

HiPortfolio Job Queue Server

hipQueue Version 1.0

Table Of Contents

1	Introduction	2
2	Processing phases.....	2
2.1	New	2
2.2	Syntax Check	2
2.3	Add to queue	2
2.4	Wait	2
2.5	Write Fin file.....	2
2.6	Error	3
2.7	Archive	3
2.8	Processing log	3
3	Requests.....	3
3.1	Model job	3
3.2	Batch check.....	3
3.3	Time Out.....	3
3.4	Remark.....	4
3.5	Set statements.....	4
3.6	Command summary.....	4
3.7	Examples.....	4
3.7.1	IDX Cash import request.....	4
3.7.2	IDX On Market Importation request.....	4
3.7.3	RG request	4
4	hipQueue Error handling	5
4.1	Know Error States Which hipQueue Can't Handle	5
4.2	I: must be mapped to HiPortfolio drive	5
4.2.1	Suppressing HiPortfolio 'Error' messages	5
5	Specification file.....	5
6	Main display.....	5
6.1	Logon	6
6.2	Start/Stop.....	6
6.3	Log	6
7	System options	7
7.1.1	Sample rate	7
7.1.2	Trace On.....	7
7.1.3	HiPortfolio settings.....	7
8	HiPortfolio Job Queue.....	7
8.1	Load queue.....	8
8.2	Save to CSV file.....	8
8.3	Clear queue	8
8.4	Save to folder	8
8.5	Test model job	8
8.6	Test all model jobs.....	8
8.7	Refresh model jobs.....	8
8.8	Same model job.....	8
8.9	Same model job1	8
8.10	Convert saved to model jobs	8
9	Completed job queue.....	8
10	Creating a model job.....	8

1 Introduction

To reduce the development cost and maximize usability:

- All XML processing is within hipQueue
- No XML knowledge is required to run API via hipQueue

The Program:

- Translates a simple CSV file to XML based HiPortfolio request and calls API
- Interprets and translates HiPortfolio API responses and error messages
- Returns simple FIN files
- Automatically updates its own batches
- Waits for job completion and reports it
- Logs all activity to two files and memory based session log which are auto-condensed

Job queue can be:

- Loaded and displayed
- Cleared
- Downloaded as XML files to a folder
- Used to display a single job as XML file
- Used to create a trial model job from job and then test it

2 Processing phases

hipQueue creates XML control file in its processing folder (XML file is stored as XXXX.cnt). Then it goes through the following processing phases:

2.1 New

New request

2.2 Syntax Check

The request syntax and validity of supplied tokens is checked

2.3 Add to queue

HiPortfolio XML based request is added to the job queue

2.4 Wait

hipQueue checks if a job is completed by querying the job completed queue. If request specifies wait for batches update then it will wait for ALL batches to be updated.

2.5 Write Fin file

HiPortfolio/hipQueue result is written to a fin file in the format
Ok or Error | HiPortfolio job Id | Comment |time taken in seconds

For example

```
Error|-1|j:\SYSTEM\HIPORT\XAPI.Dll not found|20
```

- Job Id is either -1 (not added to the job queue) or Job Id from the job queue
- Comment is either BLANK or contains error message, time out and/or other hipQueue information
- Time taken is the number of seconds it took to complete request. Please note that hipQueue will query job completed queue every 20 seconds. This time is an overestimate of the actual completion time

2.6 Error

An implied phase is the error state. If hipQueue detects any errors be it syntax or HiPortfolio based then it will return Error Fin file

2.7 Archive

The final processing phase is archiving of all files created from request to HAM archive.

2.8 Processing log

Control file contains all of information it requires to process its request. It also contains a log of all processing.

3 Requests

Request to hipQueue is a pipe (|) delimited file in the following format:

Command |Token |Value

File should have extension .txt and its name must be unique across all requests.

