Contents

Basics

Tasks

Below is a list of tasks which are supported by Hitek Software's products. Please note that all tasks listed below may not be available in your program. For example, the Command task is available in Automize, but not in AbleFtp.

	Automize Standard	Automize Professional	Automize Enterprise	JaSFtp	AbleFtp	AutoKrypt
Scripting	N	Y	Y	N	N	N
SFtp tasks	Optional	N	Y	Y	N	N
SFtp browser	Optional	N	Y	Y	N	N
Command-line module	Optional	Optional	Y	Optional	Optional	Optional
Open PGP Encryption	Optional	N	Y	N	N	Y
Scheduler	Y	Y	Y	Y	Y	Y
Task Chaining	Y	Y	Y	Y	Y	Y
NT service	Y	Y	Y	Y	Y	Y
Task Synchronization	Y	Y	Y	Y	Y	Y
Email notification	Y	Y	Y	Y	Y	Y
Dynamic variables	Y	Y	Y	Y	Y	Y
Ftp tasks	Y	Y	Y	Y	Y	N
Ftp browser	Y	Y	Y	Y	Y	N
Command task	Y	Y	Y	N	N	N
WinCommand task	Y	Y	Y	N	N	N
Synchronize task	Y	Y	Y	Y	Y	Y
Copy task	Y	Y	Y	Y	Y	Y
Zip task	Y	Y	Y	Y	Y	Y
Unzip task	Y	Y	Y	Y	Y	Y
Delete task	Y	Y	Y	Y	Y	Y
Print task	Y	Y	Y	N	N	N
TextSearch task	Y	Y	Y	N	N	N
Ping task	Y	Y	Y	N	N	N
Telnet tasks	Y	Y	Y	N	N	N
Email task	Y	Y	Y	N	N	N
Check email task	Y	Y	Y	N	N	N
Bulk email task	Y	Y	Y	N	N	N
Web task	Y	Y	Y	Y	Y	N
File monitor task	Y	Y	Y	Y	Y	Y
Dir monitor task	Y	Y	Y	Y	N	N
Dir change task	Y	Y	Y	N	N	N
Logfile monitor task	Y	Y	Y	N	N	N
SQL task	Y	Y	Y	N	N	N
Open Documents (W)	Y	Y	Y	N	N	N
Shutdown (W)	Y	Y	Y	N	N	N
MS Office tasks (W)	Y	Y	Y	N	N	N
Alarm task	Y	Y	Y	N	N	N
Message task	Y	Y	Y	N	N	N
Echo task	Y	Y	Y	N	N	N
Wakeup task	Y	Y	Y	N	N	N
Delay task	Y	Y	Y	N	N	N
Send Keystrokes task	Y	Y	Y	N	N	N
Maintenance task	Y	Y	Y	Y	Y	Y
Email logs task	Y	Y	Y	Y	Y	Y

Administrator Password

You can set a password to lock the user interface. **IMPORTANT: Please remember, or write down this password. If you forget the password, you cannot access the user interface.** If you forget the password, you can contact Hitek Software to reset this password. Hitek Software will only provide this information to the licensee of this software.

Filename Format for Append To Filename option

Many of the tasks have the 'Append to filename' option. You can append Date/Time to the filename. You can also add your own Custom Text to your filename. You can also set the Date/Time format. Use the options indicated below, in the Settings Menu / FilenameFormat window.

Appending Custom Text to your filename:
You can add your own Custom Text to your filename, according to the rules below:
1) Text will add 'Text' to the end of the filename (after extension).
2) P1::Text will add 'Text' before the filename
3) P2::Text will add 'Text' before the extension
4) P3::Text will add 'Text' at the end (after extension if an extension exists)
5) P4:: will add the Date to filename (before extension if an extension exists)
6) P5:: will add the Date/Time to filename (before extension if an extension exists)
7) You can add multiple Text to your filename using + after each rule used
8) You can also enter task or system variables to be added to the filename, except for the P4:: & P5:: formats, which already add the date or date/time.

IMPORTANT: P1:: , P2:: etc., should be entered exactly as shown (case sensitive)
Examples:

Filename	Custom String Entered	Resulting Filename
test.txt	-java-$%DATE::ppddyy%$	test.txt-java-012704
test.txt	P1::pre_	pre_test.txt
test.txt	P2::_beforeExt	test_beforeExt.txt
test.txt	P2::_$%DATE::ppddyy%$	test_012704.txt
test.txt	P3::_afterExt-$%DATE::ppddyy%$	test.txt_afterExt-012704
test.txt	P2::_beforeExt_+P4::	test_beforeExt_01-27-04.txt
test.txt	P1::pre_+P5::	pre_test01-27-04_12:32.txt

Appending Date or Date/Time
If you select 'Date', the current date is appended to the filename. If you select 'Date/Time', the current date and time are appended to the filename. This is useful for archiving purposes. The day and month are always two-digit numbers. If time is appended, the hour and minute are always two-digit numbers.

The standard date time separators like / or : are not used. These separators cannot be used in pathnames or filenames, on some operating systems. Hence we chose -, _ , or none as the separators.

Date Format Sequence:
This allows you to select the sequence of day, month, and year.
Year Format:
You can set a 2-digit year or a 4-digit year, or choose not to have the year appended to your filename.
Date Time Separator:
This sets the separator between the date and time, if the 'Date/Time' 'Append to Filename' option is selected.
Date Component Separator:
This sets the separator between the day, month and year.
Time Component Separator:
This sets the separator between the hour and the minute fields, and between the minute and the second fields.
Include Seconds:
Select this option to append the second's field to the time.

Archive Logs

This administrative task allows you to archive the Output logs, Activity logs and all Task logs.

Target Directory
Enter the full path name of the directory, where you need to put the log files.

Place files in new subdirectory
If you select "None", no new subdirectory is created within the Target Directory.
If you select Date, a new subdirectory based on the date is created within the Target Directory. All files are placed in this directory.
If you select Date/Time, a new subdirectory based on date and time is created within the Target Directory. All files are placed in this directory.

Append to Filename
To archive each log file based on current time/date, choose the append file option.

Array Loop

This task allows you to loop through data from an array variable. You can either loop through all cells or loop by rows.

With every loop, the current cell values are set as variables.
If you loop by cell, the cell value is returned as a variable.
If you by rows, each columns cell values in the row is returned as variables. Also, the entire row is saved as a variable.

With every loop, you can run another task.

Array Read

This task allows you to read data from an array variable. You can either read a certain cell or an entire row.

If you read a cell, the cell value is returned as a variable.
If you read a row, each columns cell values in the row is returned as variables. Also, the entire row is saved as a variable.

Arrays

Arrays are a new feature since 9.0.

An array is a special variable that stores 1-D or 2-D data in a file within the ..data\variables folder. The array files are named using the following format:
ARRAY--tasktitle--variablename
Example:
ARRAY—copytest--Filenames

You can get the values of an array variable using the the following format:
$%ARRAYNAME::FIRSTROW%$

$%ARRAYNAME::LASTROW%$

$%ARRAYNAME::ALLROWS%$

$%ARRAYNAME::FIRSTCELL%$

$%ARRAYNAME::LASTCELL%$

$%ARRAYNAME::ROW::rownumber%$

$%ARRAYNAME::CELL::rownumber::columnnumber%$

examples:

$%ARRAY—taskTitle--variable::FIRSTROW%$

$%ARRAY—dirlisting--DirectoryListing::FIRSTROW%$

You can view the contents of an array using the Variables menu / Task Variables window.

An array is typically 2 dimensional. The most common array is the Filenames variable in the Copy or Ftp tasks. When a variable value can potentially become very large, it is stored in its own array file. The Filenames variable in the Copy,Ftp,SFtp tasks, list each filename in a new row (line). Each row contains multiple columns for the files properties like name, size, timestamp and directory path etc..

There are various tasks that work with arrays:
1) Array read task - read the value of any cell or row within the array.
2) Array loop task – loop and read value of all cells/rows and run another task every loop
3) ArrayToFilename task – Converts an array into a filename variable (#file1^#file2^#file3)
4) FilenameToArray task – Converts a filename variable into a 1-D array. One filename per line (row).

Array To Filenames

This task allows you to convert an array variable containing list of file information, to a variable which can be used in the Filenames field.

Example Array:
test.txt, c:\data\test.txt, 09-08-10 4:50 AM, 1546
test3.txt, c:\data\test3.txt, 09-09-10 4:52 AM, 1978

could be converted to a variable suitable for use in filename fields:
#test.txt^#test3.txt

In the task settings, you have to specify the column in which the filename appears. In the above example, the filename is in column 1. This is the default for most arrays.

Auto Backup

This administrative task allows you to backup all the user settings, task table data and schedule table data.

Target Directory
Enter the full path name of the backup directory.

Place files in new subdirectory
If you select "None", no new subdirectory is created within the Target Directory.
If you select Date, a new subdirectory based on the date is created within the Target Directory. All files are placed in this directory.
If you select Date/Time, a new subdirectory based on date and time is created within the Target Directory. All files are placed in this directory.

Backup and Restore

1) This feature allows you to backup all the user settings, task data, schedule data, and logs.

2) Use this feature to reinstall the software on a new system, or to move the program to a new system. In this case, you can backup to a temporary directory. Then install the program on the new system, and use the 'Restore' option.

Backup Directory

Enter the full path to a valid empty directory on your system or backup media. On your system, we recommend using "USERHOME/Prgram_Name_Backup" as the backup folder, where USERHOME is the current user's home directory.

Restoring data

When you need to restore the program data, use the Restore button. The last backup directory will be shown.

Bulk Email

The email task allows you to send email automatically to multiple recipients in sequence. This allows you to hide the identity of recipients from each other. This also allows you to send email to a large audience using an email list file. So you do not need to manually type in the email addresses. You need to specify the recipients in a text file. Each address needs to be on a new line. To create a bulk email task, select the "Create Task" menu and then the "Bulk Email" menu item.

Email Server Profile

Select the profile name for the email server you need to connect to. You should have previously created an email profile, using the Email Profiles menu item.

Recipient list file and additional parameters for mail merge

Select the recipient list file. The file should contain the list of email addresses. One email address per line. Additionally, if you wish to pass multiple parameters for each email address, enter them after the email address separated by ^. Then in any field (subject, message body, headers or attachment paths) you can specify a parameter using %%1, %%2 etc..
Example of Recipient list File:
rob@test.com^Rob Gibson^London^60
jane@test.com^Jane Doe^New York^90
In your email, you can access the Name using %%1
In your email, you can access the City using %%2
You can also access the recipient email address using %%0
Example Body:
Dear %%1, The temperature in %%2 is %%3 F. Your email address is %%0.
This would result in the following body sent to rob@test.com:
Dear Rob Gibson, The temperature in London is 60 F. Your email address is rob@test.com.

Subject

Enter the subject. This field also supports dynamic variables. For example, you could automatically parse the current date or computer IP address into the subject.

Delay

This is the time interval in seconds between successive emails. Normally a value between 5 and 60 seconds should be used.

Message Body

1) Enter the message body directly into the text field
2) This field also supports dynamic variables. For example, you could automatically parse the current date or computer name into the message.
3) Or, You can let the software dynamically parse the message text from a file on your system using the following format:
$$$[message_file_path].
for example on windows, if the message file is c:\data\message.txt, use:
$$$c:\data\message.txt

Adding Attachments

1) To add an attachment, type in the full path name of the file, into the Attachment Text Box. You can also use the "Browse" button to select the file. The format for the attachments is: AbsoluteFilePath1, AbsoluteFilePath2,......
Example: c:\data\test.txt,c:\temp\data.htm
2) You can also specify multiple filenames in multiple directories, by using the following format:
[DIR]c:\your_dir[FILE]name_filter,...............

Enter the filename_filter for the files you wish to transfer, using the following wildcard rules
Example: to attach all files that end with .txt from the c:\temp folder, use:
[DIR]c:\temp[FILE].txt
3) This field also supports dynamic variables.
Example:
To attach all files in the c:\temp folder, which has the current date (example: 06-21-02) in their filename, use:
[DIR]c:\temp[FILE]$%DATE::pp-dd-yy%$
See dynamic variables for more details on formatting the date

Headers

You can specify headers to add to the email. This will allow you to add mail priority, or other specific flags. An email server can decide not to display all the headers you specify; an email server may not understand the headers you specify. For example, if you attempt to specify Email priority level, an email server may accept any, or all, or none of the following flags: X-Priority, X-MSMail-Priority, Importance, and Priority.

To specify headers, use the following syntax:
header1=value1^header2=value2^header3=value3

Header Examples:
X-Priority=1^ Priority=Urgent
Importance=high
Reply-To=test@test.com

Chains

This feature allows you to run multiple tasks in sequence. You can run programs, email, web, ftp and all other tasks from the Chain. If a certain task fails, you can choose to stop the Chain. You can also skip ahead to another task in the chain.

Creating a Chain

From the front panel: click on the Chain button.

Adding Tasks to your Chain

The table on the upper right, displays the tasks you can add to your Chain.
Select the task you need to add.
Click the "Add Task" button. Your task will be added to the table, on the bottom of the window.

Email Notification of tasks in the chain

By default, only the Chain itself can trigger email notification based on exit code. The tasks in a chain do not trigger email notification, even if they satisfy an email notification profile. You should select this option, in order to allow tasks in the chain to trigger email notification. The chain will move on to the next step, only after the email notification has been completed.

Starting Chain at specified step

You can specify a step to start at, by entering the step number, or the step title in this field. By default, the chain will start at step 1, if this field is left blank.

Deleting Tasks from your Chain

Select the task in the bottom table and click the delete button.

Editing Tasks in your Chain

Select the task in the bottom table and click the edit button. The task parameters will be loaded into the window. You can then make any changes you require and click the "Update Task" button.

Positioning a Task within your Chain

Select the task in the bottom table and use the move-up or move-down buttons as required.

Enable Step

You can enable or disable a step. If a step is disabled, the task will move on to the next step.

Skip to Step options

You can choose to skip to certain step in the chain, if a certain exit code criteria is met. If you wish to stop the chain, because a certain exit code criteria is met, enter a Skip to Step value, which is greater than the total steps in the chain.

Skip to Step (or Title)

1) You can enter either the step number in the chain, or the task title. This task title has to be a valid title in the chain list.
2) You can also run multiple tasks simultaneously using the following rules:
a) Separate each Task Title by &. Example: TaskTitle1&TaskTitle2&TaskTitle3 (no spaces allowed)
b) You can only enter Task Titles, NOT step numbers.
c) TaskTitle1 has to be a valid Task Title in the chain list
d) TaskTitle1 CAN NOT be the last step in the chain list. You can add a dummy delay task as the last step in the chain, if TaskTitle1 happens to be your last step.
e) TaskTitle2, TaskTitle3 etc. DO NOT have to be in the chain list. They have to be valid tasks in the task list.
f) In this simultaneous mode, all tasks (TaskTitle2, TaskTitle3. etc.) will be triggered simultaneously. The chain will then continue normally at TaskTitle1.

Rerun Options

You can choose to rerun the task, if it meets certain exit code criteria. You can also set the maximum number of times to rerun a task. The task will rerun, while the rerun criteria is valid, until the maximum rerun count has been met.

Maximum Wait Time

This is the maximum time (**T1**) the Chain will wait for your task to finish. Enter 0 to wait forever until the task finishes its run. ***Please note that the Chain will not kill or terminate your task because it is unsafe to do so.*** It just waits for your task to complete normally. If the task does not complete normally in the allotted time, the chain will assume that the task has failed. The email, web, ftp and most other tasks will terminate normally.
For the **DirChange** and **Command** tasks, please ensure that you can Terminate the task within the Task settings itself. Both these tasks have a safe Termination time (**T2**) that can be set within the task. Please set **T1 T2**. If you have set the Command or DirChange task to rerun on failure, please set the **T1 (T2+Td)*n**, where **Td** is the delay between reruns, and **n** is the number or reruns.

Why not just Kill a task when the wait time is exceeded

Short answer from us: In the chain, each task is run as a thread. Killing a thread is unsafe, and may corrupt your system hours or days in the future.
Long answer from Java documentation: Because it is inherently unsafe. Stopping a thread causes it to unlock all the monitors that were locked. (The monitors are unlocked as the Thread Death exception propagates up the stack.) If any of the objects previously protected by these monitors were in an inconsistent state, other threads may now view these objects in an inconsistent state. Such objects are said to be damaged. When threads operate on damaged objects, arbitrary behavior can result. This behavior may be subtle and difficult to detect, or it may be pronounced. Unlike other unchecked exceptions, Thread Death kills threads silently; thus, the user has no warning that his program may be corrupted. The corruption can manifest itself at any time after the actual damage occurs, even hours or days in the future.

Check Email

This task allows you to download email messages from your email server to a user-specified folder. Only messages within your INBOX folder are searched. You can specify conditions based on date, sender or subject. Only messages that match your specified conditions, will be downloaded. Any attachments are also downloaded.

Conditions

Condition 1 is required. Condition 2 is optional.
For **date** comparison, enter the number of days required. To download all messages that have been received in the last 3 days, enter **3** in the text box.
For conditions based on **subject** or **sender**, t**he comparisons are always case sensitive.**

and/or

Choose **and,** if both Condition l and Condition 2 should be satisfied. Choose **or,** if either Condition 1 or Condition 2 can be satisfied.

Email Folder

Enter a valid folder path on your local system. All emails will be downloaded to this folder. The format of the email names is the name_x_ , where name is your Check Email task title, and x is the message number that has been downloaded. For example, if your task title is "emailTest", and 3 messages were downloaded, the 3 email files will be emailTest_0_ , emailTest_1_ , and emailTest_2_ . All attachments will also be downloaded to this folder, and will maintain their original names.

Append to filename

You can save downloaded messages and attachments to a different filename based on date, or date/time, or custom code you set.

Email Server Type

Consult your email service providers documentation for information on your email server type. POP3 is currently more popular. IMAP is the newer protocol, and many email servers will move to IMAP in the future.

Email Server Name

This is the most difficult parameter to set in this task. **Please consult** your email client software for this value. Look for POP3, IMAP, or incoming email server name. The exact name is required. The name is case sensitive. URL's are not allowed. If you use email clients such as: Netscape, Outlook, Lotus, or Eudora, please copy this setting from the software itself.

Email Server Port

The default POP3 email port is usually 110.

Username

The username is case sensitive. No white space at the end.

Password

The password is case sensitive. No white space at the end.

Command

This feature allows you to run executable programs. To run shell commands, batch/script files, you will need to launch the appropriate shell as an executable. Then use the shell to run the command.

Command Line

Executables: to run an executable, browse to it, or enter it manually. Example:
c:\winnt\notepad.exe
If the path to your executable includes spaces, enter the path to the executable in quotes. Example:
"c:\program files\myexe.exe"
To enter a commandline parameter to pass to your executable, enter a space between the path and the parameter. Leave spaces between any subsequent parameters. Example:
"c:\program files\myexe.exe" par1 par2

For Shell commands, and batch or script files: the syntax required vary depending on operating systems. Examples of batch commands for Win 95/98, WinNT/2000pro, Unix, and Mac OS X are listed at the end of this section.

This field also supports dynamic variables. For example, to launch a batch file, and pass it the current time, use:
c:\data\test.bat $%DATE::hh:mm:ss%$

Working Directory

The working directory is required for your program, command or script to run correctly. This is very important if the program or script uses relative paths to access files.

Environment Variables

This is an array of strings. Each element of this array has environment variable settings in format name=value. You should separate each name=value pair by a delimiter character. You can choose the delimiter to separate each element. The default delimiter is ^. If any of your variable name=value pairs contains the ^ character, then you should set another character as your delimiter. The delimiter cannot be "=" or any other character that appears in your name=value pairs.
Example of correctly formatted environment variables on windows systems:
PATH=c:\test;c:\test2;c:\test3^TEMP=c:\temp

This field also supports dynamic variables. For example, to launch a batch file and set the variable 'CUSTOM_DATE', use:
CUSTOM_DATE=$%DATE::Qqq dd, YY HH:mm:ss am_pm%$
This will set the variable CUSTOM_DATE in the batch file to: Aug 21, 2002 10:21:23 AM etc..

Termination Time

The termination time is the maximum time that your program or script can run. The task will terminate your program, if it is still running, after the maximum allowed time. Enter a value = 0, if you do not want to terminate your program. In this case, the task will wait indefinitely, for your task to finish.

Polling interval

While your program or script is running, the task continuously polls your program/script. The task will check if the program has completed. The task will obtain the Exit Code value, when your program exits. You can select the polling interval (default = 15 seconds). The minimum interval is 1 second. We recommend using 15 seconds.

Command line Examples

Win 95/98

1) Executable programs:

Command line = c:\windows\notepad.exe
Working Directory = c:\windows

2) Batch files (uses command.com, located usually in c:\windows or in c:\)
Command line = c:\windows\command.com /c c:\test\test.bat
Working Directory = c:\test

NOTE: you can also try and run batch files, just like an executable, depending on your system setup.
Command line = c:\test\test.bat

3) Shell commands (uses command.com, located usually in c:\windows or in c:\)
Command line = c:\windows\command.com /c set c:\test\set.txt
Working Directory = c:\windows
(this example will output all your environment variables to the file set.txt)

WinNT/2000pro

1) Executable programs:
Command line = c:\winnt\notepad.exe
Working Directory = c:\winnt

2) Batch files (uses cmd.exe, located usually in c:\winnt or c:\winnt\system32)
Command line = c:\winnt\system32\cmd.exe /c c:\test\test.bat
Working Directory = c:\test

NOTE: you can also try and run batch files, just like an executable, depending on your system setup.
Command line = c:\test\test.bat

3) Shell commands (uses cmd.exe, located usually in c:\winnt or c:\winnt\system32)
Command line = c:\winnt\system32\cmd.exe /c set c:\test\set.txt
Working Directory = c:\winnt\system32
(this example will output all your environment variables to the file set.txt)

Unix (Solaris, Linux, MacosX or other)

To run shell commands or scripts, you will need to launch the appropriate shell as an executable. Then use the shell to run the command.
Example to run the script 'myscript' from the /home/name directory, using the 'sh' command interpreter:

```
Commandline = sh myscript
Working Dir = /home/name
```

To see the options available to run your script, use man sh, man csh, or man bash.

Another option to run a shell script or command, is to use the Telnet Task. Using the Telnet task, login to the local system, 'cd' to the desired director, using a first telnet command. Then run the shell script or command, using a second telnet command.

Mac OS X

1. **AppleScripts:** Use the 'osascript' command to launch your AppleScripts
   Example:
   Commandline = /usr/bin/osascript /user/Name/YourScript.scpt
   Working Dir = /user/Name

2. **Launch Applications or files:** You can also use the 'open' command in MacOS X, to open a document or application

   Example:
   Command Line = open Stickies.app
   Working Dir = /Applications

3. **Command line**: To run shell commands or scripts, please see the section on Unix above.

Convert Previous Data

This feature allows you to import previous version data.
All the 7.x, 8.x, 9.x data is stored in the 'data' folder.
The 'data' folder is located within the main installation folder
Example: c:\program files (x86)\Automize9\data
Before you click the import button, make sure that you point to the correct 'data' folder of Version 7.x, 8.x or 9.x.

Version 4.x, 5.x, and 6.x data was stored in the 'inputs' folder.
The 'data' folder is located within the main installation folder
Example: c:\program files\Automize6\inputs
Before you click the import button, make sure that you point to the correct 'inputs' folder of Version 4.x, 5.x or 6.x.

Copy Files

The CopyFile task allows you to copy files from a directory or a directory tree. The 'move' option deletes the original files, after a successful copy operation. The 'rename' option renames the original file (or files) to the new name (or names).

Source Directory
Enter the full pathname of the directory where your source files exists.

Target Directory
Enter the full pathname of the directory where you need to put the target files.

Filename
For the Copy and Move options only, enter the filename filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time in addition to the normal wildcard search.
For the Rename option, the above wildcard options only apply for the file to pass the filename filter. If the file passes the filename filter, then its newly renamed name will depend on the Rename Filename field.

Rename Filename
This is a required field for the rename option. Rename Help

Folder Path Filter
Leave this field blank if you want all folders to be processed. If you need specific folders to be processed, then enter the folder name or path, using the following wildcard rules. If a folder passes the filter, then all its subfolders will also pass the filter.
Simple Example: Folder = c:\test , subfolders = john, john\old, sam, sam\old, joe, joe\old
Folder path filter = john . john and john\old are processed
Folder path filter = jo . john, john\old, joe, joe\old are processed
Folder path filter = old . sam\old, john\old, joe\old are processed
Folder path filter = test . all files in c:\test and its sub folders are processed
Folder path filter = john\old. only john\old is processed.
Folder path filter = c:\test\john . john and john\old are processed

Place files in new subdirectory
If you select "None", no new subdirectory is created within the Target Directory.
If you select Date, a new subdirectory based on the date is created within the Target Directory. All files are placed in this directory.
If you select Date, a new subdirectory based on date and time is created within the Target Directory. All files are placed in this directory.

Include subdirectories
If you select this option, all subdirectories and files within the Source Directory are copied to the Target Directory. If a subdirectory does not exist within the Target directory, then it is automatically created.

Append to Filename
If you are regularly coping a file and wish to save each copy to a different filename based on current time/date, choose the append file option which best suits your copying frequency and file naming needs.

Date Filter
If you need to filter files based on file modified date, select this option. For the between option, enter 2 values separated by '-' (2-4 etc.). Examples:
Older than 5 Minute, Newer than 2 Day, Between 3-5 Day, Between 1-4 Hour

Delete empty subfolders after Move

The move option first moves all files that meet the filename criteria, and other criteria. By default, empty subfolders are not deleted, after all files are moved out. This is done by design. Many production systems may need these empty subfolders to remain in place. This is for other processes to move new files into these empty subfolders.
If you need to delete empty subfolders after the Files have been moved, select this option. Please note that all empty subfolders in the source folder will be deleted. This is true even if there were no files in those subfolders, at the start of the move option.

Check for critical folder

This option is added to prevent accidentally processing files within critical folders. Many of our users use dynamic variables for the source and target folders. Sometimes, inadvertently, the dynamic result could be c:\ or / etc which may result in undesirable consequences.
The folders checked include:
1) Root folders (c:\ or d:\ or / etc..)
2) User home folder
3) Folders in the system library path.

If you have to use these folders (c:\ or d:\ or / etc..), and the folder filter or filename filter, you can disable this option if you are absolutely sure that no critical system or data files will be adversely effected..

CSV Loop

This task allows you to loop through data from a CSV file. You can either loop through all cells or loop by rows.

With every loop, the current cell values are set as variables.

If you loop by cell, the cell value is returned as a variable.
If you by rows, each columns cell values in the row is returned as variables. Also, the entire row is saved as a variable.

With every loop, you can run another task, and use the saved variables in that task.

CSV Read

This task allows you to read data from a CSV file. You can either read a certain cell or an entire row.

If you read a cell, the cell value is returned as a variable.
If you read a row, each columns cell values in the row is returned as variables. Also, the entire row is saved as a variable. You can then use these variable values in other tasks.

CSV Write

This task allows you to write data to a CSV file. You can either write a certain cell or an entire row. You can use this task to write a variable value into a cell.

Database SQL

You should test your database connection and SQL statements using the Database Test utility, before you make or schedule SQL tasks.

SQL statement
Enter your SQL statement here. It can be a select, insert, update, or delete statement. This field also supports dynamic variables.

Directory to place Results
Enter the full pathname of the directory where you need to put the SQL query results. Note that the pathname is case sensitive on Unix systems.

Results Filename
Enter the filename for query results. If no filename is entered, a default filename of your Task Title is used. This field also supports dynamic variables. Using dynamic variables, you can save the file to a customized name based on date/time or username etc...

Append to Filename
To save each query to a different filename, based on current time/date, choose an append file option.

Database JDBC driver
For your database, you need to obtain the JDBC driver from your database vendor or other 3rd party vendor. Many of the database vendors have free JDBC drivers available at their web site. The JDBC driver is generally in the form of a .zip or .jar file. **You need to place the JDBC driver in the following folder, depending on your installation method. Then restart the engine and the user interface.**
For installs including JVM : ...[Install_Folder]/jre/lib/ext
For installs without JVM : ...jre/1.3/lib/ext or JDK1.3/jre/lib/ext.

Driver class name
Enter the class name for the driver. This information will be available in the driver documentation. For Microsoft Access database, simply enter "access". Also, the JDBC driver required for MSAccess is included within Java itself.

Database Url
Enter the Url connection information. This information will be available with the Java JDBC driver documentation. The Url format will vary with vendor. The format includes the server name, port, database name, username, and password.

Authorization
If your database requires authentication, then select this option and enter your username and password.

Username
Enter your username if required for database access. It is case sensitive.

Password
Enter your password if required for database access. It is case sensitive.

Database Stored Procedure

You should test your database connection and SQL statements using the Database Test utility, before you make or schedule stored procedure tasks.

Stored Procedure Name
Enter the name of the stored procedure. This field also supports dynamic variables.

Stored Procedure parameters
Enter the parameters for the stored procedure separated by , . The parameters should be entered in the same order that they are declared in the database stored procedure. Leave this field blank if your stored procedure does not require parameters. This field also supports dynamic variables.

Example:
Consider the stored procedure:
addNameAge (IN name varchar(25), IN age varchar(3))
To: pass the values for Jim aged 75, you would enter for the parameters field:
Jim,75

Directory to place Results
Enter the full pathname of the directory where you need to put the SQL query results. Note that the pathname is case sensitive on Unix systems.

Results Filename
Enter the filename for query results. If no filename is entered, a default filename of your Task Title is used. This field also supports dynamic variables. Using dynamic variables, you can save the file to a customized name based on date/time or username etc...

Append to Filename
To save each query to a different filename, based on current time/date, choose an append file option.

Database JDBC driver
For your database, you need to obtain the JDBC driver from your database vendor or other 3rd party vendor. Many of the database vendors have free JDBC drivers available at their web site. The JDBC driver is generally in the form of a .zip or .jar file. **You need to place the JDBC driver in the following folder, depending on your installation method. Then restart the engine and the user interface.**
For installs including JVM : ...[Install_Folder]/jre/lib/ext
For installs without JVM : ...jre/1.4/lib/ext or JDK1.4/jre/lib/ext.

Driver class name
Enter the class name for the driver. This information will be available in the driver documentation. For Microsoft Access database, simply enter "access". Also, the JDBC driver required for MSAccess is included within Java itself.

Database Url
Enter the Url connection information. This information will be available with the Java JDBC driver documentation. The Url format will vary with vendor. The format includes the server name, port, database name, username, and password.

Authorization
If your database requires authentication, then select this option and enter your username and password.

Username
Enter your username if required for database access. It is case sensitive.

Password

Enter your password if required for database access. It is case sensitive.

Database Test

This is the most difficult task to implement correctly. Please read this entire documentation very carefully.

This utility helps you to test your connection to a database. It connects to your database using an intermediate JDBC (Java Database Connectivity) driver. A database vendor or third party vendor provides this driver. **Detailed JDBC driver and connection information for popular databases like MSAccess, Oracle, Informix, MySql, mSql, Postgres are included in our knowledge base. The knowledge base is online at www.hiteksoftware.com**. Go to our main support page and use the knowledge base link. Our knowledge base has information on these databases, and other databases provided by our users.

Detailed information on JDBC drivers, that are available for most databases, is provided by www.javasoft.com. Please read the JDBC driver documentation very carefully. The driver documentation will specify the values for the Driver Class Name and Database Url.

Database JDBC driver
For your database, you need to obtain the JDBC driver from your database vendor or other 3rd party vendor. The commercial database vendors have free JDBC drivers available at their web site. Legal issues may prevent us from redistributing some of these free JDBC drivers, in our software or, at our web site.

The JDBC driver is generally in the form of a .zip or .jar file. **You need to place the JDBC driver in the following folder depending on your installation method. Then restart the engine and the user interface.**
For installs including JVM : ...Automize/jre/lib/ext
For installs without JVM : ...jre/1.3/lib/ext   or JDK1.3/jre/lib/ext etc.. depending on your system

Driver class name
Enter the class name for the driver. This information will be available, in the driver documentation. For Microsoft Access database, simply enter "access". Also, the JDBC driver required for MSAccess is included within java itself.

Database Url
Enter the Url connection information. This information will be available, in the Java JDBC driver documentation. The Url format will vary with vendor, but generally includes the server name, port, and database name, username, and password.

