



PPL Release Notes

Implementation Restrictions

Version 1.01
February 3, 2005

Copyright © IP Fabrics, Inc. 2005

IP Fabrics Corporate Headquarters
14964 NW Greenbrier Parkway
Beaverton, OR 97006
Telephone (main line): 503 444-2400
Telephone (FAX line): 503 444-2401
Website: <http://www.ipfabrics.com/>

Information in this document is furnished in connection with IP Fabrics products. No license, express or implied, to any intellectual property rights is granted by this document. This document and the software described in it are furnished under license and may only be used or copied in accordance with the terms of the license.

Copyright © 2005, IP Fabrics, Inc. All rights reserved.

Packet Processing Language™, PPL™ and PPL-VM™ are owned and copyrighted by IP Fabrics, Inc. Microsoft®, Windows® and Windows® XP are registered trademarks of Microsoft Corporation.

Linux® is a registered trademark of Linus Torvalds.

Red Hat® is a registered trademark of Red Hat, Inc.

MontaVista® is a registered trademark of MontaVista Software Inc.

Intel® and Pentium® are registered trademarks of Intel Corporation.

*Other brands, trademarks and names are property of their respective owners.

PPL Release Notes

This document summarizes release updates and defines restrictions in this release of the Packet Processing Language (PPL) compiler, virtual machine, accompanying software and supported hardware. One should also refer to the PPL specification in which there is a section entitled "Currently Unimplemented Language Parts and Restrictions."

The CDROM contains all the software that is required to load and run the system on the Intel IXP28x Network Processor. The supplied software can be used directly on an Intel IXDP2801 or IXDP2851 development system hardware platform. If you are receiving a hardware system from IP Fabrics then it may be configured to include a Processor PMC daughter card as its Linux boot host to make a single-blade ATCA solution platform.

CDROM

- PPL Virtual Machine software for the Intel IXP28xx NPU
 - Microcode software
 - "Target binaries" for XScale boot and execution
 - Scripts
 - XScale utilities (source and binary)
 - Statistics "exporter" for use with remote display application
 - Sample PPL binary programs
- PPL Compiler suite for Linux (on Pentium)
- PPL Compiler suite for Windows
- PPL sample source code
- Documentation
- License

Once the system is installed you should only need to provide your own system to telnet into the evaluation system along with a (Gbit) Ethernet packet data source in order to run the sample PPL programs and create your own PPL applications.

Sample programs that run on the XScale Linux system are provided. You do not need to create any XScale Linux software to develop PPL applications. However, if you do want to experiment with this part of the product by writing your own XScale programs then you may also need to acquire tools from Monta Vista (Monta Vista Linux Professional 3.1) www.mvista.com and from Intel Corporation (IXA SDK 4.0 Development Tools) www.intel.com. The Intel tools are included as part of Intel's offering for development systems and their IXP2xxx family of chip products.

The Linux kernel and other associated Linux utilities are covered by the General Public License (GPL). The text of this license is included for your reference.

Documentation

PPL - Packet Processing Language

Reference manual for the PPL language. This document describes the language concepts, usage with examples.

PPL IXP DeviceMap

Reference manual for the PPL DEVICEMAP statement for the Intel IXP2xxx family of Network Processors. This command is used to describe implementation and platform specific characteristics to the PPL compiler and Virtual Machine.

PPL Installation Guide

This document describes the specific setup procedures required to install the PPL system, the contents of the release package and external resources that are required to begin writing and executing PPL programs.

PPL Tutorial

This document is a simple step by step tutorial that guides you through the stages involved in writing, compiling, loading and running a simple PPL program.

PPL Release Notes and Implementation Restrictions

This document summarizes release contents and describes implementation restrictions in this release of the Packet Processing Language (PPL) compiler, virtual machine, accompanying software and supported hardware.

API description for XScale software

This document describes the public API calls that are used with the PPL-VM technology. It is accompanied by a README file that serves as reference material for each of the sample XScale applications which also are useful command line utilities used during development of PPL programs.

PPL Compiler – Choices for Development Environment

This release includes both Linux and Microsoft Windows versions of the PPL Compiler. This program can be run from the command line in either OS. IP Fabrics has also provided support for the Eclipse graphical user interface (“GUI”) environment. To use the GUI environment you must have a Java Runtime Environment (“JRE”) and the Eclipse environment installed on your development computer. Information on obtaining this software is provided in the Appendix section of the PPL Tutorial document. Portions of the GUI are covered by the Common Public License.

Implementation Release Restrictions

This section addresses PPL language features not in the initial 1.0 product release.

PPL Events.

- Event numbers. The event numbers must be in the set [0-253, 998, 999]. The total count of EVENT statements must be 127 or less.

PPL Rules.

- The sum of all rules in all events must be less than 4096.

PPL Policies.

- Future policies. The Cipher, Crypto, Decrypt, Discrypto, Superpacket, Newsuperpacket, Patterns and Ruleset policies are not included in this release.
- Packet Policy. The FORWARD keyword is not supported in the Packet policy. One can use the CURRENT keyword when applying the Packet policy and follow it with a FORWARD action to achieve the same result.
- Newpacket creation. ETHERNET header_type packets must be 8-byte aligned if they are to be sent out using the FORWARD command for L3 routing. Violation of this rule in an executing PPL program will result in PPL Exception 33, an implementation specific exception. Any alignment is allowable if L2 FORWARDing is used. Other header_types are supported at any frame_offset alignment.
- Program policy data values. The maximum number of data values that can be sent is 14. If the target is an Event then the total length of the data values cannot exceed 32 words. If the target is a ring then the total length of the data values cannot exceed 16 words. Data values beyond these limits are ignored.
- ASSOCIATE and CONNECTION policy number of entries. The implementation may set the size of the table to be larger than the constant supplied in the NUMBER keyword. If your program depends on knowing the exact table size then use a constant value that is 32^*P where P is a power of 2. E.g. Valid number of entries are 32, 64, 128, 256, 512, 1024, ... etc.

