**Package Model**

**Part Addressing**

[Comment from CRA: This document was created from Murata-san’s draft as of May 8, 2016. Included are clauses 8.3 and 8.4. Although I chose to retain formatting, clause numbers and some other list formatting was lost. Please let me know if anything is not clear. End comment]

**General**

**This subclause is informative.**

This part of ISO/IEC 29500 defines a way to use IRIs (RFC 3987) to reference part resources inside a package. In particular, the scheme "pack" is introduced in accordance with the guidelines in RFC 4395.

[*Note:* Schemes are represented in an IRI by the prefix before the colon. A well-known example is "http". *end note*]

An example of an IRI in the pack scheme is:

pack://http%3c,,www.openxmlformats.org,my.container/a/b/foo.xml

The substring between the double slash and the first single slash represents an IRI in the http scheme for a package, transformed to allow embedding within an IRI in the pack scheme.

References from outside a package are absolute IRIs of the pack scheme, while those from inside are relative IRIs, which are resolved to absolute IRIs of this scheme.

The following terms are used in accordance with RFC 3986: *scheme*, *authority*, *path*, *segment*, *reserved characters*, *sub-delims*, *unreserved characters*, *pchar*, *pct-encoded characters*, *query*, *fragment*, and *resource*.

**End of informative subclause.**

**Pack Scheme**

This part of ISO/IEC 29500 defines a specific scheme used to refer to parts in a package: the pack scheme. An IRI that uses the pack scheme is called a *pack IRI*.

The syntax of pack IRIs is defined by the ABNF (see RFC 4234) as follows:

pack\_IRI = "pack://" iauthority [ "/" | ipath ]

iauthority = \*( iunreserved | sub-delims | pct-encoded )

ipath = 1\*( "/" isegment )

isegment = 1\*( ipchar )

In this ABNF, sub-delims and pct-encoded are defined in RFC 3986 and iunreserved and ipchar are defined in RFC 3987.

The authority component (iauthority) contains an embedded IRI that points to a package. See 8.3.4 for the procedure for transforming the IRI for the package to permit embedding in the pack IRI as the authority component. The authority component shall not reference a package embedded in another package.

The optional path component (ipath) identifies a particular part within the package. When the path component is missing, the resource identified by the pack IRI is the package as a whole. [M7.2]

A pack IRI might have a query component (as specified in RFC 3986). A query component in a pack IRI is not used when resolving the IRI to a part.

A pack IRI might have a fragment component (as specified in RFC 3986). If present, this fragment applies to whatever resource the pack IRI identifies.

[*Example:*

Example 8–. Using the pack IRI to identify a part

The following IRI identifies the “/a/b/foo.xml” part within the “http://www.openxmlformats.org/my.container” package resource:

pack://http%3c,,www.openxmlformats.org,my.container/a/b/foo.xml

*end example*]

[*Example:*

Example 8–. Equivalent pack IRIs

The following pack IRIs are equivalent:

pack://http%3c,,www.openxmlformats.org,my.container

pack://http%3c,,www.openxmlformats.org,my.container/

*end example*]

[*Example:*

Example 8–. A pack IRI with percent-encoded characters

The following IRI identifies the “/c/d/bar.xml” part within the “http://myalias:pswr@www.my.com/containers.aspx?my.container” package:

pack://http%3c,,myalias%3cpswr%40www.my.com,containers.aspx%3fmy.container  
/c/d/bar.xml

*end example*]

[Note: The pack scheme is a historical scheme in the IANA-maintained registry of schemes located at https://www.iana.org/assignments/uri-schemes/. It was a provisional scheme, but was changed to a historical scheme. See https://www.iana.org/assignments/uri-schemes/historic/pack. The change in status was due to a mistake in the registration proposal. The definition of the authority component in RFC 3986 requires that the colon character (:) in the embedded IRI be escaped (percent-encoded) as %3c. However, in the proposed registration of the pack scheme, an unescaped colon (:) character was mistakenly used. end note]

**Resolving a Pack IRI to a Resource**

The following is an algorithm for resolving a pack IRI to a resource (either a package or a part):

1. Parse the pack IRI into the potential three components: scheme, authority, path, as well as any fragment identifier.

In the authority component, replace all commas (“,”) with forward slashes (“/”).

Un-percent-encode ASCII characters in the resulting authority component.

The resultant authority component shall be a valid IRI for the package as a whole. If it is not, the pack IRI is invalid.

If the path component is empty, the pack IRI resolves to the package as a whole and the resolution process is complete.

A non-empty path component shall be a valid part name. If it is not, the pack IRI is invalid.

The pack IRI resolves to the part with this part name in the package identified by the authority component.

