Abstract: This document defines the release notes for the 2.14.0 release of the OpenPegasus CIM Server. The purpose of this PEP is to summarize the characteristics of this release, point to other documentation that defines the release in more detail, and provide additional information about this release that is not available in the other Pegasus documentation.

Contents

1. Availability of this Release
2. OpenPegasus Functionality of This Release
3. Compatibility Considerations
4. Notes on Specific OpenPegasus Functionality
5. Relationship to CIM/WBEM Standards
6. OpenPegasus Supported Platforms
7. Conformance with DMTF Specifications
8. OpenPegasus Bugs
9. OpenPegasus Release Control and Version Definition Documentation
10. General Documentation

Availability of this Release

This release is available in a number of forms including:

- **Source release** in both ZIP (and ZIP EXE) and TAR formats. These can be downloaded from the OpenPegasus Web site at [www.openpegasus.org](http://www.openpegasus.org).
- **OpenPegasus source RPMs** on the OpenPegasus web site. A source RPM is available on the OpenPegasus web site that can be used to build binaries for most LSB-compliant RPM-based Linux
distributions and versions.

- **The OpenPegasus CVS repository.** The CVS tags representing this release and subsequent snapshots that are made available in the Pegasus CVS are defined on the OpenPegasus CVS repository web page (see the OpenPegasus CVS page for information) using the tags defined on the release snapshots page and the OpenPegasus WIKI Release Status Page.

The instructions for acquiring the released code are on the Pegasus WEB site and the OpenPegasus WIKI. Installation instructions are part of the README in the root of the Pegasus source tree.

**OpenPegasus Functionality for this Release**

OpenPegasus overall status by release is defined by a Feature Status WEB Page that is available on the OpenPegasus web site. That web page shows the Pegasus users and developers the status of the various features found in OpenPegasus using a simple color coded key (white, red, yellow, green) and a description of each feature.

OpenPegasus 2.14.0 is a major release, extending the previous Pegasus release in selected areas as described in these release notes.

NOTE: OpenPegasus releases are categorized as follows:

- First number of version changes (ex. 2.x.x to 3.0.0) - Major release includes major functionality change and incompatible behavior changes and/or incompatible public API changes,
- Second number changes (ex 2.12.0 2.14.0) - Minor Release and includes new functionality but maintains behavior and public API compatibility, 
- 3rd number changes (2.14.0 to 2.14.1) - Revision(also called point release) release and only includes bug fixes.

**ALL** changes for each release are documented in the OpenPegasus bug data base by tagging each bug(keyword field of bug). No change to Pegasus is committed with out this tag on the bug. Changes for this minor release are tagged 2.14.0_APPROVED. All changes for this minor release can be viewed through this link to the OpenPegasus bugzilla [Pegasus 2.14.0_APPROVED bug list](https://bugzilla.redhat.com). Bugs marked as enhancements can be reviewed through the this [OpenPegasus bugzilla 2.14.0 enhancement bugs](https://bugzilla.redhat.com) . Bugs fixed for subsequent revision releases (ex. 2.14.1) would also be viewed through corresponding Bugzilla tags for each revisions (ex 2.14.1_APPROVED).

The major areas of development including both enhancements and bug fixes for this release were as follows. Note that the bugs listed represent only those features incorporated in bugs without PEPS that are considered enhancements, not all bugs incorporated in the release. For more information access the PEP/BUG for each change defined below:

**Table of Major Changes this Release. Review Bugzilla 2.14.0 Enhancements and approved(above) for a complete list**

<table>
<thead>
<tr>
<th>BUG #</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>9601</td>
<td>Support only TLS V1.2 Protocol for Security Compliance</td>
</tr>
</tbody>
</table>

9676, 9819 Support DMTF defined Pull Operations (per DMTF specification DSP 0200 and 0201). This is a major extension to OpenPegasus and implements all of...
the DMTF defined pull operations with NO changes to providers. This includes extensions to both the server and client code as well as new operations implemented in cimcli to allow execution of pull operations and pull operations with FQL.

9721 Fix issue causing failures when cmpi returned instances that do not have a class. This was causing problems with the Jobs profiles which specifically return information for which there is no class.

9724 Dynamic disable of the reliable Indication feature

9812 FootPrint Reduction

9831 Generate mini-CA and signed certificate instead of self-signed certificates

9832 Include cimcli in release packages

9883 support HTTP Negotiate Authentication in OpenPegasus

9892 Reduce overhead of use of PAM by use of a SessionCookie. This also introduces a new runtime configuration parameter to control the session timeout

9928 Incorporate FQL (Filter Query Language) support for the pull operations (Experimental)

9880, 9778 Several minor extensions to the APIs (ex. extend String class methods)

9737 Improve CLANG compiler support

9219 Initial CIMRS and Web Admin support (This is Experimental and the CIMRS function will be modified for CIMRS V2 specification)

9853 Correct old error in the Memory Resident Repository logic. This logic did not work in several previous versions because of the issues documented in this bug.