Authorization
If your database requires authentication, then select this option and enter your username and password.

Username
Enter your username if required for database access. It is case sensitive.

Password
Enter your password if required for database access. It is case sensitive.

Debugger

The debugger is used to help us solve bugs/issues directly on the users computer.

You should NOT user this feature yourself.
When required Hitek Software will provide you with the script.
Paste the script into the top window of the debugger. (Right mouse click / Paste)
Click the Test Script button.
The script output will be displayed in the lower pane of the debugger.

Decrypt Files

The Decrypt task allows you to decrypt files from a directory or from a directory tree.
Important: Normally an encrypted file has an extension like .pgp or .gpg at the end. To remove this extension, you need to Fill BOTH the Filename and Rename Filename fields. Please see the example in the Rename Filename section.

Profile

Select the decryption profile that will be used to decrypt the file (or files). You should create a decryption profile before creating a decryption task. Decryption profiles for password decryption, or for private key decryption are supported. This task also supports signature verification using public key.

Source Directory

Enter the full pathname of the directory where your source files exist.

Filename

Enter the filename filter for the files you wish to decrypt using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time. The above wildcard rules only apply for the file to pass the filename filter. If the file passes the filename filter, then its new name will depend on the Rename Filename field.

Rename Filename

It is recommended that you always fill in this field for the decrytion task. Please see Rename Help.

Examples:

Actual Filename = test.txt.gpg
Filename field entry = .txt.gpg
Rename Filename field entry = .txt
Resulting Decrypted Filename = test.txt

Actual Filename = test.txt.gpg
Filename field entry = .gpg
Rename Filename field entry = (blank)
Resulting Decrypted Filename = test.txt

Actual Filename = test.txt.gpg
Filename field entry = test.txt.gpg
Rename Filename field entry = (blank)
Resulting Decrypted Filename = test.txt.gpg

Actual Filename = test.txt.gpg
Filename field entry = test.txt.gpg
Rename Filename field entry = test.txt
Resulting Decrypted Filename = test.txt

Folder Path Filter

Leave this field blank if you want all folders to be processed. If you need specific folders to be processed, then enter the folder name or path, using the following wildcard rules. If a folder passes the filter, then all its subfolders will also pass the filter.
Simple Example: Folder = c:\test , subfolders = john, john\old, sam, sam\old, joe, joe\old
Folder path filter = john . john and john\old are processed
Folder path filter = jo . john, john\old, joe, joe\old are processed
Folder path filter = old . sam\old, john\old, joe\old are processed

Folder path filter = test . all files in c:\test and its sub folders are processed
Folder path filter = john\old. only john\old is processed.
Folder path filter = c:\test\john . john and john\old are processed

Target Directory

Enter the full path name of the directory where you need to put the target files.

Place files in new subdirectory

If you select "None", no new subdirectory is created within the Target Directory.
If you select Date, a new subdirectory based on the date, is created within the Target Directory. All files are placed in this directory.
If you select Date/Time, a new subdirectory based on date and time is created within the Target Directory. All files are placed in this directory.

Include subdirectories

If you select this option, all subdirectories and decrypted files (within the Source Directory) are created in the Target Directory. If a subdirectory does not exist within the Target directory, then it is automatically created.

Date Filter

If you need to filter files based on file modified date, select this option. For the between option, enter 2 values separated by '-' (2-4 etc.). Examples:
Older than 5 Minute, Newer than 2 Day, Between 3-5 Day, Between 1-4 Hour

Delay

This task delays for the entered time in milliseconds (1000 ms = 1 second).
1) This is useful in the Chain tasks if you need to put a delay between tasks.
2) This is also useful in the monitoring tasks when you need to launch more than 1 tasks in series. You can add a delay task between consecutive tasks.

Delete Files

The Delete task allows you to Delete file (or files) within a single directory.

Directory
Enter the full pathname of the directory where your files exist.

Filename
Enter the filename filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Date Filter
If you need to filter files based on file modified date, select this option. For the between option, enter 2 values separated by '-' (2-4 etc.). Examples:
Older than 5 Minute, Newer than 2 Day, Between 3-5 Day, Between 1-4 Hour

Folder Path Filter
Leave this field blank if you want all folders to be processed. If you need specific folders to be processed, then enter the folder name or path, using the following wildcard rules. If a folder passes the filter, then all its subfolders will also pass the filter.
Simple Example: Folder = c:\test , subfolders = john, john\old, sam, sam\old, joe, joe\old
Folder path filter = john . john and john\old are processed
Folder path filter = jo . john, john\old, joe, joe\old are processed
Folder path filter = old . sam\old, john\old, joe\old are processed
Folder path filter = test . all files in c:\test and its sub folders are processed
Folder path filter = john\old. only john\old is processed.
Folder path filter = c:\test\john . john and john\old are processed

Check for critical folder
This option is added to prevent accidentally processing files within critical folders. Many of our users use dynamic variables for the source and target folders. Sometimes, inadvertently, the dynamic result could be c:\ or / etc which may result in undesirable consequences.
The folders checked include:
1) Root folders (c:\ or d:\ or / etc..)
2) User home folder
3) Folders in the system library path.

If you have to use these folders (c:\ or d:\ or / etc..), and the folder filter or filename filter, you can disable this option if you are absolutely sure that no critical system or data files will be adversely effected..

Directory Change

The Directory Change task monitor polls a single directory (or file) for changes. The Directory Change task can be used in three ways:
1) If a change is detected in this task, you can run another task and exit.
2) You can set this task to run as a daemon, which continuously polls the directory/file for changes, and keeps triggering another task. This is very useful in production environments, where files are continuously moving into a folder. For example, when a change is detected, you can continuously launch a file processing script. Then continue polling for changes.
3) The Directory Change task can also be embedded within a Chain task. The Directory Change task is useful for conditional task processing. If you want to use this task in a chain for conditional processing, select 'None' as the 'Task To Run'. If the criteria is satisfied, then an exit code = -100 is thrown. Else, an exit code = 0 is thrown.

IMPORTANT: This task only detects changes while the task is running. So, if you start/restart the engine at 10:00 AM, and if your next DirChange task is at 10:30 AM, any directory changes between 10:00 AM - 10:30 AM will go undetected. This task is best used on servers, where you do not have to restart the engine often.

Directory

Enter the directory or file you need to monitor.

Task To Run

Select the desired task to run if any of the selected criteria are met. If you want to use this task in a chain for conditional processing, select 'None'. You can also select multiple tasks, to run in sequence or, to run simultaneously. To run tasks in sequence, use taskTitle1|taskTitle2|taskTitle3. To run tasks simultaneously, use taskTitle1&taskTitle2&taskTitle3.

Maximum wait time

Set the maximum allowed time to wait for a change to occur (**T1**).

Polling Interval

Enter the polling interval (in seconds). The default value of 60 seconds is recommended. You can choose a smaller interval if you need to.

Use as Daemon

Select this option if you need to continuously poll a directory for changes and trigger another task to run. This demon option should **not** be used when this task is running in a chain.
To monitor a directory continuously, do the following:
Set the maximum wait time (**T1**) for the task = 60 minutes. Schedule this task to run once an hour. i.e. when the previous schedule ends, a new schedule is started.

Using Directory Change Task in a Chain

- Create a "DirChange" task with the directory or the file you need to monitor for changes.
- Create a Chain Task with the above "DirChange" task as the 1st step.
- Set the maximum wait time (**T2**) to be greater than **T1** above.
- Use the Exit Code criteria as (if Exit Code != -100 Go to Step 10) i.e. stop chain if exit code != -100
- Add a second task after this step.
- Set this Chain to run at time = X.
- Set another task type like (ftp, program etc.) to run at time = X+1 minutes
- On successful completion, the other task type (ftp, program etc.) should generate/copy/move/delete a file in the desired directory.
- If a directory change is detected, the DirChange task will exit with Exit Code = -100 and the Chain will run the

2nd task.

- If the task wait time (**T1**) is exceeded, the DirChange task will exit with Exit Code != -100 and the Chain will terminate without running the 2nd task.

Difference between Directory Change and Directory Monitor Tasks

Directory Change task keeps polling the directory for the maximum time set. When the maximum time expires, or a change is detected, this task either exits, or continues polling as a daemon. This task can only monitor a single directory, or a single file.

Directory Monitor task stores (in a log file), the modified date information for all files within the directory. This log file is read, and updated, with every run. The Directory Monitor task compares the modified date (stored in the log file) with the actual modified date of the file. This task can also monitor sub directories.

Directory Listing

This task allows you to list the contents of a directory into an array variable for use in logic For or While tasks, or Array loops. This task is mainly useful for Automize, since it has logic and array support. However, since this is a useful debugging task, it is availabe in AbleFtp, JaSFtp, and AutoKrypt also.

Directory Loop

The Directory loop task allows you to monitor a directory based on filename, file size or file date. For every file that satisfies the criteria, the Directory loop task will run another task. This other task can be a processing task that works on the current file. See the example at the end of this help file on how to use this task.

The Directory Loop task sets the following variables for use in the other tasks:
CurrentFilename, CurrentFilesize, CurrentFilepath

Directory
Enter the directory where your files are located.

Filename
Enter the filename filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Task To Run
Select the desired task to run, if any of the selected criteria are met. If you want to use this task in a chain for conditional processing, select 'None'. You can also select multiple tasks to run in sequence, or to run simultaneously. To run tasks in sequence, use taskTitle1|taskTitle2|taskTitle3. To run tasks simultaneously, use taskTitle1&taskTitle2&taskTitle3.

Maximum loops
This is the maximum number of files that will be processed in each run. If 100 files pass the criteria, you can process the first 5 files in the list. Enter 0 to process all files that pass the criteria.

Delay between loops
You should enter a delay time between loops. The delay must be long enough to ensure the processing task is completed before the next instance of the processing task is called.

Options
The Task will be run, if the selected criteria is satisfied. You can select 1 criteria.
1) File (or files) which satisfy the Filename criteria exists
2) File (or files) which satisfy the file size criteria. The file size is entered in kb, but you can specify 0.001 to 1000000 kb.
3) File (or files) which satisfy the date modification criteria.

Example
Create the following 2 tasks:

1) Echo task
Task Title = echoTest
message = The current filename is $%loop::CurrentFilename%$

2) Directory Loop
Task Title = loop
directory = [temp or test directory]
filename =
Maximum loops = 5
Delay = 15
Task To Run = echoTest
Criteria = Run Task if File exists

Run the Directory Loop task. The directory loop task will run the Echo task for the first 5 files in the directory.

In real situations, you would use either a copy task or a delete task etc.. For the filename field in these tasks, use:
$%loop::CurrentFilename%$

Directory Monitor

The Directory Monitor task is a special type of task which can be used in two ways:
1) If a change is detected in this task, you can run another task, or
2) The Directory Monitor task can also be embedded within a Chain task, and is useful for conditional task processing. If a change is detected, then an exit code = -100 is thrown. Else, an exit code = 0 is thrown.

The Directory Monitor task also outputs a variable called 'TaskTitle::Filenames". This variable contains the list of all files that have been modified, since the last time this task was run. The format of this variable is #file1^#file2^#file3, where file1, file2, file3, are the names of the files, that have been modified. You can use this variable directly as the filename in other copy tasks, ftp tasks, or delete tasks, to process only these files.

Directory to Monitor
Enter the directory or file you need to monitor.

Include Subdirectories
To monitor all sub directories, select this option

Filename
Enter the filename filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Task To Run
Select the desired task to run if any of the selected criteria is met. If you want to use this task in a chain for conditional processing, select 'None'. You can also select multiple tasks to run in sequence, or to run simultaneously. To run tasks in sequence, use taskTitle1|taskTitle2|taskTitle3. To run tasks simultaneously, use taskTitle1&taskTitle2&taskTitle3.

Using Directory Monitor Task in a Chain

- Create a Directory Monitor task with the directory or file you need to monitor for changes.
- Create a Chain Task with the above "Directory Monitor" task as the 1st step.
- Use the Exit Code criteria as (Exit Code != -100 Go to Step 10) i.e. stop chain if exit code != -100
- Add a second task after this step.
- If a directory change is detected from comparison with the previous run of this task, the Directory Monitor task will exit with Exit Code = -100 and the Chain will run the 2nd task.
- If no change is detected, the Directory Monitor task will exit with Exit Code = 0 and the Chain will terminate without running the 2nd task.

Difference between Directory Change and Directory Monitor Tasks
Directory Change task keeps polling the directory for the maximum time set. When the maximum time expires or a change is detected, this task either exits or continues polling as a daemon. This task can only monitor single directory.

Directory Monitor task stores (in a log file), the modified date information for all files within the directory. This log file is read and updated with every run. The Directory Monitor task compares the modified date (stored in the log file) with the actual modified date of the file. This task can also monitor sub directories.

Echo

This task adds user defined messages or comments to the logs.

Email

The email task allows you to send email automatically to multiple recipients. You can also attach files.

Email Server Profile
Select the profile name for the email server. You should have previously created an email profile, using the Email Profiles menu item.

Adding a recipient to the current email message
You can add a recipient in 2 ways.
1. Enter the email address directly into the recipient text box. Click the New button to add the recipient to the recipient list.
2. Select the recipient in the address book, then select the recipient type (To:, cc:, bcc:); and click the paste button. This will add the recipient to the recipient list box.

Adding a recipient to the email address book
You can add a new recipient to the email address book, using the New button. Enter the desired name and email address. You have to press enter, for any changes in the address book to be validated.

Subject
Enter the subject. This field also supports dynamic variables. For example, you could automatically parse the current date, or computer IP address, into the subject.

Message Body
1) Enter the message body directly into the text field.
2) This field also supports dynamic variables. For example, you could automatically parse the current date, or computer name, into the message.
3) You can let the task dynamically parse the message text from a file on your system using the following format:
$$$[message_file_path].
for example on windows, if the message file is c:\data\message.txt, use:
$$$c:\data\message.txt
NOTE:: The message body field has to start with $$$[message_file_path] AND you cannot use any other text in the message body.

Adding Attachments
1) To add an attachment, type in the full path name of the file into the Attachment Text Box, or use the "Browse" button to select the file. The format for the attachments is: AbsoluteFilePath1,AbsoluteFilePath2,......
Example: c:\data\test.txt,c:\temp\data.htm

2) You can also specify multiple filenames in multiple directories, by using the following format:
[DIR]c:\your_dir[FILE]name_filter
Enter the name_filter for the files you wish to transfer, using the following wildcard rules
To attach all files that end with .txt from the c:\temp folder, use:
[DIR]c:\temp[FILE].txt

3) This field also supports dynamic variables.
Example:
To attach all files in the c:\temp folder, which has the current date (example: 06-21-02) in their filename, use:
[DIR]c:\temp[FILE]$%DATE::pp-dd-yy%$
The date format in the filename has to match the date format entered above. See dynamic variables for more details

Append to Filename
You can append the date/time or a customized date to the attachments based on current time/date, or custom code.

Headers

You can specify headers to add to the email. This will allow you to add mail priority, or other specific flags. An email server may not display all the headers that you specify, the server may not understand the headers you specify. For example, if you attempt to specify Email priority level, an email server may accept any, or all, or none of the following flags: X-Priority, X-MSMail-Priority, Importance, Priority.

To specify headers, use the following syntax:
header1=value1^header2=value2^header3=value3

Header Examples:
X-Priority=1^ Priority=Urgent
Importance=high
Reply-To=test@test.com

Email Logs

This administrative task will send you an email with the output log and activity log attached. This is useful if you deploy the software on a remote machine.

Email Server Profile
Select the profile name for the email server. You should have previously created an email profile by using the Email Profiles menu item.

Email Recipients
Enter the email address(or addresses) to receive the email message. Separate multiple addresses by a comma ,.

Email Notification

The software can automatically send you an email notification message if your tasks fail. The output log is attached to the email message.

Email Server Profile
Select the profile name for the email server you need to connect to. You should have previously created an email profile using the Email Profiles menu item.

Selecting Exit Code Criteria
1. Generally, if your tasks fail, the Exit Code > 0.
2. Many tasks also sets an exit code of -100 or -101, if a change is detected during monitoring tasks such as: ftp, local or remote directory monitor. If you also want to get a message when an exit code of -100 is obtained, you should select the Exit Code != 0 criteria.

Email Recipient
Enter the email address to receive the email message. To send notification to multiple recipients, separate email addresses by a comma ',' .

Email Subject
If the email subject starts with [DEFAULT], then the default email notification subject is used. You can enter a customized subject based on special variables as follows:
Task_Title = task title for the task
Task_Type = task type
Exit_Code = exit code for the task
Additionally, you can use dynamic variables like $%IPADRESS%$ or $%SYSNAME%$

Example subject input = The Task_Type task, Task_Title failed with exit code = Exit_Code
resulting subject = The Copy Directories task, copydata failed with exit code = 3

Email Message
If the email message starts with [DEFAULT], then the default email notification message is used. You can enter a customized subject based on special variables as follows:
Task_Title = task title for the task
Task_Type = task type
Exit_Code = exit code for the task
Additionally, you can use dynamic variables like $%IPADRESS%$ or $%SYSNAME%$

Example Message Input:
The Task_Type task, title = Task_Title, failed with exit code = Exit_Code.
Host = $%SYSNAME%$
IP Address = $%IPADDRESS%$

Resulting message:
The Copy Directories task, copydata failed with exit code = 3.
Host = win3
IP Address = 192.168.0.3

Headers
You can specify headers to add to the email. This will allow you to add mail priority, or other specific flags. Note that a mail server may not display all the headers you specify, or the mail server may not understand the headers you specify. For example, if you attempt to specify Email priority level, an email server may accept any, or all, or none of the following flags: X-Priority, X-MSMail-Priority, Importance, Priority.

To specify headers, use the following syntax:
header1=value1^header2=value2^header3=value3

Header Examples:
X-Priority=1^ Priority=Urgent
Importance=high
Reply-To=test@test.com

Email Profiles

Email Server Name
Enter the name of your SMTP server. Check your email client software for this value. Look for the outgoing server or smtp server name.

Email Server Port
The default SMTP email port is usually 25.
For secure email using TLS (example: GMail), the port used is 587.
For secure email using SSL (example: Yahoo mail), the port used is 465.

Sender Name and Sender Email
Enter your email sender name and email. The email address should be a valid email address on the smtp server.

SMTP Authentication
Some SMTP servers require user authentication to send email. This is a good method to prevent unsolicited email relaying and spam. Please talk with your SMTP server admin, regarding the type of authentication required. We provide the following forms of authentication:
smtp : sends your username and password information, when the SMTP connection is made.
java : sends your username and password information, when requested by server.
pop : first logs into your pop3 email server, and then sends the email via the SMTP server.
tls : sends your email credentials and message securely using TLS. Example: GMail
ssl : sends your email credentials and message securely using SSL. Example: Yahoo mail

Email Support Generator

This feature allows you to send your data files to Hitek Software for analysis. This helps us to recreate your tasks, schedule, and settings on our test systems. It also includes output logs and error logs, and helps us to debug your issue very quickly.

You can also add individual task logs to the data files that you send to us. Select the task logs to add, using the task logs button. You can select multiple task logs to attach from the list, using the CTRL button. You can also manually enter the task title in the task logs field. You can also enter multiple task titles separated by ','. Example: task1,task2 etc. Do not leave any white space after the comma ','.

1) **ZIP button:** This will create a zip file of all your relevant data. Please email this zip file to us using your own email client.

2) **EMAIL button:** If you are setup to send email via our software, click on the EMAIL button. This will automatically create a zip file of your data, and email it to us.

Encrypt Files

The Encrypt task allows you to encrypt files from a directory, or a directory tree.

Profile

Select the encryption profile that will be used to encrypt the file (files). You should create an encryption profile, before creating an encryption task. Encryption profiles for password encryption, or public key encryption are supported. This task also supports private key signing.

Source Directory

Enter the full pathname of the directory where your source files exists.

Filename

Enter the filename filter for the files you wish to encrypt using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Folder Path Filter

Leave this field blank if you want all folders to be processed. If you need specific folders to be processed, then enter the folder name or path, using the following wildcard rules. If a folder passes the filter, then all its subfolders will also pass the filter.

Simple Example: Folder = c:\test , subfolders = john, john\old, sam, sam\old, joe, joe\old
Folder path filter = john   . john and john\old are processed
Folder path filter = jo   . john, john\old, joe, joe\old are processed
Folder path filter = old   . sam\old, john\old, joe\old are processed
Folder path filter = test   . all files in c:\test and its sub folders are processed
Folder path filter = john\old.   only john\old is processed.
Folder path filter = c:\test\john   . john and john\old are processed

Append to Filename

It is recommended that you add an extension for the encrypted filename. For PGP encryption, add .pgp as the extension. For signing, normally .sig is appended. If you regularly encrypt a file, and wish to save each encryption to a different filename based on current time/date, choose an append file option.

Target Directory

Enter the full pathname of the directory, where you need to put the target files. Note that the path is case sensitive on Unix systems.

Place files in new subdirectory

If you select "None", no new subdirectory is created within the Target Directory.

If you select Date, a new subdirectory based on the date, is created within the Target Directory. All files are placed in this directory.

If you select Date/Time, a new subdirectory based on date and time, is created within the Target Directory. All files are placed in this directory.

Include subdirectories

If you select this option, all subdirectories and encrypted files within the Source Directory, are created in the Target Directory. If a subdirectory does not exist within the Target directory, then it is automatically created.

Date Filter

If you need to filter files based on file modified date, select this option. For the Between option, enter 2 values separated by '-' (2-4 etc.). Examples:

Older than 5 Minute, Newer than 2 Day, Between 3-5 Day, Between 1-4 Hour

Encryption Tutorial

This very brief tutorial is for beginners who have not used encryption tools before.

Introduction
There are two types of encryption methods: symmetric and asymmetric.

1) **Symmetric Password Based Encryption** - This is the simplest form of encryption. It is a symmetrical encryption method. The same password is used to encrypt and decrypt the data (or the file). This method is useful to encrypt sensitive information for yourself, or for family, or for a few trusted friends or coworkers.

2) Symmetric Secret Key Based Encryption
This is the simplest form of key based encryption. It is a symmetrical encryption method. The same secret key file is used to encrypt and decrypt the data (or the file). This is not a very commonly used technique.

3) Assymetric public/private Key Based Encryption
A public key file is used to encrypt the data. The corresponding private key file is used to decrypt the data. Only you should have access to your private key. You can distribute your public key to anyone who needs to send you data. This is the technique that is most commonly used in corporations.

OpenPGP encryption
This is the most popular encryption method. Many corporations employ OpenPGP encryption to exchange or store sensitive data.

OpenPGP keys
To get started you would need to create an OpenPGP Key pair. The private key should be stored carefully. The public key should be distributed to your friends, collegues, customers or clients etc..

Encrypting and Decrypting files
Your customers or clients can use your public key to encrypt a file.. Then email the encrypted file to you. The encrypted file can only be decrypted by your private key. If the email is intercepted, no one will be able to decrypt the file since they do not have your private key. Similarly if you need to send a file to a client, you should use the clients public key to encrypt the file. Only your client has his private key to decrypt the file.

Signing and Verifying files
To confirm that you have sent the file, you should use your private key to sign the file. Your clients can then use your public key to verify that the file was sent by you.

Encrypt & Sign
OpenPGP software normally support one step encrypt and sign. This allows you to encrypt a file using the recipients public key. At the same time, this also signs the file with your private key. When the recipient receives this file, he can use his private key to decrypt the file, and use your public key to verify that it was indeed sent by you.

Key Generator

This feature allows you to generate keys (key files) that can be used for encryption/decryption.

OpenPGP Private/Public Key pair
The public key is normally used to encrypt a file; and the private key is used to decrypt a file. This feature only generates the OpenPGP private and public keys. The OpenPGP key files normally have the extension '.asc'. This feature does not generate the OpenPGP secret and public key rings.

Key Sizes
1) DSA signing keys can only be 1024

2) RSA signing keys can be 1024-4096

3) DSA-ELGAMAL Sign and Encrypt keys
DSA signing part can only be 1024
El Gamal encryption part can be 1024-4096

Encryption Profiles

This feature allows you to create encryption or decryption profiles. An encryption profile should be used in an encryption task to encrypt files. A decryption profile should be used in a decryption task to decrypt files. There are various types of encryption methods supported.

OpenPGP Password Based Encryption & Decryption
This is a symmetrical encryption/decryption method, i.e. the same password is used for encryption and decryption. Two separate profiles are needed, one encryption profile and another decryption profile. Files encrypted using this method can be decrypted by this application and by other applications that support OpenPGP. Similarly, files encrypted in other applications using OpenPGP password based encryption can be decrypted in this software.

OpenPGP Public Key Encryption
This is an asymmetrical encryption method. You should create an encryption profile based on a Recipients Identity (using his OpenPGP public key file). Files encrypted using this OpenPGP public key can only be decrypted by the Recipient using his own private key.

OpenPGP Private Key Decryption
This is an asymmetrical decryption method. You should create a decryption profile based on an OpenPGP private key file. This OpenPGP private key can only decrypt files which were encrypted using its corresponding public key.

OpenPGP Private Key Signing
This method is used to sign a file using a private key. During signing, the file is encrypted with a Senders private key. The corresponding OpenPGP public key can be used by anyone to verify the Senders signature.

OpenPGP Public Key Verification
This method uses a public key to verify a signed file. The Senders public key is used to verify that the file was signed by the Sender. This method does not automatically decrypt and output the original file. Use a Decrypt profile to automatically decrypt the original Signed file.

Example of Encrypt and Decrypt:
John's customers need to send John important files securely via email.
Only John should be able to read these messages.
If these files are intercepted by a malicous 3rd party, they should not be able to decrypt/read these files.
To do this, John creates a Private & Public key pair.
John keeps his Private key securely to himself.
John distributes his Public key to his customers.
John's customers then encrypt these files using John's Public key.
John's customers send these files to John via email.
Only John can decrypt these encrypted files because only he has the Private key

Example of Sign and Verify:
John's needs to send all his customers very important secret messages (in text files).
His customers need proof that these messages are indeed from John.
To do this, John creates a Private & Public key pair.
John keeps his Private key securely to himself.
John distributes his Public key to his customers.
John then signs his secret messages using his Private key.
John sends these signed messages via email to all his customers.
His customers use his Public key to verify (and decrypt) John's messages.

Key Store

This feature allows you to manage, import and export keys into the Key Store or from the Key Store. You need to import keys into the Key Store before you can use them to create encryption profiles and tasks.

Key Types
1) DSA / EL Gamal - this is the common key type used for Encryption and Decryption. It can also be used for 'Encrypt and Sign' and 'Decrypt and Verify' methods. The DSA key is the signing key and is fixed at 1024. The EL Gamal sub key is the encryption key and the size can be varied from 1024 - 4096.

2) RSA - used for signing only

3) DSA - used for signing only

Import Keys
You can import existing keys into the key store. **Once you import a public key, you need to Set the keys Trust level to Ultimate. To do this, right mouse click on the key row. Select "Edit Trust" and follow instructions.**

Export Keys
You can select and export keys from the key store to any folder on your system.

Encryption Panel

The encryption panel contains the following:
1) File explorer pane - Allows a user to manually encrypt (or decrypt) files.
2) Encryption settings pane - Allows a user to create encryption profiles, generate keys, or manage keys.

File Explorer Pane
This pane contains the favorite encryption folders. You can add local or network folders to this pane. You can select a file (or files) or folder to encrypt (or decrypt) manually. When you click on the Encrypt (or Decrypt) button, an options dialog will appear. You can select the profile, and other options to be used for encryption (or decryption).

Encryption Settings Pane
This pane contains all encryption settings in various tabs.
The Encryption Profile tab allows you to create and modify profiles. You can use these profiles in your encryption (or decryption) tasks.
The Key Generator tab allows you to generate various types of keys used in encryption.
The Key Manager tab allows you to import, export and manage your encryption keys.

Encryption Settings - Favorites Tab

This tab allows you to specify favorite folders
1) You can add local folders or network folders to this pane using the 'Add Favorite' button.
2) You can remove a favorite folder using the 'Remove Favorite' button.

You can select a file(s) or folder to encrypt (or decrypt) manually, or create an encryption (or decryption) task. When you click the Encrypt (or Decrypt) button, an options window will appear. You can select the profile, and other options, to be used for encryption (or decryption).

Windows ONLY: The View button allows you to view a file using the default file viewer in Windows. If you select a Folder, then the folder is opened in Windows Explorer.

Exit Codes

The program attempts to obtain exit codes for your tasks. If your task fails, the task exit code > 0. In this case, please view the task and output logs for the cause of the error.

If a task exits normally without generating any error, Exit Code = 0. For some tasks which involve monitors or file downloads, the Exit Code = -100, if the tasks monitoring or download criteria are met.

EXIT CODE

DESCRIPTION	
0	No Error.
>0	Error occurred. Please check output log and or task logs for error details.
-100	Normally returned by monitoring tasks, if monitoring condition has been met.

Run task as external process

Normally, every task is run in a separate thread within the Engine. However, you can choose to run each task as a separate process outside the Engine.

Wait Time: Enter the wait time in seconds. Enter 0 to wait forever. If the wait time has been exceeded, the engine will terminate the external task process. If the engine is shutdown before the external task process completes or wait time is exceeded, then the external task process could run forever, unless it exits normally.

NOTE: Every task run as an external process will show up in the Windows Task Manager as 'javaw' and in the Unix/MacOSX process list as 'java'.

NOTE: You will see two entries for the Start Instance and End Instance. Each task run will be shown to take up 2 instances. The first instance corresponds to the external task runner which will launch the task as an external process. The second instance corresponds to the actual task. Example:

```
Aug 3, 2006 10:53:25 AM Alarm - alarm - Manual Run
Aug 3, 2006 10:53:25 AM Alarm - alarm - Start Instance = 1002
Aug 3, 2006 10:53:26 AM External Task Runner:: running task: alarm
Aug 3, 2006 10:53:26 AM Alarm - alarm - Manual Run
Aug 3, 2006 10:53:26 AM Alarm - alarm - Start Instance = 1003
Aug 3, 2006 10:53:37 AM Alarm - alarm - End Instance = 1003 , Exit Code = 0 , Runtime (ms) = 10875
Aug 3, 2006 10:53:41 AM Alarm - alarm - End Instance = 1002 , Exit Code = 0 , Runtime (ms) = 15187
```

File or Directory Properties

This task allows saves important information about a file or directory into variables for use in other tasks.

Properties that are saved into variables include timestamp components, size, and permissions.

File or Directory Permissions

This task allows you to set the Read, Write, or Execute permissions of a file or directory.

You can specify if the permission to be set is only for you (Owner) or for everyone.

File or Directory Timestamp

This task allows you to set the timestamp. You have to specify the timestamp in the correct format. It is best to use the SHORT format. It is fairly difficult to enter the MEDIUM or LONG format correctly.

You should specify the format according to your locale. i.e. The format will be different if you are in the US or in France etc..

Examples of entering format in US:
SHORT - 05/17/10 9:14 AM
or using variables:
$%DATE::pp/dd/yy hhx:mmx am_pm%$

File Monitor

The File Monitor task is a special type of task which can be used in two ways:
1) If a change is detected in this task, you can run another task.
2) The File Monitor task can also be embedded within a Chain task, and is useful for conditional task processing. If the monitoring criteria is satisfied, then an exit code = -100 is thrown. Else, an exit code = 0 is thrown.

The File Monitor task also outputs a variable called 'TaskTitle::Filenames". This variable contains the list of all files that satisfy the monitoring criteria. The format of this variable is #file1^#file2^#file3, where file1, file2, file3 are the names of the files, that have been modified. You can use this variable directly as the filename in other tasks (such as: copy, ftp, delete), to process only these files.

Directory
Enter the directory where your files are located.

Filename
Enter the filename filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Task To Run
Select the desired task to run, if any of the selected criteria are met. If you want to use this task in a chain for conditional processing, select 'None'. You can also select multiple tasks, to run in sequence, or to run simultaneously. To run tasks in sequence, use taskTitle1|taskTitle2|taskTitle3. To run tasks simultaneously, use taskTitle1&taskTitle2&taskTitle3.

Options
The Task will run, if any of the following selected criteria are satisfied. You can select 1 or more criteria.
1) File (or files) which satisfy the Filename criteria exists
2) File (or files) which satisfy the Filename criteria, as well as file size criteria. The file size is entered in kb, but you can specify 0.001 to 1000000 kb.
3) File (or files) which satisfy the Filename criteria, as well as date modification criteria
4) Total number of files in directory is greater or less than a certain value. This option does not update any of the dynamic variables.
5) Total size of files in directory is greater or less than certain value. The total size is entered in kb, but you can specify 0.001 to 1000000 kb.

