

Release Notes

Version 1.4.0

Revision History

Revision	Description of Change	Date
v1.0.1	Initial creation	5/2016
v1.1.0	Updates for 1.1 release	3/2017
v1.2.0	Updates for 1.2 release	8/2017
v1.3.0	Updates for 1.3 release	1/2018
v1.3.1	Updates for 1.3.1 release	3/2018
v1.4.0	Updates for 1.4.0 release and moved previous to appendices	10/2018

Table of Contents

I Enhancements	5
I-1 Support for Ettus E310 Platform	5
I-2 Component Unit Test Enhancements	5
I-3 Multi-Slave Proxy Support	5
I-4 Transmit Blanking / Powerdown for DACs	5
I-5 <code>ocpidev</code> Improvements	6
I-6 VHDL-Native Pattern and Capture HDL Workers	6
I-7 New Documentation RPM	6
I-8 Zynq Parts Defined	6
II Changes and Notes	6
II-1 I/Q Data Ordering	6
II-2 Upgrade Notes	7
II-3 Significant SD Card Layout Changes	7
II-4 Intel Quartus and Quartus Pro Version Change	7
II-5 New Input Port ZLM-Handling Primitive	7
II-6 Change in QDAC/QDAC_TS Component Spec	7
II-7 Change in <code>ocpidev</code> Default Output and Scope	8
II-8 Rounding and Truncation in <code>qdac-spec</code> Workers	8
II-9 HDL Workers Failing Unit Tests That Previously Passed	8
II-10 New Framework Build System	8
II-11 New RCC Platform Configuration	8
IIIMajor Known Deficiencies	8
III-1 ZedBoard IP Required Even In Standalone Mode	8
III-2 Connection Direction In GUI Assumes XML order	8
III-3 OnlyPlatforms/ExcludePlatforms Does Not Work For Individual Unit Test Cases	8
III-4 Some HDL Workers Will Need To Be Updated When <code>props_in.raw.address</code> Width Fixed	9
III-5 Default <code>python</code> Must Be Python2	9
III-6 Some <code>ocpi.assets</code> Workers Have Flow Control Issues	9
III-7 Previously Documented Deficiencies	9
Appendices	10
A Release 1.3.1	10
A-I Enhancements	10
A-II Changes and Notes	10
A-IIIMajor Known Deficiencies	10
A-III.1 PCI Platforms Can Only Transmit Using One Channel	10
B Release 1.3	11
B-I Enhancements	11
B-I.1 OSS Release	11
B-I.2 Python and GNU Radio Integration (Technology Preview)	11
B-I.3 AD9361 Support	11
B-I.4 GUI Improvements	11
B-I.5 Project Registry	12
B-I.6 Project Migration	12
B-I.7 <code>ocpidev</code> Improvements	12
B-I.8 RCC Platforms	12
B-I.9 Remote Containers (Technology Preview)	12
B-I.10 OpenCL Containers (Technology Preview)	12
B-II Changes and Notes	12
B-II.1 Major Project Restructuring	12
B-II.2 Upgrade Notes	13
B-II.3 Prerequisite RPMs Built With Newer Language Standards	13

B-II.4	Legacy Platform Names Continue to be Deprecated	13
B-II.5	OCPI_XILINX_VIVADO_DIR Variable Changed	13
B-II.6	ocpidev and Requesting RCC Platform To Match HDL Platform	13
B-II.7	zero_padding Component in ocp.assets deprecated	13
B-III	Major Known Deficiencies	14
B-III.1	Previously Documented Deficiencies	14
C	Release 1.2	15
C-I	Enhancements	15
C-I.1	Vivado Support for Applicable Xilinx-Based Platforms	15
C-I.2	New command option “ocpidev build”	15
C-I.3	GUI Now Delivered as RPM	15
C-I.4	Artifact Version Checking	15
C-I.5	Miscellaneous Improvements	15
C-II	Changes and Notes	15
C-II.1	rx-spec and tx-spec Support Property Interdependence	15
C-II.2	Vivado Support	16
C-III	Major Known Deficiencies	16
C-III.1	Driver Failing to Allocate Memory on Zynq-Based Platforms	16
C-III.2	HDL Workers Cannot Send SOM Without Valid Data	16
C-III.3	SDP Cannot Handle More Than One ZLM Per Application Run	16
C-III.4	File_Read and File_Write Cannot Handle ZLM Mid-file	16
C-III.5	PCI-Based Platforms Not Always Detected in Remote Tests	16
C-III.6	Built Projects Cannot Be Moved	16
C-III.7	Properties Before Raw Properties Cannot Have Parameterized Lengths	16
C-III.8	XML Parser Mixed-Case Issue	16
C-III.9	Simulation of Primitive Cores Needs Makefile Modified	16
C-III.10	Lime RX/TX Proxy Tests on CentOS6	17
C-III.11	Previously Documented Deficiencies	17
C-IV	Addenda to Previous Release Notes	17
D	Release 1.1	18
D-I	Enhancements	18
D-I.1	Debug Symbols for Framework Now Available	18
D-I.2	Kernel Driver Memory Arguments	18
D-I.3	HDL File_Read and File_Write CWD	18
D-II	Changes and Notes	18
D-II.1	Simulators Renamed and Work Out-of-the-Box	19
D-II.2	HDL Width Adapters Not Required	19
D-II.3	Platform Name Changes	19
D-II.4	RCC Cross-compilation No Longer Uses OCPI_CROSS_HOST	19
D-II.5	Updating a 1.0.x Unit Test for the 1.1 Framework	19
D-II.6	Prerequisite RPM Changes	20
D-II.7	Deprecated RCC controlOperations	20
D-II.8	64-Bit Build Host Issues (“Value too large for defined data type”)	20
D-III	Major Known Deficiencies	20
D-III.1	Maximum Message Length in PCI-Based Platforms	20
D-III.2	MAC Address on ZedBoard	21
D-III.3	Platform Development Guide	21
E	Release 1.0.1	22
E-I	OpenCPI HDL Assembly Filename Constraint	22
E-II	32-bit Architecture File Size Limit	22
E-III	Eclipse Project Description File	22
E-IV	Refreshing the ANGRYVIPER IDE	22
E-V	Non-default Xilinx Installation Locations	22
E-VI	Zero-Length Message Finite State Machines	22

