

Batch Loader Command Line Tool

The command line version of the Batch Loader tool greatest advantage is that it allows the use of a known configuration in a regularly scheduled process, such as loading User info into a database. It could even be used to provide a set of sample data to be applied to any database with the same configuration. It is as flexible as the UI but can save on setup time by reusing the previous configuration.

Command Line Options

There are several elements to the command line tool. First, is the executable file for the batch load tool itself, BatchLoaderCMD.exe. The details of the various options for running the executable at the command line are as follows:

```
Usage: BatchLoaderCmd.exe -d {data file} [optional arguments]
where:
-d {data file}          - path to a delimiter-separated data file
```

Optional arguments can either be specified in the command line or in the configuration file; if not defined in either place a default value is assumed (if a default value is mentioned below):

```
-c {config file}      - path to configuration file; default 'config.xml'
-t {template file}    - path to template file; default 'template.xml'
-l {log file}         - path to log file
-ll {log level}       - verbosity of logging (1 - 3); default 1
-e {encoding}         - name of data file encoding; e.g. 'us-ascii'
-s {url}              - Innovator server URL
-db {db name}         - Database name
-u {user name}        - Innovator user login name
-p {user password}    - Innovator user password
-th {threads}         - Number of threads in a worker process; default 1
-pr {processes}       - Number of worker processes; default 1
-lpp {lines}          - Lines processed by each worker process; default 1000
-fr {row number}      - First row to process in the data file; default 1
-lr {row number}      - Last row to process in the data file; default -1
-crtloc {certificate location} - Certificates store location; default CurrentUser
-crtstore {certificate store}   - Certificates store name; default My
-crtftype {certificate find type} - Find type for certificate search; default FindBySubjectName
-crtfvalue {certificate find value} - Find value for certificate search
-crtvalid {certificate is valid} - Certificate should be valid; default is true
```

Important



Most of these parameters are optional and are generally included in the xml configuration file. However, the command line arguments override any values in the configuration file.

Connection Mapping in the Configuration File

The configuration file that is referenced by the batch load application is a straightforward xml file that defines the parameters for the batch load activity. The configuration file is used to configure the Aras Innovator server and data source, as well as the process parameters. The following describes the different parameter tags that are included in the XML configuration file.

```
<BatchLoaderConfig>
<server>http://localhost/InnovatorServer</server>
<db>InnovatorSolutions</db>
<user>admin</user>
<password>innovator</password>
<max_processes>1</max_processes>
<threads>1</threads>
<lines_per_process>250</lines_per_process>
<delimiter>\t</delimiter>
<encoding>utf-8</encoding>
<first_row>2</first_row>
<last_row>-1</last_row>
<log_file>C:\User</log_file>
<log_level>3</log_level>
<certStoreLocation>CurrentUser</certStoreLocation>
<certStoreName>My</certStoreName>
<certFindType>FindBySubjectName</certFindType>
<certFindValue>SomeSubjectName</certFindValue>
<certValidOnly>true</certValidOnly>
</BatchLoaderConfig>
```

The only parameter defined in the configuration file that cannot be defined in the command line is the delimiter. Any other parameter tag may be left out of the configuration file.

- **server** – The connection URL for Aras Innovator. If all the defaults were taken during the Aras Innovator installation, the path should be something such as: http://localhost/InnovatorServer. If unsure, check in IIS to get the exact path. This URL should not include reference to the /Client folder
- **db** – The database to which the data is loaded. Selecting this field after defining the connection URL makes this a pick list of available databases.
- **user** – The user login to be used for connecting to and loading data into the database.
- **password** – The user password in plain text.



- **max_processes** – The number of worker processes to be used by the Batch Loader while loading data. Recommend using the default of 1.
- **threads** – The number of threads per worker process. Recommend using the default of 1.
- **lines_per_process** – The number of lines in the data file that are loaded by a single worker process. If the worker process reaches the end of its line and the data file has not completed processing, then a new worker process is started.
- **delimiter** – the delimiter used to separate data, usually a tab (\t), or a space (), or a comma (,).
- **encoding** – Encoding (or codepage number) of data file
- **first_row** – The number of the row where the actual data starts. Sometimes the first row is used for row headings. In that case, the data starts in the second row. (See FirstRowIsColumnNames property below.)
- **last_row** – The number of the row where the actual data stops. Default of -1 indicates that the file should be read until the end of the file
- **log_file** – The fully specify the name of the log file where all information and errors are to be written by the Batch Loader.
- **log_level** – The level of detail included in the logging.
 - 1 – Low, recommended for automated jobs with low risk on failure. Details about start and stop, and how many items succeeded.
 - 2 – Medium, recommended for use while developing new Batch load job. Logs details about failure, as well as details logged in low mode.
 - 3 – High, recommended for debugging. Provides detail and AML about every line loaded.
- **certStoreLocation** – the name of the certificate's store location (CurrentUser, LocalMachine) used for Client Certificate Authentication.
- **certStoreName** – the name of the certificate's store used for Client Certificate Authentication.
- **certFindType** – the type of a search criteria that will be used to find a client certificate in the specified store. Here is the list of available values:
<https://docs.microsoft.com/dotnet/api/system.security.cryptography.x509certificates.x509findtype?view=net-6.0>
- **certFindValue** – the value of a search criteria that will be used to find a client certificate in the specified store.
- **certValidOnly** – parameter that determines whether to accept only valid certificates (true) or any certificate (false).