Each hipQueue request is analyzed for:

- Syntax
- Correctness
- Token presence in the model job

Please note that:

- Token value is not checked. Its up to requestor to make sure that all values are correct.
- Tokens and values are case sensitive
- API expects dates in format yyyy-MM-dd

3.1 Model job

Selects a model job stored within the system.

Job | Model job name

For example

Job| IDX CASH IMPORT

3.2 Batch check

Check if there are any batches and updates them if any are present.

Batch Check | Yes or No

Default value is Check batches set to No

The return fin contains 'No batches detected' in the comment field.

3.3 Time Out

Time out | ddd

Where 'ddd' is the number of minutes after which the request is timed out.

If request is timed out Fin file contains Error message. System default value is 180 minutes (3 hours)

For example
 Time out |100
 Will time out this request after 100 minutes.

3.4 Remark

Remark | This is a comment

3.5 Set statements

Set | TOKEN |Value

hipQueue will match TOKEN within request to the model job it will then replace the model job value by the specified value .

If no Value statements are present within a request them hipQueue will send the model job to Hiport

Please note TOKEN and Value are case sensitive.

3.6 Command summary

Command	Syntax	Example 1	Example 2
Job	Job Model job name	Job CASH IMPORT	J CASH IMPORT
Batch Check	Batch Check Yes or No	Batch Check Yes	B Yes
Time out	Time out ddd	Time out 100	T 400
Remark	Remark This is a comment	Remark Comment	R Comment
Set	Set TOKEN Value	Set FUND XX	S FUND XX

Please note that 1st letter of any command is also acceptable

3.7 Examples

3.7.1 IDX Cash import request

Job| CASH IMPORT
 Batch Check | YES
 Set |FUND|XX

Please note that tokens and their values are case sensitive.

3.7.2 IDX On Market Importation request

J |On Market Import
 B| YES
 S|FILE-EXTENSION|TXT
 S|FILE-NAME|TEST
 S|FILE-PATH|...\HIPQUEUE\TEST
 S|MOVE-IMPORT-FILE|YES
 S|REJ-FILE-EXTENSION|ERR

3.7.3 RG request

J|TAX RG
 S|TASKNAME|*XX222
 S|PORTFOLIO|FRED
 S|FUND|XX

S|FILEPATH|..\..\TEST
S|FILENAME|FRED.txt
S|STARTDATE|2008-01-01
S|ENDDATE|2009-04-05
S|ASATDATE|2009-04-05

4 hipQueue Error handling

hipQueue is a robust system. It can handle most of the error conditions. All requests are syntax and content checked to reduce the risk of failure.

For example:

hipQueue can be abended, stopped and restarted without any loss of processing.

4.1 Know Error States Which hipQueue Can't Handle

The following error states can't be handled by hipQueue

- API DLL locks up both itself and hipQueue (Usually caused by network outage). Since hipQueue can't execute its code it can't handle this type of lockup.
- HiPortfolio API login password is expired, suspended etc.

4.2 I: must be mapped to HiPortfolio drive

Call to SLAVE API is "hard-wired" in hipQueue (there is no alternative) and it points to J: drive. Any other mapping of HiPortfolio drive will not work.

4.2.1 Suppressing HiPortfolio 'Error' messages

File HiPortfolioIgnoreMessages.txt contains messages that HiPortfolio falsely labels as errors.

These are:

```
No batches found to update  
No records to print  
Batch no longer exists. Maybe already updated  
No clients in this group
```

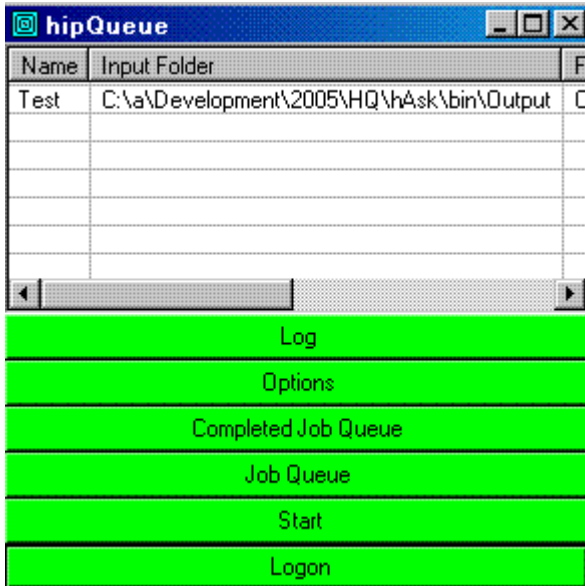
hipQueue will classify all messages that start with above strings as 'Ok'