Overview

This document contains corrections, changes, and enhancements made to OpenCPI since the previous release. It can never encapsulate *every* change or bug fix, but attempts to assist previous users in getting up and running as swiftly as possible. The *general* format of this document is to answer user questions:

- Section I (“Enhancements”) describes “*What’s New?*” about the current release.
- Section II (“Changes and Notes”) is “*What do I need to be aware of as a **previous** user of the Framework?*”
- Section III (“Major Known Deficiencies”) is for *all* users to answer “*What bugs should I be aware of?*”

The appendices contain the Release Notes from previously released versions of the Framework, roughly following the same format, e.g. B-I, being Section I of the v1.3 Appendix, is the “Enhancements” that were added to Release 1.3. ***It is highly recommended that a new user reviews the appendices as well; not only the main document concerning the latest release.***

I Enhancements

I-1 Support for Ettus E310 Platform

The Ettus Research USRP E310 is now supported within the OpenCPI framework. This includes the changes to the AD9361 device workers for using a CMOS interface, increasing the number of modes that the framework supports¹. There is currently a limitation of a single TX and RX channel being supported; see the *E3xx Getting Started Guide* for more information.

I-2 Component Unit Test Enhancements

The provided Component Unit Test infrastructure has been enhanced to automatically incorporate elements that address common issues when developing HDL workers:

1. Input message metadata manipulation applied to the Unit Under Test’s (UUT) input port(s) helps ensure an HDL Worker can properly handle various message-protocol edge conditions, *e.g.* start-of-message without data until a later clock. **This manipulation is *not* enabled by default.**
2. Output message throttling applied to the UUT’s output port(s). Not respecting “backpressure” is often the reason an HDL Worker fails on a “real” hardware platform after passing all tests within a simulation. **This backpressure is now enabled in *all* test benches by default.**

Exercising Workers in this manner during the simulation design phase has been shown to greatly increase the likelihood of success on physical hardware. More information can be found in the *Component Development Guide*.

I-3 Multi-Slave Proxy Support

Previously, a Proxy Worker could only represent a single Worker but not a collection of Workers that had related functions, *e.g.* the transmit and receive paths of a single RF link. This resulted in various workarounds that are not officially supported by the Framework. For example, one user had a Proxy “peeking” into the Application to find peer Workers: this does not work with Remote Containers with the Application on another physical machine. Now a Proxy can (optionally) define more than one slave Worker with full interface compatibility with previous implementations². More information can be found in the *Component Development Guide*.

I-4 Transmit Blanking / Powerdown for DACs

Both the AD9361 and LIME DACs are now blanked / powered down³ when not in use. To perform this on-demand, they use a new “tx_event” Port/Protocol, as well as properties exposed in their proxies. See the respective Worker datasheets for more details. **This added capability resulted in a change to the qdac-spec that may affect current user applications.** If an application used `connect=qadc` (connect attribute), that application will require an additional `to` attribute specified directing it to explicitly use the IN input port.

¹AD9361 LVDS signaling has been supported since 1.3; cf. B-I.3.

²The interface of using an object named “slave” remains. The type of the object is considered an implementation detail and *has* changed.

³Implemented by TX RF carrier suppression

I-5 ocpidev Improvements

The `ocpidev` tool has been significantly improved, including:

- run some Applications
- run Unit Tests
- show Unit Tests within a Project
- show additional Project information

See II-7 for `ocpidev` command changes.

I-6 VHDL-Native Pattern and Capture HDL Workers

The `pattern.v2` and `capture.v2` Workers have been written from the ground up in VHDL to replace the machine-generated Verilog-sourced `pattern` and `capture` Workers. **These latter Workers are now considered deprecated.** The new Workers include fully-functional test benches and data sheets. Platform and BSP Developers will find these Workers to be very useful when debugging various issues related to low-level OpenCPI development.

I-7 New Documentation RPM

In addition to the PDF-based documentation available at github.io, a new RPM is now included “`opencpi-doc`,” which includes most of that documentation. Most of the PDFs are installed into `/usr/share/doc/`⁴, but a convenience symlink can be found at `/opt/opencpi/documentation.html`. This path should not change across multiple OpenCPI versions and can be passed directly to most browsers, e.g. “`xdg-open /opt/opencpi/documentation.html`”. If you receive the RPM directly from the AV team, it may include BSP documentation that is not available on GitHub.

I-8 Zynq Parts Defined

All Zynq-based Xilinx parts that are supported by the current Framework, but not necessarily used in an OpenCPI-supported platform, have been defined within the Framework build system. This means that an end user creating a BSP for a “larger fabric” Zynq-7000, e.g. `xc7z100`, will not need to change any Framework files and have all required changes isolated to a single Project. This does *not* include additional platforms that have different internal configurations, e.g. Zynq-UltraScale+ devices.

II Changes and Notes

II-1 I/Q Data Ordering

As noted in the *HDL Development Guide*, when a Protocol contains a Struct Argument, the first Argument Member defined in the Protocol’s XML is *always* in the *least significant* bits of the resulting Port. A good example of the importance of this is the scenario where a Port’s default data width is overridden (in the OWD) to present all of a Struct’s Members in parallel within a single clock cycle. For example, `iqstream_protocol`’s default ordering is a 16-bit interleaved I/Q data: “ $I_{t=0}, Q_{t=0}, I_{t=1}, Q_{t=1}, I_{t=2}, \dots$ ”. However, if the Worker configures its data Ports to be a width of 32 bits, then the I/Q data is presented as a parallel I/Q sample pair with “I” in the *lower* 16 bits and “Q” in the *upper* 16 bits, i.e.:

$$\begin{aligned} I_0 &= InPort_0[15 : 0]; Q_0 = InPort_0[31 : 16] \\ I_1 &= InPort_1[15 : 0]; Q_1 = InPort_1[31 : 16] \end{aligned}$$

The following Application and Workers were identified as improperly handling `iqstream_protocol` complex data (I/Q) presented in the manner noted above and have been corrected:

- `assets/components/dsp_comps`:

⁴The Fedora/Red Hat standard documentation location.

cic_dec.hdl	dc_offset_filter.hdl	pr_cordic.hdl
cic_int.hdl	fir_complex_sse.hdl	rp_cordic.hdl
complex_mixer.hdl ⁵	iq_imbalance_fixer.hdl	
complex_mixer.rcc ⁵	phase_to_amp_cordic.hdl	

- assets/hdl/devices:

ad9361_dac.hdl	lime_dac.hdl	lime_dac_ts.hdl
----------------	--------------	-----------------

- assets/applications

FSK	Rx_App
-----	--------

It is important to note that this list includes various DACs as well as the `complex_mixer`, meaning that **user's RF Applications from previous versions of the Framework *may* need to be updated**, *e.g.* to handle this change in the DAC's interpretation of data presented for transmission. Also related to this, the `phs_inc` Property of `complex_mixer` may require inversion when compared to previous versions.

