffic type. At least one expected header must be chosen for a particular traffic type.

The figure below shows the **Traffic Type** page.

Figure 8. Traffic Type page

As shown in the figure above, a **Frame View** page appears for each defined traffic type that describes, in the OSI layer style, the fields of the protocol headers that the traffic type is expected to contain. For more details, see Frame View page on page 19.

This section contains the following subsection:

1.2.2.1 Use proprietary header definitions

This section explains how to use a custom protocol header definition.

You can choose header types from a known list of standard headers (for example, Ethernet, UDP). However, in some cases, the expected traffic does not comply with any standard protocol definition. In such cases, the DPAA block allows you to load a custom protocol header definition (see figure below) written in NetPDL, a mark-up language designed to describe protocols used in OSI layers 2-7.

After a custom protocol header definition file is loaded, the protocol header defined in the file gets added to the list of available headers that can be selected when defining traffic types. You can load up to two proprietary header definitions. If the NetPDL definition of the proprietary header is not correct, then an error message is displayed and also a notification icon is added in the left side of the text box showing path to the header definition file.

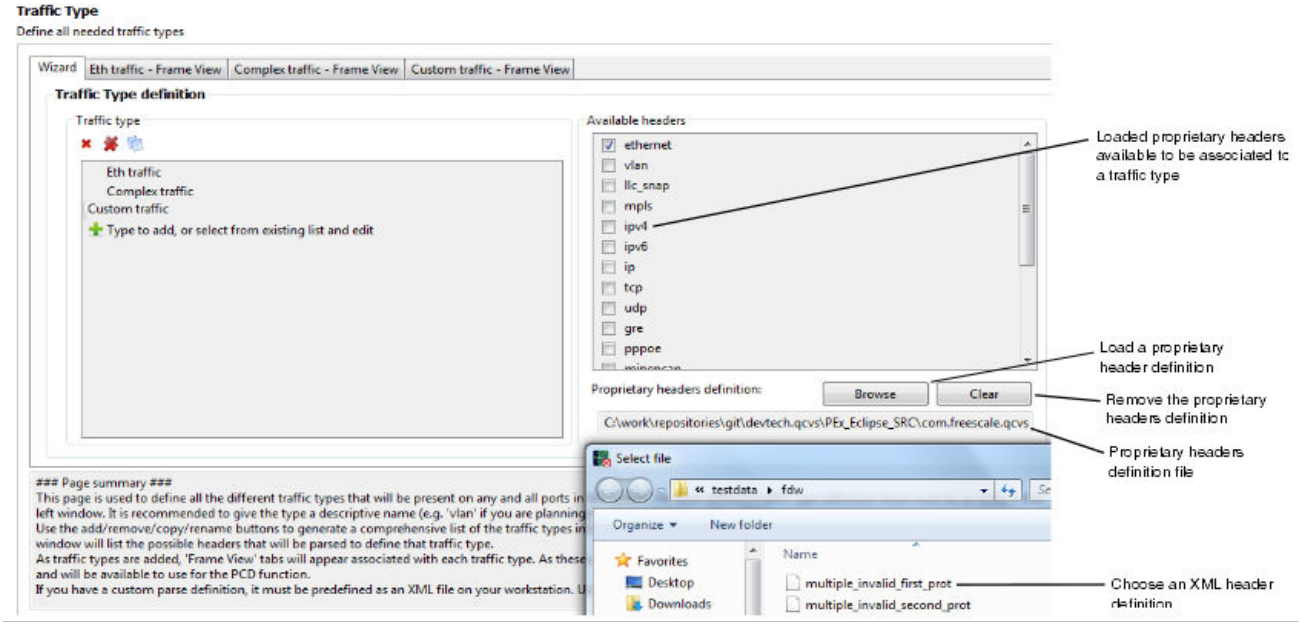

Figure 9. Use proprietary header definitions

1.2.3 Map traffic types to FMan ports

The **Traffic Mapping** page of FDW allows you to bind the traffic types to the selected FMan ports.

By making this binding, you specify that on a specific port, a particular traffic type is expected to be received and processed in the PCD stages.

You can map multiple traffic types to one FMan port. The figure below shows the **Traffic Mapping** page.