9926 Update default CIM Schema to CIM 2.41

Add scripts for load testing of OpenPegasus (see the directory src/Unsupported/Scripts/UinxLoadTests)

10012 Fix issue with the configuration parameters for numberOfTraceFiles and traceFileSizeKbytes.
The status of several components of Pegasus functionality change status between Pegasus versions. Features that have changed status can be identified by a change in color from the previous release to this release on the Feature Status WEB Page.

Notes on Specific OpenPegasus Functionality Changes

This section documents specific issues that the OpenPegasus team feels are important to OpenPegasus users with the current release (and possibly older releases).

Add new RunTime Variables to control trace file size and Rotation(OpenPegasus 2.13.0) (bug 9550)

As of OpenPegasus 2.14.0, several new runtime variables have been added that can be manipulated with the cimconfig utility:

* **pullOperationsMaxObjectCount** - Defines the maximum allowed value of the maxObjectCount argument on open and pull operation requests which defines the maximum number of instances or paths that will be returned in a single open or pull response. Requests that have a value higher than this configuration parameter will be rejected. The default is 10000. This may be set to any value between 1 and 10000.

* **pullOperationsDefaultTimeout** - Defines the default timeout in seconds between the time a pull open response is sent by the server and a subsequent pull or close response received from the client if the value is not set by the client in the request. This can be set to any value between 1 and 90 seconds. Default=30

* **pullOperationsMaxTimeout** - Defines the maximum allowable value for the maxTimeout argument that is part of the pull open... (openEnumerationInstances, etc) requests. Any value higher than the value set in this configuration parameter will cause the request to be rejected by the CIMServer. The default=90 seconds.

TODO - Explain more on all of these above

* **httpSessionTimeout** - Allows using a cookie to temporarily bypass the PAM authentication in favor of the cookie provided by the server during a single session. This is a performance enhancement. If this variable is set to zero (default) the cookie bypass is disabled, the server will not send nor accept the id cookie. If set to an integer, a client session will maintain the cookie for the number of seconds defined by the value of this variable. Enabled with a build parameter PEGASUS_ENABLE_SESSION_COOKIES.

* **sslBackwardCompatibility** - Limits OpenPegasus support to TLS 1.2/OpenSSL1.01. Can be set to allow backward compatibility with older versions of TLS/OpenSSL.
The following sections describe in more detail some of the more significant changes in this release as well as carrying forward some of the same information for the last release to be sure users are aware what these changes affect.

**DMTF Pull Client operations and FQL Query Language (OpenPegasus 2.14.0)**

As of DMTF DSP0200 version 1.4, new client operations (generally named the pull operations have been part of the CIM/XML protocol). This consists of a set of new operations (open..., pull..., close) that allow getting enumerations, associations, references (and their corresponding name operations) as a sequence of operations (an enumeration sequence) rather than a single monolithic operation.

This will provide several advantages including:
1. Better control of client memory since the client can determine the size of each response.
2. Clearly separates the return of errors from data.

Pegasus 2.14 supports all of these operations in both the CIM/XML client and the server. For more information on the implementation and use of these operation see the following documents:

1. OpenPegasus Wiki page for this project.
2. PEP documenting the Pull operations and FQL

**HTTP Sessions (OpenPegasus 2.14.0)**

Allows using a cookie to enhance performance of PAM. This function is enabled with the build variable PEGASUS_ENABLE_SESSION_COOKIES. When the option is enabled, the OpenPegasus CIM server will use HTTP cookies for session management (RFC 6265).

After a successful client authentication the client is given a cookie. The client is then not asked for re-authentication as long as it provides the same cookie in all subsequent requests and until the session expires. Session expiry is configurable using httpSessionTimeout configuration option. Nothing changes for clients that do not support HTTP cookies (RFC 6465) - their requests are authenticated as usual, either using Basic or Negotiate authentication mechanisms.

The timeout for a session is determined by the runtime parameter httpSessionTimeout. If this runtime parameter is set to zero the HTTP cookies option is disabled.

This option has proven to represent a significant improvement in performance over using PAM for every client operation.

**WEBAdmin (OpenPegasus 2.14.0)**

OpenPegasus has included on an experimental basis a web server that acts as an administration tool to replace the command line tools. This option presents the same information as the major command line tools as web pages so that parameters can be modified without using the command line tools. It is compiled by default and may be disabled with the build option PEGASUS_ENABLE_PROTOCOL_WEB=false. See the file readme.webadmin in the OpenPegasus source tree for more detailed information on using WEBAdmin.