II-2 Upgrade Notes

If you have previous versions of the RPM-based OpenCPI RPMs installed, they must all be manually removed, including the prerequisites. **There is no automatic upgrade path.** Manually performing “`sudo rm -rf /opt/opencpi/prerequisites`” after uninstalling **all** previous RPMs *may* also be needed.

II-3 Significant SD Card Layout Changes

There have been significant changes to the layout of the Framework and its support files. These are mostly implementation details that the end user does not need to be concerned with. However, scripts and paths that are used for embedded development (both locally and using NFS mounts) have changed enough that users should wipe any SD cards they have used and follow the directions given in the respective platform's *Getting Started Guide* to create a new one.

II-4 Intel Quartus and Quartus Pro Version Change

This release has been tested with Quartus Standard 17.1 which is a newly supported version of the tool. Version 15.1 should still work, but future versions will be tested with 17.1. Additionally, preliminary support for Quartus Pro 17.0.2 is also available within the Framework, but is not used by any platforms at this time.

II-5 New Input Port ZLM-Handling Primitive

For an HDL Worker to handle the *de facto* end-of-stream signaling used by the Component Unit Test infrastructure (a ZLM with OpCode 0), a convenience primitive is now available and found at `projects/core/hdl/primitives/util/zlm_detector.vhd` with an example implementation shown in the `capture_v2.hdl` Worker.

II-6 Change in QDAC/QDAC_TS Component Spec

The QDACs now have two ports. The “IN” ports have not changed, but some applications may need to be updated if they did not explicitly connect to them. See I-4 above for more information.

⁵See last paragraph as well concerning Property changes.

II-7 Change in ocpidev Default Output and Scope

Some of the output from `ocpidev`'s `show` has changed to default to being more “human friendly” and reporting in a tabular fashion (`--table`). Many of these commands retain a `--simple` parameter for quick parsing. Most also support `--json`, which is the recommended and most stable method of parsing `ocpidev` output.

Some of `ocpidev`'s `show` commands now have the concept of “scope” where the user is requesting a list of *somethings* provided by a single project, or *all somethings* available within all registered Projects. At this time, no commands support both of the scoping arguments `--global-scope` and `--local-scope`. However, to ensure future `ocpidev` enhancements remain compatible with user scripts, some commands will force the user to explicitly specify a scope if the “reasonable” not-yet-implemented default is not yet ready. For example, in a future release, “`ocpidev show tests`” will report *all* Unit Tests that are available. That report is not yet available, so the user *must* currently explicitly scope the request with “`ocpidev show tests --local-scope`”. However, something like “`ocpidev show registry --local-scope`” would make no sense, so will never be supported.

II-8 Rounding and Truncation in qdac-spec Workers

Various DACs were found to be improperly casting *signed* data to *unsigned* during certain operations, resulting in an overflow when the maximum value was transmitted.

II-9 HDL Workers Failing Unit Tests That Previously Passed

If a user experiences Unit Test failures that were not present in previous releases, the most likely cause is that “backpressure” is now automatically asserted by default; see I-2 for details.

II-10 New Framework Build System

The OSS OpenCPI Project has migrated the Framework build system to a new custom-made system that produces RPMs directly⁶ and removes the need for “prerequisite” RPMs. If upgrading from a previous RPM-based installation, the prerequisite RPMs might require manual deletion. Any mention of prerequisite RPMs in the Appendices (*e.g.* D-II.6) below can likely be ignored.

II-11 New RCC Platform Configuration

Prior to 1.4, RCC Platforms were defined in `/projects/core/rcc/platforms`. With 1.4, they can be in the `/rcc/platforms` directory of any Project. *The Platform Development Guide* defines how RCC Platforms are *now* defined and a condensed migration guide can be found at `projects/core/rcc/platforms/README.migrating`.

III Major Known Deficiencies

III-1 ZedBoard IP Required Even In Standalone Mode

Applications (including XML-only ones) fail if there is not an IP address assigned to the ZedBoard, even when in “standalone mode.” If the network connection is not used, forcing a MAC (cf. Section D-III.2) and IP addresses is required. To set a temporary IP address, the command “`ifconfig eth0 192.168.244.244`” can be used. This problem was found late within the 1.4 release cycle and should be addressed with the next major release.

III-2 Connection Direction In GUI Assumes XML order

When importing hand-written XML into the GUI, the `to` and `from` attributes are ignored on connections. This does not affect any XML generated by the GUI.

III-3 OnlyPlatforms/ExcludePlatforms Does Not Work For Individual Unit Test Cases

You can include and exclude platforms at a full-test level, but individual cases currently ignore the directives.

⁶If a user was building custom RPMs (from the `releng` directory), the process has changed considerably.

III-4 Some HDL Workers Will Need To Be Updated When `props_in.raw.address` Width Fixed

The “raw” address scheme for HDL workers is documented as a 32-bit bus. In testing, it was found that this is currently implemented as only 16 bits. While this has *not* yet been fixed, users should review their own libraries for any Workers that make assumptions concerning the size of the bus. Any `raw.address` access should be written in more generic VHDL, *e.g.*:

```
-      event_rawprops.raw.address <= x"0040"; --TXRF modules enable register
+      event_rawprops.raw.address <= (6 => '1', others => '0'); --TXRF modules enable register (0x40)
```

III-5 Default python Must Be Python2

According to the PEP-394 specification, system-level `python` *should* point to a 2.x version of the Python interpreter. Some scripts within the Framework and Unit Tests *require* this configuration.