[*Example:*

Example –. Resolving a pack IRI to a resource

Given the pack IRI:

pack://http%3c,,www.my.com,packages.aspx%3fmy.package/a/b/foo.xml

The components:

<authority>= http%3c,,www.my.com,packages.aspx%3fmy.package

<path>= /a/b/foo.xml

are converted to the package IRI:

http://www.my.com/packages.aspx?my.package

and the path:

/a/b/foo.xml

Therefore, this IRI refers to a part named “/a/b/foo.xml” in the package at the following IRI: http://www.my.com/packages.aspx?my.package.

*end example*]

**Composing a Pack IRI**

The following is an algorithm for composing a pack IRI from the IRI of an entire package resource and a part name.

In order to be suitable for creating a pack IRI, the IRI of a package resource shall conform to RFC 3986 requirements for absolute IRIs.

To compose a pack IRI from the absolute package IRI and a part name, the following steps shall be performed, in order:

1. Remove the fragment identifier from the package IRI, if present.

Percent-encode all percent signs (“%”), question marks (“?”), at signs (“@”), colons (“:”) and commas (“,”) in the package IRI.

Replace all forward slashes (“/”) with commas (“,”) in the resulting string.

Append the resulting string to the string “pack://”.

Append a forward slash (“/”) to the resulting string. The constructed string represents a pack IRI with a blank path component.

Using this constructed string as a base IRI and the part name as a relative reference, apply the rules defined in RFC 3986 for resolving relative references against the base IRI.

The result of this operation is the pack IRI that refers to the resource specified by the part name.

[*Example:*

Example –. Composing a pack IRI

Given the package IRI:

http://www.my.com/packages.aspx?my.package

and the part name:

/a/foo.xml

The pack IRI is:

pack://http%3c,,www.my.com,packages.aspx%3fmy.package/a/foo.xml

*end example*]

**Equivalence**

The package implementer shall consider pack IRIs equivalent if:

1. The scheme components are octet-by-octet identical after they are both converted to lowercase; *and*
2. The IRIs, decoded as described in 0 from the authority components, are equivalent (the equivalency rules by scheme, as per RFC 3986); *and*
3. The path components are equivalent part names as defined in [M7.3]

[*Note*: In some scenarios, such as caching or writing parts to a package, it is necessary to determine if two pack IRIs are equivalent without resolving them. *end note*]

[*Drafting note*: Text of last item in published Part 2 is: The path components are equivalent when compared as case-insensitive ASCII strings. Need to modify previous text to deal with non-ASCII normalization. Could this be handled with a reference back to 8.2.2.3? *end note*]

**Base IRIs**

This subclause defines a procedure for determining base IRIs for resolving relative references within parts in packages.

Note: Section 5.1 of RFC 3986 provides four general methods, in order of precedence, for establishing base IRIs for resolving relative references.  The procedure in this subclause provides an OPC-specific method corresponding to the second general method (5.1.2 Base URI from the Encapsulating Entity).

Note: Base IRIs determined by the procedure in this subclause may be overridden by ways 3 or 4 in RFC 3986.

The base IRI to use with a relative reference depends on where that reference occurs within the package. There are three possible cases:

Case 1: Relative reference occurs within a non-relationship part

The base IRI shall be the pack IRI created from the IRI of the package and the name of the part within which the relative reference occurs.

[*Example:*

Consider a part /a/b/foo.xml in a package available at

http://www.mysite.com/my.package

The base IRI is

pack://http%3c,,www.mysite.com,my.package/a/b/foo.xml

*end example*]

Case 2: Relative reference occurs within a relationship part for some part

The base IRI within a relationship part shall be the pack IRI created from the IRI of the package and the source part name.

[*Example:*

Consider a relationship part /a/b/\_rels/foo.xml.rels in a package available at

http://www.mysite.com/my.package

The base IRI is

pack://http%3c,,www.mysite.com,my.package/a/b/foo.xml

*end example*]

Case 3: Relative reference occurs within the relationship part for the entire package (/\_rels/.rels)

The base IRI for a relative reference within /\_rels/.rels shall be the pack IRI created from the IRI of the package.

[*Example:*

Consider the relationships part /\_rels/.rels in a package available at http://www.mysite.com/my.package.

The base IRI is

pack://http%3c,,www.mysite.com,my.package/

*end example*]

**Resolving Relative References**

**This subclause is informative.**

Relative references in parts are resolved as specified in RFC section 5.2, as extended in RFC 3987, section 6.5. This part of ISO/IEC 29500 introduces no changes to the resolution procedure, but Annex A introduces a preprocessing sequence for generating relative references.

