

MDQ Client Software

This table captures current and future state of client software capable of requesting and consuming per-entity metadata via the [Metadata Query Protocol](#).

Client Software	Supports MDQ protocol?	Notes on current capability	Security Model(s)	Known future capabilities or enhancements?
Shibboleth SP (current: V2.6.0)	Yes	See the Dynamic MetadataProvider topic in the Shibboleth wiki. This feature (first introduced in SP V2.0) is probably the most mature client implementation available.	XML Signature, TLS validation against explicit anchors	New "file://" feature in SP V2.6.0
Shibboleth IdP (current: V3.2.1)	Yes	See the DynamicHTTPMetadataProvider topic in the Shibboleth wiki. This feature (new in IdP V3.0.0) is mostly untested (which means there are probably bugs).	XML Signature, TLS validation against explicit anchors	New "file://" feature in IdP V3.3.0 ?
SimpleSAMLphp (current: V1.14.8)	Yes	MDQ metadata handler merged on March 16, 2015. There is no formal documentation (search for "MDQ" in config.php). This feature is mostly untested.	XML Signature (via cert fingerprint)	
ADFS 2.0 (Server 2008 and Server 2008 R2) *	No	ADFS will fetch and cache a single SAML EntityDescriptor at a configured endpoint location beginning with "https://"	TLS	
ADFS 3.0 (Server 2012 R2) *	No	ADFS will fetch and cache a single SAML EntityDescriptor at a configured endpoint location beginning with "https://"	TLS	
ADFS 4.0 (Server 2016 Tech Preview) *	No	ADFS will fetch and cache a single SAML EntityDescriptor at a configured endpoint location beginning with "https://"	TLS	This version may load an aggregate
Ping	No	Ticket filed for next release to enable the needed 'Accepts' header value.	TLS	

* Information is not verified by the vendor/developer and may not be accurate

MDQ Tutorial

A simple shell script ([mdq_url.sh](#)) illustrates the client side of the Metadata Query Protocol:

Using mdq_url.sh

```
# the base URL of the mdq-beta.incommon.org server
$ export MDQ_BASE_URL=http://mdq-beta.incommon.org/global

# an entityID for testing
$ entityID=https://idp.ncsa.illinois.edu/idp/shibboleth

# compute the MDQ protocol URL
$ mdq_url.sh -v $entityID
Using base URL http://mdq-beta.incommon.org/global
http://mdq-beta.incommon.org/global/entities/https%3A%2F%2Fidp.ncsa.illinois.edu%2Fidp%2Fshibboleth

# retrieve a signed entity descriptor
$ curl --silent $( mdq_url.sh $entityID )
...
```

A fully functioning client (like Shibboleth or simpleSAMLphp) will of course verify the signature on the entity descriptor. Other security models are possible (as noted in the table).