Note: This option is considered experimental for 2.14
Using GCC 4.7 Compiler on Linux OS (OpenPegasus 2.13.0)

As of 2.14.0, OpenPegasus has been updated to replace the custom atomic operations with GCC built-in atomic operations if GCC version 4.7 or greater is used as the compiler. It is recommended that if possible this version of the GCC compiler be used since the change provides significant performance improvements.

OpenPegasus and OpenSLP V 2.0 (OpenPegasus 2.13.0)

As of 2.13.0, OpenPegasus slp fully supports IPV6 and has been tested with OpenSLP 2.0 (Recently released by OpenSlp.org)

While testing OpenPegasus with OpenSLP 2.0 the OpenPegasus team found two issues with this version of OpenSLP. The patches documented below allow this version of OpenSLP to work with OpenPegasus

1) SLPReg used TCP connection previously, but it now uses UDP. Therefore, a BUFFER_OVERFLOW exception occurs. This is because UDP cannot handle attribute size greater than network MTU size. In such cases, according to the specification, OpenSLP should automatically switch to TCP. Since this has not been implemented in OpenSlp 2.0, the workaround used by OpenPegasus to force use of TCP may be found at http://sourceforge.net/p/openslp/bugs/139/.

2) On Windows, slpd service fails to start when IPV6 is enabled and throws the error message "Error 1067: the process terminated unexpectedly". More details and the fix for this issue are in the bug http://sourceforge.net/p/openslp/bugs/140/.

Interop namespace name (OpenPegasus 2.13.0)

Effective with the release of OpenPegasus 2.13 the "interop" namespace support for existing or older repositories which used "root/PG_InterOp" has been added via a new option in the "reupgrade" utility. Users with existing repositories may use this option to migrate the repository from "root/PG_InterOp" to "interop" namespace. See bug 9414 (PEP304) for details.

For this to work, the build option "PEGASUS_INTEROP_NAMESPACE" has to be set to "interop". Only then will the "-i" option in repupgrade be enabled. Once the repository is upgraded with "-i", the namespace "root/PG_InterOp" will cease to exist. Clients using "root/PG_InterOp" must now use "interop".

Prior to version 2.12 OpenPegasus used an OpenPegasus internal variable to define "root/PG_InterOp" as the name for the OpenPegasus namespace with no defined way to change this variable. Note that an unsupported method involving editing one file and rebuilding OpenPegasus was defined as a workaround in the OpenPegasus wiki FAQs.

However, since the definition of a standard name for this namespace in DMTF and SNIA specifications ("interop" or "root/interop" with "interop" being the preferred alternative) the use of "root/PG_InterOp" has become an issue. Support for forward-compatibility is a fundamental design principle for the OpenPegasus project. As a community, our goal is for well-behaved OpenPegasus providers or clients, using only the externally defined OpenPegasus interface, to continue to work with each version upgrade of OpenPegasus.

Effective OpenPegasus 2.12.0 a new build variable was defined (PEGASUS_INTEROP_NAMESPACE) which allows an OpenPegasus builder to define an alternate name for the Interop namespace to either "interop" or "root/interop" (ex. PEGASUS_INTEROP_NAMESPACE = interop). Once this build variable is defined and OpenPegasus compiled, the actual name of the interop namespace is what was defined in this variable and all components of OpenPegasus use this as the interop namespace name include the complete OpenPegasus test suite. The only name for the interop namespace will be the name defined through this build variable. This is the logical approach for implementations starting a new repository.
In order to supexport PEGASUS_CLIENT_TRACE=keyword:keywordport users with existing repositories, in 2.13.0, migration of existing repositories was integrated into the "repupgrade" utility to allow existing OpenPegasus environments to smoothly move the server from use of the old name "root/PG_InterOp" to the preferred name "interop".

**NOTE:** If the interop namespace name is redefined with PEGASUS_INTEROP_NAMESPACE clients using the "root/PG_InterOp" namespace will not connect with the updated version of OpenPegasus. The OpenPegasus team searched for a solution to the issue supporting the old and new name at the same time and concluded that there were so many issues that it could not effectively be done.

### Configure Script to Control OpenPegasus Builds (OpenPegasus 2.13.0)(Bug 9592)

OpenPegasus is moving from the use of environment variables to a configure script to control build. This is both to make it easier to define the build variables, etc. and to be more compatible with the Linux/Unix standard build environment. However, because OpenPegasus must build on a number of platforms that do not support the full Linux/Unix configuration tool set, this is an OpenPegasus defined configure script. The first very experimental version is part of OpenPegasus 2.13.0 but this version has a number of limitations (see the OpenPegasus Bugzilla) and will be expanded in the next versions of OpenPegasus.

