Interoperability Conventions 8

Interoperability Conventions

CGE PEX 5.1 utilizes many of the PEX interoperability conventions in its design to ensure interoperability across all servers. The conventions are provided here for those interested in the interoperability rules. As its background information, you may choose to ignore this information:

1. X Consortium vendors wishing to define their own individual extension (for example, enumerated types (ET), enumerated type values (ETV), escapes (ESC), output commands (OCs), table types (TTs), GDPs, or GSEs should obtain a unique vendor id (a number) by the MIT registry.

   Vendor id numbers range from 1 to 126. The numbers 0 and 127 are reserved and are not distributed.

   If all vendor id numbers are distributed, a new vendor wishing a number needs to negotiate with an existing registered vendor and share that number space; otherwise, the new vendor has to wait until a release of PEX that expands vendor space.

2. A vendor-specific ET, ETV, OC, or TT has the high bit set (bit 15) and the vendor id encoded in the high bits (bits 14 through 8) of the type or value.
   - Bit 15 a one
   - Bits 14..8 vendor id
   - Bits 7..0 the type or value (a number in the range of 0..255)

3. Vendor-specific GDP, GSE, and ESC ids have the high bit set (bit 31), and the vendor id encoded in the high bits of the id:
   - Bit 31 a one
   - Bits 30..23 zeros
   - Bits 22..16 vendor id
   - Bits 15..8 zeros
   - Bits 7..0 the id (a number in the range of 0..255)
4. All vendor-specific ETs, ETVs, GDPs, GSEs, and ESCs can be inquired via the PEXGetEnumeratedTypeInfo request (the PEXlib function is PEXGetEnumTypeInfo). Because the PEXGetEnumerateTypeInfo interface is only 16 bits, GDP, GSE, and ESC ids are packed into 16 bits only for purposes of returning the values through PEXGetEnumeratedTypeInfo. Therefore all ETs, ETVs and GDP, GSE and ESC ids, as returned by PEXGetEnumeratedTypeInfo, are encoded:
   - Bit 15 a one
   - Bits 14..8 vendor id
   - Bits 7..0 type, value, or id (a number in the range of 0..255)
   - Specifically, the GDP, GSE, and ESC ids are packed as follows:
   - Bit 31 packs into bit 15
   - Bits 22..16 packs into bits 14..8
   - Bits 7..0 packs into bits 7..0
5. The list of vendor-specific OCs can be inquired via the PEXGetEnumeratedTypeInfo request using the PEXExtETOC enumerated type.
6. The list of vendor-specific table types can be inquired via the PEXGetEnumeratedTypeInfo request using the PEXExtETLUT enumerated type. An alternative way to determine if a vendor-specific table type is supported is to make a PEXGetTableInfo request with the table type (PEXlib function is also named PEXGetTableInfo). If the request is successful, then the table type is supported. If an error (either LookupTable or Value) is generated, then the table type is not supported.
7. Defining a separate X extension is the only way to add events or errors to PEX. XQueryExtension can be used to determine the base offsets for any new errors or events.

8. New requests can be added via the PEXEscape request. See Items 1-4 regarding the encoding of escape identifiers.

9. Not all servers support Big Request Extension (BRE). If BRE is supported by the server, large requests (greater than the maximum request size) can be sent to the server; however, individual OCs will still be limited to 64K words.

10. OCNil is encoded as opcode 0xFFFF. (This is the current PEX-SI OC opcode, and would conform to the encoding specified, if PEX-SI had vendor id 127: 0xFFFF = [1,vendor 127,value 255]). OCNil is valid only for element searching.

11. If facet normals are not given, the server should compute them as described in the *PHIGS PLUS International Standard* (ISO/IEC 9592-4:1992, pp. 37-39, clauses 4.5.5.2 - 4.5.5.4, Facet Normal, Facet Orientation, and Reflection Normal). The server should not modify any geometric or reflectance facet normals that the client explicitly defines.

12. The direction vector associated with the WCS_Vector and WCS_Spot light types points in the direction the light would travel from these sources.

13. Specular exponents are used as specified in Annex C of the *PHIGS PLUS International Standard* (ISO/IEC 9592-4:1992, pp 163-165).

14. Not all servers support all drawables. Use the PEXMatchRendererTargets request to determine supported drawable types (PEXlib function is PEXMatchRenderingTargets). The ordering of supported rendering targets returned by servers should not be interpreted as a recommendation for usage because the list may not necessarily be ordered.

15. Double-buffering with the PHIGS workstation resource, if supported, requires that the resource be created with the double-buffering mode enabled. To determine if double-buffering can be enabled, use the PEXGetImpDepConstants request (PEXlib functions also call PEXGetImpDepConstants).

16. Servers return their native floating point format as the first entry in the list of supported floating point formats returned by PEXGetEnumeratedTypeInfo.

17. Implementations that do not support the pattern interior style do not guarantee support for the pattern lookup table (a LookupTable error is returned from PEXCreateLookupTable and PEXGetTableInfo). Regardless of support for the pattern interior style, servers which support the PHIGS workstation resource also support the pattern LUT.

18. Implementations return a Value error from PEXCreateLookupTable and PEXGetTableInfo for attempted use of vendor specific lookup tables that are not supported.

19. Double buffering via MBX might not be available across all devices. The application should always first inquire whether MBX can create more than one image buffer. If only one MBX image buffer can be created, then the application could also inquire support for rendering to pixmaps which, if supported, could be used to accomplish the same effect as double-buffering by rendering the pixmap and copying to the window.

20. Not all servers support arbitrary color approximation settings. To address this, these conventions should be followed.

    a. A PEX server should possess either or both of the X standard colormap properties: RGB_BEST_MAP and RGB_DEFAULT_MAP. These properties contain entries that describe IDs of Colormaps that are supported for PEX Color Approximation by that server. PEX clients can use these properties to find Colormaps to share, and/or to derive PEX Color Approximation setups that are supported. In all respects, the content and usage of these properties under PEX should conform to the ICCCM.

    b. A PEX server that does not support arbitrary valid Color Approximation configurations should provide the QueryColorApprox escape. PEX clients can use this escape to verify support for a specific Color Approximation configuration, and/or to obtain a list of one or more supported configurations.