Using File Monitor Task in a Chain

- Create a File Monitor task with the directory and file you need to monitor for changes.
- Create a Chain Task with the above "File Monitor" task as the 1st step.
- Use the following Exit Code criteria: If Exit Code != -100, Go to Step 10. This will stop the chain task if exit code != -100
- Add a second task after this step.
- If a change is detected from comparison with the previous run of this task, the File Monitor task will exit with Exit Code = -100. The Chain will run the 2nd task.
- If no change is detected, the File Monitor task will exit with Exit Code = 0. The Chain will terminate without running the 2nd task.

Filename Selection

We provide two separate custom wildcard systems. Combining these two systems provides more flexibility than either the DOS or Unix wildcard systems.
System 1 - Custom wildcard -
System 2 - Regular Expressions (Regex) There are lots of Regex tutorials online. To use Regex, your filename should start with 'REGEXF=' or 'REGEXM='.

You can use either System 1 OR System 2 for the filename. Please note that System1 and System 2 have very little in common. The * , ! and ^ used in System 1 have completely different meanings in System 2 (Regex).

System 1 - Custom wildcard
1) Multiple sets of filenames can be selected using ^.
2) Each filename can contain 1 or more string tokens, separated by *.
3) Each * also stands for one character (or more).
4) Filenames containing all individual tokens entered, in the correct sequence, are selected.
5) The index of an individual token in the filename selected cannot be less than the index of that token in the filename entered.
6) The length of the filename selected should always equal or exceed the length of the filename entered.
7) If * (or multiple *) is at the end, **it is ignored**. However, the length of the filename selected, will equal or exceed the length of the filename entered.
8) For exact filename search, use # at start of filename. This ensures only one file with the exact name, will be selected.
9) To exclude filename (or filenames), use ! at the start of filename. The exclusion filename **should be followed by at least one other token** separated by ^. The exclusion filename (filenames) should also be listed before any normal filename. Examples: !.zip^* or !.zip^.txt or !.txt,!.zip,.htm,.gif
10) To specify a case insensitive token, use (?i) at the start of the filename. Example: (?i)test.txt

Consider a directory with the following files:
junk, test, temp, TEMP, temp1, temp.txt, temp.htm, Test, Temp.txt, teeempju.txt, activityLog.txt, debugLog.txt, test.zip, test.gif

Filename entered	Files Selected
	All files selected
*	All files selected
te	test, temp, temp1, temp.txt, temp.htm,teeempju.txt
ju	junk,teeempju.txt
T	Test, Temp.txt
.txt	temp.txt, Temp.txt, teeempju.txt, activityLog.txt, debugLog.txt
te*htm	temp.htm
te*emp	teeemp.txt
te*htm^debug	temp.htm, debugLog.txt
te*htm^te*emp^debug	temp.htm, teeemp.txt, debugLog.txt
temp*.txt	No files selected (rule 5)
***********	teeempju.txt, activityLog.txt, debugLog.txt (rule 6)
est**	No files selected (rule 6)
est*	test, Test (rule 6 and 7)

#temp	temp (rule 8)
!.zip^*	All files except test.zip (rule 9)
!.txt^temp	temp,temp1,temp.htm (rule 9)
(?i)temp	temp,temp1,temp.htm, TEMP, Temp.txt (rule 10)

System 2 - Regular Expressions (Regex)
Regex is a very large subject. It contains lots of syntax that cannot all be listed here. Please search the Internet for tutorials on Regex.

REGEXF method scans the filename to find the next subsequence that matches the pattern entered in filename field.
REGEXM method attempts to match the entire filename against the pattern entered in the filename field.

We recommend that you use the REGEXF method first.

Java Regex has the following defined metacharacters`([{\^$|)?*+.` These are special characters that can affect the way a pattern is matched. See the examples below for the use of these metacharacters.

Consider directories with the following sets of files:
bat.txt, cat.txt, rat.txt, hat.txt, mat.txt
1.txt, 2.txt, 3.txt, 4.txt, 5.txt, 6.txt, 7.txt, 8.txt, 9.txt
a.txt, b.txt ... to z.txt & A.txt, B.txt .... to Z.txt
aa.txt, aaa.txt, aaaa.txt
123, 123123, 123123123

Filename entered	Files Selected
REGEXF=.	All files selected. The metacharacter "." means "any character"
REGEXF=txt	all .txt files are selected
REGEXF=[rcb]at	bat.txt, cat.txt, rat.txt are selected characters within the [ ] metacharacters form a set
REGEXF=[^bcr]at	hat.txt, mat.txt are selected ^ within [ ] is the negation metacharacter. Hence, bat.txt, cat.txt, rat.txt are rejected
REGEXF=[0-4].txt	1.txt, 2.txt, 6.txt, 7.txt, 8.txt are selected. - is the range metacharacter.
REGEXF=[a-m[A-M]].txt	a.txt to m.txt, A.txt to M.txt are selected. [a-m[A-M]] is UNION of [a-m] and [A-M]
REGEXF=[a-m&&[d-z]].txt	d.txt to m.txt are selected. [a-m&&[d-z]] is INTERSECTION of [a-m] and [d-z]
REGEXF=[a-g&&[^cde]].txt	a.txt, b.txt, f.txt, g.txt are selected [a-m[^cde]] is SUBTRACTION of [a-m] and [cde]
REGEXF=(?i)a.txt	a.txt, A.txt are selected (?i) enables case-insensitive matching
REGEXF=a{3}.txt	aaa.txt, aaaaa.txt are selected a{3} looks for 3 or more a's in a row
REGEXF=[0-9]{1,}[^a-z]	123, 123123, 123123123 are selected [0-9] looks for numbers, {1,} means 1 or more numbers [^a-z] rejects all files with alphabets in them. ex: all .txt

REGEXF=(123){2}	123123, 123123123 are selected (xxx) implies a group. (123) looks for the 123 group, {2} means 2 or more groups of 123

Predefined character shortcuts:
`\d` matches all digits = `[0-9]`
`\D` matches non-digits = `[^0-9]`
`\s` matches spaces = `[ \t\n\x0B\f\r]`
`\S` matches non-spaces = `[^\s]`
`\w` matches word characters = `[a-zA-Z_0-9]`
`\W` matches non-word characters = `[^\w]`

Filenames to Array

This task allows you to convert a variable typically used in the Filenames field, to an array variable containing list of files, one per row.

Example: variable suitable for use in filename fields:
#test.txt^#test3.txt

Resulting Array:
test.txt
test3.txt

Filing Variables

This feature allows you to output the value of a variable to a local file on your system. You can then use this file in other tasks, or in your own programs. You can also insert the variable into various locations of a file, depending on the criteria you select. This feature can be useful in many situations.
Examples:
1) appending one file to another.

Variable name
Select the variable you need to monitor from the list. For some variables like DATE, you need to manually enter the format. In version 7.x, you need to enter the variable name surrounded by $%…%$. In Version 5.x, the surrounding $%…%$ was not required. By adding the $%…%$ into the format, it is possible to set the Variable Name field as an entire line of text which includes the variable.
Examples of Variable Name Field entries:
Today's date is $%DATE::qqq-dd_hh:mm%$
My username is $%USERNAME%$
This ip address = $%IPADDRESS%$

Filepath
Enter the full filepath of the file, which you want to save the variable to. This field supports dynamic variables.

Filing Options
Choose your filing criteria. You can insert the value of the variable into various locations. If you choose the 'Line Number' or one of the 'String containing' options, you should enter the correct 'Option Value' also.

Option Value
This Option Value depends on the Filing Option above.

Examples:
Variable Name = $%DATE::qqq-dd_hh:mm%$
Filing Option = Insert at Line Number
Option Value = 10
This will insert the date/time at line 10 in the file
or
Variable Name = $%DATE::qqq-dd_hh:mm%$
Filing Option = Insert before first line containing
Option Value = error
This will insert the date/time in the file, before the first line that contains the word 'error'

Types of variables listed:
Task Variables: Task variables values are generated / updated when the tasks run. Example: Tasks exit codes are stored as the following variable: TaskTitle::ExitCode
System Variables: System variables which are resolved when the current task runs, examples are DATE, USERNAME etc..
Java Variables: Java variables that are available to the Java Virtual Machine that the program runs in. You can get the value of Java variables from the Utilities menu / Java Variables
User Variables: Users can add their own variables via batch files, scripts or programs. Users can link output from their own programs to this program.

Front Panel

Table Description

The front panel contains the Tasks and Schedule lists. The Tasks list contains all the lists of tasks. The Schedule list contains scheduling data for the tasks that have been scheduled to run.

Run Software in the Background

Click the Program icon, and the user interface will run in the background. The icon will appear on your desktop. Clicking on the icon on your desktop will open the Front Panel again. You can set the icon to appear at various locations of your screen, by using the "Options" menu, "Icon Location" menu item.

Chain Tasks

Use this button to create a new chain task. Chain tasks can run other tasks in sequence.

Start / Stop Scheduler

If the button is green, the scheduler engine is running. If the button is red, the scheduler is shutdown. This button allows you to start the scheduler engine, or to shutdown the scheduler engine.

Status Bar

Status bar is at the bottom of the panel. The status bar provides information regarding user input.

Ftp

Before creating an Ftp task, we recommend that you use the FTP browser utility to get a connection. The ftp task allows you to transfer files from/to a server.

Ftp Profile

Select the ftp profile for the ftp server you need to connect to. You should have previously created an ftp profile, using the Ftp Profiles menu item.

Ftp Option

Use 'Get' to transfer files from the server to your local system. Use 'Put' to transfer files from your local system to the server. Use 'Delete' to delete files from the server.

Remote Directory

This is the most difficult parameter to enter correctly. We recommend that you use the FTP browser utility to determine the value of the full pathname of the remote directory. Execute privilege is required for the remote directory for all Ftp functions. Read privilege is required for Ftp Get. Write privilege is required for Ftp Put, and Ftp Delete functions. **If you are transferring files from the Default FTP login directory, you can leave this field blank.**

Local Directory

Enter the full pathname of the local directory, where you need to put or get files.

Backup to Local Directory

Enter the full pathname of the local directory, where you need to backup files. The Backup to local directory option does not support subdirectories. All files will be placed in the backup directory, without creating new subfolders in it.

Backup to Remote Directory

Enter the full pathname of the remote directory, where you need to backup files. The Backup to remote directory option does not support subdirectories. All files will be placed in the backup directory. No subfolders are created in the backup directory.

Use Staging

If you select this option, the file is first transferred to the local (get) or remote (put) folder entered. This folder acts as a staging folder. After the Ftp transfer is complete, the file is moved to the final destination folder. This option prevents processes from working on files, while the Ftp transfer is in progress. In the staging field, you should enter the full pathname of the final destination directory. The staging option does not support subdirectories. All files will be placed in the final destination directory. No subfolders are created in the final destination directory.

Filename

Enter the filename filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Folder Path Filter

Leave this field blank if you want all folders to be processed. If you need specific folders to be processed, then enter the folder name or path, using the following wildcard rules. If a folder passes the filter, then all its subfolders will also pass the filter.

Simple Example: Folder = c:\test , subfolders = john, john\old, sam, sam\old, joe, joe\old
Folder path filter = john   . john and john\old are processed
Folder path filter = jo   . john, john\old, joe, joe\old are processed
Folder path filter = old   . sam\old, john\old, joe\old are processed
Folder path filter = test   . all files in c:\test and its sub folders are processed
Folder path filter = john\old.   only john\old is processed.
Folder path filter = c:\test\john   . john and john\old are processed

Rename Filename

Please see Rename Help

Transfer Type

File transfer type can be either Ascii or Binary. Binary mode will transfer .exe, .jpg or other binary files. Binary mode should be the default transfer mode. Ascii mode can be used to transfer text or html files. If you transfer binary files in ascii mode, you will not get the entire file.

Append date/time To filename

You can choose to append either the date, or the date/time, or custom code. If your filename has an extension, the date or date and time is appended before the extension. This feature is useful for archive purposes.

Maintain Timestamp (for Get only)

The task can reset the transferred local files timestamp to match the remote files timestamp. This applies ONLY to the GET option.

Reset Local Timestamp (for Put only)

There is no Ftp standard to reset the remote files timestamp, to match the local files timestamp after an Ftp PUT. Hence, we provide the option to reset the source local files timestamp, to match the newly transferred remote files timestamp. There is a new proposal, to reset the remote file timestamp, using the MFMT (Modify file modification time) command. However, not many Ftp servers support the MFMT command. (Many Ftp servers incorrectly allow setting the remote file timestamp, using the MDTM command. This MDTM command is not an Ftp standard to set the remote file timestamp. The Ftp task does not reset the timestamp through this MDTM command.)

Delete Source file

You can choose to delete the source file, after a successful transfer. If the ftp transfer fails, the source file will not be deleted.

Transfer with temporary extension

The file is transferred with a temporary extension, and then renamed to the desired name, after the transfer is complete. This is useful if you have automated scripts/programs, which continuously work on transferred files with certain extensions. This ensures that the automated script will not work on a partially transferred file.

Log Transfer

You can choose to log the transfer into the Ftp Log.

Transfer Modified files

If you choose this option, files are transferred only if the source file is newer than the target file. You can specify an offset to be added to the remote ftp files timestamp, while comparing the timestamps.

Ftp server time offset

Sometimes, a Ftp server is in a different time zone from your local system.   You can specify an offset to be considered while determining which file (local or remote) is newer. The remote file timestamp is not changed. Enter a positive number to add an offset to the actual remote ftp file timestamp. Enter a negative number to subtract an offset from the actual remote ftp file timestamp. For example, consider a file with:
Actual Local file timestamp = June 1, 2005, 10:00 PM
Actual Remote file timestamp = Jun 1, 2005, 10:30 PM

Offset = 0
Put Option: File will not be transferred, since remote file is newer than local file
Get Option: File will be transferred, since remote file is newer than local file

Offset = -60 (Remote file timestamp is now calculated as 9:30 PM)
Put Option: File will be transferred, since calculated remote file timestamp is older than local file timestamp.

Get Opton: File will not be transferred, since calculated remote file timestamp is older than local file timestamp.

Transfer Modified files using Ftp Log

If you choose this option, files are transferred only if the source file date is newer than the source file date entry in the Ftp Log. For this option to work correctly, you should also enable the 'Log Transfer' option. If the source file path is not found in the Ftp log, then the file is always transferred.

Date Filter

If you need to filter files based on file modified date, select this option. For the between option, enter 2 values separated by '-' (2-4 etc..). Examples:
Older than 5 Minute, Newer than 2 Day, Between 3-5 Day, Between 1-4 Hour

File Permissions

For the PUT option, you can reset the permissions of the file, after the file is transferred to the server. The site must support the "SITE chmod" command. This is normally valid only on UNIX-based systems.
Examples for field entry:
644 = rw- for USER, r-- for GROUP, r-- for WORLD
755 = rwx for USER, r-x for GROUP, r-x for WORLD
754 = rwx for USER, r-x for GROUP, r-- for WORLD
where r = read, w = write, and x = execute permissions.

FTP Browser

ToolBar Buttons

Create New Ftp Connection
Click this button to create a new Ftp connection. A connect window will be displayed, where you can either create new settings, or connect to an existing Ftp site.

Close the current Ftp connection
Click this button to logout, and to disconnect from the current Ftp connection.

Create new Ftp task
Click this button, to create a new Ftp task. If you have a Ftp connection open, some of the ftp settings will be transferred to the new task window.

Open new Ftp browser
Click this button to open a new Ftp Browser. This allows you to connect and to view multiple Ftp sites simultaneously.

Transfer Panel

The local directory listing is on the left panel, and the remote directory listing is on the right panel

Ftp Commands

This task allows you to connect to an ftp server and issue commands. There are two types of commands supported:
1) The standard ftp commands as defined in RFC 959. Please see list of ftp commands. Only the most useful commands are listed here. Please search the Internet for RFC 959 if you need more information on all the standard Ftp commands.
2) Custom functions provided by the Ftp Command task. Please see list of custom functions.

Please note that the ftp commands, that you normally use in the ftp command-line tools in MSDOS, MacOSX or Unix etc. are not necessarily supported. For example, the LS, CD, Get or Put commands, in many command-line ftp tools are not valid RFC 959 ftp commands. Get and Put are commands created by many command-line ftp tools to implement the RETR and STOR commands, of RFC 959. To put or get files, you can use the custom functions $$GET& $$PUT that are provided by this task.

This task is multithreaded. So you can run multiple Ftp Command tasks simultaneously.

This task also outputs dynamic variables for each response provided by the Ftp server. The variables syntax is: TaskTitle::ResponseX. X is the response from the server for each of your Ftp commands. You can use these response variables in a Variable Monitor task to do conditional processing.

Host
Enter the host name or IP address of the Ftp server. If you have to connect through an Ftp proxy server, or firewall, enter the host name, or IP address of the firewall, or proxy server.

Port
Enter the desired port. Default Ftp port = 21

Password
In the password field, enter your password required to connect. Then in the command list, use the following syntax to send the hidden password: $$LOGIN::username::$PASSWORD$; or if you need to manually login, use: PASS $PASSWORD$. The string $PASSWORD$ will be replaced by the value in the password field. If you have no reason to hide your password, you do not need to fill the password field.

Commands
Enter your ftp commands, 1 per line. The custom commands are case sensitive. First line should normally be the USER command, and second line should normally be the PASS command. This field also supports dynamic variables. Using dynamic variables, you can enter a command which will be dynamically resolved at runtime.
example: cd /users/$%USERNAME%$

Example1

USER tom	sends username
PASS $PASSWORD$	sends password
ACCT production	sends account information
PWD	prints current remote ftp directory
CWD /users/tom	sets /users/tom as the current ftp directory
$$LCD::c:\test	sets c:\test as the current local directory
$$MPUT::test	puts all files from c:\test, which include 'test' in their name, to /users/tom
$$MGET::	gets all files from /users/tom to c:\test

Example2

$$LOGIN::user::$PASSWORD$	logon information
PWD	prints current remote ftp directory
CWD /users/tom	sets /users/tom as the current ftp directory

$$LCD::c:\test	sets c:\test as the current local directory
$$PASV	sets the passive mode of transfer
$$GET::test.txt	gets test.txt from /users/tom to c:\test
$$PUT::test.txt	puts test.txt from c:\test to /users/tom

List of custom commands

$$LOGIN	**Description:** Simple logon using username and password information **Syntax:** $$LOGIN::username::password **Examples:** $$LOGIN::jim::jim04 or $$LOGIN::jim::$PASSWORD$ **Comments:** To hide password, use $PASSWORD$ and enter password in password field For more complicated logons or to connect through firewalls, you may need to use as required: USER, PASS, ACCT, SITE, OPEN etc..
$$LCD	**Description:** Sets the current local folder **Syntax:** $$LCD::path_to_local_folder **Example:** $$LCD::c:\data\test **Comments:** Very important. Should always be set before transferring files **Comments:** Internal module use only. Ftp server has no idea about current local folder **Comments:** Used internally by $$PUT, $$GET, $$MPUT, $$MGET, $$MDEL functions
$$PASV	**Description:** Sets the passive mode of transfer. **Syntax:** $$PASV
$$PUT	**Description:** Puts a single file from current local folder, to current remote ftp folder **Syntax:** $$PUT::filename **Example:** $$PUT::test.txt **Comments:** Only single file is put, file should exist in current local folder
$$GET	**Description:** Gets a single file from current remote ftp folder to current local folder **Syntax:** $$GET::filename **Example:** $$GET::test.txt **Comments:** Only single file is got, file should exist in current remote ftp folder
$$PPUT	**Description:** Puts a single file from local path to ftp path **Syntax:** $$PPUT::path_to_local_file::path_to_remote_file **Example:** $$PPUT::c:\data\test.txt::/users/tom/test5.txt **Comments:** Only single file is put, file path should exist in the local system
$$PGET	**Description:** Gets a single file from ftp path to local path **Syntax:** $$PGET::path_to_remote_file::path_to_local_file **Example:** $$PGET::/users/tom/test5.txt::c:\data\test.txt **Comments:** Only single file is got, file path should exist in the remote system
	Description: Puts multiple files from current local folder to current ftp folder **Syntax:** $$MPUT::filename_filter **Example:** $$MPUT:: (will put all files from current local

$$MPUT	folder) **Example:** $$MPUT::test (will put all files which include test in their name) **Example:** $$MPUT::.txt^.htm (will put all files which have .txt or .htm extensions) **Comments:** Enter the filename_filter using the following wildcard rules
$$MGET	**Description:** Gets multiple files from current ftp folder, to current local folder **Syntax:** $$MGET::filename_filter **Example:** $$MGET::* (will get all files from current ftp folder) **Example:** $$MGET::test (will put all files which include test in their name) **Example:** $$MGET::.txt^.htm (will put all files which have .txt or .htm extensions) **Comments:** Enter the filename_filter using the following wildcard rules **Comments:** Automatically detects directory listing style and get directory information
$$MSGET	**Description:** Gets multiple files from current ftp folder to current local folder **Syntax:** $$MSGET::* (will get all files from current ftp folder) **Example:** $$MSGET::test (will put all files which include test in their name) **Example:** $$MSGET::.txt^.htm (will put all files which have .txt or .htm extensions) **Comments:** Enter the filename_filter using the following wildcard rules **Comments:** Only obtains names of objects in current ftp directory This function does not try to detect if object is a File or a Directory. It will also try to GET directories, if they satisfy the filename filter. This results in the following error being output, for every Directory in current ftp folder: 550 dir_name: [Error information varies by ftp server]. This is expected behavior.
$$MDEL	**Description:** Deletes multiple files from current ftp folder **Syntax:** $$MDEL::filename_filter **Example:** $$MDEL:: (will delete all files from current remote ftp folder) **Example:** $$MDEL::test (will delete all files which include test in their name) **Example:** $$MDEL::.txt^.htm (will delete all files which have .txt or .htm extensions) **Comments:** Enter the filename_filter using the following wildcard rules
$$MLDEL	**Description:** Deletes multiple files from current local folder **Syntax:** $$MLDEL::filename_filter **Example:** $$MDEL:: (will delete all files from current local folder) **Example:** $$MDEL::test (will delete all files which include test in their name) **Example:** $$MDEL::.txt^.htm (will delete all files which have .txt or .htm extensions) **Comments:** Enter the filename_filter using the following wildcard rules
	Description: Sends a SITE specific command

$$SITECOMMAND	**Syntax:** $$SITECOMMAND::command_arguments **Example:** $$SITECOMMAND::QUOTE SITE PRIMARY 2 **Comments: Site should support that command**

List of useful ftp commands - RFC 959

HELP	**Description:** Returns list of commands or details for a single command **Syntax:** HELP [command] **Example:** HELP returns a list of commands supported by the ftp server **Example:** HELP USER returns details on the USER command
USER	**Description:** Sends the username to begin the login process **Syntax:** USER userID **Example:** USER tom
PASS	**Description:** After sending the USER command, send this command to complete login **Syntax:** PASS user_password **Example:** PASS ergts56r
PASV	**Description:** Sets the passive mode of transfer. **Syntax:** PASV
TYPE	**Description:** Specifies Ascii or binary mode of data transfer **Syntax:** TYPE mode **Example:** TYPE I or TYPE A **Comments:** A - ASCII text, E - EBCDIC text, I - image (binary data)
PORT	**Description:** Specifies host & port to which the ftp server should connect for next data transfer **Syntax:** PORT h1,h2,h3,h4,p1,p2 **Example:** PORT 192,168,0,121,15,196 **Comments:** ip address = h1.h2.h3.h4, port = p1*256 + p2
ACCT	**Description:**Sends the account required for login **Syntax:** ACCT account_name **Comments:** normally sent after the PASS command. Uncommon command
CWD	**Description:** Sets the current ftp directory on remote ftp server **Syntax:** CWD remote_ftp_path **Example:** CWD /users/tom/data
CDUP	**Description:** Sets the parent of the current ftp folder to be the current ftp folder **Syntax:** CDUP
DELE	**Description:** Deletes a single file from the current ftp folder **Syntax:** DELE filename **Example:** DELE test.txt
RNFR	**Description:** Used to rename a file. Specifies the file to be renamed **Syntax:** RNFR original_name **Example:** RNFR test.txt **Comments:** Should be followed by RNTO command
	Description: Used to rename a file. Specifies the new name for the file

RNTO	**Syntax:** RNTO new_name **Example:** RNTO new.txt **Comments:** Should be preceded by RNFR command
SITE	**Description:** Used to issue a site specific command **Syntax:** SITE command **Example:** Use 'HELP SITE' to see list and syntax of supported commands **Comments:** May not be supported by many Ftp servers
RETR	**Description:** Transfers file from remote host **Syntax:** RETR filename **Example:** RETR test.txt **Comments:** Should be preceded by the PORT or PASV commands
STOR	**Description:** Transfers file to the remote host **Syntax:** STOR filename **Example:** STOR test.txt **Comments:** Should be preceded by the PORT or PASV commands

Ftp Commands

This task allows you to connect to an ftp server and issue commands. There are two types of commands supported:
1) The standard ftp commands as defined in RFC 959. Please see list of ftp commands. Only the most useful commands are listed here. Please search the Internet for RFC 959 if you need more information on all the standard Ftp commands.
2) Custom functions provided by the Ftp Command task. Please see list of custom functions.

Please note that the ftp commands, that you normally use in the ftp command-line tools in MSDOS, MacOSX or Unix etc. are not necessarily supported. For example, the LS, CD, Get or Put commands, in many command-line ftp tools are not valid RFC 959 ftp commands. Get and Put are commands created by many command-line ftp tools to implement the RETR and STOR commands, of RFC 959. To put or get files, you can use the custom functions $$GET& $$PUT that are provided by this task.

This task is multithreaded. So you can run multiple Ftp Command tasks simultaneously.

This task also outputs dynamic variables for each response provided by the Ftp server. The variables syntax is: TaskTitle::ResponseX. X is the response from the server for each of your Ftp commands. You can use these response variables in a Variable Monitor task to do conditional processing.

Profiles
Enter the profile name for the ftp server.

Commands
Enter your ftp commands, 1 per line. The custom commands are case sensitive. This field also supports dynamic variables. Using dynamic variables, you can enter a command which will be dynamically resolved at runtime.
example: cd /users/$%USERNAME%$

Example1

PWD	prints current remote ftp directory
CWD /users/tom	sets /users/tom as the current ftp directory
$$LCD::c:\test	sets c:\test as the current local directory
$$MPUT::test	puts all files from c:\test, which include 'test' in their name, to /users/tom
$$MGET::	gets all files from /users/tom to c:\test

Example2

PWD	prints current remote ftp directory
CWD /users/tom	sets /users/tom as the current ftp directory
$$LCD::c:\test	sets c:\test as the current local directory
$$PASV	sets the passive mode of transfer
$$GET::test.txt	gets test.txt from /users/tom to c:\test
$$PUT::test.txt	puts test.txt from c:\test to /users/tom

List of custom commands

$$LCD	**Description:** Sets the current local folder **Syntax:** $$LCD::path_to_local_folder **Example:** $$LCD::c:\data\test **Comments:** Very important. Should always be set before transferring files **Comments:** Internal module use only. Ftp server has no idea about current local folder

	Comments: Used internally by $$PUT, $$GET, $$MPUT, $$MGET, $$MDEL functions
$$CD	**Description:** Sets the current working ftp directory **Syntax:** $$CD::path_to_remote_directory **Example:** $$CD::/test/backup **Comments:** Very important. Should always be set before transferring files .
$$PASV	**Description:** Sets the passive mode of transfer. **Syntax:** $$PASV
$$PUT	**Description:** Puts a single file from current local folder, to current remote ftp folder **Syntax:** $$PUT::filename **Example:** $$PUT::test.txt **Comments:** Only single file is put, file should exist in current local folder
$$GET	**Description:** Gets a single file from current remote ftp folder to current local folder **Syntax:** $$GET::filename **Example:** $$GET::test.txt **Comments:** Only single file is got, file should exist in current remote ftp folder
$$MPUT	**Description:** Puts multiple files from current local folder to current ftp folder **Syntax:** $$MPUT::filename_filter **Example:** $$MPUT:: (will put all files from current local folder) **Example:** $$MPUT::test (will put all files which include test in their name) **Example:** $$MPUT::.txt^.htm (will put all files which have .txt or .htm extensions) **Comments:** Enter the filename_filter using the following wildcard rules
$$MGET	**Description:** Gets multiple files from current ftp folder, to current local folder **Syntax:** $$MGET::filename_filter **Example:** $$MGET::* (will get all files from current ftp folder) **Example:** $$MGET::test (will put all files which include test in their name) **Example:** $$MGET::.txt^.htm (will put all files which have .txt or .htm extensions) **Comments:** Enter the filename_filter using the following wildcard rules **Comments:** Automatically detects directory listing style and get directory information
$$MSGET	**Description:** Gets multiple files from current ftp folder to current local folder **Syntax:** $$MSGET::* (will get all files from current ftp folder) **Example:** $$MSGET::test (will put all files which include test in their name) **Example:** $$MSGET::.txt^.htm (will put all files which have .txt or .htm extensions) **Comments:** Enter the filename_filter using the following wildcard rules **Comments:** Only obtains names of objects in current ftp directory This function does not try to detect if object is a File or a Directory. It will also try to GET directories, if they satisfy the filename filter. This results in the following error being output, for every Directory in current ftp folder: 550 dir_name: [Error information varies by ftp server]. This is expected behavior.
	Description: Deletes multiple files from current ftp folder **Syntax:** $$MDEL::filename_filter **Example:** $$MDEL:: (will delete all files from current remote ftp

$$MDEL	folder) **Example:** $$MDEL::test (will delete all files which include test in their name) **Example:** $$MDEL::.txt^.htm (will delete all files which have .txt or .htm extensions) **Comments:** Enter the filename_filter using the following wildcard rules
$$MLDEL	**Description:** Deletes multiple files from current local folder **Syntax:** $$MLDEL::filename_filter **Example:** $$MDEL:: (will delete all files from current local folder) **Example:** $$MDEL::test (will delete all files which include test in their name) **Example:** $$MDEL::.txt^.htm (will delete all files which have .txt or .htm extensions) **Comments:** Enter the filename_filter using the following wildcard rules

List of useful ftp commands - RFC 959

HELP	**Description:** Returns list of commands or details for a single command **Syntax:** HELP [command] **Example:** HELP returns a list of commands supported by the ftp server **Example:** HELP USER returns details on the USER command
USER	**Description:** Sends the username to begin the login process **Syntax:** USER userID **Example:** USER tom
PASS	**Description:** After sending the USER command, send this command to complete login **Syntax:** PASS user_password **Example:** PASS ergts56r
PASV	**Description:** Sets the passive mode of transfer. **Syntax:** PASV
TYPE	**Description:** Specifies Ascii or binary mode of data transfer **Syntax:** TYPE mode **Example:** TYPE I or TYPE A **Comments:** A - ASCII text, E - EBCDIC text, I - image (binary data)
PORT	**Description:** Specifies host & port to which the ftp server should connect for next data transfer **Syntax:** PORT h1,h2,h3,h4,p1,p2 **Example:** PORT 192,168,0,121,15,196 **Comments:** ip address = h1.h2.h3.h4, port = p1*256 + p2
ACCT	**Description:**Sends the account required for login **Syntax:** ACCT account_name **Comments:** normally sent after the PASS command. Uncommon command
CWD	**Description:** Sets the current ftp directory on remote ftp server **Syntax:** CWD remote_ftp_path **Example:** CWD /users/tom/data

CDUP	**Description:** Sets the parent of the current ftp folder to be the current ftp folder **Syntax:** CDUP
DELE	**Description:** Deletes a single file from the current ftp folder **Syntax:** DELE filename **Example:** DELE test.txt
RNFR	**Description:** Used to rename a file. Specifies the file to be renamed **Syntax:** RNFR original_name **Example:** RNFR test.txt **Comments:** Should be followed by RNTO command
RNTO	**Description:** Used to rename a file. Specifies the new name for the file **Syntax:** RNTO new_name **Example:** RNTO new.txt **Comments:** Should be preceded by RNFR command
SITE	**Description:** Used to issue a site specific command **Syntax:** SITE command **Example:** Use 'HELP SITE' to see list and syntax of supported commands **Comments:** May not be supported by many Ftp servers
RETR	**Description:** Transfers file from remote host **Syntax:** RETR filename **Example:** RETR test.txt **Comments:** Should be preceded by the PORT or PASV commands
STOR	**Description:** Transfers file to the remote host **Syntax:** STOR filename **Example:** STOR test.txt **Comments:** Should be preceded by the PORT or PASV commands

Ftp

Before creating an Ftp task, we recommend that you use the FTP browser utility to get a connection. The ftp task allows you to transfer files from/to a server.