The goal is to replace the setting of most of the OpenPegasus build environment variable with options in the pegasus/configure script where those options are compatible with the Linux standard for defining configure options and those options control the entire build process including placement of output and choice of build options.

### Tracing Client Requests and Responses(OpenPegasus 2.13.0)

OpenPegasus has long contained a hidden build controlled function to allow tracing at the client through an environment variable (PEGASUS_CLIENT_TRACE_ON). In 2.13.0 this was a) make a permanent part of the environment (Bug 9564) and the calling convention slightly changed to make it simpler to use. The original format for the environment variable was:

```
export PEGASUS_CLIENT_TRACE=keyword:keyword
```

where the keyword was "con" | "log" | "both"

This was changed to:

```
keyword:Keyword separately define Client input and output
Keyword: Client Input only
:Keyword Client Output Only
Keyword Client Input and output defined by keyword
```

so that the normal setup for a console output trace would be

```
export PEGASUS_CLIENT_TRACE=con
```

### Information on IPV6 Support and OpenPegasus(PEP 291)(OpenPegasus 2.7.0)

**NOTE:** This functionality was incorporated in OpenPegasus 2.7.0 but it was felt that the information below was worth repeating in subsequent Release Notes

IPv6 Support for OpenPegasus is documented in [PEP 291](https://example.com/pep291).

The IPv6 support in OpenPegasus is controlled by the PEGASUS_ENABLE_IPV6 build variable. The default for this variable is "true"; setting this to 'false' before building OpenPegasus will disable the IPv6 support.
The following subsections provide some information on IPv6 support for the Linux and Windows platforms.

Note that in the 2.6.1 release there is no automatic run-time detection of IPv6 support on a platform by OpenPegasus. If you build with PEGASUS_ENABLE_IPV6=true, your platform must support IPv6 or you will get a build failure (missing header files), a run-time bind() failure, or possibly some other failure. This applies to both the CIM Server and Listener components. For more information on IPv6 support for your specific platform, refer to the documentation for that platform.

All OpenPegasus externals that support eticompilerher a hostname or an IP address as input have been updated to allow an IPv6 address to be specified. If the required input is just an IP address (eg. no optional or required port number), then the IPv6 address must be specified without brackets. For example, the OpenPegasus osinfo client (which returns information about the OS running on a host system) takes separate host and port options. In this case a host with an IPv6-configured network interface would be specified as:

```
  osinfo -h 3ffe:1:2:1 -p 5988
```

But the cimcli command, which takes an optional "location" option including an optional port number, requires the IPv6 address to be delimited with brackets, for example:

```
  cimcli ns -l [3ffe:1:2:1]
```

or

```
  cimcli ns -l [3ffe:1:2:1]:5989
```

For more information on specifying IPv6 addresses, refer to IETF RFC 2373 - IP Version 6 Addressing Architecture and IETF RFC 2732 - Format for Literal IPv6 Addresses in URL's.

**IPv6 on Linux**

Modern Linux distributions already contain IPv6-ready kernels, the IPv6 capability is generally compiled as a module, but it's possible that this module is not loaded automatically on startup.

**Note:** you shouldn't use kernel series 2.2.x, because it's not IPv6-up-to-date anymore. Also the IPv6 support in series 2.4.x is no longer improved according to definitions in latest RFCs. It's recommend to use series 2.6.x now.

To check whether your current running kernel supports IPv6, take a look into your /proc filesystem. The entry /proc/net/if_inet6 must exist. A short automated test looks like:

```
  test -f /proc/net/if_inet6 && echo "Running kernel is IPv6 ready"
```

For more iSchema version default update to CIM 2.36nformation on enabling IPv6 in the Linux kernel and configuring network interfaces, refer to The Linux Documentation Project's IPv6 HOWTO.

**Warning:** There is currently an open issue with RedHat to address a problem that is seen intermittently on RHEL5 and SLES10 systems. This problem is described in Pegasus bug 6586 and RedHat bug 248052, and manifests itself as an intermittent IPv6 socket failure. You should be aware that OpenPegasus with IPv6 enabled may be significantly impacted in these environments.
IPv6 on Windows

Microsoft provides supported IPv6 implementations for Windows Server 2003, Windows XP with Service Pack 1 (SP1), Windows XP with Service Pack 2 (SP2), and Windows CE .NET 4.1 and later.

On Windows XP, you can use the `ipv6.exe` tool to install, uninstall, and query your IPv6 configuration. For example:

```
ipv6 install
ipv6 if
```

Windows Vista and Windows Server 2008 support an integrated IPv4 and IPv6 implementation known as the Next Generation TCP/IP stack. Note that the OpenPegasus IPv6 support has not yet been tested on Windows Vista or Windows Server 2008.

