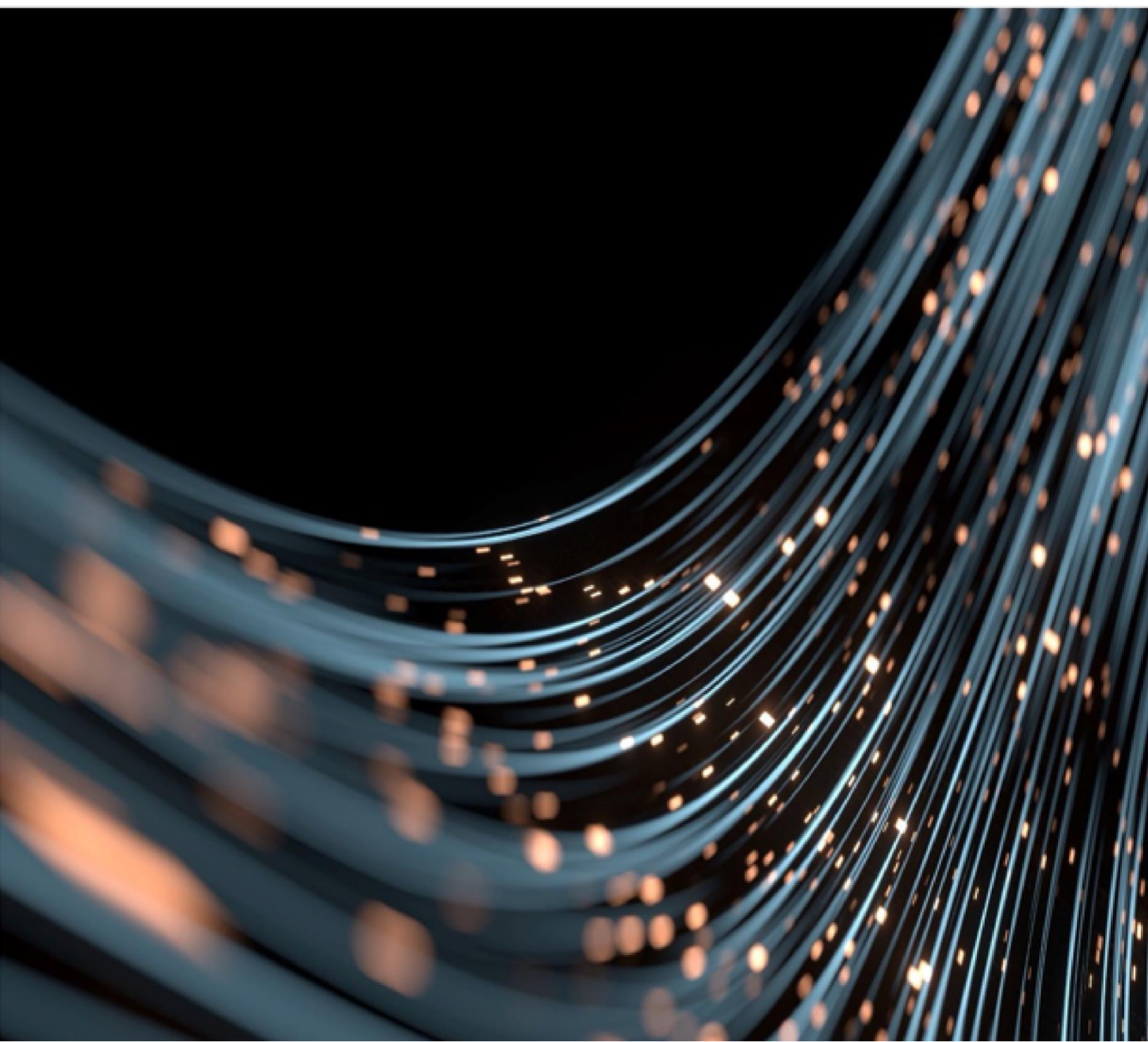


SML Profile

Version 1.1

November 2023



1 Table of Contents

1	Table of Contents.....	2
2	Version History.....	3
3	Introduction.....	3
3.1	Scope.....	3
3.2	Conformance.....	3
3.3	Terms and Definitions.....	3
3.4	Disclaimers and Copyright.....	3
4	Use of DNS	4
4.1	SML DNS domain name.....	4
4.1.1	Production SML	4
4.1.2	Test SML	4
4.2	Constructing the DNS name	4
4.3	The NAPTR record.....	5
4.4	Time to live (TTL).....	6
	Appendix A: Python DNS name construction example (non-normative).....	7
	Appendix B: Example shell script to return SMP service from SML query (non-normative).....	8