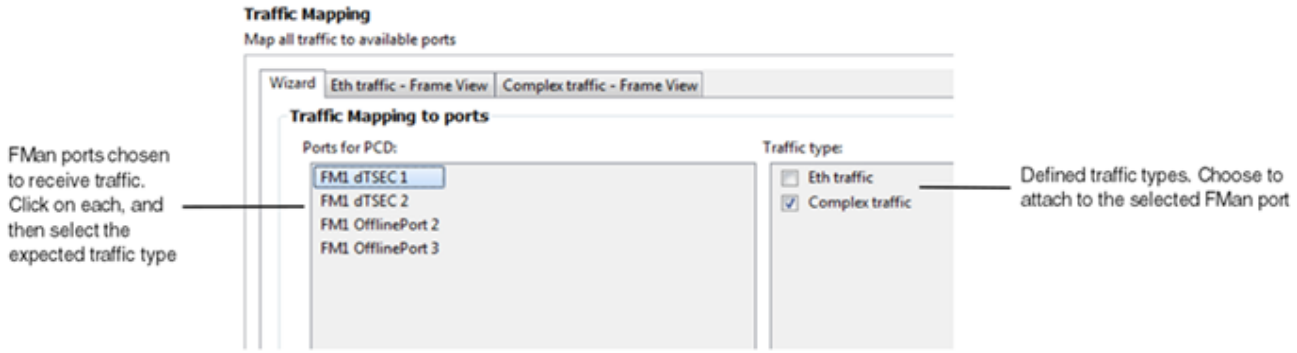

Figure 10. Traffic Mapping page

1.2.4 Define FMan port traffic processing elements

The **Examining packet headers** page of FDW allows you to define the type of processing that the traffic assigned to each FMan port will undergo.

For defining the type of processing, the following three methods are available:

- Hash Flow definition strategy (H)

  A hash flow definition will use a key (selected as one or more header fields later) as an input to a hash algorithm that will enable a consistent but psuedo random spreading of traffic among queues.

- Protocol presence verification (PPV)

  A protocol presence verification allows traffic to be routed to a queue based on the fact that a particular header was identified by the parser. In addition, a protocol presence verification can be used in conjunction with a table lookup. In this case, the protocol presence that is verified is typically the protocol(s) corresponding to the field(s) used in the table lookup.

- Table lookup (TL)

  A table lookup definition allows you to perform a precise match of header fields to define the traffic associated with a queue. There are "if/then" mechanisms available with table definition that will be configured later.

You can choose any combination of these operations; however, choosing at least one operation is necessary. The figure below shows the **Examining packet headers** page.

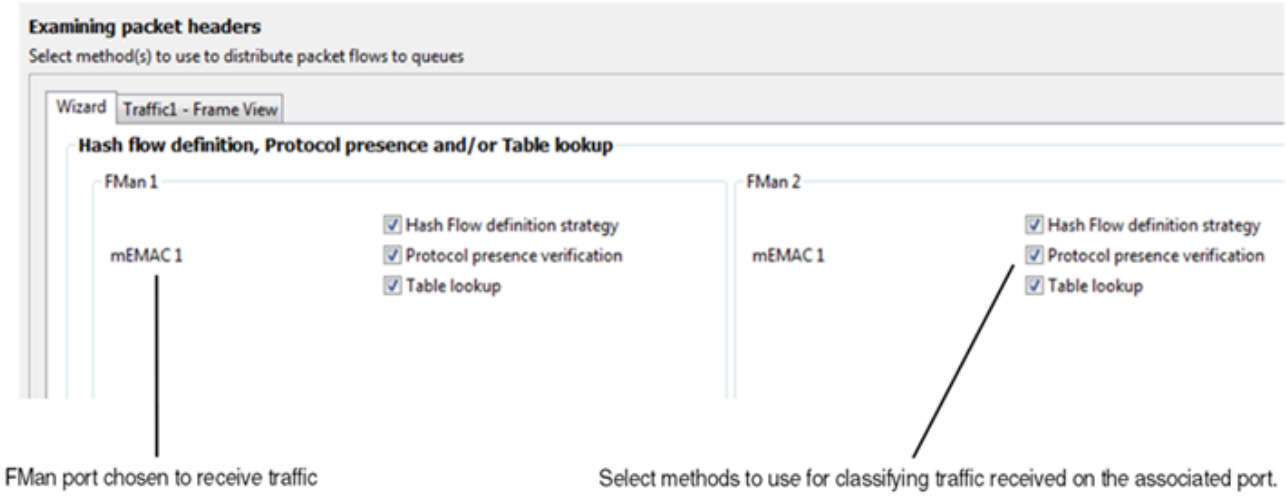

Figure 11. Examining packet headers page

1.2.5 Define flows

The **Flows definition** page of FDW lists the flow names that will be attached to each FMan port.

