**IEX Monitor – Remotely Control & Monitor IEX Servers**

IEX Monitor is a (WCF) Web Service hosted on every server machine, one instance per PC, and listens for requests for control and monitoring operations on IEX server on the local machine

Client Machine

IEX Machine

Monitor Service

Monitor Client

IEX 1

IEX 2

WCF

**API – detailed description follows**

bool StartIEX(int iexNumber, bool shutdownUponErrors, out string servicesState);

void StopIEX(int iexNumber);

bool RestartIEX(int iexNumber, bool shutdownUponErrors, out string servicesState);

ServerState GetServerState(int iexNumber);

string GetServicesState(int iexNumber);

**Control APIs**

bool StartIEX (out string servicesState, int iexNumber, bool shutdownUponErrors)

Parameters

*iexNumber* the server number to start. Must be between 1 and 8

*servicesState* out, holds the services state after server completes loading. Check this state to get load error details if server loaded with errors

*shutdownUponErrors* shutdown the server if one of the services has errors

Returns

*true* if server loaded without errors

*false* if server loaded with errors

Exceptions

Invalid IEX number

Timeout – server failed to load in a given timespan

void StopIEX (int iexNumber)

Parameters

*iexNumber* the server number to start. Must be between 1 and 8

Exceptions

Invalid IEX number

Server didn’t stop after 10 seconds of attempts

bool RestartIEX (out string servicesState, int iexNumber, bool shutdownUponErrors)

Parameters

*iexNumber* the server number to start. Must be between 1 and 8

*servicesState* out, holds the services state after server completes loading. Check this state to get load error details if server loaded with errors

*shutdownUponErrors* shutdown the server if one of the services has errors

Returns

true if server loaded without errors

false if server loaded with errors

Exceptions

Invalid IEX number

Timeout – server failed to load in a given timespan

**Monitoring APIs**

ServerState GetServerState (int iexNumber)

Parameters

*iexNumber* the server number to start. Must be between 1 and 8

Returns

*ServerState*

Enumeration values: NotRunning, LoadingServices, Running, Unknown

Exceptions

Invalid IEX number

Timeout – server failed to load in a given timespan

string GetServicesState (int iexNumber)

Parameters

*iexNumber* the server number to start. Must be between 1 and 8

Returns

*string* in XML format containing a list of the services and their state

Exceptions

Invalid IEX number

IEX is not running

**Client API**

**Usage: IEX.Server.Monitor.Client.exe** Action Host IexNumber [options]

**Important Note**: the following files must be located in the executable folder

IEX.Server.Monitor.Client.exe.config

IEX.Utilities.dll

**Arguments**

*Action* required. One of StartIEX, StopIEX, RestartIEX, GetServerState, GetServicesState

*Host* required. The host machine. default: localhost

*IexNumber* required. Between 1 and 8

*/ShutdownUponErrors* not required whether to shutdown the IEX if one of the services is in “Faulted” state. If not provided, the default is false

*/ResultsPath*:path not required. directory in which the result files will be written to. Default: executable path. **Note**: if the path is invalid the error file will be created in the executable location

**Console Output**

PASS will be written to the console if no errors occurred

FAIL will be written to the console if **any** errors occurred. See detailed errors in the result.txt and error.txt files

**Input Parameters Error file**

A file named **error.txt** is created in *ResultsPath* location with all detected input (e.g. parameters) errors

Note: The file is overridden every run. The user must use the file before the next run

**Result file**

A file named **result.txt** is created in *ResultsPath* location holding the invocation result

Notes: The file is overridden every run. The user must use the file before the next run. Any **input errors** are written to error.txt, see the previous section

Result file structure

Action: the requested action

Result: success if the call completed without errors. Failure in case an exception occurred (e.g. monitoring host is not running)

Return: the return value of the invocation (true, false in case of StartIEX, RestartIEX; ServerState enumeration in case of GetServerState)

State: the services state (XML string. Relevant in StartIEX, RestartIEX, GetServicesState)

**Monitoring Components**

**Monitoring Service**  - IEX.Server.Monitor.dll

Holds the actual WCF service to control the servers

**Monitoring Service Host** – IEX.Server.Monitor.Host.exe

Hosts the monitoring service

One instance per machine

Must run at all times

Launched at machine startup / user login

Launched every time a server is launched

**Monitoring Client**  - IEX.Server.Monitor.Client.exe

GUI mode: launch without parameters to get the GUI for invoking the monitoring service

Command line mode: launch the arguments, as described above