For more information on installing, configuring, and using IPv6 on Windows platforms, refer to the document [IPv6 for Microsoft Windows: Frequently Asked Questions](#).

**Testing OpenPegasus IPv6 support**

As part of the OpenPegasus automated tests, the TestClient, g11ntest, and IndicationStressTest test clients were modified to run tests using the IPv6 loopback address (::1) if PEGASUS_ENABLE_IPV6=true.

You can also perform manual tests using the IPv6 loopback or a real IPv6-configured network interface. For example, using the IPv6 loopback on a system with OpenPegasus running on the WBEM standard http port of 5988:

```
osinfo -h ::1
```

**OpenPegasus Compatibility Considerations**

Support for forward-compatibility is a fundamental design principle for the OpenPegasus project. As a community, our goal is for well-behaved OpenPegasus providers or clients, using only the externally defined OpenPegasus interface, to continue to work with a minor version upgrade of OpenPegasus. However, there are certain classes of errors (e.g., non-compliance with a standard that significantly affects interoperability) that may require the community to make potentially incompatible changes. The following table contains a list of defects/fixes that MAY impact, even well-behaved, OpenPegasus providers or clients when upgrading to this OpenPegasus release.

<table>
<thead>
<tr>
<th>Bugzilla #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bug 8830</td>
<td>Starting with OpenPegasus 2.11.0 the requirement for providers (CMPI and C++) to filter properties from instances has been relieved. The server will do the filtering at the protocol adapter level. To avoid a negative impact on performance the CIMInstance::filter() and CMPIInstanceFT.setPropertyFilter() functions have been changed to a NOP. This may be a change in behavior for provider relying in some unknown way on the instance being filtered through these functions. To allow users to actually filter properties from an instance in a provider, a new function (CIMInstance::filterInstance()) will actually filter properties from an instance</td>
</tr>
</tbody>
</table>
exactly as the CIMInstance::filter() did in previous versions of Pegasus. This is not required since the CIMServer will do the filtering but allows the provider to prefilter properties if it so desires, in particular where properties might be very large or there would be a significant number of properties.

**NOTE: This bug fixed in OpenPegasus 2.13.0.** Reregistering of an indication provider requires that the cimserver be restarted to send the enableIndication request to the providers. This issue has existed starting with OpenPegasus 2.10 to version 2.12. If the cimserver is not restarted the indication will not be enabled for the reregistered provider.

### OpenPegasus Relationship to CIM/WBEM Standards

Pegasus was designed and implemented to comply with the DMTF CIM/WBEM specifications. The following table defines the level of the DMTF specifications to which this version of Pegasus aims to conform today.

<table>
<thead>
<tr>
<th>DMTF Specification</th>
<th>Specification Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSP 0004 - CIM Infrastructure Specification</td>
<td>Version 2.3 Final, 4 October 2005</td>
</tr>
<tr>
<td>DSP 0200 - CIM Operations over HTTP</td>
<td>Version 1.4, 26 August 2013</td>
</tr>
<tr>
<td>DSP 0201 - Representation of CIM in XML</td>
<td>Version 2.4.0 16 January 2014</td>
</tr>
<tr>
<td>DSP 0202 - CIM Query Language Specification</td>
<td>1.0.0, preliminary, 9 December 2004</td>
</tr>
<tr>
<td>DSP 0205 - WBEM Discovery using SLP</td>
<td>1.0, preliminary, 27 January 2004</td>
</tr>
<tr>
<td>DSP 0206 - WBEM SLP Template</td>
<td>1.0.0, Preliminary, January 2004</td>
</tr>
<tr>
<td>CIM Schema</td>
<td>2.36.0 Final (default build Schema)</td>
</tr>
<tr>
<td>DSP0226 - Web Services for Management</td>
<td>Version 1.1.1</td>
</tr>
<tr>
<td>DSP0227 - WS-Management CIM Binding Specification</td>
<td>Version 1.2.0</td>
</tr>
<tr>
<td>DSP0230 - WS-CIM Mapping Specification</td>
<td>Version 1.1.0</td>
</tr>
<tr>
<td>DSP0212 - Filter Query Language</td>
<td>Version 1.0.1, 22 August 2013</td>
</tr>
</tbody>
</table>
DMTF/SNIA SMI Profiles Supported

Today OpenPegasus includes support for several DMTF profiles including:

1. DMTF ProfileRegistration Version DSP 1033, Version 1.0.0
2. DMTF Indications DSP 1054, Version 1.1.0. Since this is effectively the same as the SNIA SMI Indication profile it also supports this profile. This profile is enabled by compiling OpenPegasus with the build variable PEGASUS_ENABLE_DMTF_INDICATION_SUPPORT.
3. SNIA Profile Registration 1.0.0
4. SNIA Server 1.1.0 & 1.2.0