On the subsequent FDW pages, you will define the hash flow definition, protocol presence verification, and table lookup operations to be associated with each flow name.

Multiple flows can be attached to an FMan port. By default, the flows are displayed in the order they are created but this order can be changed via up and down buttons (see Figure 12. Flows definition page on page 14). The order of flows specified on the **Flows definition** page is used by FMan to associate an Rx frame with a flow.

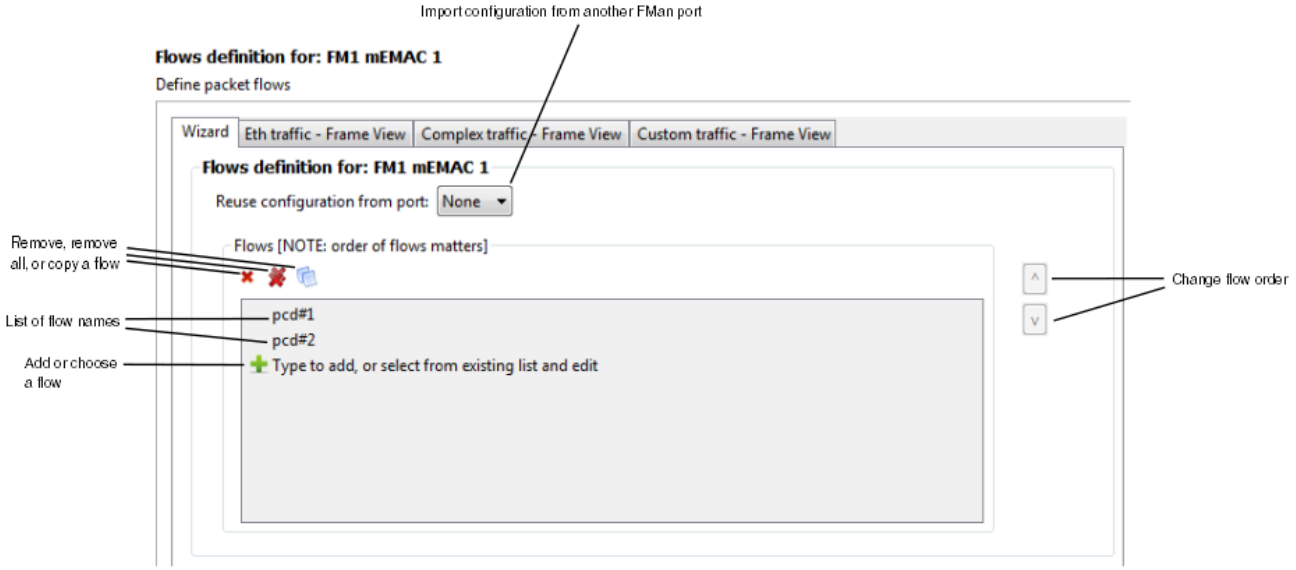

Figure 12. Flows definition page

This section contains the following subsection:

- Decide whether to use single or multiple flows on page 14

1.2.5.1 Decide whether to use single or multiple flows

This section presents some design considerations for deciding the number of flows to be used.

When deciding how many flows will be used in the system, you need to take into consideration the traffic and overall system design. The following use case presents some of these design considerations.

Suppose that you want to process Ethernet and UDP traffic coming into the same FMan port 1. Then, you have two ways to design the flows.

The first approach is to:

1. Define one traffic type with both Ethernet and UDP headers in it and assign it to FMan port 1.
2. Define one flow in which table lookup and hashing will be performed on the combined Ethernet and UDP bit fields.

The second approach is to:

1. Define two traffic types, one for Ethernet and the other for UDP.
2. Define two flows, one performing PCD operations for Ethernet traffic, and the other for UDP traffic. The UDP flow should be listed first, followed by the Ethernet flow. Then, when Ethernet or UDP traffic arrives on FMan port 1, each traffic type will be handled by a separate flow.

Therefore, the advantage of using separate flows for separate types of traffic is that it results in a much cleaner design. However, if many types of traffic are expected to be processed, then using one flow for each individual traffic type may not be optimal from an acceleration resources utilization point of view.