III-6 Some `ocpi.assets` Workers Have Flow Control Issues

The Component Unit Test infrastructure has been enhanced (cf. Section I-2) and deficiencies within some Workers have been revealed during regression testing. The following `ocpi.assets` Workers are currently known to have defects which will be fixed in a future release:

<code>bias_ver.hdl</code>	<code>cic_int.hdl</code>	<code>fir_real_sse.hdl</code>
<code>cic_dec.hdl</code>	<code>fir_complex_sse.hdl</code>	

III-7 Previously Documented Deficiencies

The following deficiencies are still present in 1.4:

- E-II: 32-bit Architecture File Size Limit
- E-I: HDL Assembly Filename Constraint
- D-III.1: Maximum Message Length in PCI-Based Platforms
- C-III.7: Properties Before Raw Properties Cannot Have Parameterized Lengths
- D-III.2: MAC Address on ZedBoard
- C-III.5: PCI-Based Platforms Not Always Detected in Remote Tests
- C-III.10: Lime RX/TX Proxy Tests on CentOS6
- A-III.1: PCI Platforms Can Only Transmit Using One Channel

The following previously-documented deficiencies have been fixed in this release:

- C-III.8: XML Parser Mixed-Case Issue⁷
- D-III.3: Platform Development Guide

⁷The opening and closing tags must still match, *e.g.* `<Element>` pairs with `</Element>` but not `</element>`.

Appendices

The Release Notes from previous releases are included here as reference for still-existing bugs, relevant upgrading notes for users of very old versions, etc.

A Release 1.3.1

A-I Enhancements

- All documentation has now been updated to cover 1.3 capabilities and features
- Bugfixes and enhancements for AD9361
- Workarounds for bug noted in Section A-III.1
- USB Blaster udev rules fixed
- ocpizynq tool fixed

A-II Changes and Notes

None to report.

A-III Major Known Deficiencies

- *Note:* no other deficiencies from Section B-III have been mitigated and all remain.

A-III.1 PCI Platforms Can Only Transmit Using One Channel

The current implementation of the PCI-e Specification (ALST4 and ML605 platforms) only allows DMA transaction from the host to the platform on the *first External Port defined in the Assembly or Container XML*. This means the *input* to any Assembly/Container should be listed first. See `projects/assets/hdl/assemblies/empty/cnt_1rx_1tx_bypassasm_fmcomms_2_3_hpc_LVDS_ml605.xml` as an example. *This does not affect any Zynq-based platforms.* This problem was found late within the 1.3 release cycle and should be addressed with the next major release.

B Release 1.3

B-I Enhancements

B-I.1 OSS Release

ANGRYVIPER Team's build of OpenCPI is now available on GitHub under the OpenCPI Organization. The first release is in a git branch, but full integration is planned in the future. This includes all of the previously-proprietary "ocpiassets / Assets" Project, including the Board Support Package (BSP) for Epiq Solutions' Matchstiq Z1.

B-I.2 Python and GNU Radio Integration (Technology Preview)

There are now SWIG bindings available to allow a top-level Python script to use the ACI to manipulate an Application in the same manner as from C++⁸. One use case is a GNU Radio block that allows GNU Radio Companion to handle the entire lifecycle of an OpenCPI Application. This block can handle data flows of complex I/Q streams or any primitive data type (*e.g.* floats or bytes) in and out of the Application. A reference implementation of this block is provided in OpenCPI's fork of GNU Radio Companion.

This fork of GNU Radio Companion also allows for Application and HDL Assembly building, as well as deployment independent of the ANGRYVIPER GUI. Some of this capability was presented at the GNU Radio Conference GRCon17; the video is available on YouTube.

B-I.3 AD9361 Support

The popular Analog Devices, Inc's AD9361 RF transceiver IC is now supported via a series of Device Workers. The modularity of these workers eases AD9361 transceiver integration with future FPGA systems. Most of the major AD9361 features are supported, including RX/TX RF digitization/data streaming and RF tuning/gain control. The AD9361 Device Workers currently only fully support some operating modes, but contain infrastructure in anticipation of future expansion. These device workers utilize the vendor's No-OS software library for high-level control.

There is also "card" support for the FMCOMMS2 and FMCOMMS3 evaluation/prototyping RF transceiver FMC cards centered on the AD9361. The newly released `ocpi.assets` Project includes a single-channel RX narrowband I/Q stream to file (`applications/rx_app`) and single-channel RX/TX FSK modem (`applications/FSK`) reference applications, which each support the FMCOMMS2 and FMCOMMS3 card on the zed platform.

B-I.4 GUI Improvements

Since 1.2, the GUI has multiple improvements, some of which are:

- **The ability to build OpenCPI Assets** - The Component and Application developer can now build without the command line. The build tool allows RCC and HDL compilation from numerous levels in a Project from individual assets to the entire Project. The tool also allows the developer to build and run Component unit tests while allowing multiple concurrent builds in separated tabbed consoles.
- **New Perspective** - This Eclipse Perspective is the new presentation of the build tool. It consists of a number of views where three are specific to OpenCPI.
- **Project Explorer view** - This view allows simple navigation to select items to build and provides a way forward to provide OpenCPI features without having to explore the file system.
- **OpenCPI Operations Panel** - Selections can be placed in this panel and built for all available RCC and HDL Targets and/or Platforms.
- **Build Status View** - Provides a simple graphical presentation of the launched build or test run and its status (*e.g.* running). The user may also re-run build, clean, or run executions from this view and bring the build console window forward.
- **New HDL Platform Editor** - The HDL developer can now create and view HDL platform XML in a graphical presentation.

⁸This integration is only available on CentOS 7 platforms.

B-I.5 Project Registry

Release 1.3 introduces a new concept of “Project Registry” to organize and interrelate a set of Projects which allows Projects to be moved (cf. B-I.6) as well as present an explicit relationship between them, *e.g.* “*this* copy of `ocpi.assets` was built based upon *that* copy of `ocpi.core`.” Registry manipulation is performed using `ocpidev` (cf. B-I.7).

The default configuration is that every Project on a machine shares a single Registry (located at `/opt/opencpi/project-registry`). A user must be in the Unix group “`opencpi`” to manage this global Registry.

This new capability deprecates the previously-used `OCPI_PROJECT_PATH` build-time variable as well as the scripts `freeze_project.sh` and `thaw_project.sh` noted in C-III.6.