Ftp Profile

Select the ftp profile for the ftp server you need to connect to. You should have previously created an ftp profile, using the Ftp Profiles menu item.

Ftp Option

Use 'Get' to transfer files from the server to your local system. Use 'Put' to transfer files from your local system to the server. Use 'Delete' to delete files from the server.

Remote Directory

This is the most difficult parameter to enter correctly. We recommend that you use the FTP browser utility to determine the value of the full pathname of the remote directory. Execute privilege is required for the remote directory for all Ftp functions. Read privilege is required for Ftp Get. Write privilege is required for Ftp Put, and Ftp Delete functions. **If you are transferring files from the Default FTP login directory, you can leave this field blank.**

Local Directory

Enter the full pathname of the local directory, where you need to put or get files.

Backup to Local Directory

Enter the full pathname of the local directory, where you need to backup files. The Backup to local directory option does not support subdirectories. All files will be placed in the backup directory, without creating new subfolders in it.

Backup to Remote Directory

Enter the full pathname of the remote directory, where you need to backup files. The Backup to remote directory option does not support subdirectories. All files will be placed in the backup directory. No subfolders are created in the backup directory.

Use Staging

If you select this option, the file is first transferred to the local (get) or remote (put) folder entered. This folder acts as a staging folder. After the Ftp transfer is complete, the file is moved to the final destination folder. This option prevents processes from working on files, while the Ftp transfer is in progress. In the staging field, you should enter the full pathname of the final destination directory. The staging option does not support subdirectories. All files will be placed in the final destination directory. No subfolders are created in the final destination directory.

Filename

Enter the filename filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Folder Path Filter

Leave this field blank if you want all folders to be processed. If you need specific folders to be processed, then enter the folder name or path, using the following wildcard rules. If a folder passes the filter, then all its subfolders will also pass the filter.

Simple Example: Folder = c:\test , subfolders = john, john\old, sam, sam\old, joe, joe\old
Folder path filter = john   . john and john\old are processed
Folder path filter = jo   . john, john\old, joe, joe\old are processed
Folder path filter = old   . sam\old, john\old, joe\old are processed
Folder path filter = test   . all files in c:\test and its sub folders are processed
Folder path filter = john\old.   only john\old is processed.
Folder path filter = c:\test\john   . john and john\old are processed

Rename Filename

Please see Rename Help

Transfer Type

File transfer type can be either Ascii or Binary. Binary mode will transfer .exe, .jpg or other binary files. Binary mode should be the default transfer mode. Ascii mode can be used to transfer text or html files. If you transfer binary files in ascii mode, you will not get the entire file.

Append date/time To filename

You can choose to append either the date, or the date/time, or custom code. If your filename has an extension, the date or date and time is appended before the extension. This feature is useful for archive purposes.

Maintain Timestamp (for Get only)

The task can reset the transferred local files timestamp to match the remote files timestamp. This applies ONLY to the GET option.

Reset Local Timestamp (for Put only)

There is no Ftp standard to reset the remote files timestamp, to match the local files timestamp after an Ftp PUT. Hence, we provide the option to reset the source local files timestamp, to match the newly transferred remote files timestamp. There is a new proposal, to reset the remote file timestamp, using the MFMT (Modify file modification time) command. However, not many Ftp servers support the MFMT command. (Many Ftp servers incorrectly allow setting the remote file timestamp, using the MDTM command. This MDTM command is not an Ftp standard to set the remote file timestamp. The Ftp task does not reset the timestamp through this MDTM command.)

Delete Source file

You can choose to delete the source file, after a successful transfer. If the ftp transfer fails, the source file will not be deleted.

Transfer with temporary extension

The file is transferred with a temporary extension, and then renamed to the desired name, after the transfer is complete. This is useful if you have automated scripts/programs, which continuously work on transferred files with certain extensions. This ensures that the automated script will not work on a partially transferred file.

Log Transfer

You can choose to log the transfer into the Ftp Log.

Transfer Modified files

If you choose this option, files are transferred only if the source file is newer than the target file. You can specify an offset to be added to the remote ftp files timestamp, while comparing the timestamps.

Ftp server time offset

Sometimes, a Ftp server is in a different time zone from your local system. You can specify an offset to be considered while determining which file (local or remote) is newer. The remote file timestamp is not changed. Enter a positive number to add an offset to the actual remote ftp file timestamp. Enter a negative number to subtract an offset from the actual remote ftp file timestamp. For example, consider a file with:
Actual Local file timestamp = June 1, 2005, 10:00 PM
Actual Remote file timestamp = Jun 1, 2005, 10:30 PM

Offset = 0
Put Option: File will not be transferred, since remote file is newer than local file
Get Option: File will be transferred, since remote file is newer than local file

Offset = -60 (Remote file timestamp is now calculated as 9:30 PM)
Put Option: File will be transferred, since calculated remote file timestamp is older than local file timestamp.

Get Opton: File will not be transferred, since calculated remote file timestamp is older than local file timestamp.

Transfer Modified files using Ftp Log

If you choose this option, files are transferred only if the source file date is newer than the source file date entry in the Ftp Log. For this option to work correctly, you should also enable the 'Log Transfer' option. If the source file path is not found in the Ftp log, then the file is always transferred.

Date Filter

If you need to filter files based on file modified date, select this option. For the between option, enter 2 values separated by '-' (2-4 etc..). Examples:
Older than 5 Minute, Newer than 2 Day, Between 3-5 Day, Between 1-4 Hour

File Permissions

For the PUT option, you can reset the permissions of the file, after the file is transferred to the server. The site must support the "SITE chmod" command. This is normally valid only on UNIX-based systems.
Examples for field entry:
644 = rw- for USER, r-- for GROUP, r-- for WORLD
755 = rwx for USER, r-x for GROUP, r-x for WORLD
754 = rwx for USER, r-x for GROUP, r-- for WORLD
where r = read, w = write, and x = execute permissions.

Ftp Log

The ftp log maintains a list of the files that have been transferred. The exact nature of what is logged depends on the options you chose when you create an Ftp task. If the logging option is selected, the task will continuously update the log with every file transfer.

Managing Log Size

You can set the maximum Ftp log size. The Ftp log has a default maximum size of 100,000. This allows about 700-900 rows. The FTP log is automatically trimmed back to 50%, once the maximum log size is exceeded.

Saving Log to Disk

Enter a valid directory on your system, in the directory text box. Click the save button. The file will be saved to the selected directory, with the filename 'FtpLog'.

Ftp Monitor

You can monitor a remote ftp folder for changes or other criteria. This task can be used in two ways:
1) If a change is detected in this task, you can run another task, or
2) The Ftp Monitor task can also be embedded within a Chain task, and is useful for conditional task processing. If the criteria is satisfied, then the exit code = -100. Otherwise, the exit code = 0.

The Ftp Monitor task also outputs a variable called 'TaskTitle::Filenames". This variable contains the list of all files that satisfy the monitoring criteria. The format of this variable is #file1^#file2^#file3, where file1,file2,file3 are the names of the files, that have been modified. You can use this variable directly, as the filename in other copy, ftp, delete tasks to copy / ftp /delete only these files. This variable is only updated if the filename, filesize, or filedate criteria options are selected.

Directory
Enter the remote folder that you wish to monitor.

Filename
Enter the filename filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Task To Run
Select the desired task to run, if any of the selected criteria are met. If you want to use this task in a chain for conditional processing, select 'None'. You can also select multiple tasks to run in sequence, or to run simultaneously. To run tasks in sequence, use taskTitle1|taskTitle2|taskTitle3. To run tasks simultaneously, use taskTitle1&taskTitle2&taskTitle3.

Options
The Task will run, if any of the following selected criteria are satisfied. You can select 1 or more criteria.
1) Any change is found in the directory since the last run of this task. This option does not update any of the dynamic variables.
2) File (or files) which satisfy the Filename criteria exists
3) File (or files) which satisfy the Filename criteria, as well as file size criteria
4) File (or files) which satisfy the Filename criteria, as well as date modification criteria
5) Total files in directory are greater/less than certain value. This option does not update any of the dynamic variables.

Ftp Server Address
The Ftp server address is the name or the IP address of the Server on the network. Please find this value from your network administrator, or from Internet service provider, or from Ftp server administrator.

Server Port
The default ftp command port is 21.

Username and Password
If the Ftp server is private, you will typically need a username/password to transfer files. Please check with the Ftp server administrator for your user account information.

Directory Listing Style
Ftp servers provide directory listing in various formats. The most common is the Unix style. We currently support Unix and DOS listing styles. If your Ftp server provides a listing in some other style, the task may fail. However, most Ftp servers have an option to provide directory listing in the Unix style. Please request your Ftp administrator, to set up your Ftp account to receive listing in Unix style, if possible.

If your Ftp server does not provide a listing in the Unix or DOS style, you can try the NameOnly option. Here, only a list of names in the directory is obtained. There is no information on the size, date, or if the name corresponds to a file, or a directory. Hence, you cannot transfer files based on date or size. Also, since the file or directory information is not available, subdirectory (tree) transfers are not possible in this mode.

Passive Mode

In normal mode, the Ftp server creates the data connection. In Passive mode, your local system will create the data connection. You may need to use passive mode, if you are behind a firewall/proxy server, or if you get erratic transfers, or if you get failed data channel errors.

Using Ftp Monitor Task in a Chain

- Create a Ftp Monitor task with the remote directory you need to monitor for changes.
- Create a Chain Task with the above "Ftp Monitor" task as the 1st step.
- Use the Exit Code criteria as (Exit Code != -100 Go to Step 10) i.e. stop chain if exit code != -100
- Add a second task after this step.
- If any of the selected criteria are met, the Ftp Monitor task will exit with Exit Code = -100 and the Chain will run the 2nd task.
- If no change is detected, the Ftp Monitor task will exit with Exit Code = 0 and the Chain will terminate without running the 2nd task.

Ftp Monitor

You can monitor a remote ftp folder for changes or other criteria. This task can be used in two ways:
1) If a change is detected in this task, you can run another task, or
2) The Ftp Monitor task can also be embedded within a Chain task, and is useful for conditional task processing. If the criteria is satisfied, then the exit code = -100. Otherwise, the exit code = 0.

The Ftp Monitor task also outputs a variable called 'TaskTitle::Filenames". This variable contains the list of all files that satisfy the monitoring criteria. The format of this variable is #file1^#file2^#file3, where file1,file2,file3 are the names of the files, that have been modified. You can use this variable directly, as the filename in other copy, ftp, delete tasks to copy / ftp /delete only these files. This variable is only updated if the filename, filesize, or filedate criteria options are selected.

Directory
Enter the remote folder that you wish to monitor.

Filename
Enter the filename filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Task To Run
Select the desired task to run, if any of the selected criteria are met. If you want to use this task in a chain for conditional processing, select 'None'. You can also select multiple tasks to run in sequence, or to run simultaneously. To run tasks in sequence, use taskTitle1|taskTitle2|taskTitle3. To run tasks simultaneously, use taskTitle1&taskTitle2&taskTitle3.

Options
The Task will run, if any of the following selected criteria are satisfied. You can select 1 or more criteria.
1) Any change is found in the directory since the last run of this task. This option does not update any of the dynamic variables.
2) File (or files) which satisfy the Filename criteria exists
3) File (or files) which satisfy the Filename criteria, as well as file size criteria
4) File (or files) which satisfy the Filename criteria, as well as date modification criteria
5) Total files in directory are greater/less than certain value. This option does not update any of the dynamic variables.

Ftp Server Address
The Ftp server address is the name or the IP address of the Server on the network. Please find this value from your network administrator, or from Internet service provider, or from Ftp server administrator.

Server Port
The default ftp command port is 21.

Username and Password
If the Ftp server is private, you will typically need a username/password to transfer files. Please check with the Ftp server administrator for your user account information.

Directory Listing Style
Ftp servers provide directory listing in various formats. The most common is the Unix style. We currently support Unix and DOS listing styles. If your Ftp server provides a listing in some other style, the task may fail. However, most Ftp servers have an option to provide directory listing in the Unix style. Please request your Ftp administrator, to set up your Ftp account to receive listing in Unix style, if possible.

If your Ftp server does not provide a listing in the Unix or DOS style, you can try the NameOnly option. Here, only a list of names in the directory is obtained. There is no information on the size, date, or if the name corresponds to a file, or a directory. Hence, you cannot transfer files based on date or size. Also, since the file or directory information is not available, subdirectory (tree) transfers are not possible in this mode.

Passive Mode

In normal mode, the Ftp server creates the data connection. In Passive mode, your local system will create the data connection. You may need to use passive mode, if you are behind a firewall/proxy server, or if you get erratic transfers, or if you get failed data channel errors.

Using Ftp Monitor Task in a Chain

- Create a Ftp Monitor task with the remote directory you need to monitor for changes.
- Create a Chain Task with the above "Ftp Monitor" task as the 1st step.
- Use the Exit Code criteria as (Exit Code != -100 Go to Step 10) i.e. stop chain if exit code != -100
- Add a second task after this step.
- If any of the selected criteria are met, the Ftp Monitor task will exit with Exit Code = -100 and the Chain will run the 2nd task.
- If no change is detected, the Ftp Monitor task will exit with Exit Code = 0 and the Chain will terminate without running the 2nd task.

Ftp Profiles

This window allows you to set ftp host settings and firewall settings. The host profiles and firewall profiles are used in the Ftp tasks and Ftp monitor task.

The Ftp-j module supports ssl and certificates. The certificates have to be imported into a java keystore as shown here.

Save Button - Saves the current data. If you are on the Host tab, this will save the host data. If you are on the Firewall tab, this will save the firewall data.

New Button - Creates a new profile. If you are on the Host tab, this will create a new host profile. If you are on the Firewall tab, this will create a new firewall profile.

Delete Button - Deletes the selected profile. If you are on the Host tab, this will delete the host profile. If you are on the Firewall tab, this will delete the firewall profile.

Host Tab

Host: Enter the exact host name, or IP address, of the Ftp server, that you need to connect to. Examples of host names are ftp.hiteksoftware.com or ftp.microsoft.com. Examples of IP addresses are: 192.168.0.1, or 10.0.0.5. If you do not know the host name, or IP address, of the Ftp server, please contact your network administrator.

Port: The ftp server listens on a given socket port, for client connections, and commands. You should enter the FTP port used. The default ftp port is 21, and most Ftp servers will listen on port 21.

Username: If you have an account on the Ftp server, please enter your username, or user identification name. If you are trying to connect to a public ftp server, that allows anonymous or guest access, enter 'anonymous' as your username.

Password: Enter your password that has been setup, with your user identification name. If you are trying to connect to a public ftp server that allows anonymous access or guest access, enter your email address as your password.

Account: Some Ftp servers require you to enter an account, before they can authenticate you. Most Ftp servers do not need any account information.

Local Dir: Enter the local directory that you want displayed after login. If you leave this field blank, the current user's local home directory is displayed.

Remote Dir: Enter the remote directory that you want displayed after login. If you leave this field blank, the current users default ftp directory is displayed.

Firewall: If you need to connect to your Ftp server through a firewall, select the firewall profile name.

Passive: Passive mode of transfer may be required if you are behind a firewall, or other gateway. Using Passive Mode ensures that the local host initiates all connections. This avoids the situation, where the Firewall rejects incoming connection attempts from the Remote Host. We recommend that you use passive mode, even if you are not behind a firewall.

Directory Listing Style: Ftp servers provide directory listing in various formats. The most common format is the Unix style. We currently support Unix and DOS listing styles. If your Ftp server provides a listing in some other style, the task may fail. However, most Ftp servers have an option, to provide directory listing in the Unix style. Please request your Ftp administrator to set up your Ftp account to receive listing in Unix style.

If your Ftp server does not provide a listing in the Unix or DOS style, you can try the Names-Only option. For the names-only option, a list of names in the directory is obtained. There is no information on the file size and file date. Hence, you cannot transfer files based on file date or file size. Also, since the file or directory information is not available, subdirectory (tree) transfers are not possible in this mode. When using the interactive Transfer window with the Names-Only option, since the directory or file information is not available, you have to manually move to a new directory using either the Go Up, Home, or Change To directory buttons.

Connection Tab

Encoding: Sets the character encoding used by the FTP control connection. Some FTP servers require that commands be issued in a non-ASCII encoding like UTF-8 so that filenames with multi-byte character representations (e.g, Big 8) can be specified.

Data Port Range: This setting is supported in FTP-J module. This field sets the data port range for incoming non-passive (active) connections. If you leave this field blank, the server will set the data port ranges. You have to enter both the start port and end port using the following format: startPort,endPort. Example: 5000,5050

SSL Tab

The Ftp-j module support ssl and certificates. The certificates have to be imported into a java keystore as shown here.

Firewall Tab

Host: Enter the name or IP address of the firewall server.

Port: Enter the port that the firewall server will listen for FTP connections. The default is 21.

Username: Enter your user identification name, if your firewall server requires authentication.

Password: Enter your user identification password, if your firewall server requires authentication.

Firewall Type: Please check your Firewall for details on how it accepts FTP connections. We currently support the following types of ftp firewalls:
1) USER 2) SITE 3) OPEN 4) Custom

Firewall Type	
Logon Details	
USER	1) Connects to FtpFirewall at FtpFirewall port 2) Sends USER command: USER username@FtpHost 3) Sends PASS command: PASS password
	1) Connects to

OPEN	FtpFirewall at FtpFirewall port 2) Sends OPEN command: OPEN FtpHost 3) Sends USER command: USER username 4) Sends PASS command: PASS password
SITE	1) Connects to FtpFirewall at FtpFirewall port 2) Sends SITE command: SITE FtpHost 3) Sends USER command: USER username 4) Sends PASS command: PASS password
	1) Connects to FtpFirewall at FtpFirewall port 2) Sends your custom commands which are entered in the Custom field If you want to hide your Ftp password, use the following syntax: PASS $PASSWORD$ $PASSWORD$ will be replaced by the password value in the password field of the Host Tab If you need to use a Firewall password and want to hide your ftp Firewall password, use the following syntax: PASS $FIREPASSWORD$ $FIREPASSWORD$

Custom	will be replaced by the password value in the password field of the Firewall Tab **Example1:** Firewall requires authentication, Ftp site supports USER logon USER firewall_username PASS $FIREPASSWORD$ USER username@FtpHost PASS $PASSWORD$ **Example2:** Firewall requires authentication, Ftp site supports OPEN logon USER firewall_username PASS $FIREPASSWORD$ OPEN FtpHost USER username PASS $PASSWORD$

Ftp SSL and Certificates

The Ftp-j module supports explicit and implicit SSL. It also supports x509 certificates. The certificates (client or server) are handled internally by Sun's Java api. The certificates are managed directly by the user via the commandline using java's **keytool** utility.

Keytool is located in the [install_directory]\jre\bin subdirectory. Open a DOS command prompt on windows, or terminal or unix/macosx and type in:
cd [install_directory]\jre\bin
then at the prompt, type in:
keytool -help

if you see a list of commands then keytool is installed and working correctly.

Setting up a Key Store
The Key Store contains all the certificates (client or server) that you can use to connect to Ftp Servers. If no Key Store exists, a default Key Store will be automatically created when you first import or create a certificate using the -import or -genkey commands. The default keystore file name is **.keystore** and is located in the users home directory.
In the Ftp SSL tab, you will be required to enter the full path to this keystore file.

Importing a Certificate
use the following command to import an existing x509 certificate:
keytool -import -alias joe -file c:\......\certificate_file.cer

Make sure that the certificate_file.cer exists and the full entered path to it exists. The import command will automatically create a keystore. It will also prompt you for a keystore password. Make sure to enter a password and **DO NOT FORGET** this password.
In the Ftp SSL tab, you will be required to enter this keystore password.

Generating a Certificate
If you do not have an existing Certificate and private key, use the following command to generate a new key pair & certificate:
keytool -genkey -alias joe

Then follow instructions. Keytool will request a password for the private key. **DO NOT FORGET** this password. This command creates a self-signed certificate that includes the public key information. This certificate is associated with the private key in the keystore entry referred to by the alias "joe". The default key created by the genkey command is dsa 1024.

Exporting a Certificate
Use the following command to export the public certificate for the alias joe:
keytool -export -alias joe -file *certificate_file_path*

Then give the public certificate file to the Ftp Server administrator. ***The format of the key is very important (dsa or rsa).***

Getting Started

Creating a new task

From the front panel, select the 'New Tasks' button.
From the Task Type explorer, select the 'Administrative' folder. Then select 'Archive Logs'. This task archives the programs output logs and activity logs.
Enter a Title for this 'Archive Logs' task. Select a local directory to archive the logs.
Click on the 'Save' button. The task will appear in the Tasks list.

Running a task

Select the task in the Tasks List, and click the 'Run' button.
The Scheduler Engine will run the task.
Click on the Logs menu of the front panel. Then click on the 'Output Log' menu item. You will see the output from the Archive Logs task.

Scheduling a task

Select the task in the Tasks list, and click the 'Schedule' button.
The scheduler window will pop up. Set the schedule type as 'Minute'.
Click 'Save' to save this schedule. The task will appear in the Schedules list.
Wait for a minute, and the 'Archive Logs' task will run again. This task will run every minute. In the output log viewer, click the Refresh Button to view the latest Run output.
You can stop the schedule by disabling the 'Run' check box, in the first column of the 'Scheduled Tasks' table.

Editing a task

To edit your task, select the task in the Tasks list, and click the 'Edit' button. You can change your task parameters, and click 'Save' again. The original task data will be updated.

Editing a task schedule

To edit your task, select the task in the Schedules list, and click the 'Edit' button. You can change your schedule parameters, and click the 'Save' button. The original schedule data will be updated.

Deleting a task

To delete the task, select the task in the Tasks list, and click the 'Delete' button.

Deleting a task schedule

To delete the schedule for a task, select the task in the Schedules list, and click the 'Delete' button.

Key Store, import and export keys into the Key Store or from the Key Store.

GnuPG setttings

In version 7.x, the encryption module of AutoKrypt and Automize Enterprise is a command-line wrapper around GnuPG 1.4.x.

Key Store Folder: This is the default GnuPG home folder normally located in the users home directory. This setting is currently read only.

Shell Command: This is the shell used on the operating system to launch and run GnuPG commands.
For Windows NT family (NT, 2000, 2003, XP) the shell command used is cmd.exe (or cmd). The /c switch tells the command interpreter to run the command and return immediately.
For earlier windows versions (95, 98, ME) the command is command.com.
For MacOSX and Linux, the default shell used is bash.

Passphrase method: When Signing or Decrypting files or for password based symmetric encryption decryption, the password has to be sent to GnuPG via the command line. The 2 techniques available are:
1) GnuPG reads the password from a password file. This password file is dynamically created before the encryption / signing / decryption process. When the process is completed, the file is deleted.
2) GnuPG reads the password from System.IN. Here the password is piped to the GunPG process using the echo command.

Command Time: This the maximum time to wait for the GnuPG process to run. The default is 300 seconds.

How To ….

1. How to send email notification if task fails

Generally, if your task exit code =< 0, then the task has passed. If exit code > 0, then task generally has failed. Use the Email Notification feature (from the Settings menu) to email yourself if a task fails.

2. How to send email notification depending on whether task fails or passes

Generally, if your task exit code =< 0, then the task has passed. If exit code > 0, then task generally has failed.

1. Setup your desired task and test it.
2. Setup two email tasks to email yourself. One with the failure notification message, and another with success notification message. Test your email tasks.
3. Setup a chain task that has 3 steps.
   1st step should be your desired task.
   2nd step should be the email failure notification.
   3rd step should be email success notification.
4. For your desired task in the 1st step, in the chain options, set the criteria:
If Exit Code < 1, Go to Step 3.
5. For the email failure notification task in step 2, Select the Check Box "Always stop chain at this Step".
6. If your task passes, the exit code = 0 or exit code < 0, the chain will move to Step 2. A success notification email will be sent, and chain will stop here.
6. If your task fails, the exit code = 0. The chain will move to step 3, and failure notification email will be sent.

Import Export task and schedule data

BEFORE USING THIS FEATURE, PLEASE BACKUP YOUR DATA USING FILE MENU / BACKUP!!!

This feature allows you to export, import, or add tasks and schedules. The import feature will add/overwrite the existing task or schedule data.

Delimiter: The delimiter should be carefully chosen. The delimiter should not exist in the task data, otherwise the import will fail. It is preferable to use a four-character delimiter, such as @%%@.
Single characters should not be used as delimiters, because most of these characters exist in the task data. Also, the following characters should not be used:
1) $ or $$ or $$$. (Email task message body uses $$$.)
2) |#| . (Chain data is separated by |#|.)
3) $ or %. (Dynamic variables are surrounded by $%...%$.)

Export:
1) Select the table type (task or schedule)
2) Enter filepath to export data to
3) Select the column delimiter carefully
4) Click the export button

Import:
1) The data should be in the correct format. To obtain the correct format, use the Export button first and export the task data to file. Then look at the file format. The format is self-explanatory.
2) Select the type (task or schedule)
3) Enter filepath to import data from
4) Select the column delimiter. This should have been the delimiter you used to export data initially
5) Click the import button

Introduction

This software is designed to automatically execute tasks. The software can be run on windows, macintosh, linux, and on other operating systems.

Tasks can be scheduled on a daily, weekly, monthly, or hourly basis. Tasks can also be scheduled by the minute, or by second. Tasks include local and Internet applications, file download and monitoring.

This software is available on a 30-day free trial basis. You can test it during this period. After 30 days, the software expires, and you will need to register the software. If you do not want to use the software, you can uninstall it from your computer. The uninstall procedure depends on the operating system. Typically, an uninstall program is also loaded onto your computer. Please see your help file for uninstall instructions.

If you need to use this software for more than 30 days, please register it. You can purchase online by credit card, check, money order, Paypal, wire transferor purchase order. Multiple copy discounts, and site licenses are available. See the Hitek Software web page, at "http://www.hiteksoftware.com" for details.

Private/public key generator

Use the keygen utility to

1. create a new private/public key pair,
2. change key passphrase,
3. convert a public key from SECSH to OpenSSH format and vice versa.

The SSH2 add-on module uses the private key to logon, and the public key is used by the SSH server. Your server administrator should be able to correctly install your public key. Since each server is different, we cannot provide information on the exact details of public key installation.

Most SSH servers accept keys either, in the SECSH format, or in the OPENSSH format. Please use the correct key format, when generating a new key.

Key Strokes

This task allows you to send key strokes to the current application window which is in focus. You can use this task in a chain after a Command, or Open Documents, or MacLauncher task to launch an application/document, and send it key strokes.

The key strokes should be entered one on each line, in the Command area. If more than one key is being entered on a line, they should be separated by '+'.

You can also specify, a delay time in milliseconds (1000 milliseconds = 1 second), between commands, using the following syntax:
delay=5000 (this would delay for 5 seconds)

Examples:
ctrl+P
delay=3000
alt+x
shift+a (for A)
t+e+s+t (for test)
shift+t+e+s+t (for TEST)
alt+f+c (this will close an in focus Internet Explorer window)

Table of supported keys

Alphabets (A-Z)	Enter in lower case (a-z)
Numbers (0-9)	Enter (0-9)
Function keys (F1-F24)	Enter in lower case (f1-f24)
Number pad keys (0-9)	Enter in lower case (numpad0 - numpad9)
The following keys have to be entered exactly as shown. The names are self-explanatory. All keys may not be supported on your system. ctrl, shift, alt, enter, accept, add, again, alphanumeric, alt_graph, back_quote, back_slash, back_space, braceleft, braceright, cancel, caps_lock, clear, colon, comma, compose, convert, copy, decimal, delete, divide, down, end, enter, equals, escape, final, find, greater, help, home, insert, left, less, meta, minus, modechange, multiply, num_lock, open_bracket, page_up, page_down, paste, pause, period, plus, printscreen, quote, quotedbl, right, scroll_lock, semicolon, separator, shift, slash, space, stop, subtract, tab, underscore, undo, up	

Set / Get Configuration/INI file Key Value

The Set/Get tasks allows you to set/get a key value in a configuration or INI file. Typical examples of such files include .ini files on windows, java properties files, unix configuration files. The format of these files is:

```
key1=value1
key2=value2
......
```

Log Files

Task data is logged into log files on your disk. The logs are available via the Logs menu. The log files are automatically trimmed back to 0%, 25%, or 50%, when they exceed the maximum size limit set by user.

Output log
This is the most used log. This log displays any output from your tasks. The data includes run times, any output data, and error data and exit codes and end times.

Debug log
This log displays any error from your tasks, and the exact location in our program code, where this error occurred. This log provides the most important information, and helps us quickly determine the cause of your task error. We can determine if it is a user input problem, system problem or other problem, or a bug. Almost 100% of our code is in error catching blocks. We keep refining our code when we analyze your debug log. This has helped the program to be extremely stable.

Activity log
This log displays a list of tasks run times, and startup times, and shutdown times.

Task logs
Each individual task that you create will output data into task log files. The log files have the same name as the task title. The data that is displayed in these logs include run times, output data, error data, and exit codes.

Logic For

This task allows you to run upto 3 tasks within a For loop. The number of loops is defined by the start value, end value and increment. Optionally, you can set Continue or Break Conditions.

Continue skips the next steps in current loop and moves to next loop.
Break exits the For loop completely.

The flow diagram example:

```
For( Start Value , End Value, Increment Value)
{
Continue Statement (optional)
Run Task 1
Continue/Break Statements (optional)
Run Task 2 (optional)
Continue/Break Statements (optional)
Run Task 3 (optional)
Break Statement (optional)
}
```

Logic IF

This task allows you to run task A when the IF condition is satisfied. Optionally, you can run task B when the IF condition is not satisfied.

```
The flow diagram example:
IF ( Condition Is True )
{ Run Task A}
else
{ Run Task B}
```

Logic IF AND OR

This task allows you to run 3 tasks A,B,C depending on the results of two IF conditions.

The flow diagram is:

```
IF ( Condition 1 Is True AND Condition 2 is True)
{ Run Task A}
else IF ( Condition 1 Is True OR Condition 2 is True)
{ Run Task B}
else IF ( Condition 1 Is False AND Condition 2 is False)
{ Run Task C}
```

Logic WHILE

This task allows you to run upto 3 tasks within a While loop. The number of loops is defined by the start value, end value and increment. Optionally, you can set Continue or Break Conditions.

Continue skips the next steps in current loop and moves to next loop.
Break exits the For loop completely.

The flow diagram example:

```
While( Condition is True )
{
Continue Statement (optional)
Run Task 1
Continue/Break Statements (optional)
Run Task 2 (optional)
Continue/Break Statements (optional)
Run Task 3 (optional)
Break Statement (optional)
}
```

Log File Monitor

The Log file monitor task allows you to monitor a log file for certain alert terms. If the alert term is found, then the timestamp for the log entry is determined and archived. If the timestamp is newer than the previously archived timestamp, you can choose to run another task (or tasks). Also, an exit code = -100 is set when a new alert is found.

File Path

Enter the full path to the log file, that needs to be monitored.

Alert Terms

Enter the alert term required. If you have more than 1 alert term, separate them by a character, not included within the alert term. Example: Searching^Java^Automation . This field also supports dynamic variables. Using dynamic variables, for example, you can search a log file for today's date/time, and run a task if that entry is found.

Alert Term Separator

Enter the separator character used between alert terms, if you have more than 1 alert term.

Logging Direction

New logging entries can either be inserted at the top, or added at the bottom of the log file.

Task To Run

Select the desired task to run, if any of the selected criteria are met. If you want to use this task in a chain, for conditional processing, select 'None'. You can select multiple tasks to run in sequence or simultaneously. To run tasks in sequence, use taskTitle1|taskTitle2|taskTitle3. To run tasks simultaneously, use taskTitle1&taskTitle2&taskTitle3.