1.2.6 Configure flow elements

This section explains how to configure the traffic processing elements for the chosen flows.

Depending on the options chosen in Define FMan port traffic processing elements on page 13 (that is, Hash Flow definition strategy, Protocol presence verification, or Table lookup), a series of pages dedicated to the configuration of each PCD option will appear. For each chosen FMan port, the H, TL, or PPV dedicated pages are dynamically added to the wizard. You can specify additional settings on the H, TL, and PPV pages.

Consider the following points while configuring flow elements:

- Some of the configuration options are mandatory, such as the FMan ports used, the traffic type to be processed, and the flows to be used for processing that traffic. Other configuration options are optional, and if not configured, they will be handled as default, which usually means that the traffic will not be processed for these options and it will be sent as is to the next processing phase.
- All PCD elements (H, TL, and PPV) are configured per flow for each FMan port selected in Choose FMan ports on page 10. The number of wizard pages is dynamically adjusted to drive you through the PCD configuration for all FMan ports.
- If the flows for one FMan port have been configured, then that configuration can be reused for another port. A menu is available at the top of the H, TL, or PPV page that allows you to reuse the configuration from another port (see Figure 13. Hashing definition page on page 16).

The following subsections explain how PCD elements are configured using FDW:

- Hashing on page 15
- Protocol presence verification on page 16
- Table lookup on page 17

1.2.6.1 Hashing

The **Hashing definition** page of FDW allows you to specify hash settings for each flow assigned to an FMan port.

A hash key is formed by selecting the header fields from the expected traffic types. The **Hashing definition** page only appears if this method was selected in Define FMan port traffic processing elements on page 13.

For example, suppose that for FMan port 1, a traffic type with Ethernet header was defined. Then, you will select a hash key from the Ethernet header fields (ethernet dst, ethernet src, and ethernet type). One or more fields can be chosen to form the hash key.

The destination of the traffic matching the hashing key is set by specifying the frame queue range (see Figure 13. Hashing definition page on page 16). When the traffic reaches a frame queue from that range, it will be dequeued to one of the possible destinations (a GPP core, an accelerator, or an FMan offline port). Queue manager (QMan) is responsible for managing the enqueueing and dequeueing of data between FMan, GPP cores, and accelerators. The figure below explains the different actions that can be performed on the hashing page.

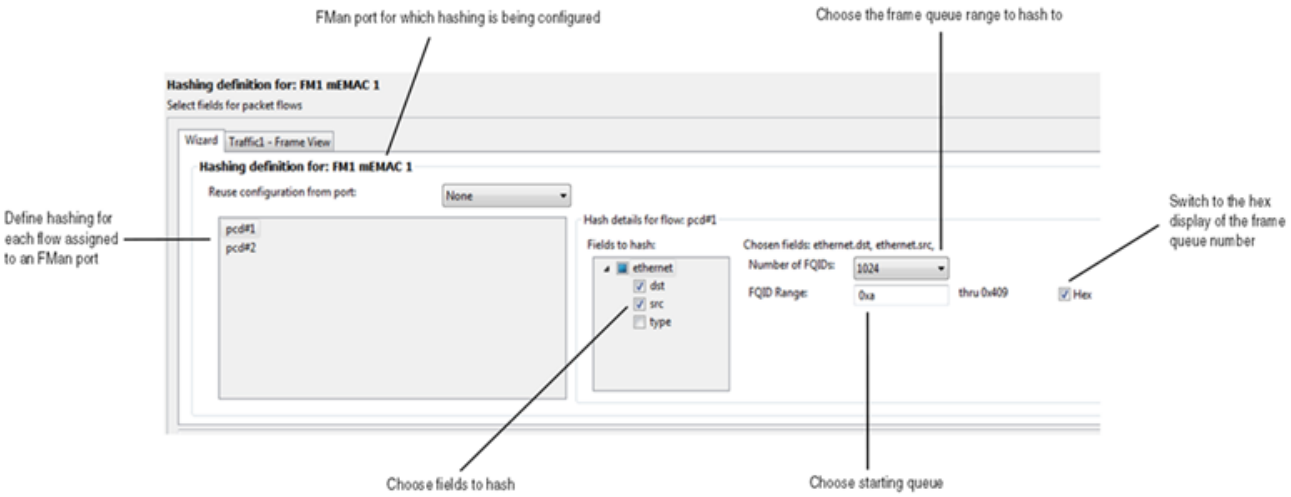

Figure 13. Hashing definition page

1.2.6.2 Protocol presence verification

The **Protocol presence verification** page of FDW allows you to verify whether or not a specific protocol header exists in a frame.