Conformance Exceptions to DMTF Specifications

It is a major goal of the OpenPegasus project to both drive and utilize the DMTF CIM/WBEM specifications. However, today there are a number of known differences. As of this release, some of the differences include the following:

1. **Provider Registration** - Pegasus supports registration through a set of Pegasus Provider registration classes rather than support the DMTF defined Provider Qualifier. Pegasus will not support this qualifier in the future. See the readme for more information on provider registration. This is not really a deviation from the specifications since there is no specification or profile for provider registration today but is a different method of registration than many CIM Servers and means that the provider qualifier is unused in Pegasus.

2. **Lifecycle Indications** - Today Pegasus supports process iPegasusindications and lifecycle indications when defined specifically by providers. Pegasus does not support lifecycle indications if they are not defined by the provider or for changes to CIM Classes. For more information see the Pegasus CVS file: pegasus/src/Unsupported/LifecycleIndicationProvider/README.LifecycleIndications.html.

3. **IncludeQualifiers option on Instance Operations** - The DMTF specifications have deprecated the use of qualifiers on instance operations with the 1.2 CIM Operations Over HTTP Specification (DSP0200) and stated specifically that the client should NOT depend on the includeQualifiers parameter of the CIM/XML instance operations returning qualifiers. Some CIM Client implementations expect instances returned from the CIM Server to include the qualifiers defined in the class definition (such as KEY qualifiers on properties). Pegasus today expects the provider to complete the qualifiers on instances in accordance with the specification and generally the interpretation is that Pegasus applies class level qualifiers when the includeInstance parameter is supplied on instance operations. However, Pegasus today is inconsistent in the return of qualifiers so that the user should NOT depend on Pegasus accurately honoring the includeQualifier parameter for instance operations. In the future it is expected that Pegasus will deprecate the use of qualifiers on instance operations completely. When Pegasus uses object normalization, the normalizer does apply qualifiers to returned instances if the includeQualifiers operation parameter is set.

4. **LocalOnly option on instance Operations** - With the 1.1 version of Specification for CIM Operations over HTTP, the definition of the LocalOnly parameter for the GetInstance and Enumerate Instances operations was modified. This change was incorrect, resulted in reduced functionality and introduced a number of backward compatibility issues. As a result of issues introduced by this change, we strongly recommend that CIM Clients set `LocalOnly = false` and do not rely on the use of this parameter to filter the set of set of properties returned. See Appendix C of this specification for more information.

5. **MultiRequest CIM/XML Option** - Pegasus does not support the DMTF defined MultiRequest Operation option (See DSP0200).
6. **MOF Namespace Pragma** - The Pegasus compiler does not support the namespace pragma defined in DSP 0004. Namespaces for the OpenPegasus repository are defined through arguments for the MOF Compiler.

7. **CIMOperation Trailer** - Pegasus implements chunking based on the DSP0200 1.2.0 preliminary spec. including the operation trailer defined in that specification. Refer to bug 6760 for more details. This trailer was completely removed from the DSP0200 1.2 final specification as a non-compatible change so that if chunking is used Pegasus is returning a noncompliant response to enumerate type requests.

8. **Whitespace in XML value Element** - Pegasus trims the leading and trailing whitespace from XML value elements that are of the Type String. This behavior is documented in Pegasus bug 8773 and there is a patch available for users. However, since the change is considered a change to behavior this will not be patched until pegasus does a major version update. NOTE: There are some documented bugs in Pegasus such as this that the OpenPegasus team feels cannot be incorporated into the code until a major version update. These bugs are documented with the Bugzilla tag "3.0_TARGET".

9. **modifyInstance operation behavior in the Pegasus repository does not match the DMTF specification**. Under some conditions (ex. if no propertyList is supplied) it modifies all properties rather than just the properties supplied with the request thereby Nulling out existing values. See bug 8752 for more information.

10. **OpenPegasus Interop Namespace** - The default interop namespace name in OpenPegasus is "root/PG_Interop". In versions of OpenPegasus prior to 2.12.0 this could be changed during build by modifying a string definition in the code. Effective version 2.12.0 this has become a build option with the build configuration environment variable PEGASUS_INTEROP_NAMESPACE = <name for this namespace> however, the default if built without this change is still "root/PG_InterOp". Effective with OpenPegasus 2.13.0 the reupgrade utility has been extended to allow existing repositories to be converted from the use of "root/PG_Interop" to "interop" (See discussion above). The default in the CVS source code is still "root/PG_InterOp".