5 Specification file

File Specification.csv contains specification of all hipQueue input (for request) and output folders.

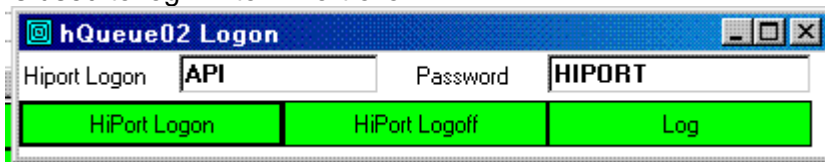
There is no limitation on the number of hipQueue In/Out folders.

6 Main display



6.1 Logon

Is used to login into HiPortfolio API



6.2 Start/Stop

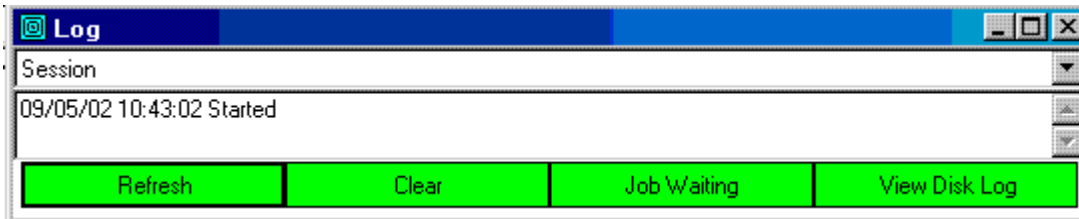
Runs or stops the program. While running the program will read requests and submit API calls to HiPortfolio.



Please note that while program running all buttons except for log are disabled.

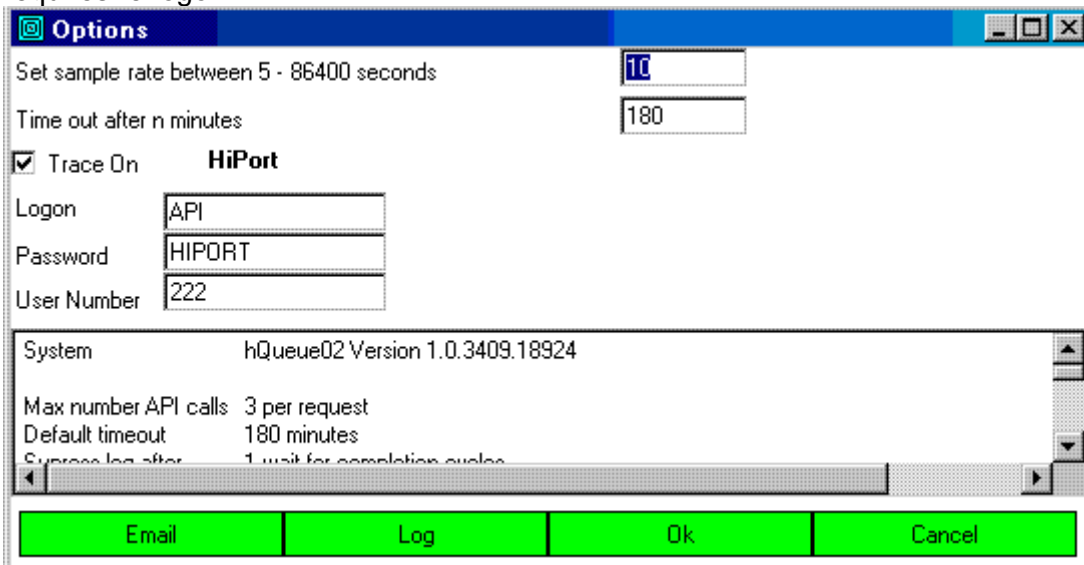
6.3 Log

All system logs can be displayed



7 System options

This form sets/changes system options. Please note that any changes HiPortfolio setting requires re-login.