Protocol presence verification can be configured to check for the protocol headers defined in the traffic types of the FMan port that the flow belongs to. The **Protocol presence verification** page only appears if this method was selected in Define FMan port traffic processing elements on page 13.

If the TL operation was also selected, then the option to do PPV over the protocols used in building the table lookup keys will become available. Therefore, you can quickly sync PPV with TL by just selecting a checkbox.

If the hashing operation was not selected, then you need to provide the frame queue ID for the frame queue where the frames, which passed the protocol presence verification test, will be dispatched.

The figure below explains the different actions that can be performed on the **Protocol presence verification** page.

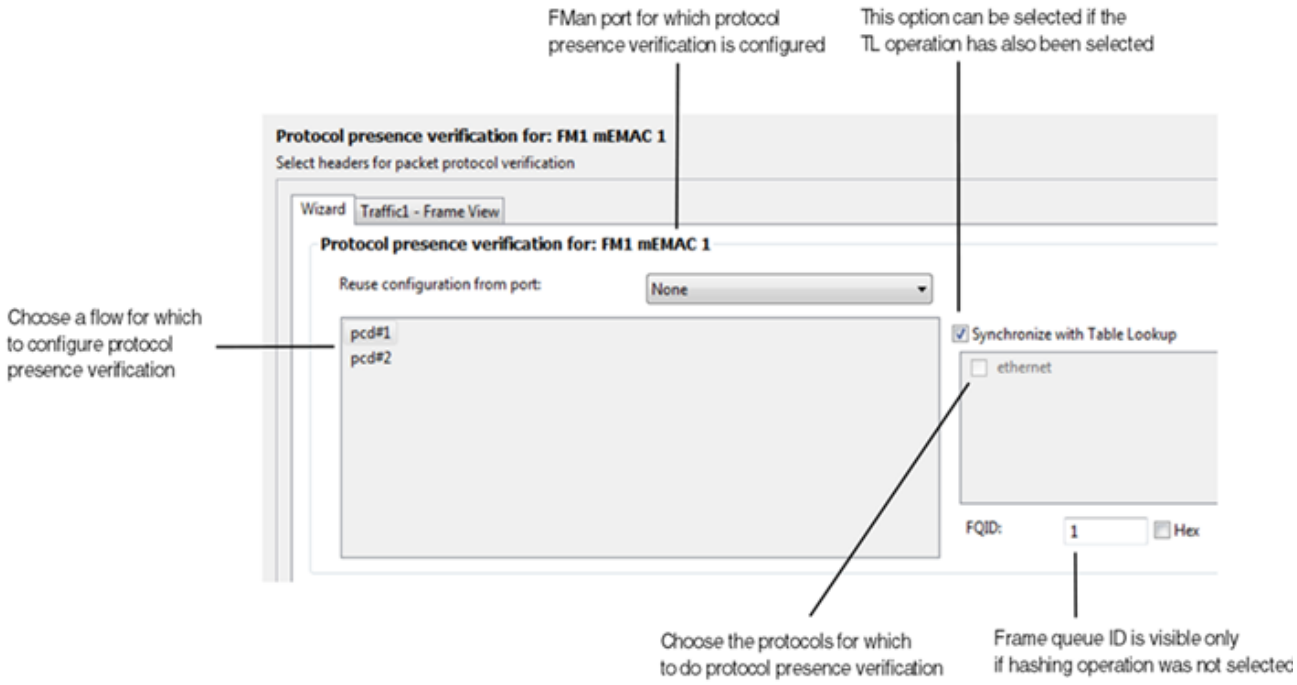

Figure 14. Protocol presence verification page

1.2.6.3 Table lookup

The **Table lookup** page of FDW allows you to define a set of actions to be performed for each frame when a condition is met.

In short, if a frame content meets a condition, then dispatch it to the next processing element or drop it.

The defined conditions are resolved in the order of their definitions. The **Table lookup** page only appears if this method was selected in Define FMan port traffic processing elements on page 13.

The table lookup processing elements can be nested. As per one flow, multiple lookup nodes can be defined. You can nest the lookup nodes using the `Goto` action explained later in this section.