11. **OpenPegasus WSMAN protocol on windows does not handle NaN, INF, or -INF** However, t (bug 8836) - If requests or responses include properties or parameters with floating point property types (Real32 or Real64) with the special states NaN, INF, or -INF on a Windows platform, the input is not accepted because these special strings are not understood by the decoder on Windows.

12. **OpenPegasus outputs Nan, INF, -INF special values for Real32 and Real64 property and Parameter types for the CIM/XML protocol** - (bug 9392). Today the DMTF Specification DSP0201 allows only numeric information in the format definition for this type and does not allow all the special strings Nan, etc. However, OpenPegasus actually outputs the special strings if that is what is defined in the internal values. OpenPegasus does not allow these special Strings on input however,

13. **OpenPegasus does not handle covered properties** - (bug 3571) Covered properties (properties which have the same name in a superclass and subclass without overriding the property in the superclass) are not allowed in OpenPegasus. While the requirement for this functionality was added to DSP0004 subsequent to version 2.4, the general agreement is that the requirement itself is not supportable and particularly not with the existing OpenPegasus APIs. Further, the requirement is expected to be removed in the future (i.e CIM 3.0).

14. **FQL Query Language** (bug 9956) incorporated has some limitations with respect to the DMTF defined Query Language (does not support full regex defined in the specification and does not support embedded instance comparison).

---

**OpenPegasus Supported Platforms**

Each OpenPegasus release lists as 'active' platforms those hardware/software platforms that have recognized ports for the Pegasus code base including a maintainer for the platform that will be willing to regularly document issues and/or fix defects as the Pegasus code base is changed. Pegasus may be operable on other platforms (ex. Windows 98) but without a group to provide the role of test and maintenance, correct
operation of Pegasus cannot be assured. In all cases, including the active platforms, it is the responsibility of the entity that packages and/or compiles OpenPegasus for shipment or deployment, to read, and actively monitor the Pegasus Bugzilla database for a list of relevant defects that affect their platform. The platforms that are considered ported are shown in the two tables below. The first table represents platform for which testing is done on a regular basis and reported to the Pegasus Nightly Build Test WEB Page. Those platforms which have been ported but do not have test results that are current at the time of the release are reported in the second table.

**Actively Supported Platforms (Nightly Tests Reported for this release)**

<table>
<thead>
<tr>
<th>Platform and OS</th>
<th>Compilers</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP-UX PA_RISC and Itanium</td>
<td>HP aC++ B3910B</td>
</tr>
<tr>
<td>Linux on Power</td>
<td>gcc</td>
</tr>
<tr>
<td>zLinux</td>
<td>gcc</td>
</tr>
<tr>
<td>Linux Itanium</td>
<td>gcc</td>
</tr>
<tr>
<td>Linux IA-32</td>
<td>gcc (versions 3.xx, 4.7) clang(The clang compiler usage is considered production effective OpenPegasus 2.13.0)</td>
</tr>
<tr>
<td>Linux X86_64</td>
<td>gcc (versions 3.xx, 4.7) clang(The clang compiler usage is considered production effective OpenPegasus 2.13.0)</td>
</tr>
<tr>
<td>z/OS V1.7 and up</td>
<td>XL C/C++ from z/OS Version 1.7 and up</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Platforms not Actively supported for this release (No current Nightly Build Test Results)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform and OS</td>
</tr>
<tr>
<td>--------------------------</td>
</tr>
<tr>
<td>MacOS version 10.3 and higher</td>
</tr>
<tr>
<td>Solaris 8</td>
</tr>
<tr>
<td>Solaris 9</td>
</tr>
<tr>
<td>HP OpenVMS 8.3 or later Alpha</td>
</tr>
<tr>
<td>HP OpenVMS 8.3 or later IA64</td>
</tr>
</tbody>
</table>

Microsoft Visual C++ Ver. 6 and Microsoft .Net compiler Version 7.1. Note: Visual C++ Ver. 6 no longer being regular tested.

CC Compiler Sun Studio 5.11

Platform patches

The following is a list known of platform patches Pegasus requires.

1. RHAS 2.1 needs the update to fix Red Hat Bugzilla 98815.
2. RHEL 4.4 multithreading bug in getpwuid-r could cause a CIM Server failure (Bugzilla 6232). This is fixed in RHEL 4.5
3. Red Hat and SUSE Linux multiple versions - multithreaded client intermittently fails connecting to IPv6 (Pegasus Bugzilla 6586) (Red Hat bug 248052)
4. OpenSLP Version 2.0 - This release of OpenSLP requires the patches for OpenSLP version 2.0 documented above.