Logging Date Format

Enter logging date format, using the following rules below. Please read this entire section very carefully, and please read all the Examples, at the bottom of this section. If you make any mistake in entering the format, the task will not be able to read the timestamp and the task will fail.

Rules:
1) The log file should have a standardized log format. The timestamp should be in the same column location for all lines, otherwise the timestamp in that particular line will not be parsed.
2) The format is: variable1=xxxx(-)variable2=zzzz(-)….etc.
3) One of these types of formats can be specified a) columnformat or b) tokenformat

formattype = columnformat/tokenformat

For tokenformat, use the following variables

formattype = tokenformat

tokens = separators that are used to separate fields. Multiple separators can be specified, one after another. Example: tokens=SPACE*&-_(-) would use the following characters as separators: whitespace * & - _

tokcol_x = specify the format for tokcol_x. Please see examples at bottom of this page for details.

For columnformat, use the following variables

formattype = columnformat

colsep = separator between columns in the log file. Enter SPACE if the column separator is white-space or Enter TAB if the columns are separated by tabs. Example: colsep=SPACE(-) or colsep=,(-)

datecol = Column where date is logged. Example: datecol=4(-)

timecol = Column where time is logged. Example: timecol=4(-)

dateformat = date format used. Example: dateformat=YY-qq-dd(-)

timeformat = time format used. If datecol is same as timecol, leave this field empty. Example: timeformat= hh:mm:ss(-)

tokens = separators between fields. Use all separators between date components and time components. If datecol is same as timecol, add the separator between datecol and timecol.

Examples:
if date is like 03-15-2004, and time is like 09:15:05, tokens=-:(-)
if date and time are logged into same column like, 03-15-2004_09:15:05, tokens=_-:(-)

Specifying date/time formats
Use the same rules, as that used for the DATE variable. Example:
If date time logging output = Feb 3, 2004 4:44:07 PM
Date time format required = Qqq ddx, YY hhx:mm:ss am_pm

YY = 4 digit year is always output (2001 etc..)
yy = 2 digit year is always output (00-99)
pp = 2 digit month is always output (01-12)
ppx = 1 or 2 digit month can be output(1-12)
qqq = 3 character month is always output (jan-dec)
QQQ = 3 character month is always output (JAN-DEC)
Qqq = 3 character month is always output (Jan-Dec)
dd = 2 digit date is always output (01-31)
ddx = 1 or 2 digit date can be output(1-31)
HH = 2 digit hour based on 12 hour clock is always output (01-12) (should generally be used with am_pm)
HHx = 1 or 2 digit hour based on 12 hour clock can be output(1-12)
hh = 2 digit hour based on 24 hour clock is always output (00-23)
hhx = 1 or 2 digit hour based on 24 hour clock can be output (0-23)
mm = 2 digit minute is always output (00-59)
mmx = 1 or 2 digit minute can be output (0-59)
ss = 2 digit seconds is always output (00-59)
ssx = 1 or 2 digit seconds can be output (0-59)
am_pm = AM is output for AM hours, PM is output for PM hours

Examples of Logging date format field entries
1) NCSA Combined / NCSA Extended / Extended Log Format / Microsoft Extended Format.
Example log entry:
2667 avi.start.com - Authorize 401 /pi.admin 1998-03-09 15:21:37 2345

ColumnFormat Logging date format field should be:
formattype=columnformat(-)colsep=TAB(-)tokens=:-(-)datecol=7(-)timecol=8(-)dateformat=YY-dd-qq(-)timeformat=hh:mm:ss(-)

TokenFormat Logging date format field should be:
formattype=tokenformat(-)tokens=-:TAB(-)tokcol_7=YY(-)tokcol_8=pp(-)tokcol_9=dd(-)tokcol_10=hh(-)tokcol_11=mm(-)tokcol_12=ss(-)

2) NCSA - Common Log File format:
Example log entry:
255.255.255.255 - REDMOND\doug [07/Jun/1996:17:39:04 -0800] "POST /iisadmin/default.htm?-, HTTP/1.0" 200 3401

ColumnFormat Logging date format field should be:
formattype=columnformat(-)colsep=space[(-)tokens=:/(-)dateformat=dd/Qqq/YY:dd:mm:ss(-)timeformat=(-)datecol=4(-)timecol=4(-)

TokenFormat Logging date format field should be:
formattype=tokenformat(-)tokens=[/:SPACE (-)tokcol_5=dd(-)tokcol_6=Qqq(-)tokcol_7=YY(-)tokcol_8=hh(-)tokcol_9=mm(-)tokcol_10=ss(-)

3) Microsoft IIS Log Format
Example log entry:
255.255.255.255, user_name, 03/20/98, 23:58:11, MSFTPSVC, SALES1, 255.255.255.255, 60, 275, 0, 0, 0, PASS, intro.htm

ColumnFormat Logging date format field should be:
formattype=columnformat(-)colsep=,SPACE(-)tokens=:/(-)dateformat=dd/Qqq/YY(-)timeformat=dd:mm:ss(-)datecol=3(-)timecol=4(-)

TokenFormat Logging date format field should be:
formattype=tokenformat(-)tokens=,SPACE/:(-)tokcol_3=pp(-)tokcol_4=dd(-)tokcol_5=YY(-)tokcol_6=hh(-)tokcol_7=mm(-)tokcol_8=ss(-)

4) WebSTAR Log format
Example log entry:
03/09/98 15:21:37 ok avi.start.com :pi.admin 2667

ColumnFormat Logging date format field should be:
formattype=columnformat(-)colsep=TAB(-)tokens=:/(-)dateformat=dd/pp/yy(-)timeformat=hh:mm:ss(-)datecol=1(-)timecol=2(-)

TokenFormat Logging date format field should be:
formattype=tokenformat(-)tokens=TAB/:(-)tokcol_1=pp(-)tokcol_2=dd(-)tokcol_3=yy(-)tokcol_4=hh(-)tokcol_5=mm(-)tokcol_6=ss(-)

5) No tokens format, date and time in same column
Example log entry:
03292004-152137 Test task - exit=0
Only ColumnFormat Logging can be used as follows:
formattype=columnformat(-)colsep=SPACE(-)tokens=(-)dateformat=ppddYY-hhmmss(-)timeformat=(-)datecol=1(-)timecol=1(-)
tokens should always be blank, if either the date or time components, are not separated by a separator character, i.e. tokens=(-)

6) No tokens format, date and time in different columns

Example log entry:
03292004 152137 Test task - exit=0
Only ColumnFormat Logging can be used as follows:
formattype=columnformat(-)colsep=SPACE(-)tokens=(-)dateformat=ppddYY(-)timeformat=hhmmss (-)datecol=1(-)timecol=2(-)
tokens should always be blank, if either the date or time components, are not separated by a separator character, i.e. tokens=(-)

Log Options

Font Size and Type - This sets the font size/type when viewing the logs in the logs menu / logs viewer.

Log Size - This sets the maximum log size for the activity, output, debug and task logs. The logs are trimmed to 0%, 25%, or 50% of this maximum size when this size is reached.

Enter new line between entries - Every log entry is separated by a blank line.

Log entries on same line as date/time - Every entry will be logged on the same line as the timestamp.

Combine Logs - You can choose to combine the activity, output, and/or debug.

Delete TaskLog before each run - The task logs are stored in the 'TaskLogs' folder, as simple text files, which have the same filename as the TaskTitle. If you select this option, all task logs will contain only the results of the last run. If you select this option, you may want to archive the 'TaskLogs' folder, using an Archive Logs task, or a Copy task.

Logging Date Format - Allows user to set the logging date format using YY,pp,dd,hh,mm,ss etc.
Example, if you use: YY-pp-dd hh:mm:ss, the log date format is 2008-09-18 09:30:15

Debugging Options

Email Debugging - When your Email task fails, the error information is logged into the output log. The error stack trace is logged into the debug log. For more comprehensive debugging information, you can turn on this switch. This will log all communication between the Email task and the Smtp server. This debugging information is logged into the debug log.

Directory Listing - This outputs the local directory listing for the Copy, Ftp, and SFtp tasks to the output log.

Synchronization module debugging - This outputs debugging information for the synchronization module, to the debug log.

Variable module debugging - This outputs debugging information for the variable module, to the debug log.

Scheduling module debugging - This outputs debugging information for the scheduling module, to the debug log.

MacOS X Launcher

This feature allows you to run AppleScripts, or open a document, or a folder on your system. This is equivalent to a user manually clicking on the document icon, to open the document. You can also open a folder listing by entering the folder path.

Directory
Enter the full path to the directory where your file resides.

Name
Enter the full filename with extension.

AppleScript Example : Run the 'CNN.scpt', which launches www.cnn.com in your default browser.
Directory = /Applications/AppleScript/Example Scripts/URLs
Name = CNN.scpt

Application Example : This will launch the Stickies application
Directory = /Applications
Name = Stickies.app

Folder Example: This will open the Applications folder
Directory = /Applications
Name = /Applications

Using the windows macro recorder

THIS FEATURE IS NOT SUPPORTED IN VERSION 7.x.

Automize 3.x and 4.x supported Key/Mouse macro record/replay. This feature is not in-built into Automize 5.x and later, for the following reasons:
1) Feature is not cross platform compatible
2) Very little user interest in this feature
3) Macro record/replay is always fundamentally unreliable

Windows users only can install optional files, to record/replay key/mouse macro's in Automize. These files are not packaged with the default Automize install, to reduce the size of the installation executable. If you wish to use the windows recorder, please download and install these extra files, from the Automize website. These optional files, are normally available from either the Automize download pages, support page (patches), or knowledge base. Please follow the instructions carefully to install these macro files.

Important: After you install the optional files, come back to the Settings / Add-ons window, and Select the option to enable windows macros. Then restart Automize. The macro menu is visible, only when this option is selected.

Maintenance

This feature does the following:
1) It saves all settings, task data and schedule data.
2) Stops the engine server.
3) Stops any active schedules in the old instance of the engine.
4) Re-launches a new instance of the engine.
5) The old instance of the engine will shutdown after currently running tasks complete.

NOTE: To wait for running tasks to complete before old instance of engine shuts down, use the engine menu - engine settings menu - Startup/ Shutdown tab. Select the option to wait for all running tasks to complete before shutting down engine.

You can create and schedule a Maintenance task weekly/monthly to free memory, if you notice that the memory consumption by the Engine is high. **Please create only a single Maintenance task. Make sure that no other task is scheduled, when this task has a scheduled run. Leave at least 1 minute idle time before this task.** Generally, you need to run this task only on Windows and Mac OS. This Maintenance task is also useful, if you are running tasks very frequently (every few seconds), and running out of memory. In this case, you can schedule a Maintenance task more frequently.

On Solaris and Linux, we have run tasks every 10 seconds for 1 month continuously, without running out of memory, and without the engine shutting down. The memory usage, during this accelerated reliability testing, stabilized at a peak of ~50 MB, for the entire 1 month testing period. Normally, the memory usage should cycle up and down between 15-30 MB.

About Memory Usage

This program does not handle any of the low-level memory allocation and release. Memory allocation and release is done by Java. Java prevents the programmer, from accessing all low-level memory handling routines, making memory safe. This prevents memory leaks and corruption, within the program, or in other running applications. This prevents serious problems, that can occur with C or C++ programs, like memory access violations, page faults, illegal operations, system crashes etc.
Java has automatic memory recycling, called Garbage Collection. Hence, you should see the memory usage, cycle up and down between 15-30 MB. However, Java tends to consume lots of memory, and may not release it efficiently on some Windows and Mac OS systems. The default maximum memory allowable is 64-128 MB. Sometimes, Java will keep requesting more memory, from the operating system. Java may shut down automatically, when the maximum allowable memory (64MB) is exceeded. This is due to incomplete garbage collection, by the Java machine, on certain operating systems.

Make Directory

This task allows you to make a new directory. It will create an entire directory tree if needed.

Example:

if following empty directory exists: c:\test

You can directly create c:\test\subdir1\subdir2\subdir3

MS Access

This feature allows you to run a Microsoft Access macro, or compact a database automatically. This feature is supported for versions Access 97 and Access 2000 only.

MSAccess Path
Enter the Full Path name of the MSAccess Application, in the appropriate Text Box. On a typical installation for Office, the access path is "C:\Program Files\Microsoft Office\Office\MSAccess.exe".

Database Path
Enter the Full Path name of your MSAccess database (ex: c:\..\Test.mdb), or use the browse button to select it.

Options
Select the desired option. You can run a macro, compact a database, or repair a database.

Termination Time
The termination time is the maximum time that your macro can run. Automize will terminate your macro, if it is still running after the maximum allowed time.

MS Excel

This feature allows you to run a Microsoft Excel macro automatically. This feature is supported for versions Excel97 and Excel2000 only.

MSExcel Path

Enter the Full Path name of the MSExcel Application, in the appropriate Text Box. On a typical installation of Office, the Excel path is "C:\Program Files\Microsoft Office\Office\MSExcel.exe".

Workbook Path

Enter the Full Path name of your MSExcel Workbook (ex: c:\..\Test.xls), or use the browse button to select it.

Macro Name

Enter the macro name to run.
If your macro is in the Personal macro workbook, enter the name as:
personal.xls!test,
where test is your macro name.

MS PowerPoint

This feature allows you to run a Microsoft PowerPoint slideshow, or print a presentation automatically. This feature is supported for versions PowerPoint97 and PowerPoint2000 only.

MSPowerpoint Path
Enter the Full Path name of the MSPowerpoint Application in the appropriate Text Box. On a typical installation of Office, the PowerPoint path is "C:\Program Files\Microsoft Office\Office\powerpnt.exe".

File Path
Enter the Full Path name of your MSPowerpoint file (ex: c:\..\Test.ppt) or use the browse button to select it.

Options
Select the desired option. You can run a slideshow or print a presentation.

Termination Time
The termination time is the maximum time that your slideshow can run. Automize will terminate your slideshow, if it is still running, after the maximum allowed time.

MS Word

This feature allows you to run a Microsoft Word macro automatically. This feature is supported for versions Word97 and Word2000 only.

MSWord Path
Enter the Full Path name of the MSWord Application in the appropriate Text Box. On a typical installation of Office, the Word path is "C:\Program Files\Microsoft Office\Office\winword.exe".

File Path
Enter the Full Path name of your MSWord file (ex: c:\..\Test.doc) or use the browse button to select it.

Termination Time
The termination time, is the maximum time, that your macro can run. Automize will terminate your macro if it is still running, after the maximum allowed time.

Windows Service Module

This new windows service module can only be used with versions 10.0 or later. The service module ensures that the engine will not shutdown when the user logs off. This module will also launch the engine on system startup. Versions 7.x-9.x had an older NT service module.

Install and Start the Service
1) Enter your admin level user account. It is recommended to use the same admin level user account that you installed the software with, and normally start the user interface with. The format of the user account entry is .\johndoe
2) Enter your password.
3) Click on 'Install Service' button.
4) Click on 'Start Service' button.

Service Monitoring
The Service is monitored by checking the Activity Log for updates.

IMPORTANT
You should create and schedule an echo task to run at same Frequency as Monitoring Frequency. The echo task will write entries into the Activity Log. If the engine is down/stopped then it will not update the activity log. The monitoring service will restart the engine if engine does not update the activity log.

Default Setting:
Monitoring Frequency = 24 hour.
Maximum non-updates before restaring engine = 1
It will check if Activity Log is updated within last 24 hours.
If Activity Log is not updated, then the engine is stopped and restarted.
Make sure to schedule an echo task to run every 24 hours, 7 days a week.

If you run a lot of tasks, then you can select the following settings:
Monitoring Frequency = 0.5 hour.
Maximum non-updates before restaring engine = 1
It will check if Activity Log is updated within last 30 minutes.
Make sure to schedule an echo task to run every 30 minutes, 7 days a week.

Open Documents

This feature allows you to open any document or folder on your system. This is equivalent to, a user manually clicking on the document icon, to launch or open the document.
You can enter only a program name.
You can also enter a full URL to launch the website in your default browser.
You can also open a folder listing, by entering the folder path.

File or Document
Examples:
c:\winnt\notepad.exe
notepad.exe
notepad
wordpad.exe
c:\test\test.txt
http://www.cnn.com
c:\temp

This field also supports dynamic variables. For example, to open a webpage on local system, which is downloaded every day to the current date, in the date format dd-yy (ex: 06-21.htm), use:
c:\data\$%DATE::dd-yy%$.htm

Print Option
The print option may, either print the entire file automatically, or bring up the default system Print Options Dialog. Automize cannot control this system behavior.

Options

Platform Look - This sets the user interface to look like a Java, Windows, MacOSX, or Unix (motif) interface. The user interface has to be restarted for this setting to take effect.

Select Icon - You can set the default Application icon to be dynamic or static. The user interface has to be restarted for this setting to take effect.

Startup Window - The user interface can be started up maximized, minimized, or as an icon on your desktop. The user interface has to be restarted for this setting to take effect.

Encrypt Passwords - Automize data is viewable in the task table. Hence, your ftp, email, and other passwords are visible. If you wish to encrypt passwords, please select this feature, and click the save Button in the window.

WARNING: Please remember, or write down passwords before encrypting.   Once encrypted, passwords cannot be decrypted, even if you un-select this option. This is to prevent other users from decrypting your encrypted passwords.

Password Encryption

This feature allows you to specify an encryption key used to encrypt passwords for storage

Java Key:
This key is used to encrypt the passwords. If the Java Key field is blank or ‘default’, then the encryption algorithm uses a default key to encrypt all your passwords with.

IMPORTANT: If you change this key, you will have to edit and resave all your passwords!!!

IMPORTANT: If you lose/forget this key, you will get locked out of the user interface if the user interface password is set!!!

VERY IMPORTANT: Make sure the user interface password is cleared before changing this key. After changing this key value, you can set the user interface password again.

NOTE: If you change this key, your passwords can only be used on the current machine. If you move the software (and settings) to a new machine, you will have to enter this same key into this field on the new machine.

NOTE: The advantage of using your own key is that if your data is stored on another system, then no one else on the other system can use your passwords even from our software. They will have to know this key value.

NOTE:
Prior to 10.09, there were 2 encryption schemes:
HITEK – very simple encryption scheme compatible with 5.x/6.x.
JAVA – new encryption scheme which uses java cryptography.

In version 10.09 and later, The HITEK encryption method has been removed as an option.
All passwords previously encrypted with the HITEK method will still be decrypted and used.
All new passwords will be created using the more secure JAVA method.

Password Manager

This utility allows you to manage all your Ftp or SFtp passwords.

Set Password Manager Password
This utility allows the user to view the hidden Ftp or SFtp passwords. Hence, the password manager is protected by a Password.
* If you protect the password manager using a password, then the passwords can be edited and viewed in clear text, when you click the Edit Password button.
* If you do not protect the password manager using a password, then the passwords can be edited, but they are displayed in encrypted form.

Load Passwords
To view the passwords for the file, select the file, and click the 'Load Passwords' button.

Edit Password
You can select the required password, and view or edit it, using the 'Edit' button.

Ping

This task allows you to check if a server is open to connection on the specified port. Each ping result is logged by the task. The uptime statistics for the server:port combination, can be viewed via the Logs / TaskLogs window. The statistics filename is "TaskTitle_statistics".
The "Ping" protocol is not actually implemented. Only a simple socket connection to the specified host is attempted. **For safety purposes, you can only ping a server every 1 minutes.** If the server is pinged twice in less than a minute, the socket connection will not be attempted, and an Exit Code = 5 is returned.

Host
Enter the host name or IP address.

Port
Enter the desired port. Common ports are: SMTP=25, FTP=21, HTTP=80, TELNET=23, POP3=110.

Ping Statistics
When a ping task finishes running; a special log file named "TaskTitle_statistics" is generated. You can view this statistics file by clicking on the filename, in the Logs / TaskLogs Window. This allows you to maintain uptime statistics of your server.

Resetting the Ping Statistics File
Go to the Logs menu, and select the "Manage Logs" menu item. Select the statistics file for your ping task, and click delete. The ping statistics will be cleared, and will start afresh the next time the task is run.

Print Files

The Print task allows you to print text files only.

Directory
Enter the full pathname of the directory, where your files exists. Note that the pathname is case sensitive for unix Ftp servers.

Filename
Enter the filename filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Profiles

A Profile is a very convenient feature. For example: A single email profile can be used in many email tasks. If any setting changes on the email server, you would only need to update the profile. You would not have to manually change this setting in every task.

The following Profiles are used in this software:

Synchronization Profiles

Email Server Profiles

Email Notification Profiles

Ftp Server Profiles

SFtp Server Profiles

Purchase Information

|

When you purchase this software, you will receive registration instructions via email. Follow the registration instructions very carefully, to receive the registration key. The registration key is required to unlock the software.

Please visit our web site at 'http://www.hiteksoftware.com' for latest purchasing details and pricing information. You can purchase, using one of the following methods:
1. Credit Card online. The registration instructions are automatically sent to you via Email.
2. Phone in or fax your order. Please see our web site for details
3. Send a check. Please see our web site for details.
4. Purchase Order. Processing fees will apply for purchase orders.
5. Paypal,
6. Wire transfer.

Multiple copy discounts and site licenses are available. See the Hitek Software web site at 'http://www.hiteksoftware.com' for details.

Regular Expression Tester

This utility helps to test Regex expressions before using them in the filename and rename filename fields of the copy, ftp, sftp and other tasks.

Actual Filename Field - Enter the actual full filename. Example: test.txt
Filename Field – This is what you normally enter in the filename filter field of the copy, ftp and other tasks
Rename Filename Field – Enter the rename string if you need to rename the actual file

For the Rename option **READ THIS**

Registration Key

After you purchase the software, you will receive an Order Confirmation Email, which includes:
1) A website URL, for a Registration Key Generator Form, that you can use to generate your Registration Keys.
2) A Password, which allows you to use the Online Registration Key Generator form.

Using the Online Registration Key generator form
You will need to have the following information available:
Computer Name
Order ID (available in the Confirmation Email sent to you after your Order)
Generator Form password (available in the Confirmation Email sent to you after your Order)

After you fill up and submit the Registration Key Generator form, you will get a Registration Key, required to unlock the software permanently.

If the Online Registration Key Generator is not available or down
If you are having trouble obtaining the registration keys, you can send us an email with the above information, and we will email you back the registration keys for your computer.

LICENSE COUNT
Please follow these criteria in determining the number of license copies that you need to purchase:
a) At least 1 license per system: Every system (PC, Mac, or Unix etc.) that the software is installed on requires a license. i.e. if you install and use the software on 10 systems, you would need to purchase 10 licenses.
b) Multiple users per system, multiple copies of software installed: Sometimes many users share a system, and multiple users may install the software in their own user directory etc. In this case, the total licenses required would equal the number of individual users using the software. i.e. if 10 users install and use the software on the same system, you would need to purchase 10 licenses for that system.
c) Multiple users per system, single copy of software installed: Sometimes many users share a system, but only a single copy of the software is installed. Only one user uses the software. In this case, only a single license is required for that system.
d) Within a Company, Licenses can be transferred from one user to another user, or one computer to another computer, provided the software is first removed completely from the original system.
e) Licenses cannot be transferred from one Company to another Company.

NOTES
1) Every Computer that the software is installed on requires a unique registration key.
2) The Generator Form Password may change from time to time. If you need to Generate Registration Keys few months after your order for any reason, please email us for the new generator form password.
3) Every Computer(s) that you register using the Generated Key(s) into the Register Dialog of the software, is logged to our system.

INFORMATION RECORDED
a) When you open this software up for the first time, the following information is recorded to the Hitek Software installation database:
Product Name
Product Version
Computer Date, Timezone, and Country settings
This helps us track our monthly installs by country and product, and provides us valuable feedback on our marketing efforts.

b) When you purchase a license for the software, you will be given information to generate a license key at our website. If you enter the license key generated from our website to unlock our software, NO registration data is recorded.

c) However, if you register the software using a license key which is not generated from our website, the following information about your computer is recorded into the Hitek Software User Registration Database:
Product Name and Version
Computer Date, Timezone, and Country settings
Registration Key
Computer Name
IP address
Username logged onto system
Operating System

Remote Task

This task allows you to run tasks on a remote system running the version 7.x engine. This task will wait for the task on the remote system to complete. It will detect and report the exit code of the remote system task run.

Host
Enter the host name or IP address of the remote system.

Port
Enter the remote systems engine port. Default engine port in version 7.x is 1967.

Maximum Wait Time
Enter the maximum time (minutes) this task should wait for the Remote system task to complete. If you want the task to wait indefinitely, enter 0. If the remote task does not complete after the wait time set, you will get an exit code = 1. If the remote task completes within the wait time set, the exit code is the value returned by the remote task.

Remote Task
Enter the task title on the remote system. This must be a valid task title on the remote system.

Password
Enter a password, if the user interface on the remote system is password protected.

Renaming Files

The copy, Ftp, Sftp, decrypt tasks etc. have a rename field.
If the rename field is filled, the rename is done using regular expression.
You can use 3 different options to rename:

Option 1: Filename field is used as the FIND string.
First the filename field entry is compared against the actual filename using regular expression Find method. If a match is found, then the Filename field pattern entered is replaced with the entry in the Rename Filename field.

Option 2: Rename field specifies FIND string using <REPLACE>.
Rename field format:
find_string<REPLACE>replace_string
.txt.gpg<REPLACE>.txt
[0-9]<REPLACE>number

First the filename field entry is compared against the actual filename using regular expression Find method. If a match is found, then the rename field is parsed and the 'find_string' is replaced with the 'rename_string'.

NOTE1: find_string can be a regex pattern
NOTE2: Option 2 gives more flexibility. Here you can specify the filename filter to find files in the filename field. Then, in the rename field, use completely different 'find' and 'replace' strings.

Option 3: Rename field specifies group regex pattern search/replace using <GROUP>.
Rename field format:
regex_(group)_pattern<GROUP>group_formatted_string

([0-9])<GROUP>group1.group2

( ) is the regex grouping pattern
The string **group1** will be replaced by the first group corresponding to the regex group pattern.
The string **group2** will be replaced by the second group corresponding to the regex group pattern.

Examples:

Option 1:
Actual Filenames in folder = Atest.txt.gpg , Btest.txt.gpg
Filename field entry = .txt.gpg
Rename field entry = .txt
resulting filenames = Atest.txt & Btest.txt
i.e. Both files pass the filename filter and will be processed.

Option 2:
Actual Filenames in folder = Atest.txt.gpg , Btest.txt.gpg
Filename field entry = Atest
rename field entry = .txt.gpg<REPLACE>.txt
resulting filename = Atest.txt
Here only Atest.txt.gpg passes the filename filter. Btest.txt.gpg is not processed

Option 3:
Actual Filename in folder = 9-5-7.txt
Filename field entry = .txt
rename field entry = ([0-9])<GROUP>group3_group1.txt

resulting filename = 7_9.txt

here group1 = 9, group2 = 5, and group3 = 7 because of the regex group pattern ([0-9])

Scheduler Settings

Engine server port

The software contains two separate programs. An administrator user interface, and a scheduler engine program, which runs in the background. The administrator user interface sends data to the Scheduler Engine Server Socket.

It is very unlikely that you will need to change this default port value. This default value need to be changed **ONLY** if another server is running on your system, and the other server is listening on this port. If you change the default port value, **BOTH** the administrator user interface and scheduler engine program need to be restarted. This Scheduler engine running in the background, allows for use as Service on Unix, Linux, Solaris, Windows NT , 2000, XP.

Launch wait time

This is the maximum time (in seconds) that the user interface waits for the Scheduler engine to respond on startup. If your system is slow, or you have many tasks and schedules, the scheduler engine may take longer than the default (60 seconds) to startup. In this case, you may get a message: "the scheduler engine has failed to startup". In reality, the scheduler engine may have started, but only after 60+ seconds. If you receive this message often, increase this setting to 120 seconds.

Scheduler sleep time (ms)

Every time a schedule is triggered, the scheduler will set the next schedule and run the task. You can set the scheduler to wait for a few milliseconds (1000 milliseconds = 1 second), before setting the next schedule. This helps to prevent multiple triggers on some older systems. The default is 100 milliseconds, but you can choose to set it between 0-60000 milliseconds (0-60 second). If you set a sleep time of 5000 milliseconds (5 seconds), the task will start running 5 seconds after the schedule is triggered.

Time Zone

The scheduler engine automatically detects the system time zone settings, indicated by the 'default' value. If your schedules are 1 hour off, you may have to reset the time zone to the correct value.

Thread Priority

All tasks are multithreaded in Version 7.x. Every task runs as a separate thread in the engine. You can set the thread priority to be minimum, normal, or maximum. It is recommended that you use either minimum or normal (default) priority. Maximum priority is NOT recommended.

Check for multiple triggers within 30 seconds

In version 5.x, two users had reported multiple task instances (2-6 instances), being triggered at the same time. The cause of this multiple triggers is unknown. Select this option, if you happen to experience this multiple trigger problem. This option does not apply, when the scheduling period is by 'Second'. If you select this option, and two instances of the same task are triggered within 30 seconds of each other, then the second instance is cancelled.

Scheduling Error

In version 5.x and earlier, when a scheduling error occurred, the task schedule was cancelled, and then rescheduled. In 6.x, more corrective options are provided.
1) No corrective action is taken.
2) Reschedule only the task schedule, that led to the error.
3) Reschedule all the task schedules.
4) Restart the scheduler engine. In this case, all pending schedules are cancelled, and a new instance of the scheduler engine is launched. However, if any task is still running in the current instance of the engine, it will continue to run. The current instance of the engine will shut down, after all running tasks have finished, or when the maximum wait time (default = 60 minutes) is reached.

Wait for tasks to Finish before Engine shuts down

When the engine is shut down, all pending schedules are cancelled. However, if any task is running in the current instance of the engine, it will continue to run. The current instance of the engine will shut down, after all running tasks have finished, or when the maximum wait time (default = 60 minutes) is reached.

Maximum wait time before engine stops

When the engine is stopped, the current instance of the engine will shut down, after all running tasks have finished, or when this maximum wait time (default = 60 minutes) is reached. You can change this default value as required.

Schedule Table

The Schedule List contains the data for all the scheduled tasks. You can edit the schedule using the Edit button. This will load the parameters correctly into the schedule window. You can edit the schedule settings in the schedule window.

To **edit** a task schedule, select the task in the Schedules list, and click the "Edit" button.

To **delete** a task schedule, select the task in the Schedule list, and click the "Delete" button.

To **run** a task, select the task in the Schedule list, and click the "Run" button..

Suspend / Resume Schedule

This task is available in Automize only

This task allows you to suspend or resume a schedule.
To suspend/resume all schedules, enter ALL for the task title.

NOTE:
When you suspend or resume a single schedule, the schedule enabled setting is written to disk, and will retain its state during subsequent instances of the engine. In the Schedule tab, the check mark in the schedule tree and the Next run time in the schedule table will match.

When you suspend or resume ALL schedules, the schedules enabled settings are not written to disk. When the engine is restarted, the individual schedules original enabled/disabled setting will be retained. In the schedule tab, the check marks in the schedule tree will show the original setting, but the Next run time in the schedule table will show the schedule status for the current engine instance.

Scheduling Tasks

Tasks can be scheduled by the second, minute, hour, day, week, or month. To schedule a task, select an existing task from the Task list, and click on the schedule button.

Scheduling by the Second
Tasks can be scheduled by the second. i.e. every x = 15 seconds etc. Scheduling tasks at this rate, is very likely to consume a large amount of your computer's time and resources. When you schedule by the second, the task will run every x seconds, from the time the task was created. If you restart the scheduler, the task will run first after x seconds, and then repeat every x seconds.
When scheduling by the Second, the following fields are used: Run Hours, Run Days.
When scheduling by the Second, the following fields are ignored: Run Months, Run Dates, Run Weeks.

Scheduling by the Minute
When you schedule by the minute, two rules apply:
1. If you schedule every 1, 2, 3, 4, 5, 6, 10, 12, 15, 20 or 30 minutes (i.e. divisible by 60), then the task will run at the same time each hour. Example, if you schedule a task to run every 15 minutes, and you set the run 'minute' = 0, it will run at 0, 15, 30 and 45. If you set the run 'minute' = 5, it will run at 5, 20, 35 and 50.
2. If the period is not divisible by 60, the task will run every x minutes. If you restart the scheduler, the task will run first after x minutes, and then repeat every x minutes.
When scheduling by the Minute, the following fields are used: Run Hours, Run Days.
When scheduling by the Minute, the following fields are ignored: Run Months, Run Dates, Run Weeks.

