

XnatDataClient

XnatDataClient, or **XDC**, is a command-line tool for making HTTP calls to an XNAT server. In many ways, it's similar to [curl](#) and can even be used to make requests of non-XNAT HTTP services. However, XDC adds a layer of XNAT "awareness" to its HTTP capabilities. For example, you may want to retrieve all of the scan files that compose a complete DICOM session. This may be only a few files, but is generally quite a few more. With curl, you would need to make a call to the server to get all of the files in the session, then make subsequent calls for each of the files. Using XDC, you can make a single call to retrieve the session data and then have XDC follow the URLs for all of the constituent scan files and automatically download them. The number of HTTP calls is still the same, but XDC automates the process for you.

Running XDC

If you have XNAT, you already have XDC. The script is located in the **bin** folder of your XNAT builder installation. There is also a [stand-alone installation of XDC available on the XNAT Marketplace](#). This is useful for applications that don't have XNAT installed on the local server but need to access data from an XNAT installation.

XDC is provided as a Java archive or jar file. The jar file is packaged as a Java application, with the required classpath included in the archive manifest. The practical result of this is that you can run the application just by specifying the XDC jar file (named **data-client-*version*.jar**) with the JVM call:

Running XDC

```
java -jar lib/data-client-1.6.5.jar -h
```

The pipeline engine also provides a script wrapper named **XnatDataClient** for this function. This script is located in the **xnat-tools** folder of the configured pipeline engine installation.

Options

XDC has the following command options (you can also see these options by running XDC with the **-h** option).

Short	Long	Description
ts	trustStore	Indicates the location of the trust store.
ls	listUris	Directs the client to render results as a list of files to the application output. This can be used for such applications as piping output from a call to the server to other commands or processing servers.
C	showStatusLine	Tells XDC to show the resulting HTTP status line on exit.
bs	bufferSize	Sets the buffer size option. Defaults to 256.
H	header	Indicates the HTTP method to be used for the operation. This can be one of GET, PUT, POST, or DELETE.
auu	allowUnsmoothUris	Indicates that XDC should allow "unsmooth" URLs. "Unsmooth" URLs may have contiguous path separators (i.e. forward slashes) and other artifacts from string arithmetic and operations. By default, XDC smooths URLs before calling them (i.e. allow unsmooth URLs is false).
osa	outputStreamAdapter	Specifies an output stream adapter implementation to handle redirecting the output from your application.
db	data-binary	Indicates data to be POSTed with no extra processing. If the data starts with an @, the rest of the data should be a file name from which to read data; carriage returns and newlines are preserved.
d	data	Indicates data to be POSTed. If the data starts with an @, the rest of the data should be a file name from which to read data; carriage returns and newlines will be stripped out. This is identical to -da (--data-ascii).
da	data-ascii	See -d, --data.

b	batch	Indicates that this is a batch operation. Batch operations are currently only supported for download transfers, i.e. GET calls, that retrieve JSON that contains URI or absolutePath references.
c	showStatusCode	Tells XDC to show the resulting HTTP status code on exit.
tsPass	trustStorePassword	Provides the password for accessing the trust store.
a	useAbsolutePath	Indicates that XDC should try to use the absolute path of specified resources for copy operations rather than REST calls.
o	outputName	Indicates the requested output name.
l	local	Indicates the local file to be uploaded.
m	method	Indicates the HTTP method to be used for the operation. This can be one of GET, PUT, POST, or DELETE.
k	useSymlinks	Indicates that files should be linked via symlinks rather than copied during batch operations. Note that this function is dependent on your platform supporting symlinks and the ln command being on your path.
h	help	Displays this help text.
xx	overwrite	Indicates whether the specified output file should be overwritten without prompting if it exists.
v	version	Displays the version of this application.
u	username	The user name for authentication.
s	sessionId	Indicates the session ID to be used for transactions. This replaces username/password authentication options.
r	remote	Indicates the remote resource location. This should be a properly formatted URL, although it can indicate file as well as http resources.
p	password	The password for authentication.
pp	pathPrefix	Indicates a substitution for the path prefix. This lets you replace the first part of the returned absolute path with another path. The path tokens should be separated by the pound sign (#). For example, if you know that the absolute path returned by the server will be /data/project/archive/..., but your system has the same data archive mounted at /mnt/data/archive, you would specify the value for this option as /data/project#/mnt/data. Specifying this option implies the absolutePath option set to true.
x	proxy	Indicates the server address for the proxy (if required).

Usage

Here are some handy use cases for XDC.

Batch Copy

The ability to do a batch copy by symlink using the **-k** or **--useSymlinks** parameters. This is obviously very dependent on platform and hasn't been tested extensively. It actually uses the **Runtime.exec()** method to run ln on the OS (Java 7 has a **Files.createSymbolicLink()** function, but XDC is currently back to Java 6 so XDC can't use that), so I'm not sure what performance will be on this. It should definitely be better than copying a whole bunch of files though.

Listing output with the **-ls** or **--listUriis** parameters. Instead of doing anything with the files found in a REST result, XDC just lists them. If you specify the **absolutePath** option, this will be listed by path, otherwise by server-relative URI. This is handy for piping the query results to other processing tools.