7.1.1 Sample rate

Sets the interval that hipQueue uses to look at its folders. The default value is 10 seconds - hipQueue will look at all outstanding work every 10 seconds.

7.1.2 Trace On

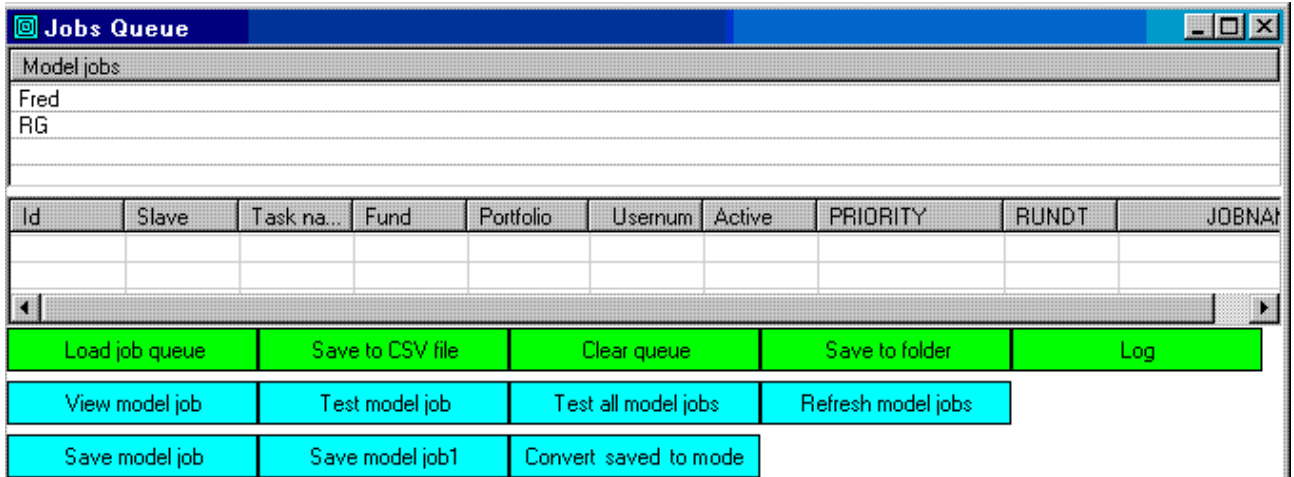
Tracing creates large number of files which will be stored in the 'trace' folder. Please use this facility to diagnose the problem. In production it's should be turned off.

7.1.3 HiPortfolio settings

Login name
Password
User number

8 HiPortfolio Job Queue

Please note double clicking on model jobs or the queue will display underlying XML



8.1 Load queue

Loads the current job queue

8.2 Save to CSV file

Saves the queue to CSV file.

8.3 Clear queue

Clears HiPortfolio job queue

8.4 Save to folder

Saves the job queue to a folder as XML files

8.5 Test model job

Adds the job to HiPortfolio job queue and checks for any errors.

8.6 Test all model jobs

Adds all jobs to HiPortfolio job queue and checks for any errors.

8.7 Refresh model jobs

Reads model jobs folder

8.8 Same model job

Creates model job from the HiPortfolio job queue **without recursion**

8.9 Same model job1

Creates model job from the HiPortfolio job queue **with recursion**

8.10 Convert saved to model jobs

Converts all saved XML files to format which can be added to HiPortfolio queue

9 Completed job queue

Displays the completed job queue

10 Creating a model job

- Place a job on the HiPortfolio queue – make sure it can't run on a slave
- Rename the job a model job
- Rename the model job and test it