Scheduling by the Hour
When you schedule by the hour, two rules apply:
1. If you schedule every 1, 2, 3, 4, 6, 8,12, or 24 hours (i.e. divisible by 24), then the task will run at the same time each day. Example, if you schedule a task to run every 6 hours, and you set the run 'minute' = 0, it will run at 12AM, 6AM, 12PM, 6PM everyday.
2. If the period is not divisible by 24, the task will run every x hours. If you restart the scheduler, the task will run first after x hours, and then repeat every x hours, at the minute of the hour you set.
When scheduling by the Hour, the following fields are used: Run Hours, Run Days.
When scheduling by the Hour, the following fields are ignored: Run Months, Run Dates, Run Weeks.

Scheduling by the Day
When you schedule by the day, the task will run on the days selected, at the exact time of the day you set.
When scheduling by the Day, the following fields are used: Run Days, Run Weeks.
When scheduling by the Day, the following fields are ignored: Run Hours, Run Months, Run Dates.

Scheduling by the Week
When you schedule by the week, the task will run every week, on the day selected, and at time of the day you set.
When scheduling by the Week, the following fields are used: Run Days, Run Weeks.
When scheduling by the Week, the following fields are ignored: Run Hours, Run Months, Run Dates.

Scheduling by the Month
When you schedule by the month, the task will run for the months that you have selected. The task will run, on the date of the month, and the time of the day you set. If you require the task to run, on the last day of the month, select the date as "32".
When scheduling by the Month, the following fields are used: Run Months, Run Dates.
When scheduling by the Month, the following fields are ignored: Run Hours, Run Days, and Run Weeks.

Start Date
This field is a Start Date filter. If the calculated next run time, is prior to the Start Date, the next run time is incremented, till it is after the Start Date. This time does not indicate the first run time.

End Date

This field is an End Date filter. If the calculated next run time, is past the End Date, the scheduled is Cancelled. This time does not indicate the Last run time.

Scheduling Tasks : More Options

Run Weeks Of Month – Select the weeks of the month that this task will run. Default value is all 5 weeks of month are selected.

Run Weeks of Year – There are 4 types of entries:

a) Odd weeks of Year

b) Even weeks of Year

c) Specify weeks manually separated by ,
Example: 1,4,9,11,35,36,47

d) Specify start week of year and period between weeks separated by -
Example: 0-3 will schedule weeks 3,6,9,12,15...
Example: 1-4 will schedule weeks 1,5,9,13,17...

NOTE1: Either Run Weeks of Month or Run Weeks of Year must be used. i.e. either entry can be blank, but both entries cannot be blank together.
NOTE1: Run Weeks of Month and Run Weeks of Year are used in **UNION**.
i.e. If Run Weeks of Month = 1,2 & Run Weeks of Year = Even , the schedules will run for the 1st and 2nd week of every month AND will also run for the 3rd,4th, 5th week of any month that also happens to be an Even week of the year.

Script

This feature is only available in Automize Enterprise.

This feature allows you to write scripts to enhance the capabilities of our software. You can write scripts in any scripting language that is supported by the JSR-223 javax.scripting interface.

The following is a list of supported languages shipped with the default installation:
java - java based scripting is supported using BeanShell which is included in Automize.
javascript - javascript support is built into Java itself using Mozilla Rhino.

Additionally, you can install the following languages:
groovy, python, ruby, Tcl, AWK, Pnuts etc..
You can find the entire list by searching (google) for 'JSR-223 supported scripting languages'.

Installing New Scripting Language
1) Download the .jar file for the scripting language that implements JSR-223.
2) Copy the .jar file into the 'Automize10\jre\lib\ext' folder.
3) Restart the Automize user interface and engine.
4) Create a new Script task.
5) You should see your scripting language listed in the Script Settings Tab

Example Scripts
Example java scripts are located in the 'example_scripts' folder within the main install folder. Please read the examples scripts via the 'Examples' button. Our website will also have more examples of scripts that you can use.

Creating a Script
Type in the script in the upper text window. Create small scripts for each task. Each script should be less than 50-75 lines. This will help in debugging. Please read the examples scripts provided, using the 'Examples' button. Our website will also have more examples of scripts that you can use.

Testing a Script
Click on the 'Test Script' button. Errors or Output will be displayed in the lower text window.

Set Task Parameter

This feature allows you to set a task parameter from an external program, script, batch file, or command-line. Select the task, and parameter value, from the drop down box. The command-line, and the working directory required to set the task parameter, will be automatically generated in the text fields. In the command-line code, you would need to replace the term 'parameter_value', with the value you need for this parameter. You can use the "Copy to Clipboard" feature, to copy these fields to clipboard, for use in your scripts, or programs.

IMPORTANT: The user interface should be closed for this to work correctly. If the user interface is open, and the user then changes any setting related to the Task Table, the changes made from the external program will be overwritten

Batch or script file example:

```
cd [working_directory]
[commandline]
```

where [working_directory] and [commandline] are generated automatically in the Task Runner fields.

Example (on windows):

```
cd c:\Data\Java\Automize
"C:\program files\Automize\jre\bin\javaw.exe" -cp .;hitek7.jar com.hitek.engine.utils.SetParameter Message 1 parameter_value
```

External program:
Use this command-line, from your external program, to launch the task. Enter the working directory, or the Java executable will not be able to run the required tasks.

From within Automize:
Use this command-line, from the Command or Windows Command tasks, to set the task parameter. Enter the working directory, or the Java executable will not be able to run the required tasks.

Set Task Parameter

This task allows you to set/change a parameter in another task.

Enter the task title, parameter number, and the parameter value to set. The parameter value field supports variables.

Example usage:
Dynamically change task parameters during a DirLoop task. i.e. For every run of the DirLoop task you can set a new filename value for a copy task parameter based on the DirLoop variable.

Set User Variable

This feature allows you to set a user variable.

Variable Name - Enter new variable name

Variable Value - Enter new value for the variable

SFtp

The sftp task allows you to transfer files from and to a SSH2 server. Also, the SSH2 server should support SFtp subsystem. Not all SSH2 servers support SFtp.

SFtp Profile

Select the SFtp profile for the SFtp server, you need to connect to. You should have previously created a SFtp profile, using the SFtp Profiles menu item.

SFtp Option

Use 'Get' to transfer files from the server to your local system. Use 'Put' to transfer files from your local system to the server. Use 'Delete' to delete files from the server.

Remote Directory

This is the most difficult parameter to enter correctly. Execute privilege is required for the remote directory, for all SFtp functions. Read privilege is required for SFtp Get. Write privilege is required, for the SFtp Put function, and for the SFtp Delete function. Note that the path is case sensitive for Unix SFtp servers. **If you are transferring files from the Default SFTP login directory, you can leave this field blank.**

Local Directory

Enter the full pathname of the local directory, where you need to put files, or get files.

Filename

Enter the filename filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Folder Path Filter

Leave this field blank if you want all folders to be processed. If you need specific folders to be processed, then enter the folder name or path, using the following wildcard rules. If a folder passes the filter, then all its subfolders will also pass the filter.

Simple Example: Folder = c:\test , subfolders = john, john\old, sam, sam\old, joe, joe\old
Folder path filter = john   . john and john\old are processed
Folder path filter = jo   . john, john\old, joe, joe\old are processed
Folder path filter = old   . sam\old, john\old, joe\old are processed
Folder path filter = test   . all files in c:\test and its sub folders are processed
Folder path filter = john\old.   only john\old is processed.
Folder path filter = c:\test\john   . john and john\old are processed

Rename Filename

Please see Rename Help

Append date/time To filename

You can choose to append either the date, or the date and time, or custom code. If your filename has an extension, the date and time, is appended before the extension. This feature is useful for archive purposes.

Backup to Local Directory

Enter the full pathname of the local directory, where you need to backup files. Please note that the Backup to local directory option does not support subdirectories. All files will be placed in the backup directory, without creating new subfolders in it.

Backup to Remote Directory

Enter the full pathname of the remote directory, where you need to backup files. Please note that the Backup to remote directory option does not support subdirectories. All files will be placed in the backup directory, without creating new subfolders in it.

Use Staging

If you select this option, the file is first transferred to the local (get), or remote (put) folders entered. This folder acts as a staging folder. After the Ftp transfer is complete, the file is then moved, to the final destination folder. This option is useful to prevent processes from working on files, while the Ftp transfer is in progress. In the staging field, you should enter the full pathname of the final destination directory. Please note that the staging option does not support subdirectories. All files will be placed in the final destination directory, without creating new subfolders in it.

Umask Permissions

The umask has be entered in 4-digit octal format. The umask command is used to set the permissions of all files that are put onto the server. Leave this field blank, to use the default Sftp server permissions, for the transferred files. The Umask is the reverse of the normal file permissions, i.e. 0777^umask = permissions.
Examples:
0077 = -rwx------ (all permissions for user, no permissions for group and world)
0000 = -rwxrwxrwx (all permissions for user, group and world)
0022 = -rwrxr-xr-x (all permissions for user, read+execute permissions for group and world)
0033 = -rwxr--r-- (all permissions for user, read permission for group and world)
0011 = -rwxrw-rw- (all permissions for user, read+write permissions for group and world)

Delete Source file

You can choose to delete the source file, after a successful transfer. If the SFtp transfer fails, the source file will not be deleted.

Transfer with temporary extension

The file is transferred with a temporary extension, and then renamed to the desired name, after the transfer is complete. This is useful, if you have automated scripts/programs, which continuously work on files, with certain extensions. This option ensures that the automated script, will not work on a partially transferred file.

Log Transfer

You can choose to log the transfer into the Ftp Log.

Transfer Modified files

If you choose this option, files are transferred only if the source file is newer than the target file.

Transfer Modified files using Ftp Log

If you choose this option, files are transferred only if the source file date is newer than the source file date entry in the Ftp Log. For this option to work correctly, you should also enable the 'Log Transfer' option. If the source file path is not found in the Ftp log, then the file is always transferred.

Place files in new subdirectory

If you select "None", no new subdirectory is created within the Target Directory.
If you select Date, a new subdirectory based on the date is created within the Target Directory. All files are placed in this directory.
If you select Date, a new subdirectory based on date and time is created within the Target Directory. All files are placed in this directory.

Transfer Mode

This option sets the transfer mode for current operations. The valid modes are:

BINARY - Files are transferred in binary mode and no processing of text files is performed (default mode).

ASCII/TEXT - For servers supporting version 4+ of the SFTP protocol files are transferred in text mode. For earlier protocol versions, the files are transferred in binary mode, but the client performs processing of text. If files are written to the remote server, the client ensures that the line endings conform to the remote EOL. The EOL mode is set using the setRemoteEOL setting below. For files retrieved from the server, the EOL policy is based upon System policy as defined by the "line.seperator" system property.

Set Remote EOL

When connected to servers running SFTP version 3 (or less) the remote EOL type needs to be explicitly set because there is no reliable way for the client to determine the type of EOL for text files. In versions 4+ a mechanism is provided and this setting is overridden.

Valid values for this method are EOL_CRLF (default), EOL_CR, and EOL_LF.

SFtp Browser

ToolBar Buttons

Create New SFtp Connection
Click this button to create a new SFtp connection. A connect dialog will be displayed. In this dialog, you can either create new settings, or connect to an existing SFtp site.

Close the current SFtp connection
Click this button to logout, and disconnect, from the current SFtp connection.

Create new SFtp task
Click this button, to create a new SFtp task. If you have a SFtp connection open, some of the sftp settings will be transferred to the new task dialog, for your convenience.

Open new SFtp browser
Click this button to open a new SFtp Browser. This allows you to connect, and view multiple SFtp sites simultaneously.

Transfer Panel

The local directory listing is on the left panel, and the remote directory listing is on the right panel

SFtp-j

The sftp-j task allows you to transfer files from and to a SSH2 server. Also, the SSH2 server should support SFtp subsystem. Not all SSH2 servers support SFtp.

SFtp-j Profile
Select the SFtp-j profile for the SFtp server, you need to connect to. You should have previously created a Sftp-j profile.

SFtp Option
Use 'Get' to transfer files from the server to your local system. Use 'Put' to transfer files from your local system to the server. Use 'Delete' to delete files from the server.

Remote Directory
This is the most difficult parameter to enter correctly. Execute privilege is required for the remote directory, for all SFtp functions. Read privilege is required for SFtp Get. Write privilege is required, for the SFtp Put function, and for the SFtp Delete function. Note that the path is case sensitive for Unix SFtp servers. **If you are transferring files from the Default SFTP login directory, you can leave this field blank.**

Local Directory
Enter the full pathname of the local directory, where you need to put files, or get files.

Filename
Enter the filename filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Folder Path Filter
Leave this field blank if you want all folders to be processed. If you need specific folders to be processed, then enter the folder name or path, using the following wildcard rules. If a folder passes the filter, then all its subfolders will also pass the filter.
Simple Example: Folder = c:\test , subfolders = john, john\old, sam, sam\old, joe, joe\old
Folder path filter = john . john and john\old are processed
Folder path filter = jo . john, john\old, joe, joe\old are processed
Folder path filter = old . sam\old, john\old, joe\old are processed
Folder path filter = test . all files in c:\test and its sub folders are processed
Folder path filter = john\old. only john\old is processed.
Folder path filter = c:\test\john . john and john\old are processed

Rename Filename
Please see Rename Help

Append date/time To filename
You can choose to append either the date, or the date and time, or custom code. If your filename has an extension, the date and time, is appended before the extension. This feature is useful for archive purposes.

Backup to Local Directory
Enter the full pathname of the local directory, where you need to backup files. Please note that the Backup to local directory option does not support subdirectories. All files will be placed in the backup directory, without creating new subfolders in it.

Backup to Remote Directory
Enter the full pathname of the remote directory, where you need to backup files. Please note that the Backup to remote directory option does not support subdirectories. All files will be placed in the backup directory, without creating new

subfolders in it.

Use Staging

If you select this option, the file is first transferred to the local (get), or remote (put) folders entered. This folder acts as a staging folder. After the Ftp transfer is complete, the file is then moved, to the final destination folder. This option is useful to prevent processes from working on files, while the Ftp transfer is in progress. In the staging field, you should enter the full pathname of the final destination directory. Please note that the staging option does not support subdirectories. All files will be placed in the final destination directory, without creating new subfolders in it.

CHMOD Permissions

The chmod has be entered in 3-digit format. The chmod command is used to set the permissions of all files that are put onto the server. To use the default Sftp server permissions for the transferred files, leave this field blank.
700 = -rwx------ (all permissions for user, no permissions for group and world)
777 = -rwxrwxrwx (all permissions for user, group and world)
755 = -rwrxr-xr-x (all permissions for user, read+execute permissions for group and world)
744 = -rwxr--r-- (all permissions for user, read permission for group and world)
766 = -rwxrw-rw- (all permissions for user, read+write permissions for group and world)

Delete Source file

You can choose to delete the source file, after a successful transfer. If the SFtp transfer fails, the source file will not be deleted.

Transfer with temporary extension

The file is transferred with a temporary extension, and then renamed to the desired name, after the transfer is complete. This is useful, if you have automated scripts/programs, which continuously work on files, with certain extensions. This option ensures that the automated script, will not work on a partially transferred file.

Log Transfer

You can choose to log the transfer into the Ftp Log.

Transfer Modified files

If you choose this option, files are transferred only if the source file is newer than the target file.

Transfer Modified files using Ftp Log

If you choose this option, files are transferred only if the source file date is newer than the source file date entry in the Ftp Log. For this option to work correctly, you should also enable the 'Log Transfer' option. If the source file path is not found in the Ftp log, then the file is always transferred.

Place files in new subdirectory

If you select "None", no new subdirectory is created within the Target Directory.
If you select Date, a new subdirectory based on the date is created within the Target Directory. All files are placed in this directory.
If you select Date, a new subdirectory based on date and time is created within the Target Directory. All files are placed in this directory.

Transfer Mode

This option sets the transfer mode for current operations. The valid modes are:

BINARY - Files are transferred in binary mode and no processing of text files is performed (default mode).

ASCII/TEXT - For servers supporting version 4+ of the SFTP protocol files are transferred in text mode. For earlier protocol versions, the files are transferred in binary mode, but the client performs processing of text. If files are written to the remote server, the client ensures that the line endings conform to the remote EOL. The EOL mode is set using the setRemoteEOL setting below. For files retrieved from the server, the EOL policy is based upon System policy as defined by the "line.seperator" system property.

Set Remote EOL
When connected to servers running SFTP version 3 (or less) the remote EOL type needs to be explicitly set because there is no reliable way for the client to determine the type of EOL for text files. In versions 4+ a mechanism is provided and this setting is overridden.

Valid values for this method are Default, line feed, carriage return, line feed + carriage return. With Default option, the server determines the EOL method.

Sftp Monitor

You can monitor a remote folder for changes, or other criteria. This task can be used in two ways:
1) If a change is detected in this task, you can run another task, or
2) The Sftp Monitor task can also be embedded within a Chain task. In the chain task, it is used for conditional task processing.

The SFtp Monitor task also outputs a variable called 'TaskTitle::Filenames". This variable contains the list of all files, that satisfy the monitoring criteria. The format of this variable is #file1^#file2^#file3, where file1,file2,file3 are the names of the files that have been modified. You can use this variable directly as the filename in other copy, ftp, delete tasks to copy/ftp/delete only these files. This variable is only updated if the filename, filesize, or filedate criteria options are selected.

Directory
Enter the remote folder that you wish to monitor

Filename
Enter the filename filter for the files you wish to copy using the following wildcard rules.
This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Task To Run
Select the desired task to run, if any of the selected criteria are met. If you want to use this task in a chain for conditional processing, select 'None'. You can also select multiple tasks, to run in sequence, or to run simultaneously. To run tasks in sequence, use taskTitle1|taskTitle2|taskTitle3. To run tasks simultaneously, use taskTitle1&taskTitle2&taskTitle3.

Options
The Task will be run, if any of the following selected criteria are satisfied. You can select 1 or more criteria.
1) Any change is found in the directory, since the last run of this task. This option does not update any of the dynamic variables.
2) Files which satisfy the Filename criteria exists
3) Files which satisfy the Filename criteria, as well as file size criteria
4) Files which satisfy the Filename criteria, as well as date modification criteria
5) Total number of files in directory are greater/less than certain value. This option does not update any of the dynamic variables.

SFtp Profile
Select the SFtp profile for the SFtp server, you need to connect to. You should have previously created a SFtp profile, using the SFtp Profiles menu item.

SFtp/SSH2 Profiles

You can create multiple SSH2/SFtp profiles for easy use in your tasks. To create a new profile, click on the "New Profile" button. Make sure to click the Save button after you enter the site parameters.

Hosts TAB

This tab allows you to set the main connection parameters to the site.   Most users only need to use the options on this tab.

Profile

Enter a name that describes this SFtp site.

Host

Enter the host name or IP address.

Port

Enter the desired port. Default SSH2 port = 22.

Username

Enter your username.

Password

Enter your password. If you are using public/private key authentication, enter your pass phrase in this field.

Use Pvt Key Authentication

Select this option if you need to use pvt key authentication.

Pvt Key file

Enter the full path to the private key file. If your SSH2 server is an OpenSSH server, you need to convert the public key to the OpenSSH format. Then send this key to the SSH server administrator.

Local Directory

Enter the full path to the local directory. This setting is only used by the SFtp Browser. This directory will be shown, after a connection is made in the manual SFtp Browser. This setting is not used in the automated SFtp tasks.

Remote Directory

Enter the full path to the remote directory. This setting is only used by the SFtp Browser. This directory will be shown, after a connection is made, in the manual SFtp Browser. This setting is not used in the automated SFtp tasks.

Firewall

Enter the Firewall profile name if you are behind a Firewall / Proxy Server. If you can directly connect to the internet, select NONE for the firewall

Firewall TAB

This tab allows you to connect to a SFtp server, if you are behind a Firewall / Proxy Server. These settings are similar to your Web Browser settings. You can copy the settings from your default browser's Proxy settings. If your web browser is not setup, please check with your Network Administrator, for the Hostname and Port settings required.

Host

Enter the host name or IP address of your proxy server / firewall.

Port

Enter the desired port. Default port for http proxy is normally 80, but every proxy server can be configured to use a different port. So, you should check with your Network Administrator, for the Hostname and Port settings required.

Username
Enter your username information, only if your proxy server / firewall requires authentication.

Password
Enter your password information, only if your proxy server / firewall requires authentication.

Type
Enter the firewall type. (http / socks4 / socks5). Check with your Network Administrator, if you are not sure about this.

Connection TAB

This tab allows you to set advanced connection properties for the SSH2 Server. Most users do not need to change any of these settings.

Context
There are two separate SSH protocol versions available, the original SSH1, and the current version SSH2. The two protocols are not compatible, and you cannot connect to an SSH1 server using an SSH2 client, and vise versa. SSH1 is considered deprecated. We recommend that you use SSH2 at all times.

Cipher
This sets the encryption algorithm, for server < client data streams.

Compression
This sets the compression method, for server < client data streams.

Message Authentication
This sets the message authentication algorithm, for each server < client data stream.

Public Key Method
This sets the public key method used, for private/public key based authentication.

Key Exchange Method
This sets the key exchange method used, for private/public key based authentication.

Maximum Channels
This sets the maximum number of channels that are allowed open at any one time.

Maximum Packet Length
This sets the maximum packet size supported by the transport layer. The default is about 131K, which should support most SFtp servers.

SFtp Synchronize Folders

This task allows you to synchronize a local folder on your workstation, with a folder on your SFtp server. This is a simple task, which is based on synchronization functionality, provided by our supplier SFTP api. Because this task is dependent on the functionality of the supplier SFTP api, it is limited in its options. Additional options will only be possible if our SFTP api supplier incorporates more options.

Option

Two synchronization modes are supported.
1) GET
All unpaired files and folders in the remote folder are copied to the local folder.
Older files in the local folder are replaced with their newer counterpart in the remote folder.

2) PUT
All unpaired files and folders in the local folder are copied to the remote folder.
Older files in the remote folder are replaced with their newer counterpart in the local folder.

Delete unpaired files on target
Use this option carefully since, this will remove all unpaired files and folders in the Target.

Remote Directory
Enter the full pathname (case sensitive) of the remote directory on the SFTP server.

Local Directory
Enter the full pathname of the local directory. Note that the pathname is case sensitive on Unix systems.

Include subdirectories
If you select this option, all subdirectories and files within the Source Directory are copied to the Target Directory. If a subdirectory does not exist within the target folder, then it is automatically created.

Umask Permissions
The umask has to be entered in 4-digit octal format. The umask command is used to set the permissions of all files that are put onto the server. Leave this field blank, to use the default Sftp server permissions, for the transferred files. The Umask is the reverse of the normal file permissions, i.e. 0777^umask = permissions.
Examples:
0077 = -rwx------ (all permissions for user, no permissions for group and world)
0000 = -rwxrwxrwx (all permissions for user, group and world)
0022 = -rwrxr-xr-x (all permissions for user, read+execute permissions for group and world)
0033 = -rwxr--r-- (all permissions for user, read permission for group and world)
0011 = -rwxrw-rw- (all permissions for user, read+write permissions for group and world)

SSH2

This task allows you to login to a SSH2 server, and issue commands. It will try and detect the presence of the command prompts, before sending the next command. If these prompts are not detected within the timeout value set, it will send the next command. This task is multithreaded, so you can run multiple SSH2 tasks simultaneously. This is useful if you are running long scripts on the server.

Ssh2/SFtp Profile

Select the Ssh2/SFtp profile for the SSh2/SFtp server. You should have previously created a SFtp profile, using the SFtp Profiles menu item.

Polling Period

Enter the desired polling time. For each SSH2 command sent, the module will poll the connection, at the Polling interval, for the reply string from the server. If the last 10 characters (default value) of the reply string from the server contain the shell prompt, then the next command is sent. This polling period should be less than the command time.

SSH2 Prompt

This is usually the last character that the server outputs, when it is done processing the current shell command. This value normally depends on the system shell you are connecting to. The task will look for shell prompt character, in the last 10 characters (default value) of the reply string from the server. As soon as this character is found, the next command is sent. If however, the shell prompt character is not found, within the command time set, the next command is then sent.

Prompt Characters

This module waits for the shell prompts, before sending the next command. It will search the last reply string from the SSH2 server, for the shell prompt. You can define how the module searches for these prompts. If you select Prompt Characters = 10 (default value), then the module will search the last 10 characters of the last reply string. If the shell prompt is found in these 10 characters, then the next command is sent. Some servers send some characters after the shell prompts (Example: MS Telnet server). In this case, you may have to increase this value to 1000, to detect the shell prompt. As soon as this character is found, the next command is sent.

Command Time

Enter the maximum time allowed for each command. If this command time has been exceeded, and the server has not replied back, then the next command is sent.
You can also set a custom command time for each step using the following case sensitive format:
STEPTIME=2^pwd
A step time = 2 minutes is used and the command is 'pwd'.

Command

Enter your commands, 1 command per line. Always enter the exit command on the last line. This field also supports dynamic variables. Using dynamic variables, you can enter a command which will be dynamically resolved at runtime. example: cd /users/$%USERNAME%$

Example of Commands to enter

```
ls -lt
cd /user/test
STEPTIME=5^./runTestScript.scr
pwd
exit
```

SSH2 Add-on Package

The SSH2 add-on package allows you to automate SSH2 and SFTP tasks. This package requires a separate license. This add-on package requires a separate download. Please visit our website to download the add-on package. Detailed instructions to install the package are provided on our website. The package is normally available via the Automize download page.

SSH2-j

This task allows you to login to a SSH2 server, and issue commands. It will try and detect the presence of the command prompts, before sending the next command. This task is multithreaded, so you can run multiple SSH2 tasks simultaneously. This is useful if you are running long scripts on the server.

Ssh2/SFtp-j Profile
Select the profile for the SSh2/SFtp server. You should have previously created a SFtp-j profile, using the SFtp-j Profiles menu item.

Delay Time
Enter the desired delay time between commands.

SSH2 Prompt
This is usually the last character that the server outputs, when it is done processing the current shell command. This value normally depends on the system shell you are connecting to.

Command Time
Enter the maximum time allowed for each command. If this command time has been exceeded, and the server has not replied back, then the next command is sent.

Command
Enter your commands, 1 command per line. This field also supports dynamic variables. Using dynamic variables, you can enter a command which will be dynamically resolved at runtime. example: cd /users/$%USERNAME%$

Example of Commands to enter

```
ls -lt
cd /user/test
./runTestScript.scr
pwd
```

Subdirectory Loop

The Subdirectory loop task allows you to monitor subdirectories within a Directory. For every subdirectory that satisfies the criteria, the Subdirectory loop task will run another task. This other task can be a processing task that works on the current subdirectory. See the example at the end of this help file on how to use this task.

The Directory Loop task sets various variables for use in the other tasks like:
CurrentSubdirectoryName, CurrentSubdirectoryPath

Directory
Enter the directory where your files are located.

Subdirectory name filter
Enter the Subdirectory name filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Task To Run
Select the desired task to run, if any of the selected criteria are met. If you want to use this task in a chain for conditional processing, select 'None'. You can also select multiple tasks to run in sequence, or to run simultaneously. To run tasks in sequence, use taskTitle1|taskTitle2|taskTitle3. To run tasks simultaneously, use taskTitle1&taskTitle2&taskTitle3.

Maximum loops
This is the maximum number of files that will be processed in each run. If 100 files pass the criteria, you can process the first 5 files in the list. Enter 0 to process all files that pass the criteria.

Delay between loops
You should enter a delay time between loops. The delay must be long enough to ensure the processing task is completed before the next instance of the processing task is called.

Options
The Task will be run, if the selected criteria is satisfied. You can select 1 criteria.
1) Subdirectories hich satisfy the Filename criteria exists
2) Subdirectories which satisfy the total file size criteria. The file size is entered in kb, but you can specify 0.001 to 1000000 kb.
3) Subdirectories which satisfy the date modification criteria.

Example
Create the following 2 tasks:

1) Echo task
Task Title = echoTest
message = The current filename is $%loop::CurrentSubdirectoryName%$

2) Directory Loop
Task Title = loop
directory = [temp or test directory]
Subdirectory name filter =
Maximum loops = 5
Delay = 15
Task To Run = echoTest
Criteria = Run Task if Subdirectories exists

Run the Subdirectory Loop task. The Subdirectory loop task will run the Echo task for the first 5 subdirectories in the directory.

In real situations, you would use either a copy task or a delete task etc.. For the local directory field in these tasks, use: $%loop::CurrentSubdirectoryPath%$

Subdirectory Monitor

The Subdirectory Monitor task allows you to monitor subdirectories within a Directory.

The Subdirectory Monitor task also outputs variables like 'TaskTitle::SubdirectoryNames' and 'TaskTitle::SubdirectoryPaths'. These variable contains the list of all subdirectories that satisfy the monitoring criteria. The format of this variable is #subdir1^#subdir2^#subdir3, where subdir1, subdir2, subdir3 are the names/paths of the subdirectories that have been modified. You can use this variable directly as the folder filter in other tasks (such as: copy, ftp, delete), to process only these subdirectories.

Directory
Enter the directory where your files are located.

Subdirectory name filter
Enter the subdirectory name filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Task To Run
Select the desired task to run, if any of the selected criteria are met. If you want to use this task in a chain for conditional processing, select 'None'. You can also select multiple tasks, to run in sequence, or to run simultaneously. To run tasks in sequence, use taskTitle1|taskTitle2|taskTitle3. To run tasks simultaneously, use taskTitle1&taskTitle2&taskTitle3.

Options
The Task will run, if any of the following selected criteria are satisfied. You can select 1 or more criteria.
1) Subdirectories which satisfy the name criteria exists
2) Subdirectories which satisfy the name criteria, as well as total size of files contained in them. The file size is entered in kb, but you can specify 0.001 to 1000000 kb.
3) Subdirectories which satisfy the name criteria, as well as date modification criteria
4) Total number of files in subdirectory is greater or less than a certain value. This option does not update any of the dynamic variables.

Task Synchronization

All tasks are multithreaded. Each individual task can be synchronized, to prevent or allow overlapping schedules. If you do not set a profile for the task title, the default behavior is to **NOT** allow overlapping schedules.

Profiles
You can set multiple synchronization profiles, with different synchronization options. For Ftp tasks, you can choose to allow a single overlapping instance. At the same time, for email tasks, you can disallow overlapping instances.

Task List
Enter task titles separated by commas. To specify all the tasks, use: All_Tasks

Options
The following synchronization options are supported:
1) Allow overlapping schedules - multiple instances of the same task title can run simultaneously.
2) No overlapping schedules, no instances wait - no instances will wait, if a previous instance of the same task is running.
3) No overlapping schedules, single instance waits - only a single instance will wait, if a previous instance of the same task is running.

Wait time
If a single instance waits, you can set the maximum wait time to wait for the previously running instance to finish running.

After Wait time
If the wait time is exceeded and the previous instance has not finished, you can choose to run the task or abort the task.

Synchronize Folders

This task allows you to synchronize two folders.

Action

Three synchronization modes are supported. Be careful when using the Unidirectional Mirror mode; because newer files can be replaced by older files, and unpaired files are deleted.

1) Bi-directional Mirror
All unpaired files and folders in the Source are copied to the Target.
All unpaired files and folders in the Target are copied to the Source.
Older files in the Target are replaced with their newer counterpart in the Source.
Older files in the Source are replaced with their newer counterpart in the Target.

2) Unidirectional Copy
All unpaired files and folders in the Source are copied to the Target.
Older files in the Target are replaced with their newer counterpart in the Source.

3) Unidirectional Mirror
All unpaired files and folders in the Source are copied to the Target.
Older files in the Target are replaced with their newer counterpart in the Source.
Newer files in the Target are replaced with their older counterpart in the Source.
All unpaired files and folders in the Target are removed.

Source Directory

Enter the full pathname of the directory, where your source files exists. Note that the pathname is case sensitive for Unix Ftp servers.

Target Directory

Enter the full path name of the directory, where you need to put the target files. Note that the path is case sensitive on Unix systems.

Include subdirectories

If you select this option, all subdirectories and files within the Source Directory are copied to the Target Directory. If a subdirectory does not exist within the target folder, then it is automatically created.

