2.2. Identifiers

2.2.1. Unique Identifier

Each entity known to the responder MUST be associated with the unique identifier of the entity, that is, the entityID attribute of the corresponding <md:EntityDescriptor> element in SAML metadata.

If a client issues a metadata query using the unique identifier of an entity (subject to the requirements of the base specification), but the responder is unable to map the identifier in the request to the entityID of a known entity, the responder MUST return an HTTP status code of 404 ("not found"). See the base specification for detailed information regarding this and other HTTP response codes.

2.2.2. Transformed Identifier

Each entity known to the responder MUST also be associated with at least one other identifier. In particular, a responder compliant with this profile MUST be able to associate each entity with an identifier matching the SHA-1 hash of the entityID. For the purposes of this profile, a responder MUST be able to treat such an identifier as equivalent to the corresponding untransformed identifier (i.e., the entityID).

An identifier is said to exhibit the {sha1} syntax if it matches the "sha1id" production of the following ABNF grammar:

```
SHA1     = %x73 %x68 %x61 %x31 ; lowercase "sha1"
DIGIT    = %x30-39
HEXDIGIT = DIGIT | %x61-66 ; lowercase a-f
sha1hex  = 40*HEXDIGIT
sha1id   = "{" SHA1 "}" sha1hex
```

In the above grammar, the "sha1hex" component encodes the 20-octet (160-bit) binary SHA-1 hash value as a sequence of 40 lowercase hexadecimal digits.

For example, the identifier

```
http://example.org/service
```

transformed by means of SHA-1 hashing becomes

```
{sha1}11d72e8cf351eb6c75c721e838f469677ab41bdb
```

From a client's point of view, the transformed identifier may be used interchangeably with the original untransformed identifier since a conforming responder MUST return identical metadata in both cases.

2.2.2.1. Use Case: SAML Artifacts

All versions of SAML include a profile based on the notion of an "artifact," a small piece of data that serves as a reference to a SAML assertion. This document profiles the use of either the SAML V1.1 Type 0x0001 Artifact [ref] or the SAML V2.0 Type 0x0004 Artifact. [ref] All other artifact formats are out of scope with respect to this profile.

Both of the artifact formats referenced in the preceding paragraph are defined to have a 20-byte SourceID component, and moreover, both artifact profiles recommend that the value of the SourceID component be the SHA-1 hash of the issuer's entityID. In this way, the receiver is able to determine who issued the artifact by mapping the SourceID to a particular entityID.

Now suppose a client receives an artifact of the appropriate type. Assuming the value of the SourceID component of the artifact is the SHA-1 hash of the issuer's entityID, the client performs the following operations:

1. Base64-decode the artifact
2. Extract the 20-byte SourceID value
3. Hex-encode the SourceID value

With the hex-encoded SourceID value in hand, the client may issue a metadata query with an identifier using the {sha1} syntax defined above.

In general, the client will not know if the value of the SourceID is a SHA-1 hash, let alone the SHA-1 hash of the issuer's entityID. If any case, if the client uses the {sha1} syntax, the responder MUST process the request subject to the following rules.

2.2.2.2. Processing Rules

Responder implementations MAY detect malformed identifiers with the {sha1} syntax. For example, the string of characters following the "}" may contain characters other than lowercase hexadecimal digits or may not be exactly 40 characters in length. If the identifier is malformed, the responder MAY return an HTTP status code of 400 ("bad request"). If not, implementations MUST process malformed identifiers as normal identifiers as follows.

If the responder is able to map an identifier with the {sha1} syntax to the entityID of a known entity, the responder returns the corresponding SAML metadata document (subject to the requirements of the base specification). In this case, the responder MUST return an HTTP status code of 200 ("ok") and the HTTP response body MUST be identical to the response body

that would have been returned if the client had used the unique identifier of the entity (i.e., the entityID).

On the other hand, if the responder is unable to map an identifier with the {sha1} syntax to the entityID of a known entity, the responder MUST return an HTTP status code of 404 ("not found"). See the base specification for detailed information regarding this and other HTTP response codes.

2.2.3. Additional Identifiers

A responder MAY associate arbitrary groups of one or more entities with other identifiers as desired, including identifiers with the {sha1} syntax. The use of such identifiers is out of scope with respect to this profile.