B-I.6 Project Migration

With the new Project Registry (cf. B-I.5), Projects can now be moved to new locations or archived and then restored on another machine with the use of the `ocpidev unregister` and `ocpidev register` commands. This mitigates the previously-documented deficiency found in C-III.6.

B-I.7 ocpidev Improvements

The “`ocpidev`” command has been improved in various ways, including:

- `bash` completion for many commands (try `ocpidev <tab><tab>`)
- Ability to show possible HDL platforms: `ocpidev show hdl platforms --table`
- Ability to show possible RCC platforms: `ocpidev show rcc platforms`
- Ability to build unit tests
- Project Registry (B-I.5) manipulations

B-I.8 RCC Platforms

With the new Project Registry (cf. B-I.5), RCC Platforms can now be defined exclusively within a Project outside of the CDK / Core Project. An example of this can be seen in the `inactive` project under `rcc/platforms/macos10_13`. This improvement allows BSPs to be self-contained within a Project.

B-I.9 Remote Containers (Technology Preview)

The full-OSS OpenCPI code base has imported previously-written code that allows Remote Containers. This code has not yet been evaluated by the ANGRYVIPER Team.

B-I.10 OpenCL Containers (Technology Preview)

The full-OSS OpenCPI code base has imported previously-written code that allows OpenCL (“OCL model”) Containers on CPUs and GPUs. This code has not yet been evaluated by the ANGRYVIPER Team.

B-II Changes and Notes

This section describes items that may cause users of previous versions to modify their code, build system, etc.

B-II.1 Major Project Restructuring

Starting with Release 1.3, the CDK is no longer a top-level Project. All Components that were previously found in the CDK, *e.g.* `file_read`, have migrated to `ocpi.core`. The previously-documented “Base Project” has been removed; it was the RPM-specific manner of separating the CDK from the core components needed to run OpenCPI. The various items (cards, devices, specs, etc) that were previously found in the Base Project have been distributed into one of three new Projects. The previously non-OSS proprietary items have been redistributed within the new Projects as well. The Projects are:

- `ocpi.core` - Core items for the Framework

- `ocpi.assets` - Items fully-supported by the ANGRYVIPER Team
- `ocpi.inactive` - “Everything else”

These items being moved will cause Application XML to possibly fail, *e.g.* anything calling for the `ocpi.file_read` spec now needs to look for `ocpi.core.file_read`. There is a similar rename of `ocpiassets` \Rightarrow `ocpi.assets`.

There is a provided utility script (`/opt/opencpi/cdk/scripts/upgradeApp_v1_3.py`) to assist in this migration. This script will modify the package information for the specs required. There are three ways it can be called:

- If given a single argument (the Application XML file name), it will write the same name with `updated_` prepended.
- If given a second argument, the modified file will be written to that file name.
- If the second argument is `same`, the original file will be overwritten.

This change also removes the `/opt/opencpi/base_project_source/new_base_project.sh` script found in earlier releases; its equivalent is now `/opt/opencpi/projects/new_project_source`, which uses the same arguments as noted in the *Getting Started Guide*.

B-II.2 Upgrade Notes

If you have previous versions of the RPM-based OpenCPI RPMs installed, they must all be manually removed, including the prerequisites. **There is no automatic upgrade path.** Manually performing “`sudo rm -rf /opt/opencpi/prerequisites`” after uninstalling **all** previous RPMs *may* also be needed.

B-II.3 Prerequisite RPMs Built With Newer Language Standards

All prerequisites are now built with the C++11 or the C98 standard implemented. This will increase interoperability and avoid ambiguous situations, *e.g.* the size of a `bool` variable.

B-II.4 Legacy Platform Names Continue to be Deprecated

The legacy suffix “`_pf`” for simulator platforms, *e.g.* `isim_pf`, is not supported when running unit tests. Unit tests were introduced after the suffix was deprecated, so the effect on end users should be minimal.

B-II.5 OCPI_XILINX_VIVADO_DIR Variable Changed

A bug in Release 1.2 required the variable to be set “deep”, *e.g.* `/data/Xilinx/Vivado/2017.1`. You *must* now set it to the top-level value, *e.g.* `/data/Xilinx`. If a specific version of Vivado is required, `OCPI_XILINX_VIVADO_VERSION` should be set, *e.g.* `2017.01`. As a reminder, these variables only apply if you have both ISE and Vivado installed in different base directories, *e.g.* ISE in `/opt/Xilinx` vs. Vivado in `/data/Xilinx`.

B-II.6 ocptidev and Requesting RCC Platform To Match HDL Platform

When building Projects with `ocptidev`, the user can tell the build system to automatically select the proper RCC Platform when given an HDL Platform. This option is `--rcc-hdl-platform` with a mnemonic of “build *this* RCC Platform for *that* HDL Platform.” Previous Releases had the flags backwards (“`--build-hdl-rcc-platform`”). This is incorrect and has been deprecated.

B-II.7 zero_padding Component in ocpi.assets deprecated

In the FSK reference application, `zero_padding.hdl` was replaced by two new Workers, `zero_pad.hdl` and `mfsk_mapper.hdl`. The change was made because `zero_padding.hdl` performed two functions: binary FSK mapping and inserting zeros between symbols. Whenever possible, functions should be separated into their own Components for increased flexibility and reuse. The `zero_padding` Component will be removed in Release 2.0.

B-III Major Known Deficiencies

B-III.1 Previously Documented Deficiencies

The following deficiencies are still present in 1.3:

- E-II: 32-bit Architecture File Size Limit
- E-I: HDL Assembly Filename Constraint
- C-III.8: XML Parser Mixed-Case Issue
- D-III.1: Maximum Message Length in PCI-Based Platforms
- D-III.3: Platform Development Guide
- C-III.7: Properties Before Raw Properties Cannot Have Parameterized Lengths
- D-III.2: MAC Address on ZedBoard
- C-III.5: PCI-Based Platforms Not Always Detected in Remote Tests
- C-III.10: Lime RX/TX Proxy Tests on CentOS6

The following previously-documented deficiencies have been fixed in this release:

- C-III.1: Driver Failing to Allocate Memory on Zynq-Based Platforms
- C-III.6: Built Projects Cannot Be Moved
- C-III.2: HDL Workers Cannot Send SOM Without Valid Data
- C-III.4: File_Read and File_Write Cannot Handle ZLM Mid-file
- C-III.3: SDP Cannot Handle More Than One ZLM Per Application Run

The following previously-documented deficiencies will *not* be fixed:

- C-III.9: Simulation of Primitive Cores Needs **Makefile** Modified - It was determined that this is not a bug. For primitive cores, if a user sets the **SourceFiles** variable, they must explicitly list all files, including ***_pkg.vhd** files. This is true for *all* platforms.