Filename

Leave this field blank if you want all files to be synchronized. Enter the filename filter for the files you wish to synchronize, using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Folder Path Filter

Leave this field blank if you want all folders to be processed. If you need specific folders to be processed, then enter the folder name or path, using the following wildcard rules. If a folder passes the filter, then all its subfolders will also pass the filter.
Simple Example: Folder = c:\test , subfolders = john, john\old, sam, sam\old, joe, joe\old
Folder path filter = john . john and john\old are processed
Folder path filter = jo . john, john\old, joe, joe\old are processed
Folder path filter = old . sam\old, john\old, joe\old are processed
Folder path filter = test . all files in c:\test and its sub folders are processed
Folder path filter = john\old. only john\old is processed.
Folder path filter = c:\test\john . john and john\old are processed

Check for critical folder

This option is added to prevent accidentally processing files within critical folders. Many of our users use dynamic variables for the source and target folders. Sometimes, inadvertently, the dynamic result could be c:\ or / etc which may result in undesirable consequences.
The folders checked include:
1) Root folders (c:\ or d:\ or / etc..)
2) User home folder
3) Folders in the system library path.

If you have to use these folders (c:\ or d:\ or / etc..), and the folder filter or filename filter, you can disable this option if you are absolutely sure that no critical system or data files will be adversely effected..

System Variables

Many task fields support variable entries. The program has system variables that you can use in these task fields.

Using variables in text fields

The variable must be formatted as: $%**Variable_Name::Option1::Option2..**%$
Options may only be required by some Variables.
Examples: $%DATE::hh:mm:yy%$ or $%USERNAME%$

Nested Variables

Single level nested variables are allowed using the format
$%....NEST-[nested_variable]-NESTEND...%$
examples :
1) $%FILE::DATE::pp-dd-YY_mm-hh-ss::NEST-fileMon::FilePath-NESTEND%$ gets the date modified time for the file path found in the variable 'fileMon::FilePath'.
2) $%FILE::DATE::pp-dd-YY_mm-hh-ss::c:\test\NEST-fileMon::FileName-NESTEND%$ gets the date modified time for the file found in the variable 'fileMon::FileName'.

Only 1 nested variable can be read in a single field.

The following is the list of System Variables

*** DATE**
You can specify the current date using the following format
$%DATE::dd/pp/yy hh:mm:ss AMPM%$

You can also choose any date, time, or date/time separators like /, _, -, or : etc...

YY = 4-digit year will be output (2001 etc..)
yy = 2-digit year will be output (00-99)
pp = 2-digit month will be output (01-12)
ppx = 1-digit or 2-digit month will be output(1-12)
qqq = 3-character month will be output (jan-dec)
QQQ = 3-character month will be output (JAN-DEC)
Qqq = 3-character month will be output (Jan-Dec)
dd = 2-digit date will be output (01-31)
ddx = 1-digit or 2-digit date will be output(1-31)
HH = 2-digit hour based on 12 hour clock will be output (01-12) (should generally be used with am_pm)
HHx = 1-digit or 2-digit hour based on 12 hour clock will be output(1-12)
hh = 2-digit hour based on 24 hour clock will be output (00-23)
hhx = 1-digit or 2-digit hour based on 24 hour clock will be output (0-23)
mm = 2-digit minute will be output (00-59)
mmx = 1-digit or 2-digit minute will be output (0-59)
ss = 2-digit seconds will be output (00-59)
am_pm = AM is output for AM hours, PM is output for PM hours
wy = 2-digit week of the year
wyx = 1-digit or 2-digit week of the year
dy = 3-digit day of the year
dyx = 1-digit, 2-digit, or 3-digit day of the year
dw = day of the week (1-7) (country dependent, ex: 1 = sunday in the US, and 1 = monday in France)
wm = week of the month (1-5)
dwm = day of week in the month (1-5)

Examples: (for date/time = June 21st, 2005, 3:45:30 PM)
$%DATE::dd-pp-YY%$ = 21-06-2005
$%DATE::dd-pp1-yy%$ = 21-6-05
$%DATE::qqq/dd/yy_hh:mm:ss%$ = jun/21/05_15:45:30
$%DATE::Qqq dd, YY, HH:mm am_pm%$ = Jun 21, 2005, 03:45 PM

*** DATES**
This variable is provided as a simpler option to the DATE variable above and it also provides more flexibility.
The DATES variable is described here.

*** DATEADD**
This variable returns the value of a single time field (date, hour, minute etc..), after adding (or subtracting) days/hours etc.. from the current time:
$%DATEADD::dd::X%$
where X is the number to add. For next day, use 1, for previous day/hr etc.. use -1.

Examples: (for date/time = June 21st, 2005, 3:45:30 PM)
$%DATEADD::dd::1%$ = 22
$%DATEADD::dd::-1%$ = 20
$%DATEADD::hh::1%$ = 16
$%DATEADD::hh::-1%$ = 14
$%DATEADD::mm::1%$ = 46
$%DATEADD::mm::-1%$ = 44

Notes:
1) You can only add or subtract and recieve one field at a time
2) Only numerical date / time values are supported. i.e. days of week (Mon., Tue. etc..) or months (jan., feb. etc..) are not supported.
3) You cannot embed a DATEADD variable within a DATE variable. You have to use all variables in sequence.
Example: If you want current month and date, but require previous hour, in the format month-date_hr, you would use
$%DATE::pp-dd_%$$%DATEADD::hh::-1%$ = 06-21_14

*** DATEADDX**
This variable returns a formatted date/time string, after adding (or subtracting) days/hours etc. from the current time:
$%DATEADDX::Format::dd::X%$
where X is the number to add. For next day/hour etc., use 1, for previous day/hour etc.. use -1. Format is the date/time format that you need the result to be returned in.

This variable allows user to add date/hour/minute to current time intelligently. Ex: adding a day to 12/31/03, rolls over to 01/01/04, not to 12/32/03

Examples: (for date/time = June 21st, 2005, 3:45:30 PM)
$%DATEADDX::YY-pp-dd::dd::1%$ = 2004-06-22
$%DATEADDX::YY-pp-dd_hh-mm::hh::-1%$ = 2004-06-21_14-45
$%DATEADDX::YY-pp-dd_hh-mm::hh::1%$ = 2004-06-21_16-45
$%DATEADDX::YY-pp-dd_hh-mm::mm::1%$ = 2004-06-21_16-46
$%DATEADDX::YY-pp-dd_hh-mm::mm::-1%$ = 2004-06-21_16-44

Notes:

1) Only the following are allowed in the 3rd field: YY, pp, dd, hh, mm
2) You can only add or subtract one field at a time
3) Only numerical date / time values are supported. i.e. days of week (mon., tue. etc..) or months (jan., feb. etc..) are not supported.

*** DATEMONTH**
This variable allows you to get the last day of the previous month, or first Sunday of next month etc.. It returns a formatted date/time string. The format required is:
$%DATEMONTH::format::month_offset::day_val::X%$
format is the date/time format that you need the result to be returned in
month_offset = 0 for current month, 1 for next month, -1 for previous month
day_val = sun, mon, tue, wed, thu, fri, sat to specify day of week. Use 'day' to specify date of the month.
X = week of month (1-5) or day of month (1-31). To specify the last week of the month, use -1. To specify the last date of the month, use -1.

Examples: (for month = April, 2004)
To get first Sunday of this month:
$% DATEMONTH::YY-Qqq-dd::0::sun::1%$ = 2004-Apr-04
To get last Sunday of this month:
$% DATEMONTH::YY-Qqq-dd::0::sun::-1%$ = 2004-Apr-25
To get first Monday of next month:
$% DATEMONTH::YY-Qqq-dd::1::mon::1%$ = 2004-May-03
To get last Monday of previous month:
$% DATEMONTH::YY-Qqq-dd::-1::mon::-1%$ = 2004-Mar-29
To get last day of previous month:
$% DATEMONTH::YY-Qqq-dd::-1::day::-1%$ = 2004-Mar-31

*** FILE**
This variable outputs information on file size, date or can parse lines from the file into the variable. This is useful if you need to monitor files for certain strings etc.. This variable can be used in the 'File Variable' task to process text in files, and output results to new files.

$%FILE::SIZE::[file_path]%$
$%FILE::DATE::[format]::[file_path]%$
$%FILE::LINE::FIRST::[file_path]%$
$%FILE::LINE::LAST::[file_path]%$
$%FILE::LINE::MID::[start_line]::[end_line]::[file_path]%$
$%FILE::LINECONTAINS::FIRST::[string]::[file_path]%$
$%FILE::LINECONTAINS::LAST::[string]::[file_path]%$
$%FILE::LINECONTAINS::ALL::[string]::[file_path]%$
$%FILE::LINENUMCONTAINS::FIRST::[string]::[file_path]%$
$%FILE::LINENUMCONTAINS::LAST::[string]::[file_path]%$
$%FILE::LINESAFTER::FIRST/LAST::[string]::[offset]::[file_path]%$
$%FILE::LINESBEFORE::FIRST/LAST::[string]::[offset]::[file_path]%$

Examples:
1) $%FILE::SIZE::c:\temp\test.txt%$
will output the size of the file in bytes

2) $%FILE::DATE::qqq-dd-yy_hh:mm:ss::c:\temp\test.txt%$
will output the last modified time of the file, in the date/time format you specify. Please see the **DATE** variable above,

for details on how to format the date. example: if June 21,2002, 3:45 pm was the last modified time of the file c:\temp\test.txt, this variable will output: jun-21-02_15:45:30

3) $%FILE::LINE::FIRST::c:\temp\test.txt%$
will output the first line in the file c:\temp\test.txt

4) $%FILE::LINE::LAST::c:\temp\test.txt%$
will output the last line in the file c:\temp\test.txt

4) $%FILE::LINE::MID::5::90::c:\temp\test.txt%$
will output the lines 5-90 in the file c:\temp\test.txt. If there are only 65 lines in the file, it will output lines 5-65.

6) $%FILE::LINECONTAINS::FIRST::error alert::c:\temp\test.txt%$
will output the first line in test.txt, that contains the string 'error alert'. It will return an empty string, if the 'error alert' is not found.

7) $%FILE::LINECONTAINS::LAST::error alert::c:\temp\test.txt%$
will output the last line in test.txt, that contains the string 'error alert'. It will return an empty string, if the 'error alert' is not found.

8) $%FILE::LINECONTAINS::ALL::error alert::c:\temp\test.txt%$
will output all the lines in test.txt, that contains the string 'error alert'. It will return an empty string, if the 'error alert' is not found in any line.

9) $%FILE::LINENUMCONTAINS::FIRST::error alert::c:\temp\test.txt%$
This variable will output the line number of the first line, that contains the string 'error alert'. It will return 1000000, if the string is not found.

10) $%FILE::LINENUMCONTAINS::LAST::error alert::c:\temp\test.txt%$
This variable will output the line number of the last line, that contains the string 'error alert'. It will return 1000000, if the string is not found.

11) $$%FILE::LINESAFTER::FIRST/LAST::[string]::[offset]::[file_path]%$
This variable will output all lines, after the line that contains the String [string] .
Use FIRST to output all lines, after the first line containing the [string]
Use LAST to output all lines, after the last line containing the [string]
use [offset] = 0 , to include the line containing the [string], use [offset] = 1, to start at next line, etc..
examples:
$$%FILE::LINESAFTER::FIRST::error::0::c:\temp\test.txt%$ (will include the line containing the string 'error')
$$%FILE::LINESAFTER::LAST::error::1::c:\temp\test.txt%$ (will not include the line containing the string 'error')

12) $$%FILE::LINESBEFORE::FIRST/LAST::[string]::[offset]::[file_path]%$
This variable will output all lines, before the line that contains the String [string] .
Use FIRST to output all lines, before the first line containing the [string]
Use LAST to output all lines, before the last line containing the [string]
use [offset] = 0 , to include the line containing the [string], use [offset] = 1, to end at previous line, etc..
examples:
$$%FILE::LINESBEFORE::FIRST::error::0::c:\temp\test.txt%$ (will include the line containing the string 'error')
$$%FILE::LINESBEFORE::LAST::error::1::c:\temp\test.txt%$ (will not include the line containing the string 'error')

*** Other Variables**

USERNAME
This variable outputs the value of the current user
example: if your username is 'john'
$%USERNAME%$ =john

USERHOME
This variable outputs the value of the current users home directory
example: if your username is 'john', your home directory on win 2000 would be
$%USERHOME%$ =c:\documents and settings\users\john

TEMPDIR
This variable outputs the current value of the temp folder
$%TEMPDIR%$ =c:\temp

OSNAME
This variable outputs name of the operating system
example: on Windows 2000
$%OSNAME%$ =Windows 2000

OSVERSION
This variable outputs the version of the operating system
example: on Windows 2000
$%OSVERSION%$ =5.0

IPADDRESS
This variable outputs the computers IP address
example: if your systems IP address is 192.168.0.1
$%IPADDRESS%$ =192.168.0.1

SYSNAME
This variable outputs the computer name on the network
example: if your computer name is Win1
$%SYSNAME%$ =Win1

DATES variable

This variable is being provided as an alternative to the DATE variable since verion 10.08. It provides more flexibility and has simpler options relative to the DATE variable.

Example use:

```
$%DATES::yyyy-MM-dd  HH:mm:ss%$   would return 2013-07-04  12:08:56
$%DATES::MMMMMMMMMM%$   would return  March or April or September or December  depending on the month
$%DATES::MMM%$   would return  Mar or Apr or Sep or Dec  depending on the month
$%DATES::MM%$   would return  03 or 04 or 09 or 12  depending on the month
```

DATES is based on the java SimpleDateFormat class. The description below is taken from the java SimpleDateFormat class.

Date and time formats are specified by *date and time pattern* strings. Within date and time pattern strings, unquoted letters from `'A'` to `'Z'` and from `'a'` to `'z'` are interpreted as pattern letters representing the components of a date or time string. Text can be quoted using single quotes (`'`) to avoid interpretation. `"''"` represents a single quote. All other characters are not interpreted; they're simply copied into the output string during formatting or matched against the input string during parsing.

The following pattern letters are defined (all other characters from `'A'` to `'Z'` and from `'a'` to `'z'` are reserved):

Letter	Date or Time Component	Presentation	Examples
`G`	Era designator	Text	`AD`
`y`	Year	Year	`1996`; `96`
`M`	Month in year	Month	`July`; `Jul`; `07`
`w`	Week in year	Number	`27`
`W`	Week in month	Number	`2`
`D`	Day in year	Number	`189`
`d`	Day in month	Number	`10`
`F`	Day of week in month	Number	`2`
`E`	Day in week	Text	`Tuesday`; `Tue`
`a`	Am/pm marker	Text	`PM`
`H`	Hour in day (0-23)	Number	`0`
`k`	Hour in day (1-24)	Number	`24`
`K`	Hour in am/pm (0-11)	Number	`0`
`h`	Hour in am/pm (1-12)	Number	`12`
`m`	Minute in hour	Number	`30`
`s`	Second in minute	Number	`55`
`S`	Millisecond	Number	`978`
`z`	Time zone	General time zone	`Pacific Standard Time`; `PST`; `GMT-08:00`
`Z`	Time zone	RFC 822 time zone	`-0800`

Pattern letters are usually repeated, as their number determines the exact presentation:

- **Text:** if the number of pattern letters is 4 or more, the full form is used; otherwise a short or abbreviated form is used if available.
- **Number:** the number of pattern letters is the minimum number of digits, and shorter numbers are zero-padded to this amount.

- **Year:** the number of pattern letters is 2, the year is truncated to 2 digits; otherwise it is interpreted as a number.
- **Month:** If the number of pattern letters is 3 or more, the month is interpreted as text; otherwise, it is interpreted as a number.
- **General time zone:** Time zones are interpreted as text if they have names. For time zones representing a GMT offset value, the following syntax is used:

  ```
  GMTOffsetTimeZone:
          GMT Sign Hours : Minutes
  Sign: one of
          + -
  Hours:
          Digit
          Digit Digit
  Minutes:
          Digit Digit
  Digit: one of
          0 1 2 3 4 5 6 7 8 9
  ```

  Hours must be between 0 and 23, and *Minutes* must be between 00 and 59. The format is locale independent and digits must be taken from the Basic Latin block of the Unicode standard.

  For parsing, RFC 822 time zones are also accepted.

- **RFC 822 time zone:** For formatting, the RFC 822 4-digit time zone format is used:

  ```
  RFC822TimeZone:
          Sign TwoDigitHours Minutes
  TwoDigitHours:
          Digit Digit
  ```

  TwoDigitHours must be between 00 and 23. Other definitions are as for general time zones.

Examples

The following examples show how date and time patterns are interpreted in the U.S. locale. The given date and time are 2013-07-04 12:08:56 local time in the U.S. Pacific Time time zone.

Date and Time Pattern	Result
`yyyy.MM.dd G 'at' HH:mm:ss z`	`2013.07.04 AD at 12:08:56 PDT`
`EEE, MMM d, ''yy`	`Wed, Jul 4, '01`
`h:mm a`	`12:08 PM`
`hh 'o''clock' a, zzzz`	`12 o'clock PM, Pacific Daylight Time`
`K:mm a, z`	`0:08 PM, PDT`
`yyyyy.MMMMM.dd GGG hh:mm aaa`	`02013.July.04 AD 12:08 PM`
`EEE, d MMM yyyy HH:mm:ss Z`	`Wed, 4 Jul 2001 12:08:56 -0700`
`yyMMddHHmmssZ`	`010704120856-0700`
`yyyy-MM-dd'T'HH:mm:ss.SSSZ`	`2013-07-04T12:08:56.235-0700`

Task account task

This task is ONLY available in Automize Windows version

This feature allows you to run tasks under other user accounts.

You will need to enter the other users domain, username, password.

Termination Time

The termination time is the maximum time that the task can run. The task will terminate, if it is still running, after the maximum allowed time. If you leave this field empty, then a default value of 60 minutes is used.

Task Runner

This feature allows you to run tasks from an external program, script, batch file, or command-line. Select the desired task from the drop down box. The command-line and working directory will be automatically generated in the text fields. You can use the "Copy to Clipboard" feature to copy these fields to clipboard for use in your scripts or programs.

Batch or script file example:
cd [working_directory]
[commandline]
where [working_directory] and [commandline] are generated automatically in the Task Runner fields.

External program:
Use this command-line from your external program to launch the task. Make sure that you also enter the working directory, otherwise the Java executable will not be able to find the correct program classes.

Archive logs task	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**
Autobackup task	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**

Task Sequence

This feature allows you to run multiple tasks in sequence. You can run programs, email, web, ftp and all other tasks from the Sequence. If a certain task fails, you can choose to stop the Sequence. You can also skip ahead to another task in the Sequence. The Task Sequence task is an extension of the 7.x Chain task. In 8.x, this task was named Visual Script, but in 9.x renamed to a more appropriate Task Sequence.

Creating a Sequence

From the front panel: click on the 'Task Sequence' button.

Adding Tasks to your Sequence

You can add new tasks or existing tasks to your Sequence.
To add a new task: In the task type tree, Select the task type and click on the Add New Task button
To add an existing task: Click on the Add Existing Task Button.

Email Notification of tasks in the Sequence

By default, only the Sequence itself can trigger email notification based on exit code. The tasks in a Sequence do not trigger email notification, even if they satisfy an email notification profile. You should select this option, in order to allow tasks in the Sequence to trigger email notification. The Sequence will move on to the next step, only after the email notification has been completed.

Starting Sequence at specified step

You can specify a step to start at, by entering the step number, or the step title in this field. By default, the Sequence will start at step 1, if this field is left blank.

Deleting Tasks from your Sequence

Select the task in the bottom table and click the delete button.

Editing Tasks in your Sequence

Select the task in the bottom table and click the edit button. The task parameters will be loaded into the window. You can then make any changes you require and click the "Update Task" button.

Positioning a Task within your Sequence

Select the task in the bottom table and use the move-up or move-down buttons as required.

Enable Step

You can enable or disable a step. If a step is disabled, the task will move on to the next step.

Skip to Step options

You can choose to skip to certain step in the Sequence, if a certain exit code criteria is met. If you wish to stop the Sequence, because a certain exit code criteria is met, enter a Skip to Step value, which is greater than the total steps in the Sequence.

Skip to Step (or Title)

1) You can enter either the step number in the Sequence, or the task title. This task title has to be a valid title in the Sequence list.
2) You can also run multiple tasks simultaneously using the following rules:
a) Separate each Task Title by &. Example: TaskTitle1&TaskTitle2&TaskTitle3 (no spaces allowed)
b) You can only enter Task Titles, NOT step numbers.
c) TaskTitle1 has to be a valid Task Title in the Sequence list
d) TaskTitle1 CAN NOT be the last step in the Sequence list. You can add a dummy delay task as the last step in the Sequence, if TaskTitle1 happens to be your last step.
e) TaskTitle2, TaskTitle3 etc. DO NOT have to be in the Sequence list. They have to be valid tasks in the task list.

f) In this simultaneous mode, all tasks (TaskTitle2, TaskTitle3. etc.) will be triggered simultaneously. The Sequence will then continue normally at TaskTitle1.

Rerun Options

You can choose to rerun the task, if it meets certain exit code criteria. You can also set the maximum number of times to rerun a task. The task will rerun, while the rerun criteria is valid, until the maximum rerun count has been met.

Maximum Wait Time

This is the maximum time (**T1**) the Sequence will wait for your task to finish. Enter 0 to wait forever until the task finishes its run. ***Please note that the Sequence will not kill or terminate your task because it is unsafe to do so.*** It just waits for your task to complete normally. If the task does not complete normally in the allotted time, the Sequence will assume that the task has failed. The email, web, ftp and most other tasks will terminate normally.
For the **DirChange** and **Command** tasks, please ensure that you can Terminate the task within the Task settings itself. Both these tasks have a safe Termination time (**T2**) that can be set within the task. Please set **T1 T2**. If you have set the Command or DirChange task to rerun on failure, please set the **T1 (T2+Td)*n**, where **Td** is the delay between reruns, and **n** is the number or reruns.

Why not just Kill a task when the wait time is exceeded

Short answer from us: In the Sequence, each task is run as a thread. Killing a thread is unsafe, and may corrupt your system hours or days in the future.
Long answer from Java documentation: Because it is inherently unsafe. Stopping a thread causes it to unlock all the monitors that were locked. (The monitors are unlocked as the Thread Death exception propagates up the stack.) If any of the objects previously protected by these monitors were in an inconsistent state, other threads may now view these objects in an inconsistent state. Such objects are said to be damaged. When threads operate on damaged objects, arbitrary behavior can result. This behavior may be subtle and difficult to detect, or it may be pronounced. Unlike other unchecked exceptions, Thread Death kills threads silently; thus, the user has no warning that his program may be corrupted. The corruption can manifest itself at any time after the actual damage occurs, even hours or days in the future.

Task List

The tasks list and tasks table display your task information.

You can edit the task using the Edit button. This will load the parameters into the task window correctly, and you can edit the settings there.

To run a task, select the task in the Tasks list, and click the "Run" button.
To schedule a task, select the task in the Tasks list, and click the "Schedule" button.
To edit a task, select the task in the Tasks list, and click the "Edit" button.
To delete a task, select the task in the Tasks list, and click the "Delete" button.
To duplicate a task, select the task in the Tasks list, and click the "Copy" button.

Task To Run

The calling monitor task or loop task will wait for called task to complete before continuing on.
If you want to use the task in a Task Sequence, and do not need the Task To Run feature, select 'none'.

You can select multiple tasks to run in sequence or simultaneously.
To run tasks in sequence, use taskTitle1|taskTitle2|taskTitle3.
To run tasks simultaneously, use taskTitle1&taskTitle2&taskTitle3.

If you do not want to wait for running task to complete, use /nowait after the task title.
example: if task title = test , use: test/nowait

Task Variables

Every task outputs values of certain variables. You can use these values in other tasks, or in the Variable Monitor task. You can view the list of Task Variables using the 'Utilities' Menu - 'Task Variables'.

Using task variables in text fields

The variable must be formatted as: $%**TaskTitle::VariableName**%$
Example: for command task with task title = TestTask
$%TestTask::ExitCode%$ = Exit code for latest run of this task
$%TestTask::ErrorMsg%$ = Error message for latest run of this task

Task Type	Variable Name	Variable Description
All Tasks	LastTaskTitleRun	Task title of last task run
All Tasks	LastTaskTypeRun LOC::LastTaskTypeRun	Task Type of last task run LOC:: returns the localized version of the variable.
All Tasks	LastExitCode	Exit Code of last task run
All Tasks	TaskType::LastTaskRun	Task title of last task run for each task type
Engine	CurrentInstanceRunningTasks	Total currently running tasks in current instance of engine. Not reset to 0 if engine shuts down while tasks are running
Engine	StoppedInstanceRunningTasks	Updated periodically during maintainance task and shutdown requests while engine is waiting for tasks to complete. Reset to 0 when engine shuts down
Eingine	StoppedInstanceKilledTasks	Total running tasks when engine shuts down due to wait time expired
All Tasks	TaskTitle::ExitCode	Task exit code value
All Tasks	TaskTitle::LastRunEndTime	Time last run finished
All Tasks	TaskTitle::LastRunStartTime	Time last run started
All Tasks	TaskTitle::LastRunRunTime	Duration of last run
Command	TaskTitle::ErrorMsg	Task error message. This variable is BLANK unless an error is detected during the task run. This variable is set to BLANK before each run.
Command	TaskTitle::OutputMsg	Task output message. This variable is BLANK unless output is detected during the task run. This variable is set to BLANK before each run.
WinCommand	TaskTitle::ErrorMsg	Task error message. This variable is BLANK unless an error is detected during the task run. This variable is set to BLANK before each run.
CopyFile	TaskTitle::TotalFiles	Number of files in directory
CopyFile	TaskTitle::PassFilename	Number of files passing filename filter check
CopyFile	TaskTitle::PassDate	Number of files passing date modified criteria
CopyFile	TaskTitle::ForceCopy	Number of older(or same) source files copied
CopyFile	TaskTitle::SourceOlder	Number of source files older (or same date) than target files (i.e. not copied)

CopyFile	TaskTitle::CopyFailed	Number of files failed to copy
CopyFile	TaskTitle::DeleteFailed	Number of files failed to delete after move/rename options
CopyFile	TaskTitle::FilesCopied	Number of files copied
DeleteFile	TaskTitle::TotalFiles	Number of files in directory
DeleteFile	TaskTitle::PassFilename	Number of files passing filename filter check
DeleteFile	TaskTitle::PassDate	Number of files passing date modified criteria
DeleteFile	TaskTitle::FilesDeleted	Number of files deleted
ZipDirs	TaskTitle::TotalFiles	Number of files in directory
ZipDirs	TaskTitle::PassFilename	Number of files passing filename filter check
ZipDirs	TaskTitle::PassDate	Number of files passing date modified criteria
ZipDirs	TaskTitle::FilesZipped	Number of files zipped
ZipDirs	TaskTitle::ZipFilePath	Zip file path
ZipDirs	TaskTitle::ZipFileSize	Zip file size
TextSearch	TaskTitle::FilesSearched	Number of files searched
TextSearch	TaskTitle::FilesFound	Number of files search term(s) were found in
TextSearch	TaskTitle::TotalFound	Total number of times search terms were found
TextSearch	TaskTitle::ResultsFile	Results filename
TextSearch	TaskTitle::TotalTermsFound	Total number of search terms found. This variable is useful for searches in single file. If you have X search terms, but only Y terms were found in 1 file, then the value of this variable = Y. But if, you have multiple files being searched, this value may be greater than the number of search terms.
TextSearch	TaskTitle::LastLineNumber	The last line number where a search term was found. Value is 1000000 if no search term is found.
Ftp	TaskTitle::TotalFiles	Number of files in directory
Ftp	TaskTitle::LocBackup	Total local files backed up
Ftp	TaskTitle::FailLocBackup	Total local files failed to back up
Ftp	TaskTitle::RemBackup	Total remote files backed up
Ftp	TaskTitle::FailRemBackup	Total remote files failed to back up
Ftp	TaskTitle::FilesDeleted	Total source files deleted
Ftp	TaskTitle::DeleteFailed	Total source files failed to delete
Ftp	TaskTitle::FailRename	Total temporary files failed to rename
Ftp	TaskTitle::PassFilename	Total files meeting filename criteria
Ftp	TaskTitle::PassDate	Total files meeting filename and date criteria
Ftp	TaskTitle::PassNew	Total files meetings newly modified files criteria
Ftp	TaskTitle::FailedXfers	Total failed transfers
Ftp	TaskTitle::FilesXfered	Total files transferred
FtpMonitor	TaskTitle::SourceDir	Remote Directory being monitored
FtpMonitor	TaskTitle::DirChanged	true if directory change is detected since last run, else false
FtpMonitor	TaskTitle::FileFound	Name of last file found that satisfies the monitoring criteria
FtpMonitor	TaskTitle::FirstFileFound	Name of first file found that satisfies the monitoring criteria

FtpMonitor	TaskTitle::FileSize	Size of file found that satisfies the monitoring criteria
FtpMonitor	TaskTitle::FileDate	Date of file found that satisfies the monitoring criteria
FtpMonitor	TaskTitle::TotalFiles	Total files in directory, if 'Total Files' criteria is selected
Ftp Monitor	TaskTitle::PassFileName	Number of files that pass the filename filter
Ftp Monitor	TaskTitle::Filenames	Files that pass monitoring criteria. The format is #file1^#file2... so that this variable can be directly used in other tasks filename field (example copy/ftp/delete etc..)
Ping	TaskTitle::LastResult	Last result for this task
Ping	TaskTitle::TestCount	Total test count
Ping	TaskTitle::UpCount	Total successfull connections
Ping	TaskTitle::UpPercent	Percent uptime
Ping	TaskTitle::LastRuntime	Last run time
Telnet	TaskTitle::Response1	The initial connection response from server + login prompt (All task variables are reset to BLANK before task runs)
Telnet	TaskTitle::Response2	password prompt
Telnet	TaskTitle::Response3	Authentication response/welcome message
Telnet	TaskTitle::Response4+	Response4, Response5 etc.. are the reply strings from the server for each command you send.
Telnet(Adv)	TaskTitle::Response1	The initial connection response from server + login prompt (All task variables are reset to BLANK before task runs).
Telnet(Adv)	TaskTitle::Response2	password prompt
Telnet(Adv)	TaskTitle::Response3	Authentication response/welcome message
Telnet(Adv)	TaskTitle::Response4+	Response4, Response5 etc.. are the reply strings from the server for each command you send.
EmailBulk	TaskTitle::Successful	Number of Successful emails
EmailBulk	TaskTitle::Failed	Number of Failed emails
EmailBulk	TaskTitle::Total	Number of Total emails
EmailCheck	TaskTitle::PassCrit1	Number of messages passing criteria 1
EmailCheck	TaskTitle::PassCrit2	Number of messages passing criteria 2
EmailCheck	TaskTitle::Messages	Number of messages being downloaded
EmailCheck	TaskTitle::Subject	Subject for the last email downloaded
EmailCheck	TaskTitle::From	Sender for the last email downloaded
Database SQL	TaskTitle::Rows	Number of rows returned or updated
File Monitor	TaskTitle::FileName	Filename for the last file that satisfied the monitoring criteria
File Monitor	TaskTitle::FirstFileName	Filename for the first file that satisfied the monitoring criteria
File Monitor	TaskTitle::FilePath	Filepath for the file that satisfied the monitoring criteria
File Monitor	TaskTitle::FileSize	File size in bytes for the file that satisfied the monitoring criteria
File Monitor	TaskTitle::FileDate	File timestamp for the file that satisfied the monitoring criteria
File Monitor	TaskTitle::TotalFiles	Total files in directory, if total files criteria is selected

File Monitor	TaskTitle::PassFileName	Number of files that pass the filename filter
File Monitor	TaskTitle::Filenames	Files that pass monitoring criteria. The format is #file1^#file2... so that this variable can be directly used in other tasks filename field (example copy/ftp/delete etc..)
Dir Monitor	TaskTitle::Filenames	Files that pass monitoring criteria. The format is #file1^#file2... so that this variable can be directly used in other tasks filename field (example copy/ftp/delete etc..)
Dir Monitor	TaskTitle::ModifiedObjects	Total modified objects (directories + files) in directory tree
Dir Monitor	TaskTitle::TotalObjects	Total objects (directories + files) in directory tree
Dir Monitor	TaskTitle::FileName	Last filename that passes monitoring criteria.
Dir Monitor	TaskTitle::FirstFileName	First filename that passes monitoring criteria.
Ftp Command	TaskTitle::ResponseX	Response1, Response2 etc.. are the reply strings from the ftp server for each command you send
Chain	TaskTitle::LastRunStepTitle	Latest step task title run in this chain task
Chain	TaskTitle::LastRunStepType	Latest step task type run in this chain task
Chain	TaskType::LastRunStepTitle	Latest step task title run in any chain task
Chain	TaskType::LastRunStepType	Latest step task type run in any chain task
Web	TaskTitle::FileSize	File size as reported by web server
Web	TaskTitle::TotalBytes	Total bytes downloaded
Web	TaskTitle::ContentType	File content type
Web	TaskTitle::ContentEncoding	File content encoding
Web	TaskTitle:: LastModifiedHeader	value of the last-modified header field
Web	TaskTitle:: DateHeader	value of the date header field
Web	TaskTitle:: ExpirationHeader	value of the expires header field
LogFile Monitor	TaskTitle::TotalAlerts	Total new alerts detected in this run
LogFile Monitor	TaskTitle::AlertLines	Variable contains all new alert lines. Use the File variable task to output lines to file.
LogFile Monitor	TaskTitle::LastAlertTime	Last alert time
Unzip Archive	TaskTitle::TotalFiles	Total files extracted
Unzip Archive	TaskTitle::TotalZips	Total zip files
Unzip Archive	TaskTitle::TotalSuccessUnzip	Total zip files successfully unzipped
Unzip Archive	TaskTitle::TotalFailUnzip	Total zip files failed to unzip
Unzip Archive	TaskTitle::FailZipnames	Zip filenames which failed to unzip
Unzip Archive	TaskTitle::SuccessZipnames	Zip filenames successfully unzipped

Task Variables via command-line

Task variables are dynamically generated when your task runs. This feature provides the command-line code required to get the value of the Task Variable. You can use this command-line code, from an external program, script, batch file, or command-line (command prompt in windows or terminal in Unix). The command-line and working directory required to get the variable will be automatically generated in the text fields.

