

SavvyDownloadScheduler

Purpose

This application allows scheduling of web downloads (FTP, HTTP, etc.) at set times or frequencies.

It currently downloads around 65,000 files per month as a result of around 150 jobs scheduled through the application.

Technical Details

This application is a SavvyPlus-developed console application written in Python.

It runs on MEL-SVR-AP-001 as a scheduled task and is designed to run continuously.

The application is located at E:\SavvyApps\SavvyDownloadScheduler\dist\SavvyDownloadScheduler and writes log files to E:\ApplicationLogs

Database connection string is set in config file SavvyDownloadScheduler.cfg

The application writes to and reads from the MarketData database. Configuration table is NEMWebFolders and file download records are stored in NEMWebFiles

The application needs certificate file cacert.pem present in application folder to run successfully

Monitoring and trouble-shooting

There is indirect monitoring of this application via a SQL Server Agent job that runs each hour. If no files have been loaded by the SavvyDataLoader in the last hour then an email is sent to the SQLAdmin email list. If such an email is received then the SavvyDownloadScheduler not running correctly is one possible cause.

If the application is not successfully downloading files then re-starting the application is the only thing generally required.

For more advanced trouble-shooting consult the log file or run the application from a dos box to see error messages.

Scheduling Downloads

If a new scheduled download is added or an existing one modified then the application needs to be re-started for the changes to take effect.

Download jobs are set up using table MarketData.dbo.NEMWebFolders. Parameters are as follows:

Parameter	Description
URL	URL from which to download. Can contain {year},{month},and/or {day} in which case the download running date will be substituted in yyyy, mm, dd format
interval_minutes	interval (in minutes) between downloads
start_time	time of day to begin the download.
active_flag	1 makes the job active, 0 makes it inactive
destination_folder	folder (on relevant server) to which downloaded files should be written
retrieval_type	<p>LINKS - Use if the specified URL has links to the actual files to be downloaded. Only NEW files will be downloaded (i.e. new files names, not ones already stored in database as being downloaded previously). Only links matching <i>filename_pattern</i> will be downloaded.</p> <p>FTP_FILES - Downloads files from an FTP folder specified by <i>URL</i>. Files matching <i>filename_pattern</i> will be downloaded. Previously downloaded file names will be ignored.</p> <p>DIRECT_FTP - Downloads the file at the named <i>URL</i> via FTP. The <i>URL</i> should include the full file name, and <i>filename_pattern</i> is used to determine the name of the file written at the destination.</p> <p>LINKS_OVERWRITE - same as LINKS but will download all matching files irrespective of whether the same named file has previously been downloaded.</p> <p>DIRECT - Downloads the file at the named <i>URL</i> via HTTP.</p>

filename_pattern	The file mask used to determine which links/files to download, or the pattern for the file name to save to (depending on <i>retrieval_type</i>). If it contains {year} or {month} or {day} then these will be replaced with the year, month or day of the download.
utc_offset_hours	The {year}, {month} and {day} will be the UTC time offset by this number of hours i.e. +10 will be the current AEST time, -14 will be yesterday's date in AEST.
user_description	Description of the download job. Not used by the system
warning_minutes	If this particular job has not downloaded anything for longer than <i>warning_minutes</i> then it indicates a problem with the data feed. Not used by the system.