C Release 1.2

C-I Enhancements

C-I.1 Vivado Support for Applicable Xilinx-Based Platforms

By far the #1 requested feature by OpenCPI users, the Framework now supports the use of Xilinx’s Vivado on the appropriate Platforms. **This toolchain is now the default** where applicable (mostly Zynq-based), and it is highly recommended that users of Xilinx-based platforms review the provided *Vivado Usage Guide*.

End users should be aware (see *Vivado Usage Guide* for more details) that:

- If you simulate with ModelSim, you may need to upgrade. Vivado requires a minimum version of 10.5c (tested by ANGRYVIPER Team using 10.6a)
- If your design requires the use of Xilinx CORE Generator IP (**coregen**) items, they must be regenerated with Vivado to be *simulated* in XSIM. Your existing **ngc** files from ISE can continue to be used for *synthesis*, but porting to new *Vivado IP* is recommended.
- There is a new platform, **zed_ise**, that is intended for testing purposes, but released with the distribution. This platform is the ZedBoard using the Xilinx ISE tool chain.

C-I.2 New command option “ocpidev build”

To help improve the end-user experience, OpenCPI 1.2 requires less command-line usage of “**make**”. The previously-available tool **ocpidev** now has a “**build**” verb available. More information can be found in the (OSS) *Component* and *Application Development Guides*.

C-I.3 GUI Now Delivered as RPM

There is now an **opencpi-ide** RPM included in the distribution. A separate installation of Eclipse for each end user is no longer required. Once installed, any user can launch with the **ocpigui** command.

C-I.4 Artifact Version Checking

While backwards compatibility is a major goal of OpenCPI, binary compatibility across versions is *not* guaranteed. Every artifact should always be rebuilt with the latest framework. To enforce this requirement, the framework will check all artifact metadata at deployment time to verify the version used to build. Version 1.1 inserted the proper metadata, but the comparison is new in 1.2, allowing 1.2 to recognize and reject 1.1 artifacts. Artifacts rejected for no other reason will issue a warning message at all debugging levels. This behavior can be overridden by setting the environment variable **OCPI_ALLOW_VERSION_MISMATCH** to “1”; *this is not recommended*.

C-I.5 Miscellaneous Improvements

A quick list of improvements not documented above include:

- Cross-platform ACI builds using **ocpidev**-generated applications previously failed
- Picoflexor T6A Software updated to vendor’s Drop 5.5.2
- **ocpiassets** project can now be imported into GUI
- Proxies can now use string properties⁹

C-II Changes and Notes

This section describes items that may cause users of previous versions to modify their code, build system, etc.

C-II.1 rx-spec and tx-spec Support Property Interdependence

Certain front-ends have different ranges based other properties, *e.g.* the current sample rate, requiring some properties that were previously “Initial” (immutable) to have their restrictions relaxed. *This compatibility break should only affect platform developers*, unless applications were incorrectly caching these values. All interfaces above that level should be unaffected.

⁹This is noted here because it is one of the first user reported bugs.

C-II.2 Vivado Support

Please see C-I.1 for modifications that *may* need to be performed.

C-III Major Known Deficiencies

C-III.1 Driver Failing to Allocate Memory on Zynq-Based Platforms

In some situations after repeated multi-model (RCC and HDL) application runs on Zynq-based platforms (e.g. ZedBoard), the kernel driver will fail to allocate contiguous memory for DMA data transfers. On x86-based platforms, memory for this use is allocated at boot time before memory fragmentation. Currently, no memory is being allocated at boot time for the Zynq-based platforms. Single ocpirun or ACI application runs from the command line do not present any errors. When running an app in a long loop, such as in the unit test framework, or an application that writes a large amount of data (approx. 1GB) to file, this error *may* occur.

C-III.2 HDL Workers Cannot Send SOM Without Valid Data

Although the HDL Component documentation claims the Framework's internal protocols support a separate Start-of-Message independent of valid data designations, this sequence causes the Component's output port to ignore any remaining data. This was found late within the 1.2 release cycle and should be addressed with the next release.

C-III.3 SDP Cannot Handle More Than One ZLM Per Application Run

The SDP transport will not propagate a second ZLM, so at this time the *only* ZLM should be an implied end-of-application transmission. This problem does not exist when communicating between RCC-only Workers. Related to, but independent of C-III.4. (See also E-VI.)

C-III.4 File_Read and File_Write Cannot Handle ZLM Mid-file

`file_read` (HDL implementation) and `file_write` (RCC and HDL implementations) will stop if a Zero-Length-Message (ZLM) is found when operating in "message mode" regardless of the location within the file. Related to, but independent of C-III.3. (See also E-VI.)

C-III.5 PCI-Based Platforms Not Always Detected in Remote Tests

Using `OCPI_REMOTE_TEST_SYSTEMS`, JTAG connection problems have been observed, possibly related to system reboots. They seem to be resolved after disconnecting and reconnecting the USB cable.

C-III.6 Built Projects Cannot Be Moved

When a Project is built, the dependencies of each artifact and other support files are captured. Some of these contain absolute paths pointing to where the Project is currently located. The distribution includes a pair of *unsupported* scripts that *may* allow movement of Projects. They can be found in `/opt/ocpncpi/cdk/scripts/` and are named `freeze_project.sh` and `thaw_project.sh`.

C-III.7 Properties Before Raw Properties Cannot Have Parameterized Lengths

"Raw properties" start at a certain property with the properties before that being "non-raw" properties. Currently, none of these "before" properties can have their length be set by an expression dependent on another parameter property.

C-III.8 XML Parser Mixed-Case Issue

The XML parser does not always recognize tags if differently-cased versions of the same tag are in the same XML file, e.g. an OAS having both `<Instance .../>` and `<instance .../>` XML tags will fail. Users of the GUI should not be affected.

