

URI Scheme: “ves:”

URI scheme “ves:” is defined in compliance with [RFC 3986](#). The purpose of “ves:” scheme is to provide unique global identifiers for objects within [VESvault](#) repository.

This document defines “ves:” scheme for two separate types of VESvault objects:

- Vaults
- Vault Items

URIs for Vaults

VES URI for Vault may take one of the following forms:

An App Vault, identified by an App Domain and an External ID. An optional *userRef* is needed to automatically create a temporary Vault Key for an App Vault that didn’t exist yet, or is used to assert that an existing App Vault belongs to the specified user:

```
“ves:” || [ “//” || { domain } || “/” || ] { externalID } || “/”  
[ || { userRef } ]
```

A Primary Vault, identified by *userRef*:

```
“ves:” || [ “///” || ] “/” { userRef }
```

A Vault Key identified by an internal VESvault ID. This syntax allows to access any types of Vault Keys, including specific temporary keys, lost keys, and recently deleted keys:

```
“ves:///” || { internalID } || “/” [ || { userRef } ]
```

In all statements above, “||” denotes concatenation, parts in [brackets] are optional.

userRef is a user’s email address, with or without personal name, formatted according to [RFC 822](#).

It is recommended to apply “%” encoding on any special characters inside *domain* and *externalID*, mandatory if any of them includes “/” | “%” | “?” | “#” characters.

userRef will contain special characters that are required to be “%” encoded according to RFC 3986, although the parser SHOULD be able to unambiguously handle unescaped characters, except for “?” and “#” since *userRef* is always the last part of the path.

Unescaped “?” and “#” should be treated as URI part separators, for compatibility with any possible future revisions.

Note: the *externalID* or *internalID* part is always followed by “/”, even if *userRef* is not supplied. It’s important for distinguishing Vault URIs from Vault Item URIs.

URIs for Vault Items

A Vault Item can be referenced by an App Domain and an External ID:

```
"ves:" || [ "//" || { domain } || "/" || ] { externalID }
```

Or, by its internal VESvault ID. This syntax allows to access internal and recently deleted Vault Items:

```
"ves:////" || { internalID }
```

Note: the *externalID* or *internalID* is always the last part and is not followed by "/", for distinguishing from Vault URIs.

Examples of Valid VES URIs:

```
ves://demo/myvault/  
(an existing App Vault)
```

```
ves:myvault/  
(an existing App Vault within the default domain for the current context)
```

```
ves://demo/myfriend/My Friend <myfriend@mydomain.com>  
ves://demo/myfriend2/myfriend2@mydomain.com  
(these App Vaults will be automatically created if didn't exist)
```

```
ves:/My Friend3 <myfriend3@mydomain.com>  
ves:////myfriend4@mydomain.com  
(Primary Vaults)
```

```
ves:///1234/  
(an App Vault Key by internal ID)
```

```
ves://demo/myitem  
(a Vault Item)
```

```
ves:myItem  
(a Vault Item within the default domain for the current context)
```

```
ves:///5678  
(a Vault Item by internal ID)
```