

Utilities

Detailed information on XINA utility applications.

- [XINA Connect](#)
- [XINA Tunnel](#)
- [XINA Import](#)
- [XINA Download](#)
- [XINA API Libraries](#)
- [XINA Struct Archive](#)
- [XINA Struct Mine](#)

XINA Connect

XINA Connect is a Java application which combines the features of the XINA Tunnel and XINA Import applications with a UI. This application is distributed as a Java jar file and requires Java 17 or greater to run. The recommended OpenJDK build is available [here](#).

Latest Version

1.1.0

[Windows Installer](#)

[Windows Portable](#)

[MacOS Installer](#)

[Jar File](#) (requires Java 17+)

```
java -jar xina_connect.jar
```

Previous Versions

1.0.0

[Windows Installer](#)

[MacOS Installer](#)

XINA Tunnel

XINA Tunnel is a command line application required for client applications to communicate with the XINA Server via the [XINA API](#). This application is distributed as a Java jar file and requires Java 17 or greater to run. The recommended OpenJDK build is available [here](#).

Latest Version

XINA Tunnel 9.2.0

[Download](#)

Usage

Windows

```
java -jar "path to xina_tunnel.jar" ^  
-host "host URL or IP address" ^  
-keyfile "path to key.json"
```

MacOS, Linux

```
java -jar "path to xina_tunnel.jar" \  
-host "host URL or IP address" \  
-keyfile "path to key.json"
```

Old Versions

XINA Import

XINA Import is a utility for importing XINA API actions as JSON files. This application is distributed as a Java jar file and requires Java 17 or greater to run. The recommended OpenJDK build is available [here](#).

Latest Version

XINA Import 9.2.0

[Download](#)

Usage

Windows

```
java -Dlog4j.configurationFile="path to log4j2.xml" ^  
-jar "path to xina_import.jar" ^  
[additional arguments...]
```

MacOS, Linux

```
java -Dlog4j.configurationFile="path to log4j2.xml" \  
-jar "path to xina_import.jar" \  
[additional arguments...]
```

Argument	Info	Default
<code>-host <hostname></code>	the XINA Tunnel host	<code>"localhost"</code>
<code>-post <port></code>	the XINA Tunnel port	<code>41746</code>
<code>-movejson <path></code>	directory to move JSON files to after import	none (files are not moved)
<code>-movefile <path></code>	directory to move other files to after import	none (files are not moved)
<code>-deljson</code>	if set, permanently delete JSON files after import	<code>false</code>
<code>-delfile</code>	if set, permanently delete other files after import	<code>false</code>
<code>-dir <path></code>	path to directory containing files to import	
<code>-watch <path></code>	path to directory to watch for files to import	
<code>-recursive</code>	if true in <code>dir</code> or <code>watch</code> mode, searches directory recursively for JSON files	<code>false</code>

XINA Import has three modes of operation:

File List

JSON files can be listed explicitly. They will be imported in the specified order.

```
java -jar "path to xina_import.jar" \  
    "path to JSON file" \  
    "path to JSON file"...
```

Directory

If the `-dir` argument is used, XINA Import will attempt to import all `*.json` files in the specified directory in alphabetical order. It is recommended to include `-movejson` or `-deljson` to track progress, in case the import is interrupted.

```
java -jar "path to xina_import.jar" \  
    -dir "path to directory" \  
    -movejson "path to different directory"
```

Watch

If the `-watch` argument is used, XINA Import will attempt to import all `*.json` files in the specified directory in alphabetical order. Once complete, the directory is watched for any new JSON files, which are imported as they become available. It is required to include `-movejson` or `-deljson`, as otherwise files would be continuously re-imported.

```
java -jar "path to xina_import.jar" \  
    -watch "path to directory" \  
    -movejson "path to different directory"
```

XINA Download

XINA Download is a utility for managing batch file downloads with XINA. This application currently works with the SAM and MOMA XINA instances for telemetry file management only.

Latest Version

Windows Installer

[xina_download.exe](#)

MacOS Installer

[xina_download.dmg](#)

XINA API Libraries

We aim to provide a variety of reference XINA API client implementations for different programming languages and environments.

Python

[xina_api_python_1.0.0.zip](#)

Installation

Prerequisites

- Python 3.8 or greater

The `xina` client communicates with XINA via the [XINA Tunnel](#) utility. You must have it running for the `xina` client to connect.

Steps

1. Extract the zip file.
2. Open a terminal in the root extracted directory, and execute the following commands:

```
python3 install .
```

Examples

Client class

```
import socket

from xina import XPClient

with socket.socket() as sock:
    sock.settimeout(5.) # Five seconds is the recommended socket timeout.
    sock.connect(('localhost', 41746))

    with XPClient(sock=sock) as x:
        x.init()
```

```
res = x.act({'action': 'version'})
```

```
print(res)
```

```
# => {'schema':100, 'host':'sandbox.xina.io', 'server':'9.1.4', 'team':100}
```

CLI

```
python -m xina action '{"action":"version"}'
```

```
# => {"schema":100,"host":"sandbox.xina.io","server":"9.1.4","team":100}
```


XINA Struct Archive

The XINA Struct Archive utility processes mnemonic buffer files into mnemonic archive files. It is primarily intended to be run as part of the automated XINA Structs data pipeline.

Arguments

Name	Req	Description	Default
task	?	task ID	
conf	?	JSON conf file path	
temp	?	temp directory path	
import	?	import directory path (for file outputs)	
post	?	post directory path (for post-import outputs)	
host		tunnel host	"localhost"
port		tunnel port	41746

Configuration

Name	Req	Description	Default
origin	?	origin group	
slice		slice length in minutes	60
t		slice time barrier (see below)	

Operation

This utility serves three main functions. First, processing raw buffer files of any supported format into well formatted, optimized xbin files. Second, merging buffer file data into time slice separated archive files. Third, processing any changes of flagged buffer files and updating associated archives as needed.

File Processing

Buffer files are typically imported in the `PENDING` state, and must be converted into xbin files, which are indicated with the `PROCESSED` state.

XINA Struct Mine