A condition can be of one of the following two types:

- on-hit: The table lookup operation determines if this condition is met by tracking the packets and checking packet content against a user-specified value (for example, `ipv4.src` matches a specific IP address range).
- on-miss: There can only be one condition of this type. This condition is true when all other conditions were not met. If you do not define the on-miss condition, it is automatically set by FDW.

The action defined for a condition takes place only if the condition is met. The action can be of one of the following types:

- Enqueue frame: Sends the frame to a specified Frame Queue ID, from where the frame is dequeued to one of the possible destinations.
- Drop frame: Ignores the frame and does not relay it further to any PCD processing element.
- Goto: Sends the frame to the next table lookup processing element, or to a completely different flow. This allows nesting of the table lookup elements, and also allows passing traffic between flows.

When the Goto action is selected, <New table> can be chosen as the destination, which allows you to write in the name of a new lookup table that you want packets to go to. By writing in a name for the table, a new table is created in the lookup tree, which can then be configured.

The figure below explains the different actions that can be performed on the **Table lookup** page.

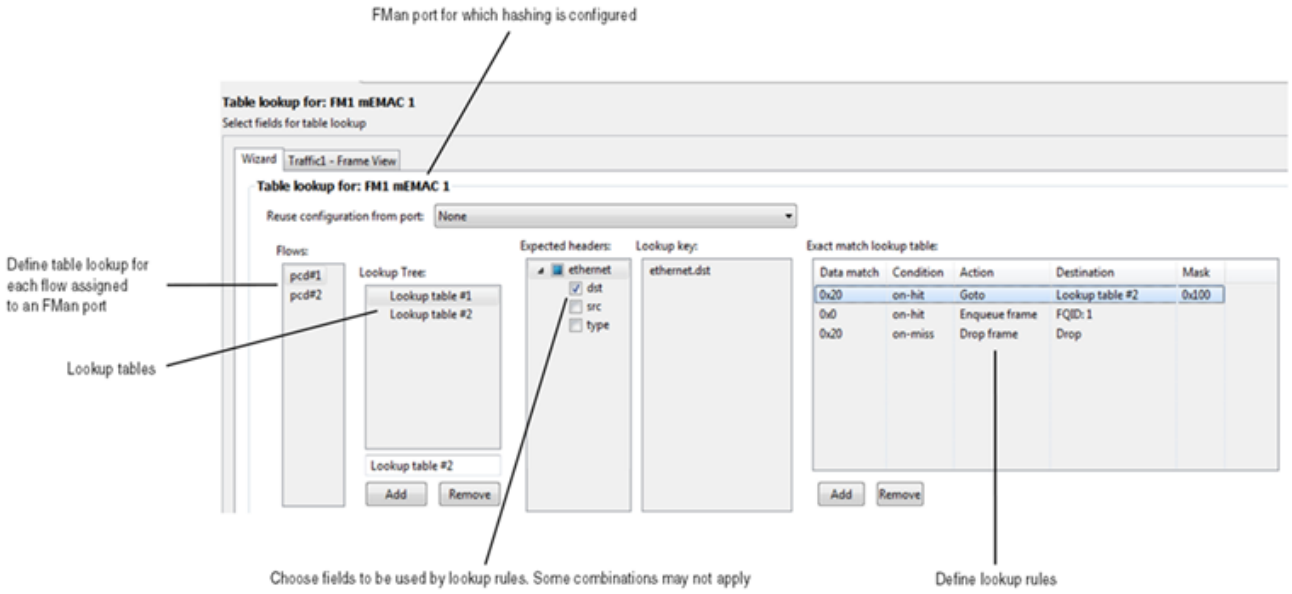

Figure 15. Table lookup page

1.2.7 Generate PCD configuration files

After the PCD configuration is completed, the last FDW page displays, in the text format, a summary of the selected PCD configurations.

To revisit or change the PCD configuration, use the **Back** and **Next** buttons.

Using these buttons, you can navigate through the previous wizard pages, which will adjust based on the changes done. For example, if you go back and add a new FMan port to use, and then start going forward, additional pages related to the new FMan port configuration will appear.

When the PCD is ready to be deployed on the target, you can proceed to generate the XML files containing the PCD definition.

Besides the PCD configuration file, a C file is also generated that contains the PCD defined as a C structure. This code can be embedded into an SDK application using FMD. Therefore, you can control at a lower level the PCD initialization that is done using FMD. The figure below shows how to generate the PCD code.