2 Version History

Revision date	Version	Change description	Editor
07/20/2023	1.0	Initial version	BPC Market Pilot Technical Committee
11/13/2023	1.1	Domain names changed	DBNA Technical Committee

3 Introduction

3.1 Scope

This specification is a profile of the Business Document Metadata Service Location (BDXL) Version 1.0 OASIS Standard (OASIS BDXL 1.0) published here:

<http://docs.oasis-open.org/bdxr/BDX-LocaBon/v1.0/BDX-LocaBon-v1.0.html>

This specification describes how Access Points must query the DBNAlliance network Service Metadata Locator (SML) to discover SMP services of Participants. The interoperability and other technical interfaces between network registrars, and between SML Service Providers, and the DNS server storage of the records is specified elsewhere and not in the scope of this specification.

3.2 Conformance

The keywords 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY', and 'OPTIONAL' in this specification are to be interpreted as described in RFC2119 and RFC 8174 when, and only when, they appear in all capitals, as shown here.

3.3 Terms and Definitions

For the purpose of this specification, all terms shall have the definitions defined in the document Terms and Definitions of the DBNAlliance version 1.0.

3.4 Disclaimers and Copyright

Views expressed here are not necessarily those of, and should not be attributed to, any particular DBNAlliance participant or organization. They are not intended to provide business or legal advice, nor are they intended to promote or advocate a specific action, payment strategy, or product. Readers should consult with their own business and legal advisors.

This specification is the work product of the DBNAlliance, and readers are free to republish this specification in whole or in part without further permission, as long as the work is attributed to the DBNAlliance, and in no way suggests the DBNAlliance sponsors, endorses or recommends any organization or its services or products. Other product names and company names referenced within this document may be either trademarks or service marks of their respective owners.

4 Use of DNS

4.1 SML DNS domain name

4.1.1 Production SML

The DNS domain names used for the DBNAlliance SML are:

Production: `sml.dbnalliance.net`

Test: `sml.dbnalliance.com`

Pilot: `sml.digitalbusinessnetworks.info`

All Participants in the network SHALL be discoverable through querying this domain. Except for exchanging test transactions as described in 4.1.2, Access Point services MUST use the Production SML domain for all Participant Discovery in the network.

The Production SML domain MUST NOT be used for test purposes, and all SMP services registered in the Production SML MUST be production services.

4.1.2 Test SML

The DBNAlliance maintains an SML environment for testing purposes, such as for testing interoperability between parties, for onboarding new users and systems, etc. The DNS domain name used for the DBNAlliance *Test* SML is:

`sml.dbnalliance.com`

The Test SML MUST only be used for testing purposes and MUST NOT be used for sending real/production transactions. SMP services registered in the Test SML SHOULD NOT be production services and users of the Test SML MUST maintain segregation between production and test environments.

4.2 Constructing the DNS name

To retrieve the SML record containing information about a Participant's SMP service, an SML client application must first know the complete party identifier and party identifier scheme of the Participant being queried. The format of the party identifier and party identifier scheme MUST follow the DBNAlliance Policy for Using Identifiers.

The DNS name MUST be constructed as follows:

1. The complete party identifier scheme and the complete party identifier MUST be concatenated and delimited by two colons ("::") as follows:

```
{lowercase identifier scheme}::{lowercase party identifier}
```

2. The SHA256 hash of the *lowercased* concatenated string is computed,
3. The SHA256 digest is Base32 encoded, removing any trailing “=” characters,
4. The final Base32 encoded string is used as the hostname component together with the SML DNS domain name.