This subclause shows examples of resolving relative references to pack IRIs in relation to two pack IRIs. One is a pack IRI "pack://http%3c,example.com,foo.opc/a/foo.xml" for a part /a/foo.xml, while the other is a pack IRI "pack://http%3c,example.com,foo.opc/" for an entire package.

[Comment from CRA: The examples below are very confusing in their presentation. I’m seeing this set of examples as you proving to yourself that the rules for relative references in RFC 3968 can be applied to Pack URIs. If so, is it not possible to say so without explaining all the examples in gory detail? Is there actually some way in which the 3986 rules do not apply? If so, we probably need to be explicit about that. I found myself wondering about a presentation that used a tabular presentation for the examples either instead of all the explanation or as a summary before the explanations. Something like

Base Pack IRI: pack://http%3c,example.com,foo.opc/a/foo.xml

|  |  |  |
| --- | --- | --- |
| Relative reference | Pack representation of resolved IRI |  |
| /b/bar/xml | pack://http%3c,example.com,foo.opc/b/bar.xml | Example 1a |
| bar.xml | pack://http%3c,example.com,foo.opc/a/bar.xml |  |
| ./bar.xml | pack://http%3c,example.com,foo.opc/a/bar.xml |  |
| ../bar.xml | pack://http%3c,example.com,foo.opc/bar.xml |  |

Base Pack IRI: pack://http%3c,example.com,foo.opc/

|  |  |  |
| --- | --- | --- |
| Relative reference | Pack representation of resolved IRI |  |
| /b/bar/xml | pack://http%3c,example.com,foo.opc/b/bar.xml | Example 1b |
| bar.xml | pack://http%3c,example.com,foo.opc/bar.xml |  |
| ./bar.xml | pack://http%3c,example.com,foo.opc/bar.xml |  |
| ../bar.xml | pack://http%3c,example.com,foo.opc/bar.xml |  |

The third column could be used to point to the explanation for each example.

I have not attempted to clarify the examples.

end comment]

Example 1: Leading slash: /b/bar.xml

1) pack://http%3c,example.com,foo.opc/a/foo.xml

Since this relative reference begins with the slash character, the path component (/a/foo.xml) of the base IRI is ignored by the algorithm in 5.2.2 of RFC 3986. The scheme and authority of the resulting IRI is the same as those of the base pack IRI. Thus, the resulting IRI is

pack://http%3c,example.com,foo.opc/b/bar.xml

2) pack://http%3c,example.com,foo.opc/

Likewise, the path component (/) of the base IRI is ignored. The rest is the same.

Example 2: No leading slash: bar.xml

1) pack://http%3c,example.com,foo.opc/a/foo.xml

Since this relative reference does not begin with the slash character, the path component （/a/foo.xml) of the base IRI and that (bar.xml) of the relative reference are merged. The "merge" routine in 5.2.3 of RFC 3986 first removes "foo.xml" from the path component of the base IRI, and emits "/a/bar.xml". Thus, the resulting IRI is a pack IRI "pack://http%3c,example.com,foo.opc/a/bar.xml".

2) pack://http%3c,example.com,foo.opc/

Since the relative reference does not begin with the slash character, the path component （/) of the base IRI and that (bar.xml) of the relative reference are merged. The "merge" routine emits "/bar.xml".Thus, the resulting IRI is a pack IRI "pack://http%3c,example.com,foo.opc/bar.xml".

Example 3: Dot segment: ./bar.xml

1) pack://http%3c,example.com,foo.opc/a/foo.xml

As in the previous case, the "merge" routine in 5.2.3 of RFC 3986 removes "foo.xml" from the path component of the base IRI, and emits "/a/./bar.xml". But the "remove\_dot\_segments" routine further removes "./" and emits "/a/bar.xml". Thus, the resulting IRI is a pack IRI

pack://http%3c,example.com,foo.opc/a/bar.xml

2) pack://http%3c,example.com,foo.opc/

The "merge" routine emits "/./bar.xml" but the "remove\_dot\_segments" routine removes "./" and emits "/bar.xml". Thus, the resulting IRI is

pack://http%3c,example.com,foo.opc/bar.xml

Example 4: Dot segment: ../bar.xml

1) pack://http%3c,example.com,foo.opc/a/foo.xml

This case is similar to the previous case, but the "remove\_dot\_segments" routine removes "a/..". Thus, the resulting IRI is a pack IRI "pack://http%3c,example.com,foo.opc/bar.xml".

2) pack://http%3c,example.com,foo.opc/

The "merge" routine emits "/../bar.xml", but the "remove\_dot\_segments" routine replaces ""/../" by "/". Thus, the resulting IRI is a pack IRI pack://http%3c,example.com,foo.opc/bar.xml".

**End of informative subclause.**