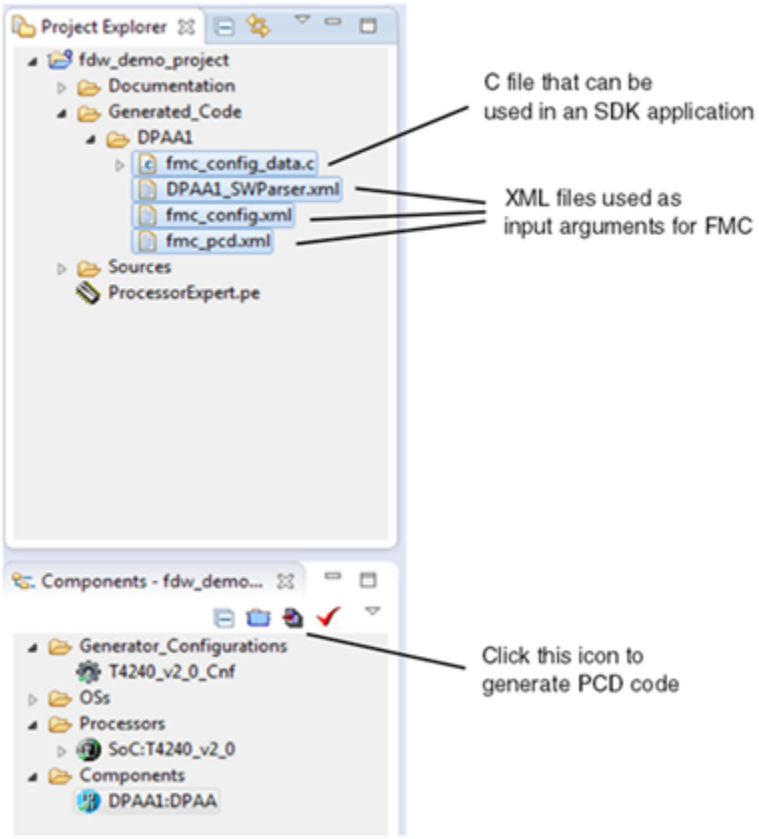

Figure 16. Generate PCD code

1.2.8 Use PCD configuration files

This section tells what you can do with the generated PCD configuration files.

The PCD configuration files can be used as follows:

- Import them back into FDW and adjust the PCD configuration
- Apply PCD to a QorIQ target using the FMC tool. See Frame Manager Configuration Tool User Guide for instructions on how to use the FMC tool to configure the PCD of a QorIQ target.

1.3 Frame View page

This section explains the **Frame View** page in detail.

The DPAA IP block is intended to be used for packet acceleration. A network packet is organized in the form of a header and payload. The header describes the nature of the packet and also the OSI model processing layer that it went through. The DPAA block processes packets at the data link, transport, and network layers.

The **Frame View** page is intended to show an OSI model layer organization of the packet, highlighting the header fields involved in the PCD operations. This page provides read-only information, and it is updated each time the PCD configuration is changed. A **Frame View** page appears for each defined traffic type while navigating the FDW pages (see Figure 17. Frame View pages on page 20).

