

# Release Notes

Version 1.3.0

*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

# Table of Contents

<b>1</b>	<b>Release 1.3</b>	<b>5</b>
1.1	Enhancements	5
1.1.1	OSS Release	5
1.1.2	Python and GNU Radio Integration (Technology Preview)	5
1.1.3	AD9361 Support	5
1.1.4	GUI Improvements	5
1.1.5	Project Registry	6
1.1.6	Project Migration	6
1.1.7	ocpidev Improvements	6
1.1.8	RCC Platforms	6
1.1.9	Remote Containers (Technology Preview)	6
1.1.10	OpenCL Containers (Technology Preview)	6
1.2	Changes and Notes	6
1.2.1	Major Project Restructuring	7
1.2.2	Upgrade Notes	7
1.2.3	Prerequisite RPMs Built With Newer Language Standards	7
1.2.4	Legacy Platform Names Continue to be Deprecated	7
1.2.5	OCPLXILINX_VIVADO_DIR Variable Changed	7
1.2.6	ocpidev and Requesting RCC Platform To Match HDL Platform	7
1.2.7	zero_padding Component in <code>ocpi.assets</code> deprecated	8
1.3	Major Known Deficiencies	8
1.3.1	Previously Documented Deficiencies	8
<b>2</b>	<b>Release 1.2</b>	<b>8</b>
2.1	Enhancements	8
2.1.1	Vivado Support for Applicable Xilinx-Based Platforms	8
2.1.2	New command option “ocpidev build”	9
2.1.3	GUI Now Delivered as RPM	9
2.1.4	Artifact Version Checking	9
2.1.5	Miscellaneous Improvements	9
2.2	Changes and Notes	9
2.2.1	rx-spec and tx-spec Support Property Interdependence	9
2.2.2	Vivado Support	9
2.3	Major Known Deficiencies	9
2.3.1	Driver Failing to Allocate Memory on Zynq-Based Platforms	9
2.3.2	HDL Workers Cannot Send SOM Without Valid Data	10
2.3.3	SDP Cannot Handle More Than One ZLM Per Application Run	10
2.3.4	File_Read and File_Write Cannot Handle ZLM Mid-file	10
2.3.5	PCI-Based Platforms Not Always Detected in Remote Tests	10
2.3.6	Built Projects Cannot Be Moved	10
2.3.7	Properties Before Raw Properties Cannot Have Parameterized Lengths	10
2.3.8	XML Parser Mixed-Case Issue	10
2.3.9	Simulation of Primitive Cores Needs <code>Makefile</code> Modified	10
2.3.10	Lime RX/TX Proxy Tests on CentOS6	10
2.3.11	Previously Documented Deficiencies	11
2.4	Addenda to Previous Release Notes	11
<b>3</b>	<b>Release 1.1</b>	<b>12</b>
3.1	Enhancements	12
3.1.1	Debug Symbols for Framework Now Available	12
3.1.2	Kernel Driver Memory Arguments	12
3.1.3	HDL File_Read and File_Write CWD	12
3.2	Changes and Notes	12
3.2.1	Simulators Renamed and Work Out-of-the-Box	13
3.2.2	HDL Width Adapters Not Required	13

3.2.3	Platform Name Changes . . . . .	13
3.2.4	RCC Cross-compilation No Longer Uses <code>OCPI_CROSS_HOST</code> . . . . .	13
3.2.5	Updating a 1.0.x Unit Test for the 1.1 Framework . . . . .	13
3.2.6	Prerequisite RPM Changes . . . . .	14
3.2.7	Deprecated RCC controlOperations . . . . .	14
3.2.8	64-Bit Build Host Issues (“Value too large for defined data type”) . . . . .	14
3.3	Major Known Deficiencies . . . . .	14
3.3.1	Maximum Message Length in PCI-Based Platforms . . . . .	14
3.3.2	MAC Address on ZedBoard . . . . .	15
3.3.3	Platform Development Guide . . . . .	15
<b>4</b>	<b>Release 1.0.1</b>	<b>16</b>
4.1	OpenCPI HDL Assembly Filename Constraint . . . . .	16
4.2	32-bit Architecture File Size Limit . . . . .	16
4.3	Eclipse Project Description File . . . . .	16
4.4	Refreshing the ANGRYVIPER IDE . . . . .	16
4.5	Non-default Xilinx Installation Locations . . . . .	16
4.6	Zero-Length Message Finite State Machines . . . . .	16

## 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. *It is highly recommended that the user review the entire document and not only the section concerning the latest release.*

## 1 Release 1.3

### 1.1 Enhancements

#### 1.1.1 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.

#### 1.1.2 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++<sup>1</sup>. 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.

#### 1.1.3 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.

#### 1.1.4 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.

---

<sup>1</sup>This integration is only available on CentOS 7 platforms.

- **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.

### 1.1.5 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. 1.1.6) 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. 1.1.7).

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 2.3.6.

### 1.1.6 Project Migration

With the new Project Registry (cf. 1.1.5), 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 2.3.6.

### 1.1.7 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 (1.1.5) manipulations

### 1.1.8 RCC Platforms

With the new Project Registry (cf. 1.1.5), 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.