C-III.9 Simulation of Primitive Cores Needs Makefile Modified

When designing a Primitive vendor-specific core that is intended for simulation, the required `*_pkg` file containing the core's stub must be explicitly listed first in the Makefile's `SourceFiles` variable.

C-III.10 Lime RX/TX Proxy Tests on CentOS6

The proxy tests for the Lime transceiver use a feature of C++11 that is not supported by the compiler in CentOS 6. The workaround is to use similar code to the following in the various places `to_string()` was used:

```
-      ClkInStr = to_string(2 * sample_rate * 1e6).c_str();  
+      std::ostringstream ostr_clkkin;  
+      ostr_clkkin << (2 * sample_rate * 1e6);  
+      ClkInStr = ostr_clkkin.str().c_str();
```

C-III.11 Previously Documented Deficiencies

The following deficiencies are still present in 1.2:

- E-II: 32-bit Architecture File Size Limit
- E-I: HDL Assembly Filename Constraint
- D-III.2: MAC Address on ZedBoard
- D-III.1: Maximum Message Length in PCI-Based Platforms
- D-III.3: Platform Development Guide

C-IV Addenda to Previous Release Notes

- The default `CWD.MAX_LENGTH` noted in D-I.3 is now set to 256.
- The Liquid RPMs referenced in D-II.6 are now **required** to build the `ocpiassets` project (`complex_mixer.rcc`). They are still considered optional when installing.

D Release 1.1

D-I Enhancements

A quick list of improvements include:

- Simulatable Data Plane: Allows HW and SW co-simulation to test integration between heterogeneous Workers
- Scalable Data Plane: Allows Platform Workers to more efficiently use wider data paths on future platforms, e.g. a 128-bit bus instead of 32-bit¹⁰
- New automatic test harness creation, including limited remote platform support¹¹
- Automatic bus width adaption between HDL Workers
- Additional Assets components to help design and debug Applications
- HDL Simulator on-the-fly launching¹²
- Extended GUI capabilities
- Official CentOS 7 driver support
- Split platform names more explicitly between hardware and software platform types
- Many fields in XML that require numerical values can now have mathematical calculations
- Added platform support (Picoflexor RCC-Only)

D-I.1 Debug Symbols for Framework Now Available

An RPM containing most debug symbols for the Framework (`opencpi-debuginfo-*.x86_64.rpm`) is now provided. This greatly enhances the debugging of RCC Workers using industry-standard debuggers such as `gdb` and `ddd`.

D-I.2 Kernel Driver Memory Arguments

In previous versions, the “`memmap`” argument was given to the Linux kernel command line. This can conflict with other software that may also require reserved memory. If the user requires the reservation of more than one memory area using the `memmap` option, the OpenCPI driver will:

- Assume the first `memmap` is for its use
- Explicitly use the range/subrange marked with an additional `opencpi_memmap`. *The original `memmap` must still be provided* for the kernel to make the reservation.

See `/opt/opencpi/driver/README` for more information, including usage examples and configuration.

D-I.3 HDL File_Read and File_Write CWD

The HDL Workers implementing the `File_Read` and `File_Write` specs distributed in 1.0 failed if used in a directory that exceeded 100 characters deep. There is an additional parameter `CWD_MAX_LENGTH` on these Workers that can be increased to allow deeper directories when needed.

D-II Changes and Notes

This section describes items that may cause users of previous versions to modify their code, build system, etc.

¹⁰Currently limited to some Zynq-based platforms.

¹¹Most `ocpiassets` have been moved to the new test system to provide examples.

¹²See Section D-II.1 for more information.

D-II.1 Simulators Renamed and Work Out-of-the-Box

The simulator platform names have been renamed to remove the “_pf” suffix, *e.g.* `isim`. The `isim`¹³ and `modelsim` simulation platforms can now be targeted once the appropriate environment variables are set, *e.g.* `OCPI_MODELSIM_DIR`. With `ocpirun` able to launch simulations, the usage of “`ocpihdl simulate`” is now deprecated and does nothing.

An example:

```
$ ocpirun -C
Available containers:
#  Model Platform      OS      OS-Version  Arch      Name
0  hdl   isim                linux    c7         x86_64    lsim:isim
1  hdl   modelsim            linux    c7         x86_64    lsim:modelsim
2  rcc   centos7              linux    c7         x86_64    rcc0
```

D-II.2 HDL Width Adapters Not Required

One of the advantages of the new *Scalable Data Plane* is the Framework’s automatic insertion of “width adapters” whenever needed between Workers in an Assembly. This means the user can remove any `ws_i16_to_32`, `ws_i32_to_16`, or `ws_i_width_adapter` Components that are instantiated in their Assemblies or Applications. It is harmless to leave them in existing designs, but future Applications should not instantiate them.

D-II.3 Platform Name Changes

To allow for more hardware and software separation in the future, there is now more distinction used in platform naming. Previously, the keyword “zed” was used to describe both software (the Xilinx-based Linux kernel) and hardware (the ZedBoard). This ambiguity led to confusion for some users and implied a nonexistent reliance on the ZedBoard. Starting with OpenCPI 1.1, the software platform for most Zynq-based hardware is labeled “`xilinx13_3`”, analogous to “`centos7`”. The “13.3” is shorthand for Xilinx’s Linux release from the Third Quarter (Q3) of 2013. This means that previous code that used an `OCPI_TARGET_PLATFORM` of `zed` must now be changed to `xilinx13.3`.

D-II.4 RCC Cross-compilation No Longer Uses OCPI_CROSS_HOST

Previous OpenCPI releases used `OCPI_CROSS_HOST` to define RCC cross-compilation. This required the usage of less-than-memorable values such as `arm-xilinx-linux-gnueabi`. The new manner is to use `OCPI_TARGET_PLATFORM`, `RccPlatforms`, or `RccHdlPlatforms`, which allow more intuitive and consistent values (See Section D-II.3). For example, the example above could be replaced with “`OCPI_TARGET_PLATFORM=xilinx13.3`”, “`RccPlatforms=xilinx13.3`”, or “`RccHdlPlatforms=zed`”.