PPL arrays.

- Array memory type. Arrays in Scratch memory are not supported.
- PPL Array memory boundary. In order for PPL arrays to be accessed by Linux usermode programs (applications) running on the XScale, they must be allocated on page boundaries which are 4KB. Arrays mapped to either physical locations or *external_name* using the PPL DeviceMap ARRAY_MAP statement, must specify the start of the array on a 4KB address boundary if that array will be accessed by a Linux usermode application using the ipf_get_array_base() API. If such arrays do not need to be accessible from usermode programs, then they must be aligned on a 16-byte address boundary. When there is no PPL DeviceMap ARRAY_MAP statement for a PPL array, the System Application will allocate the array memory on a 4KB address boundary to ensure the array is accessible from Linux usermode programs.

PPL Expressions and Actions.

- SCAN/SCANB/SCANE. The maximum pattern size is 65 bytes. Regular expressions must be smaller than 192 characters in length and have no more than 8 repeaters. The alternation operator is not supported.

- Atomic operations. The only atomic operation is an unmasked 32-bit Add action on a global register.
- Log. If a packet for which there are no true rules causes an entry in the log then the LPN for that log entry will be invalid.

PPL DeviceMap.

- NPU model. NPU model number 2300 is not supported.
- Processor allocation static. PPL processor allocation is not dynamic. The 16 PPL processors are static based on the microcode image loaded. In this release, two configurations of microcode load files (“uof” files) are provided. The PPL_PROCESSORS DeviceMap command must match the load file.

“uof” load file	GbE ports	DeviceMap PPL_PROCESSORS command
ppl_vm_2800	8	(AE(69%),BE(6%),CE(6%),TE(1%),I4(6%),ER(6%), ET(6%))
ppl_vm_2800_12ports	12	(AE(63%),BE(6%),CE(6%),TE(1%),I4(6%),ER(6%), ET(12%))
ppl_vm_2800_demo	8	(AE(69%),BE(6%),CE(6%),TE(1%),I4(6%),ER(6%), ET(6%))

- Beta tester caution. You must remove the “vm_processors” from your DeviceMap files.
- SADB MEM parameter is not supported.
- LINK only supports GE_ON_SPI. No support for CSIX, PCI, or POS LINKs.
- Ctrl_type with LINK. The controller chip name must include a dash number to indicate which instance of the chip is being described. For example, use “IXF1104-0” to describe the first interface chip and “IXF1104-1” to indicate the second interface chip.
- PROG does not support the REMOTE interface.

Input/Output and packet considerations.

- IPv6. The IPv6 implementation is incomplete. There is no support for layer-3 forwarding (i.e. FORWARD action with no parameters) of an IPv6 packet. IPv6 packets can be received and sent directly to an LPN by using the FORWARD statement with one or two parameters – e.g. to the TX block or to another Event.
- Ethernet. Jumbo frames are not implemented.
- VLAN. Packets with VLAN can be received and manipulated but FORWARD(lpn,Fuf) is the only supported means to forward packets with VLAN tags.
- Multicast packets. The L3 Forward action does a unicast IPv4 forwarding operation. If you expect to handle multicast packets then you should handle this within your PPL program and transmit them either by Forward(lpn[,id]) action or sending to your own external program or core component.

Hardware.

- Full platforms. The PPL system has been designed to run on a family of platforms. The validation hardware platform used by IP Fabrics for this release was the Intel IXDP2851. This software should run properly on this platform or the Intel IXDP2801 ATCA boards with B1 version chip. If your platform is provisioned differently then please contact IP Fabrics to discuss the requirements in more depth.
- Minimum requirements. PPL-VM technology in this release requires:
 - DRAM – at least 512 Mbytes. Three channels of at least 768 Mbytes is recommended.
 - SRAM – 3 channels (0, 1, 2) with at least 8 Mbytes each.
 - Ethernet interface. This release only supports the Intel IXF1104 interfaced via an Intel SPI3-to-SPI4 bridge.
 - Ethernet “debug” port for software loading and NFS file structure access.
 - Flash memory used for system boot via a compatible version of “RedBoot” software.

- Hardware - CRC errors with the Gigabit Ethernet Mezzanine in Master PHY mode. (Intel IXDP2801/2851 errata Reference # IGK0100039965.)
 - A problem may occur when an external gigabit Ethernet device, such as a switch, router, or tester, initializes itself as a GbE slave interface during autonegotiation. Many CRC errors occur in the loopback tests. If the external device is made the master, these errors do not occur.
 - If using Ixia tester (for example) you can workaround the problem with the following procedure:
 1. Load up your old cfg file.
 2. You cannot change this when link is up. Make links go down (i.e. unplug cable). On the Ixia main window, you will see the port status red (instead of green) when link is down.
 3. Go to each port (under chassis, card), highlight the port you are going to change. Right click your mouse and pick the last item called "properties...".
 4. A new window pop up and choose the "Auto Negotiation" tab.
 5. In the tab, there is a place called "Negotiate Master/Slave". Un-tick this box. Then you can select Ixia as Master.
 6. Click "Apply" and "OK"
 7. Do the same to other 3 ports.
 8. Go back to chassis and do a complete save to the cfg file.