For example, a company identified by the GLN number 1234567890123 would use the scheme identifier GLN. The scheme identifier is then concatenated with the party identifier to produce the string:

```
GLN::1234567890123
```

The Base32 encoding of the SHA256 hash of the above (lowercased) string is:

```
qcie7f2ny3ze5nmhqse7z5j6jerds3gc437bfjl2k6vq6minb47a====
```

This string, stripped from its trailing “=” characters and used with the DNS domain name becomes the complete DNS name:

```
qcie7f2ny3ze5nmhqse7z5j6jerds3gc437bfjl2k6vq6minb47a.sml.dbnalliance.com
```

See also Appendix A: Python DNS name construction example (non-normative).

4.3 The NAPTR record

As described in OASIS BDXL 1.0, the SML stores the SMP service endpoint information in DNS NAPTR records. To discover the SMP service endpoint of a given Participant, one must query the SML by first constructing a DNS name as described above in section 4.2 and then retrieve its NAPTR records from the DNS.

A NAPTR record consists of the following fields:

```
order    preference    flags    service    reg.exp.    replacement
```

Querying a Participant in the SML will return a structure similar to the following example:

```
IN NAPTR 100 10 "U" "oasis-bdxr-smp-2#dbnalliance-1.1"
"!^.*$!https://smp.example.com/myservice/!" .
```

The above example shows a NAPTR record with these fields and values:

```
Order          100
Preference     10
Flags          U
Service        oasis-bdxr-smp-2#dbnalliance-1.1
Reg.exp.       !^.*$!https://smp.example.com/myservice/!
Replacement    none (the NAPTR record contains an empty string)
```

The trailing “.” denotes the end of the NAPTR record.

The fields of the NAPTR record SHALL have the following definition, meaning and use in the DBNAlliance network:

Field	Definition and use
Order	No Participant can be registered in the SML with the same service value in more than one NAPTR record. The Order and Preference fields therefore have no practical use. An SML client MAY ignore these fields.
Preference	
Flags	<p>The SML NAPTR records MUST always be of type “U”, meaning that the value of the flag field MUST be set to either “U” (uppercase) or “u” (lowercase).</p> <p>The value of the flag field MUST be treated as case insensitive.</p>
Service	<p>This field describes which type of service the record points to. The value of the service field MUST be an identifier defined in a network service profile or specification. Currently only the OASIS SMP 2.0 service is defined.</p> <p>An entity MUST NOT be registered with more than one record having the same value in the service field.</p> <p>The value of the service field MUST be treated as case insensitive.</p>
Reg.exp.	<p>The value of the reg.exp. field is a “!”-delimited string containing a substitution expression consisting of a regular expression and URL:</p> <p style="text-align: center;"><code>!{regular expression}!{URL}!</code></p> <p>The URL MUST be an absolute URL to the SMP service of the Participant being queried. The regular expression MUST be exactly <code>^.*\$</code></p> <p>SMP clients SHOULD ignore the regular expression and MAY assume that the URL contains the entire fully qualified URL to the SMP service.</p> <p>The value of the reg.exp. field MUST be treated as case sensitive.</p>
Replacement	<p>The replacement field of the NAPTR record MUST be empty.</p> <p>SMP clients SHOULD ignore this field.</p>

See also Appendix B: Example shell script to return SMP service from SML query (non-normative).

4.4 Time to live (TTL)

It is RECOMMENDED that SML services set the Time to live (TTL) of DNS records to no less than 3,600 seconds (one hour) and to no more than 86,400 seconds (one day).

Access Points MUST NOT cache the DNS record for any longer than the specified TTL.

Appendix A: Python DNS name construction (non-normative)

The script constructs a proper DNS name from party identifier scheme and the complete party identifier.

The latest version of the script can be found at <https://github.com/dbnalliance/published-documents/blob/main/scripts/domain-name-constructor.py>

Usage instructions

```
> python3 domain-name-constructor.py GLN 1234567890123  
qcie7f2ny3ze5nmhqse7z5j6jerds3gc437bfjl2k6vq6minb47a.sml.dbnalliance.com
```


Appendix B: Example shell script to return SMP service from SML query (non-normative)

The latest version of the example shell script can be found at

<https://github.com/dbnalliance/published-documents/blob/main/scripts/get-smp-from-sml.sh>

Usage instructions

```
> get-smp-from-sml.sh  
qcie7f2ny3ze5nmhqse7z5j6jerds3gc437bfj12k6vq6minb47a.sml.dbnalliance.com  
  
https://smp.example.com/myservice/
```