Further information regarding Pegasus support on IBM platforms can be found at:

Further information regarding Pegasus support on HP platforms can be found at:

Pegasus Defects

The OpenPegasus Bugzilla database documents defects found in Pegasus and is available through the following link: OpenPegasus bugzilla database. Effective with the start of the OpenPegasus 2.6 work, ALL changes to the OpenPegasus CVS tree are documented through bugs. Therefore all source code changes to OpenPegasus are documented through bugs providing a complete history of changes and the reasons for those changes. Bugs reports are filed not only for bugs and their corresponding fixes but also as part of the process of committing new code representing the work on PEPs done for OpenPegasus 2.6 and all subsequent versions. Therefore, a search of the OpenPegasus Bugzilla base for bugs with the tag for a particular version (ex. 2.6.0_APPROVED, 2.6.1_APPROVED, 2.7.0_APPROVED, etc) will yield all changes to the Pegasus CVS tree for that OpenPegasus release.

Changes for this release (bugs Tagged 2.14.0_APPROVED). The link is Pegasus 2.13.0_APPROVED bug list.
Release Control and Version Definition Documentation

The OpenPegasus project is controlled largely through the CVS repository and a set of documents (PEPs) that serve both as the definition and review mechanism for new and modified Pegasus functionality and for the definition and documentation of releases.

The following documentation defines the characteristics of this Pegasus release. The documents are available in the OpenPegasus CVS repository.

- **OpenPegasus Release Definition/Status** - (See Wiki Section [OpenPegasus 2.13.x Release Status](#))
  - A section in the Pegasus wiki is used throughout the development of this version as the control point for functionality that will go into the release and points to all of the PEPs that represent the Pegasus functionality changes for this version of Pegasus.  
  - **OpenPegasus Build and Configuration Options for Selected Platforms** - In previous versions of Pegasus this information was released as a Pegasus PEP. Starting with Pegasus 2.9.0 the information is located in the OpenPegasus CVS repository as [pegasus/doc/BuildAndReleaseOptions.html](#).

- **OpenPegasus External Interfaces** - The list of the OpenPegasus interfaces that are considered external and therefore "frozen". Unless an exception is explicitly approved by the Steering Committee all subsequent releases of Pegasus MUST continue to support these interfaces. Interfaces not explicitly listed in this document, should be considered as internal and subject to change. In previous Pegasus releases this information was available as a separate PEP. Starting with Pegasus 2.9.0 this information is integrated into the Pegasus repository as [pegasus/doc/ExternalInterface.html](#).

- **OpenPegasus SDK Packaging Definition** - Defines the recommended set of files for inclusion in the OpenPegasus SDK. Starting with Pegasus release 2.11.0, this document is available in the Pegasus CVS repository as [pegasus/doc/SDKPackaging.html](#). In previous Pegasus releases this document was made available as a separate Pegasus PEP document rather than in the CVS repository.

- **Pegasus Runtime Packaging Definition** - Defines the recommended set of files for inclusion in this OpenPegasus release. Starting with Pegasus release 2.11.0, this document is available in the CVS repository as [pegasus/doc/RuntimePackaging.html](#). In previous releases this was made available as a separate Pegasus PEP document rather than in the CVS repository.

- **Pegasus Release Notes** - PEP 368 - (This document is located in the approved PEP repository and the OpenPegasus source tree root directory ([pegasus/ReleaseNotes.htm](#))

General OpenPegasus Documentation

The following documentation is available for this Pegasus release:

- **Utilities** - A combination of help generally available with the --help option for each command and HTML documentation for most of the tools.

- **API and usage documentation** - See the header files and the HTML API documentation that is on the web site. The best API usage documentation is the existing utilities and test programs and the examples in the API documentation. In particular the Common/tests unit tests contain extensive examples of the use of the Common APIs.

- **Building and Debugging Providers** - Readme.html in the Pegasus source tree Root Directory, API documentation, and documentation from the Pegasus Technical Workshop which is available on the Pegasus web site.
- **Building and Debugging Clients** - API documentation and the documentation on the Pegasus Technical Workshop which is available on the Pegasus web site.

- **PEPs** - The features of Pegasus that have been installed in this and the previous few versions are defined by Pegasus PEPs that are available on the OpenPegasus web site. While these are the original design documents largely and use in the process of defining and approving the overall characteristics of new functionality, they serve as a guide to the design and implementation of these features.

- **OpenPegasus WIKI** - This WIKI is maintained both for the use of the development team and as a user information resource. The wiki can be accessed at [https://wiki.opengroup.org/pegasus-wiki/doku.php?id=start](https://wiki.opengroup.org/pegasus-wiki/doku.php?id=start)

Licensed to The Open Group (TOG) under one or more contributor license agreements. Refer to the OpenPegasusNOTICE.txt file distributed with this work for additional information regarding copyright ownership. Each contributor licenses this file to you under the OpenPegasus Open Source License; you may not use this file except in compliance with the License.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.