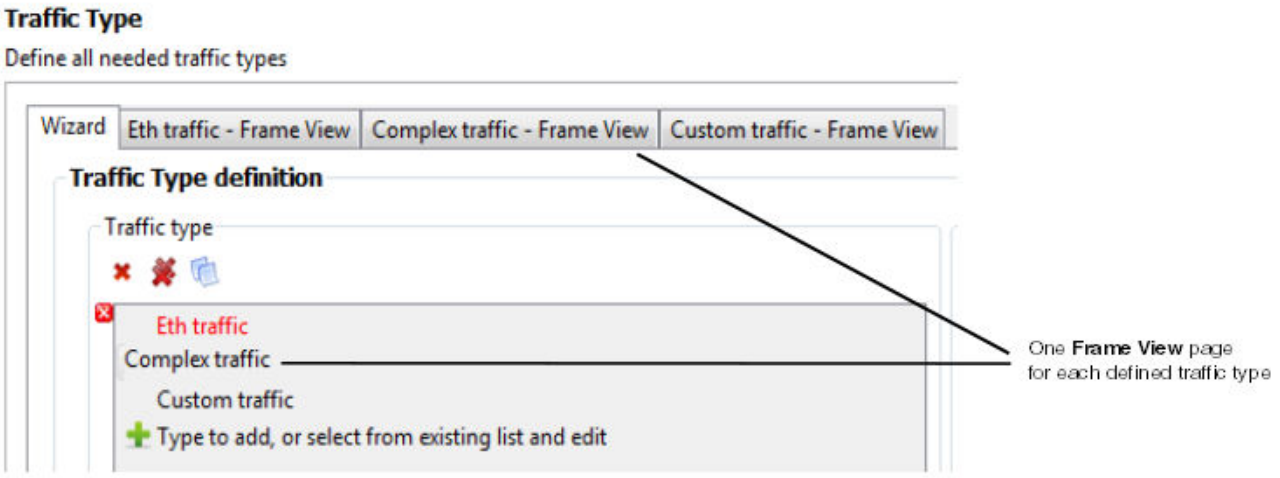

Figure 17. Frame View pages

Each time you make a modification to the PCD configuration that changes how packets from a traffic type are being processed, the **Frame View** page is updated with the impacted protocol header fields. For example, if hashing is done on some of the header fields of an Ethernet header, then those fields located at the data link layer (where Ethernet header is added) are highlighted in blue. The figure below explains the **Frame View** page content.

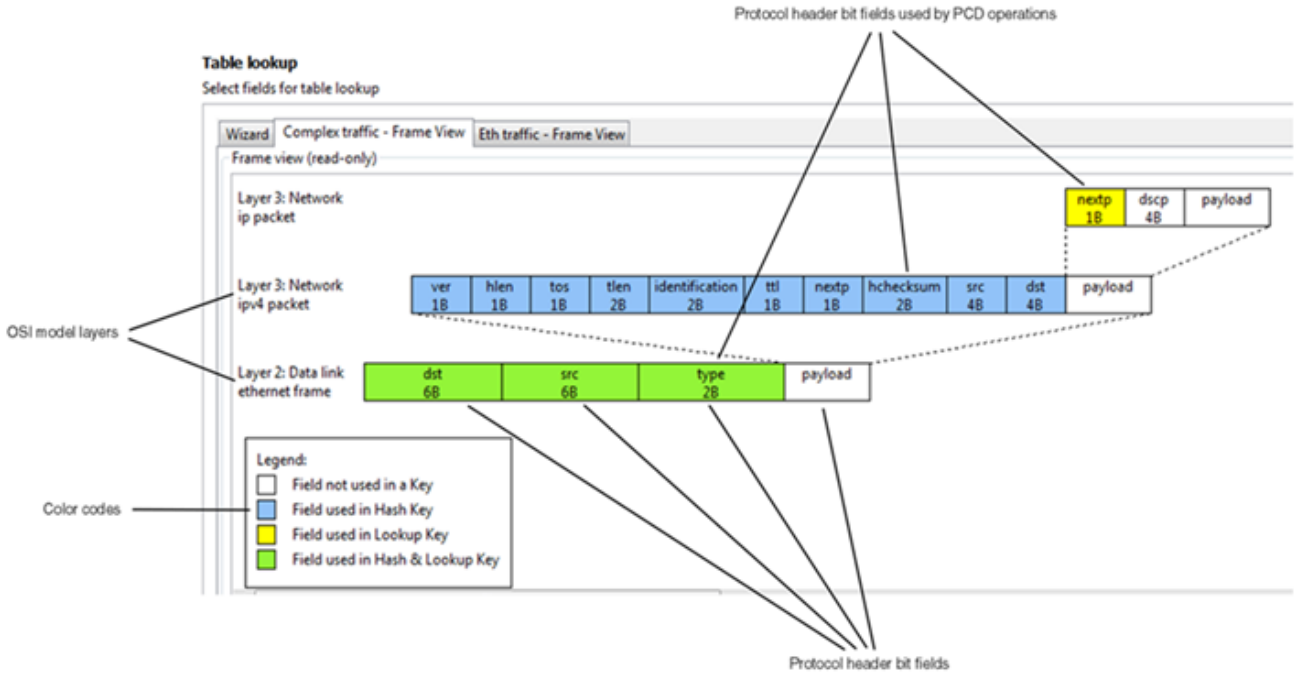

Figure 18. Frame View page content

Index

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

QCVS_FDW_User_Guide
Rev. 4.x
02/2017