The `RccPlatforms` and `RccHdlPlatforms` variables allow the targeting of multiple platforms with a single `make` call. While either format can be used for Workers, ACI Applications can only be built for a single target platform using the `OCPI_TARGET_PLATFORM` variable.

D-II.5 Updating a 1.0.x Unit Test for the 1.1 Framework

Some unit tests for Components were released with the 1.0.x version of `ocpiassets` that relied upon simulator availability or certain platform names reported by `ocpirun` (see Sections D-II.1 and D-II.3). If any users have developed their own unit tests based on these, some updates may be required for them to work using the new release.

All of the provided unit tests have been updated to work with the new release and can be used for reference. If the previous unit test is using `ocpirun`, the `Makefile` needs to be updated to add the following to any lines that are forming the execution string:

```
-P $@=$P
```

This is forcing the platform of your unit test to be set explicitly with `ocpirun`’s `-P` option.

If the unit test is using an ACI Application to run the unit test, the platform needs to be set similarly within the C++ source. The Platform will be passed from the `Makefile` to the ACI Application as a command-line argument

¹³Only version 14.7 without rebuilding the Base Project.

by adding `$P` to the end of any execution strings. The source code needs to be updated to use this new argument by adding the following to the C++ application, while changing “XXX” to your Worker’s name and “2” to the correct argument number based on your current inputs:

```
std::string pf("XXX=");
pf += argv[2];
OA::PValue params[] = {
    ...
    OA::PVString("platform", pf.c_str()),
    ...
    OA::PVEnd};
...
OA::Application app(appNameXml, params);
```

This patch is adding the platform that is passed in (via `argv[2]`) to the `params` array that is used to constrain the application’s search space.

The user could do a full conversion to the new testing framework released with 1.1, but this is expected to be more work.

D-II.6 Prerequisite RPM Changes

The Open Source XZ Utils has replaced the LZMA Utils. This has caused the ANGRYVIPER-provided RPM to change names. Additionally, new RPMs are now provided for other prerequisite packages; consult the *Installation Guide* for a list. An upgrade to the new RPMs *should* cause previous versions to be removed, but the recommended procedure is to uninstall all 1.0.x RPMs before installing 1.1.

For the first time, OpenCPI is providing a pre-packaged third-party library that is not directly required by the Framework: `liquid-dsp`. This is provided to assist end users in the creation of DSP-heavy RCC Workers that execute on all currently-supported platforms without requiring any additional configuration for cross-compilation. It is treated like any other prerequisite library; usage information can be found in the *OpenCPI RCC Development Guide*.

D-II.7 Deprecated RCC controlOperations

The RCC `controlOperations` `afterConfigure` and `beforeQuery` have been deprecated and removed from the IDE. They were never called by the Framework and the C++ interface provides the finer-granularity `XXX_read()` and `XXX_written()` notification methods.

D-II.8 64-Bit Build Host Issues (“Value too large for defined data type”)

Some 32-bit cross-compilers exhibit somewhat-rare problems on 64-bit host machines using file systems that allow 64-bit inodes, *e.g.* XFS. A new RPM, `ocpi-rereq-build-support-inode64`, provides a workaround. Additional information can be found in the ANGRYVIPER-provided Picoflexor T6A documentation.

D-III Major Known Deficiencies

This is a list of major known deficiencies at the release of 1.1 that were not previously documented in Section E. This is *not* an exhaustive list.

D-III.1 Maximum Message Length in PCI-Based Platforms

PCI-based platforms, *e.g.* ML605 and ALST4, are limited to a 32KB buffer. By default, that buffer is split into two, resulting in two 16KB buffers. The maximum message length, in bytes, is then 16KB-16¹⁴. RCC Workers will see any data beyond the limit as zero-filled.

¹⁴16 bytes required for metadata

D-III.2 MAC Address on ZedBoard

The currently used Linux release (“`xilinx13.3`”) cannot properly set the MAC address on the ZedBoard. This should only affect users who have *more than one ZedBoard* on a single network. Consult the *ZedBoard Getting Started Guide* for a workaround.

D-III.3 Platform Development Guide

While all documentation is constantly under improvement, the *Platform Development Guide* in particular is currently undergoing major changes, in part due to the SDP improvements. If an end user intends to create a Board Support Package (BSP) for a new target platform, they should contact the ANGRYVIPER Team to ensure the latest Guide is used.

E Release 1.0.1

These are “Release Notes” from Version 1.0.1 that are still relevant as of Version 1.1.

E-I OpenCPI HDL Assembly Filename Constraint

Due to a limitation with the ISE Simulator (isim), the filename of any HDL Assembly must be all lowercase letters. A capital letter anywhere in the filename will result in an error.

E-II 32-bit Architecture File Size Limit

On some software platforms, there will be an error when writing files greater than 2^{31} bytes (about 2.2 GB), especially if using NFS. This includes the ARM 32-bit architecture on Zynq-based platforms.

E-III Eclipse Project Description File

A necessary file known as the project description file (`.project`) is generated when creating a Project in the ANGRYVIPER IDE. This file is required by Eclipse and is used to describe the project in its workspace. Removal of this file will cause an error.

E-IV Refreshing the ANGRYVIPER IDE

Whenever files are being created or removed via command-line for an Project, the Project within the IDE needs to be refreshed. Examples include executing `make clean`, `make tests`, `make rcc`, etc. Many of the `make` commands will create or remove files in the Project, so it is important to refresh the Project within the IDE to reflect the changes.

E-V Non-default Xilinx Installation Locations

OpenCPI assumes Xilinx’s installation location is `/opt/`. If the Xilinx tools were installed in another location, the cross compiling environment variable can not be inferred correctly. To resolve this issue, refer to the `/opt/opencpi/cdk/env/xilinx.sh.example` file and the *Installation Guide*.

E-VI Zero-Length Message Finite State Machines

Some workers make use of a finite state machine to monitor a zero-length message (*ZLM*) from an upstream worker. This implementation was designed specifically with unit tests in mind so that simulations could be ended with an end-of-file (EOF) condition from the File Read Worker rather than waiting for a fixed amount of time. Simulation times vary across machines and Workers, so predicting how much time a Worker might need on various machines is not deterministic. In the future, the handling of zero-length message may change, but the current finite state machines for handling them are provided to complete the unit tests of the workers.