### 1.1.9 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.

### 1.1.10 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.

## 1.2 Changes and Notes

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

### 1.2.1 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*.

### 1.2.2 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.

### 1.2.3 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.

### 1.2.4 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.

### 1.2.5 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`.

### 1.2.6 ocpidev and Requesting RCC Platform To Match HDL Platform

When building Projects with `ocpidev`, 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.

### 1.2.7 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.

## 1.3 Major Known Deficiencies

### 1.3.1 Previously Documented Deficiencies

The following deficiencies are still present in 1.3:

- 4.2: 32-bit Architecture File Size Limit
- 4.1: HDL Assembly Filename Constraint
- 2.3.8: XML Parser Mixed-Case Issue
- 3.3.1: Maximum Message Length in PCI-Based Platforms
- 3.3.3: Platform Development Guide
- 2.3.7: Properties Before Raw Properties Cannot Have Parameterized Lengths
- 3.3.2: MAC Address on ZedBoard
- 2.3.5: PCI-Based Platforms Not Always Detected in Remote Tests
- 2.3.10: Lime RX/TX Proxy Tests on CentOS6

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

- 2.3.1: Driver Failing to Allocate Memory on Zynq-Based Platforms
- 2.3.6: Built Projects Cannot Be Moved
- 2.3.2: HDL Workers Cannot Send SOM Without Valid Data
- 2.3.4: File\_Read and File\_Write Cannot Handle ZLM Mid-file
- 2.3.3: SDP Cannot Handle More Than One ZLM Per Application Run

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

- 2.3.9: 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.

## 2 Release 1.2

### 2.1 Enhancements

#### 2.1.1 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.



### 2.1.2 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*.

### 2.1.3 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.

### 2.1.4 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*.

### 2.1.5 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 properties<sup>2</sup>

## 2.2 Changes and Notes

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

### 2.2.1 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.

### 2.2.2 Vivado Support

Please see 2.1.1 for modifications that *may* need to be performed.

## 2.3 Major Known Deficiencies

### 2.3.1 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.

---

<sup>2</sup>This is noted here because it is one of the first user reported bugs.

### 2.3.2 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.

### 2.3.3 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 2.3.4. (See also 4.6.)

### 2.3.4 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 2.3.3. (See also 4.6.)

### 2.3.5 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.

### 2.3.6 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/opencpi/cdk/scripts/` and are named `freeze_project.sh` and `thaw_project.sh`.

### 2.3.7 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.

### 2.3.8 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.

### 2.3.9 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.

### 2.3.10 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_clkin;
+   ostr_clkin << (2 * sample_rate * 1e6);
+   ClkInStr = ostr_clkin.str().c_str();
```

### 2.3.11 Previously Documented Deficiencies

The following deficiencies are still present in 1.2:

- 4.2: 32-bit Architecture File Size Limit
- 4.1: HDL Assembly Filename Constraint
- 3.3.2: MAC Address on ZedBoard
- 3.3.1: Maximum Message Length in PCI-Based Platforms
- 3.3.3: Platform Development Guide

## 2.4 Addenda to Previous Release Notes

- The default `CWD_MAX_LENGTH` noted in 3.1.3 is now set to 256.
- The Liquid RPMs referenced in 3.2.6 are now **required** to build the `ocpiassets` project (`complex_mixer.rcc`). They are still considered optional when installing.

## 3 Release 1.1

### 3.1 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-bit<sup>3</sup>
- New automatic test harness creation, including limited remote platform support<sup>4</sup>
- Automatic bus width adaption between HDL Workers
- Additional Assets components to help design and debug Applications
- HDL Simulator on-the-fly launching<sup>5</sup>
- 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)

#### 3.1.1 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`.

#### 3.1.2 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.

#### 3.1.3 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.

### 3.2 Changes and Notes

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

---

<sup>3</sup>Currently limited to some Zynq-based platforms.

<sup>4</sup>Most `ocpiassets` have been moved to the new test system to provide examples.

<sup>5</sup>See Section 3.2.1 for more information.

### 3.2.1 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`<sup>6</sup> 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
```

### 3.2.2 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 `wsi_16_to_32`, `wsi_32_to_16`, or `wsi_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.

### 3.2.3 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`.

### 3.2.4 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 3.2.3). 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.

### 3.2.5 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 3.2.1 and 3.2.3). 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

<sup>6</sup>Only 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.

### 3.2.6 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*.

### 3.2.7 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.

### 3.2.8 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.

## 3.3 Major Known Deficiencies

This is a list of major known deficiencies at the release of 1.1 that were not previously documented in Section 4. This is *not* an exhaustive list.

### 3.3.1 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-16<sup>7</sup>. RCC Workers will see any data beyond the limit as zero-filled.

---

<sup>7</sup>16 bytes required for metadata

### 3.3.2 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.

### 3.3.3 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.

## 4 Release 1.0.1

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

### 4.1 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.

### 4.2 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.

### 4.3 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.

### 4.4 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.

### 4.5 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` file and the *Installation Guide*.

### 4.6 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.