You can also specify the data about the client certificate in the 'blworker.exe.config' file. In this case additional CLI or configuration parameters are not required. There is an example of the file's content below:



```

<configuration>
<configSections>
<sectionGroup name="Aras">
<sectionGroup name="Net">
<section name="RequestProvider"
type="Aras.Net.Configuration.RequestProviderConfigurationSection, IOM"/>
</sectionGroup>
</sectionGroup>
</configSections>
...
<Aras>
<Net>
<RequestProvider>
<providers>
<provider uriPattern=".*">
<ClientCertificate>
<certificate sourceType="CertificateStore" storeLocation="CurrentUser"
storeName="My" findType="FindByThumbprint"
findValue="71f942b0710faa873d1d3de2a9815b3321604d5a"/>
</ClientCertificate>
</provider>
</providers>
</RequestProvider>
</Net>
</Aras>
</configuration>

```

Item Mapping in the Template file

A Batch Loader template file contains a parameterized template of the AML request that is sent to the Aras Innovator server by the batch loader session for every processed row of the input data file. Parameters of the template have format @N, e.g. @1. The N number is the integer column number from left to right in the data file.

Basic Item Mapping

Below is an example of what a template file might look like for the User ItemType.

```

<BatchLoaderPrototype>
<Item type="User" action="add">
<login_name>@1</login_name>
<password>607920b64fe136f9ab2389e371852af2</password>
<logon_enabled>@2</logon_enabled>
<first_name>@3</first_name>
<last_name>@4</last_name>
<email>@5</email>
</Item>
</BatchLoaderPrototype>

```



This file could be used to add Users into Aras Innovator, like the GUI.

Important

The password has been hard coded to a specific value of 'innovator'.

By making a couple of small changes to the file:

```
<BatchLoaderPrototype>
<Item type="User" where="login_name='@1'" action="merge">
<login_name>@1</login_name>
<!-- password>607920b64fe136f9ab2389e371852af2</password -->
<logon_enabled>@2</logon_enabled>
<first_name>@3</first_name>
<last_name>@4</last_name>
<email>@5</email>
</Item>
</BatchLoaderPrototype>
```

We make the file usable for a nightly load of user data. Notice that the password has been commented out, to prevent the batch job from overwriting the current values. Notice also, that the Item tag has a where clause included to specify how to query for existing Users. Lastly, notice that the top-level action has been changed from 'add' to 'merge'. This is so that if the User in the AML sent to the server exists, then the action is to 'edit' the existing user. However, if the User does not exist, then the action is set to 'add', so that the new user is created. This is especially useful when trying to manage process such as User data uploaded from Active Directory.

Item and Relationship Simultaneous Load

Drawing from the example in the section [Batch Loader Graphical User Interface](#) , we can make an AML template that would load better represent how the Part BOM relationship should be loaded from the same sample data file.



```

<BatchLoaderPrototype>
<Item type="Part" where="item_number='@1'" action="edit">
<Relationships>
<Item type="Part BOM" action="add">
<related_id>
<Item type="Part" action="get" select="id">
<item_number>@2</item_number>
</Item>
</related_id>
<quantity>@3</quantity>
</Item>
</Relationships>
</Item>
</BatchLoaderPrototype>

```

This AML template make the additions in the context of exiting source Part Items only and forces the part data to make updates to the Part Item history, while uploading the BOM data.

Uploading a File Item

Uploading files to Aras Innovator can be done using the Batch Loader, but Aras recommends loading the File Item in the context of the source item. The following sample template allows you to load the File Item as part of loading the source Document Item. This decreases the complexity of trying to create the correct Document File relationship if the Document and File had been loaded separately.

```

<Item type="Document" action="add">
<item_number>@1</item_number>
<description>@2</description>
<Relationships>
<Item type="Document File" action="add">
<related_id>
<Item type="File" action="add">
<actual_filename>@3</actual_filename>
<filename>@4</filename>
<Relationships>
<Item type="Located" action="add">
<related_id>67BBB9204FE84A8981ED8313049BA06C</related_id>
</Item>
</Relationships>
</Item>
</related_id>
</Item>
</Relationships>
</Item>

```

From the AML, you can see that @1 and @2 are mapped to properties of the document item. To load the File Item, we are leveraging the standard Aras Innovator IOM. This requires that we define the



following parameters, at a minimum.

On the File Item:

actual_filename (@3 above) This is the path to the file that is to be vaulted.

Example: C:\Drawing.bmp

filename (@4 above) This is the name of the file that for the database.

Example: Drawing.bmp

Uploading a File Item also requires the 'Located' relationship be defined. In the example provided, the Located relationship has a related Item of the Default vault. If the File must be vaulted to a Vault other than the Default vault, the AML template must be altered to use the id of the alternate Vault.