A) Getting Task variable values to use in scripts or batch files
From the command-line via your batch files or scripts etc,
first set the program install folder as the current directory

```
cd install_folder
```

on windows, use:

```
jre\bin\java.exe -cp .;hitek7.jar com.hitek.engine.mods.var.TaskVariable get variable_name
jre\bin\java.exe -cp .;hitek7.jar com.hitek.engine.mods.var.TaskVariable get variable_name c:\test\test.txt
```

(will output the variable to the file test.txt)

IMPORTANT: variable_name cannot include any spaces in it

on Unix/MacOSX, use:

```
java -cp .:hitek7.jar com.hitek.engine.mods.var.TaskVariable get variable_name
java -cp .:hitek7.jar com.hitek.engine.mods.var.TaskVariable get variable_name /home/test.txt
```

(will output the variable to the file test.txt)

where variable_name = to variable name, and variable_value = to variable value

The variable value will be returned as a string value in the system output stream. You should add code in your batch or script or program to read this value

Also, if the variable is an integer, the exit code of the command above is the integer value of the variable. If the variable is not an integer, the exit code of the command above = -1001. If there is any error in your syntax, the exit code = 1000.

B) Setting your own task variables for the program (Automize, AbleFtp etc..) to recognize

It is not recommended to set or change task variables from the command-line. These task variables are dynamically generated when a task runs, and will overwrite any new variable you enter. You should instead set user defined variables, using the UserVariables feature.

From the command-line via your batch files / scripts etc, first set the program install folder as the current directory
cd install_folder (i.e. cd ..\Automize or cd ..\AbleFtp etc..)

on windows, use:

```
jre\bin\java.exe -cp .;hitek7.jar com.hitek.engine.mods.var.TaskVariable set variable_name variable_value
```

IMPORTANT: variable_name cannot include any spaces in it

on Unix/MacOSX, use:

```
java -cp .:hitek7.jar com.hitek.engine.mods.var.TaskVariable set variable_name variable_value
```

where variable_name = to variable name, and variable_value = to variable value

Task Wait Time

Every task is run in a separate thread within the Engine. You can set the maximum time that the Engine will wait for the task to finish running.

Wait Time: Enter the wait time in seconds. Enter 0 to wait forever.

NOTE: ***It is not safe to kill/destroy a thread. Hence the Engine will not kill the thread if it is still running after the wait time. It will simply stop waiting for it.***

NOTE: If your task actually takes longer to run than the set wait time, you will see the Exit Code being logged before some of the task run output. Example:

```
Jul 27, 2006 4:44:13 PM Copy - CopyTest - Manual Run
Jul 27, 2006 4:44:13 PM Copy - CopyTest - Start Instance = 850
Jul 27, 2006 4:44:13 PM Source = c:\test\test.bat, Date = Jul 26, 2006 1:44:15 PM
Jul 27, 2006 4:44:13 PM File copied to: c:\test\test_06_27_07.bat
...
...
Jul 27, 2006 4:44:18 PM Copy - CopyTest - Task runtime detected as greater than set wait time. Setting exit code = 999
Jul 27, 2006 4:44:18 PM Copy - CopyTest - End Instance = 850 , Exit Code = 999 , Runtime (ms) = 15031
Jul 27, 2006 4:44:19 PM Source = c:\test\test2.bat, Date = Jul 26, 2006 1:44:15 PM
Jul 27, 2006 4:44:19 PM File copied to: c:\test\test2_06_27_07bat
...
...
```

Telnet

This task allows you to telnet to a server and issue commands. If the server is up an exit code is 0. If server is down or unreachable, exit code is 1. No other exit codes are thrown for login failures etc.

Host
Enter the host name or IP address.

Port
Enter the desired port. Default Telnet port is 23.

Delay
Enter the delay time between sending your commands. 1 second is minimum. 5 seconds is good enough, unless you are on a slow network. You can increase it to ~15 seconds, but make sure you do not exceed the hosts telnet timeout value.

You can also set a custom command time for each step using the following case sensitive format:
STEPTIME=45^pwd
A delay = 45 seconds is used and the command is 'pwd'.

Command
Enter your telnet commands, 1 per line. First line should be your username, Second line should be your password and the last line should be **exit**. This field also supports dynamic variables. Using dynamic variables, you can enter a command, which will be dynamically resolved at runtime. example: cd /users/$%USERNAME%$

Example of Commands to enter

```
myUsername
myPassword
ls -lt
STEPTIME=45^pwd
ls
exit
```

Telnet(Adv)

This allows you to telnet to a server and issue commands. This Telnet(Adv) task is more advanced than the Telnet task. It will try and detect the presence of the telnet login prompts, and the telnet command prompts, before sending the next command. If these prompts are not detected within the timeout value set, it will send the next command.

Host
Enter the host name or IP address.

Port
Enter the desired port. Default Telnet port is 23.

Username
Enter your username.

Password
Enter your password.

Polling Period
Enter the desired polling time. This polling period applies only to Telnet commands, but not to the Login commands. Polling period does not apply to username and password commands. During the Login commands, the polling period is fixed at 1 second. For the regular Telnet commands, the module will poll the connection at the Polling interval, for the reply string from the server. If the last 10 characters (default value) of the reply string from the server contain the Telnet prompt, then the next command is sent. This polling period should be less than the command time.

Login Prompt
This is usually the last character, that the server outputs, when it requests a username or password to be sent. This value is normally : but may depend on the telnet server. The task will look for this character, in the last 10 characters (default value) of the reply string from the server. If this character is found, the next command is sent. If the login prompt character is not found within the Login time set, the next command is sent.

Telnet Prompt
This is usually the last character, that the server outputs, when it is done processing the current telnet command. This value normally depends on the system shell you are connecting to. The task will look for telnet prompt character, in the last 10 characters (default value) of the reply string from the server. As soon as this character is found, the next command is sent. If the telnet prompt character is not found, within the telnet command time set, the next command is sent.

Prompt Characters
This module waits for the Login or Telnet prompts, before sending the next command. It will search the last reply string from the telnet server, for the Login prompt, or the Telnet prompt. You can define how the module should search for these prompts. If you select Prompt Characters is 10 (default value), then the module will search the last 10 characters, of the last reply string. If the Login prompt, or the Telnet prompt, is found in these 10 characters, then the next command is sent. If your server sends a bunch of characters, after the Telnet prompts (Example: MS Telnet server), then you may have to increase this value to 1000, to detect the Telnet prompt. As soon as this character is found, the next command is sent.

Login Time
Enter the maximum time allowed for each login command (username and password). If this login time has been exceeded, and the server has not replied back with a telnet login prompt, then the next command is sent.

Command Time
Enter the maximum time allowed for each telnet command. If this command time has been exceeded, and the server

has not replied back with a telnet command prompt, then the next command is sent.
You can also set a custom command time for each step using the following case sensitive format:
STEPTIME=2^pwd
A step time = 2 minutes is used and the command is 'pwd'.

Command

Enter your telnet commands, 1 command per line. Always enter the exit command on the last line. This field also supports dynamic variables. Using dynamic variables, you can enter a command which will be dynamically resolved at runtime. example: cd /users/$%USERNAME%$

Example of Commands to enter

```
ls -lt
cd /user/test
STEPTIME=2^./runTestScript.scr
pwd
exit
```

Custom Login

The Telnet(Adv) task always sends the username first, followed by the password. If your login method is different than the Telnet(Adv), task will fail. In this case, please use the following procedure to apply a Custom login method. On the first line of the commands field, enter the String CUSTOMLOGIN^ (case important) followed by the number of commands you need for your custom login. Ex: CUSTOMLOGIN^3 should be the first line if you have 3 login commands. Then enter the rest of the commands like you normally would.

Example of Commands to enter for custom login

```
CUSTOMLOGIN^3
username
group
password
ls -lt
cd /user/test
./runTestScript.scr
exit
```

Reliability and Performance Tests - version 7.x

General

Automize 7.x uses Suns Java 1.4 machine. If it runs on one system, it should **supposedly** run on all systems. The Java machine is tested rigorously on all released platforms by Sun Microsystems Java team.   However, we rigorously test all versions of Automize on the following platforms:
1) Windows (both 95 and NT families, including servers)
2) MacOSX,
3) Linux Redhat

We also provide some releases for other Java 1.4 enabled platforms. We cannot test on all other Java 1.3 platforms, however, since we confirm that it runs perfectly on windows, mac, and linux, we are very confident that it will run on all other java 1.3 platforms.

Debug Log

The debug log is the single most important information we use to figure out why your tasks fails (either user input error, system or other error or bug). We constantly refine our code when we go through your debug logs. 100% of our code is in error catching loops that write to the debug log. This has helped the program to be extremely stable. By version 7.x, we have cleaned up a lot bugs. We used information that was in the users 6.x debug log.

Scheduler engine CPU usage

The scheduler engine just sets timers for your tasks. Every task has one timer associated with it. Typically when no tasks are running, the scheduler engine will consume almost very little (~0%) of CPU time. When the task is triggered, the CPU usage will depend on the task.

Scheduler reliability testing

Conditions: Ultra1, Solaris 8.0, 256MB Ram, 166MHz, 2 Tasks (copy & delete) run every 15 seconds for 1 month.
Results: RAM usage plateaus at 60-80 MB during the entire month. CPU usage ~2-5%

Zip Task

Conditions: Win2000, 256MB Ram, 667 MHz, Source folder size is ~1.8 GB, Source files zipped is 20031
Results: Zip time is ~24 minutes, Zip file size is ~1.3GB, Average CPU is ~70-90%
Output:

```
May 17, 2002 1:23:49 PM
Zip Directories - Zip
Zip session summary:
Number of files in directory = 20031
Number of files passing filename filter = 20031
Number of files passing date modified criteria = 20031
Number of files zipped = 20031
Zip file created : C:\Documents and Settings\melvin\junk\Zip.zip
File Size = 1325729962 bytes

May 17, 2002 1:23:50 PM
Zip Directories - Zip - Exit Code is 0
```

Copy Task

Conditions: Win2000, 256MB Ram, 667 MHz, Source folder size = ~150 MB, Source files copied = 5667, Subfolders = 419
Results: Copy time = ~5 minutes, Average CPU = ~20-40%.
Output:

```
May 17, 2002 2:00:10 PM
Copy Files - Copy
Copy session summary:
Number of files in directory = 5667
Number of files passing filename filter = 5667
Number of files passing date modified criteria = 5667
Number of older source files copied (only if 'Ignore file timestamp' option is selected) = 0
Number of source files older than target files (i.e. not copied) = 0
Number of files copied = 5667

May 17, 2002 2:00:11 PM
Copy Files - Copy - Exit Code = 0
```

Ftp Task

Conditions: Client Win2000, 256MB Ram, 667 MHz. Ftp Server = Solaris Sparc 8.0, 10 Mbps network speed.
Ftp task: Source folder size = 150 MB, Source files transferred = 5642, Subfolders = 419, log transfer or backup options were NOT selected.

Results:

Ftp Get time = ~40 minutes, client CPU usage = varies 5-20%.
Ftp Put time = ~60 minutes, client CPU usage = varies 20-100%.
The Ftp transfer time and %CPU usage will be higher if the log transfer or backup options are selected.

Output:

```
May 17, 2002 5:58:36 PM
FTP session summary:
Total files = 5642
Files meeting filename criteria (C1) = 5642
Files meeting transfer criteria per date modified & criteria C1 (C2) = 5642
Files meeting transfer criteria based on new modified files & criteria C2 = 5642
Total files downloaded = 5642

May 17, 2002 5:58:37 PM
Ftp - Ftpget - Exit Code = 0
```

Text Search

This task allows you to search for a string of text within a file. If the text is found, then the line containing the text is appended to a results file. You can also choose to append surrounding lines to the results file. You can also search for multiple strings of text. Also, you can search multiple files in a directory tree.

WARNING: If you select the option to 'save lines as variables', the task will run much slower. If the search file has thousands of lines that are saved as variables then the task could even run out of memory. Please test this option with small files and slowly move to bigger files.

Source Directory

Enter the full pathname of the directory where the files you need to search are located.

Filename

Enter the filename filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Folder Path Filter

Leave this field blank if you want all folders to be processed. If you need specific folders to be processed, then enter the folder name or path, using the following wildcard rules. If a folder passes the filter, then all its subfolders will also pass the filter.

Simple Example: Folder = c:\test , subfolders = john, john\old, sam, sam\old, joe, joe\old

Folder path filter = john   . john and john\old are processed

Folder path filter = jo   . john, john\old, joe, joe\old are processed

Folder path filter = old   . sam\old, john\old, joe\old are processed

Folder path filter = test   . all files in c:\test and its sub folders are processed

Folder path filter = john\old.   only john\old is processed.

Folder path filter = c:\test\john   . john and john\old are processed

Search Term

Enter the search term required. If you have more than 1 search term, separate them by a character which is not included within the search term. Example: Searching^Java^Automation . This field also supports dynamic variables. Using dynamic variables, for example, you can search a log file for toady's date/time, and run a task if that entry is found.

Search Term Separator

Enter the separator character used between search terms, if you have more than 1 search term.

Surrounding Lines

If a search term is found, the line is appended to the results file. If you need to append surrounding lines also, select the desired number of surrounding lines to append.

Results Directory

Enter the full pathname of the directory where you need to put the results File. Note that the path is case sensitive on Unix systems.

Results File

Enter the name for the results File. If left blank, a default filename of your task title is used. This field also supports dynamic variables. Using dynamic variables, you can save the file to a customized name based on date/time.

Append to Filename

You can save each archive to a different filename based on current time/date, or based on custom code you set.

Windows specific tutorials

Windows Batch Files

Using Batch Files is a very powerful way to enhance your productivity.
Batch files are simple text files with commands.
Each command has to be entered on a new line followed by a carriage return.
The Text file is saved as 'xxxx.bat' to generate a batch file. . The maximum length of the filename is 8 characters.

Create a Simple Batch file

Open up a new text document using notepad.
Type in the Following command on the first line: Start notepad
From the 'File Menu' select 'Save As'
In the File Save Dialog Box, select 'All Files (*.*)' as the File Type.
Enter the name of the batch file to save as. Ex: Test.bat

Running a Batch file

To run a batch file, double click on its icon. For example, the Simple batch file code above, would launch notepad.

Passing Command-line parameters to a batch file

The parameters are entered after the full path to the batch file. There is a space between parameters.
Example: c:\mybat.bat command1 command2 command3

Using Command-line parameters in a batch file

Within the batch file, use %1 to refer to the first command line parameter (%2 for the second parameter etc.). This parameter is passed while launching the batch file.
Ex: To launch a batch file, and tell it to copy the file "c:\test.txt" to drive a:, we would do the following:
1) Launch your batch file(c:\mybat.bat) using: c:\mybat.bat c:\test.txt
2) In mybat.bat, use the following code: Copy %1 a:

Concept of Start/Current directory

If you plan to use relative paths in your batch file, it should know its current directory.
For example, to delete all text files in the current directory, we would use a batch file with the following code: DEL *.txt
This would work, if you run the batch file, by clicking on its icon.
However, if you used another batch file, located in a different directory, to run this batch file, then this code will not work. You need to enter the full path: DEL c:\junk\*.txt

Batch File Examples

In the examples below, Filepath1, Filepath2 refer to the full path of the executable files.
Ex: c:\program files\programs.exe or c:\windows\notepad.exe

The syntax for the Start command varies between Win95 and WinNT families.

Example 1: Starting programs simultaneously
Start Filepath1
Start Filepath2

Example 2: Starting 2nd program after 1st finishes
Start /w Filepath1
Start /w Filepath2
The /w switch tells the windows to wait for the programs to finish.

Example 3: Running program with command-line arguments
Start /w Filepath1 Command-line

You need to enter the command-line arguments after the Filepath1. A blank space is required after the Filepath1 and the first command-line argument. A blank space is also required between command-line arguments if you need to specify more than one argument.

Useful DOS Commands for Batch Files
From a MS-DOS window, type in 'command /?' for details on the commands. Some of these commands may not be available to you, depending on your system. For some commands like Ftp, type ‘ftp’ in a dos window. At the subsequent prompt, type ‘help’.

Commands which may be of use to you
attrib, break, call, command, copy, chcp, chdir (cd), chkdsk, cls, date, debug, del, deltree, dir, diskcopy, doskey, defrag, drvspace, dir, echo, edit, erase, exit, expand, fc, find, for, format, fdisk, ftp, goto, if, label, mem, mkdir (md), mode, more, move, netstat, net, path, pause, ping, prompt, rem, rename (ren), rmdir (rd), set, shift, sort, start, subst, time, type, verify, vol, xcopy

Backing up using XCopy
MS-DOS comes with a xcopy utility, which you can use, to backup files, or directories. The exact syntax varies with operating systems. Win 95, Win NT, and Win 2000 pro, have different syntax for XCopy. From a command window use xcopy /? for more information.

For example, to backup the c:\junk directory and all its sub directories to a:, and replace older files with newer files, and overwrite existing files without prompting, use
xcopy c:\junk a: /D /E /Y (the /Y switch may or may not be used depending on your system)

UNC Format

Universal Naming Convention (UNC) is a naming format, to specify the location of servers, drives, folders, printers, and other resources, on a local area network (LAN).

UNC names begin with \\ (two backslashes) and take the form: \\ComputerName\ShareName
For example, a share named 'Test' on a computer named 'Res' will have the UNC name: \\Res\Test

You can also use UNC names on Unix computers, but you have to use forward slashes rather than backslashes.
Example: //ServerName/Path

Example 1:
Remote System = win1
Remote mapped Folder = c:\test1
Mapped drive name = k:\
Share Name = test1

Use shared drive = \\win1\test1

Example 2:
Remote System = win2
Remote mapped Folder = c:\test2
Mapped drive name = m:\
Share Name = test2

If remote directory = c:\test2\temp
Mapped drive name = m:\temp

Use shared drive = \\win2\test2\temp

Unix Service

This software contains two separate programs. An administrator user interface, and a scheduler engine program which runs in the background. **By default, the scheduler engine in the background is not started with the 'nohup' command**. Hence, when you logout, the scheduler may shutdown. By using the 'nohup' command to launch the scheduler engine, the scheduler engine will keep running after user logout.

Restart engine using launch code

You can re-launch the engine using the '**Restart engine using launch code**' button. This automatically shuts down the current instance of scheduler engine. Then, the engine is restarted, using the new 'nohup' code.

Recommended nohup launch code

The recommended command to launch the engine using 'nohup', is shown in the text box. You can edit it as you wish.

Save code and always launch scheduler engine using this launch code

Select this checkbox, if you wish to always use nohup, to launch the scheduler engine.

Important Notes

1) If you are re-launching the scheduler engine, through a script or from a terminal, make sure that you shut down the scheduler engine, through the user interface first.
2) If launching Engine from command-line/script, start dir should be software install folder. Please 'cd' to the software install folder first.

Default Scheduler Engine Launch Code

If you used the install with JVM:
jre/bin/java -Xrs -cp .:help:hitek.jar:mail.jar:activation.jar:net.jar: .... com.hitek.engine.Engine
or if you used the install without JVM and java is in your path
java -Xrs -cp .:help:hitek7.jar:mail.jar:activation.jar:net.jar: .... com.hitek.engine.Engine

Look in the main install folder. All the .jar file names in the main install folder should be added after net.jar:

Unzip Archives

This task allows you to unzip files from zip archives.

Source Directory
Enter the full path name of the directory, where your zip files exists. Note that the path is case sensitive for Unix Ftp servers.

Filename
Enter the filename filter for the zip archives you wish to extract, using the following wildcard rules . This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Extract Directory
Enter the full path name of the directory, where you need to put the extracted files. Note that the path is case sensitive on Unix systems.

Place files in new subdirectory
If you select "None", no new subdirectory is created within the Extract Directory.
If you select Date, a new subdirectory based on the date, is created within the Extract Directory. All files are placed in this directory.
If you select Date, a new subdirectory based on date and time, is created within the Extract Directory. All files are placed in this directory.

CGI scripts

Beginning in version 6.x, this task has been merged into the Web Downloads task

User Defined Variables

This feature provides the command-line code to set an existing variable, add a new variable, or get the value of a variable. You can use this code, from an external program, script, batch file, or command-line (command prompt in windows or terminal in Unix). The command-line and working directory required to set the variable, will be automatically generated in the text fields. You can use the "Copy to Clipboard" feature to copy these fields to clipboard for use in your scripts or programs.

A) Setting your own variables for the program (Automize, AbleFtp etc..) to recognize

From the command-line, via your batch files / scripts etc, first set the program install folder as the current directory
cd install_folder (i.e. cd ..\Automize or cd ..\AbleFtp etc..)

on windows, use:

```
jre\bin\java.exe -cp .;hitek7.jar com.hitek.engine.mods.var.UserVariable set variable_name variable_value
```

IMPORTANT: variable_name cannot include any spaces in it

on Unix/MacOSX, use:

```
java -cp .:hitek7.jar com.hitek.engine.mods.var.UserVariable set name value
```

where variable_name = variable name, and variable_value = variable value

B) Getting User variable values for use in your scripts or batch files
From the command-line, via your batch files / scripts etc,
first set the program install folder as the current directory
cd install_folder

on windows, use:

```
jre\bin\java.exe -cp .;hitek7.jar com.hitek.engine.mods.var.UserVariable get variable_name
jre\bin\java.exe -cp .;hitek7.jar com.hitek.engine.mods.var.UserVariable get variable_name c:\test\test.txt
```

(outputs the variable to the file test.txt)

IMPORTANT: variable_name cannot include any spaces in it

on Unix/MacOSX, use:

```
java -cp .:hitek7.jar com.hitek.engine.mods.var.UserVariable get variable_name
java -cp .:hitek7.jar com.hitek.engine.mods.var.UserVariable get variable_name /home/test.txt
```

(will output the variable to the file test.txt)

where variable_name = variable name, and variable_value = variable value

Return Value:

The variable value will be returned as a string value in the system output stream. You should add code in your batch, script, or program, to read this value.

Also, if the variable is an integer, the exit code of the command above, is the integer value of the variable.

If the variable is not an integer, the exit code of the command above = -1001.

Using Help

There are three options to view the Help file.
1) Default java help system. Includes full text search and context sensitive help.
2) View HTML help using software browser.
3) View HTML help in native browser.

You can set the default Help system in the Settings menu of the software. You can view the latest online version of this help file, by visiting our web site. You can also access the online help, via the Help menu in the front panel.

Extract String Value from Variable

This task allows you to extract a part of a variable and save this part to a new variable.

Variable name
Select the variable you need to use from the list. For some variables like DATE, you need to manually enter the format.

Regex
To extract a substring from a variable, we have to use regex grouping. The values of the match groups is returned. See examples below.

The first subgroup is saved as variable taskTitle::Value1, second as taskTitle::Value2 etc.

Examples:

1) Variable value: 06-19-09 (some date)
Regex: ([0-9]{1,})
taskTitle::Value1 = 06
taskTitle::Value2 = 19
taskTitle::Value3 = 09

Explanation: ( ) is the grouping operator. [0-9] matches a number. [0-9]{1,} matches 1 or more numbers and puts it into the group.

2) Variable value: test12345test
Regex: ([0-9]+)
taskTitle::Value1= 12345

Explanation: ( ) is the grouping operator. [0-9] matches a number. [0-9]+ matches 1 or more numbers and puts it into the group. This group of numbers 12345 is extracted as the result.

3) Variable value: 12345Test6789
Regex: ([0-9]{1,})[a-zA-Z]{1,}([0-9]{1,})
taskTitle::Value1 = 12345
taskTitle::Value2 = 6789

Explanation: [a-zA-Z] finds any uppercase or lowercase letters. {1,} matches 1 or more. i.e. 1 or more uppercase or lowercase letters. ( ) is the grouping operator. [0-9] matches a number. [0-9]+ matches 1 or more numbers and puts it into the group. This group of numbers 12345 is extracted as the result.

Variables

This software supports variables using two techniques:

1) Many fields supporting variables: The value of these variables is resolved when the task runs. The fields, which support variables, are marked with . For example: to match files in a Filename field based on current month of year, you would enter the filename as: $%DATE::mm-yy%$. If the current month was June 2002, all files, which contain '06-02' in their name, would be selected.
2) Variable Monitor task: This task compares the value of the variable using a value, and a comparison criteria you define. If a match is found, you can run other tasks.

Types of variables

Task Variables: Task variable values are generated/updated when the tasks run. Example: Tasks exit codes are stored as the following variable: TaskTitle::EXITCODE
System Variables: System variables which are resolved when the current task runs, examples are DATE, USERNAME etc..
Java Variables: Java variables, which are available to the Java Virtual Machine. You can get the value of java variables from the 'Java Variables' menu item
User Variables: Users can add their own variables via batch files, scripts or programs. Users can link output from their own programs to this program.

Using variables in the text fields

The variable must be formatted as follow: $%**VariableFormat**%$

For System variables: $%**VariableName::VariableOptions**%$
Variable_Options may only be required by some Variables.
examples: $%DATE::hh:mm:yy%$ or $%USERNAME%$

For Task variables: $%**TaskTitle::VariableName**%$
examples: for TaskTitle = Test : $%Test::ExitCode%$ or $%Test::ErrorMsg%$

For User and Java variables: $%**Variable**%$
examples: for java variable user.name, use : $%user.name%$

Nested Variables

Single level nested variables are allowed using the format
$%....NEST-[nested_variable]-NESTEND...%$
examples :
1) $%FILE::DATE::pp-dd-YY_mm-hh-ss::NEST-fileMon::FilePath-NESTEND%$ gets the date modified time for the file path found in the variable 'fileMon::FilePath'.
2) $%FILE::DATE::pp-dd-YY_mm-hh-ss::c:\test\NEST-fileMon::FileName-NESTEND%$ gets the date modified time for the file found in the variable 'fileMon::FileName'.

Only 1 nested variable can be read in a single field.

List of currently available variables

To view the list of currently available variables, click on the 'Variable Monitor' task dialog (via Tasks menu / Monitors). The variables are listed in the Variable Name dropdown box. The variables are listed in the following order:
1) System variables
2) Task and user variables

3) Java variables

Manipulate Variable

This task allows you to modify and save the value of a variable. The new value of the variable is calculated using the following formula:

1) The existing variable value is scanned for the Find String. The scanning is done using regular expression classes.
2) If the Find string is found in the variable value, then it is replaced with the Replace String.

Variable name

Select the variable you need to monitor from the list. For some variables like DATE, you need to manually enter the format.

Find String

Enter the subsequence that you expect the Variable value to have. This field supports regular expressions.

Replace String

Choose the subsequence that will be used to replace the Find subsequence.

New Variable Name

Enter the name of the new variable to create. This is a required field.

Variables Syntax

This feature allows you to check the syntax of the variable. This is useful for the DATE variable etc., where extra formatting is required. It also allows you to check the current value of the variable. If the variable is not found, or if the variable is formatted incorrectly, you will receive an error message. Otherwise, the current value of the variable is displayed.

Variable Name: Enter your variable name with options, and click the 'Check Syntax' button. The value of the variable entered will be displayed in the Variable value box.

Variable Value: The current value of the variable entered is displayed, or error message if any.

Types of variables

Task Variables: Task variables values are generated/updated when the tasks run. Example: Tasks exit codes are stored as the following variable: TaskTitle::EXITCODE
System Variables: System variables values are calculated when the current task runs; examples are DATE, USERNAME etc..
Java Variables: Java variables, which are available to the Java Virtual Machine. You can get the value of java variables from the 'Java Variables' menu item
User Variables: Users can add their own variables via batch files, scripts or programs. Therefore, users can link output from their own programs to this program.

Variable Monitor

The Variable Monitor task is a special type of task which can be used in two ways:
1) If a variable criteria is met in this task, you can run another task.
2) The Variable Monitor task can also be embedded within a Chain task, and is useful for conditional task processing. If you want to use this task in a chain for conditional processing, select 'None' as the 'Task To Run'. If the criteria is satisfied, then the exit code = -100. Otherwise, the exit code = 0.

Variable name

Select the variable you need to monitor from the list. For some variables like DATE, you need to manually enter the format.

Variable value

Enter the value of the variable you want to monitor. This value will be dependent on the comparison criteria you select.

Comparison criteria

Choose your comparison criteria. This will dictate how the actual value of the system variable is compared to the variable value you entered.

For **Integer** comparisons, use: =, !=, , < . The values found will be converted to integers and compared
For **String** comparisons, use: equals, contains, starts with, ends with
For **Date** comparisons, use: Older than or Newer than. For Date comparisons, the Variable Value should specify both the value and the field type (m=minutes, h=hours, d=days). Date comparison examples:
For '5 days' use: 5 d
For '3 hours' use: 3 h
For '30 minutes' use: 30 m

Task To Run

Select the desired task to run, if any of the selected criteria are met. If you want to use this task in a chain for conditional processing, select 'None'. You can also select multiple tasks to run in sequence or to run simultaneously. To run tasks in sequence, use taskTitle1|taskTitle2|taskTitle3. To run tasks simultaneously, use taskTitle1&taskTitle2&taskTitle3.

There are 4 types of variables listed:
Task Variables: Task variables values are generated/updated when the tasks run. Example: Tasks exit codes are stored as the following variable: TaskTitle::ExitCode
System Variables: Program system variables which are resolved when the current task runs, examples are DATE, USERNAME etc..
Java Variables: Java variables, which are available to the Java Virtual Machine. You can get the value of java variables from the Utilities menu / Java Variables
User Variables: Users can add their own variables via batch files, scripts or programs. Users can link output from their own programs to this program.

Visual Script

This feature allows you to run multiple tasks in sequence. You can run programs, email, web, ftp and all other tasks from the Script. If a certain task fails, you can choose to stop the Script. You can also skip ahead to another task in the Script. The Visual Script task is an extension of the 7.x Chain task.

Creating a Script
From the front panel: click on the Script button.

Adding Tasks to your Script
You can add new tasks or existing tasks to your script.
To add a new task: In the task type tree, Select the task type and click on the Add New Task button
To add an existing task: Click on the Add Existing Task Button.

Email Notification of tasks in the Script
By default, only the Script itself can trigger email notification based on exit code. The tasks in a Script do not trigger email notification, even if they satisfy an email notification profile. You should select this option, in order to allow tasks in the Script to trigger email notification. The Script will move on to the next step, only after the email notification has been completed.

Starting Script at specified step
You can specify a step to start at, by entering the step number, or the step title in this field. By default, the Script will start at step 1, if this field is left blank.

Deleting Tasks from your Script
Select the task in the bottom table and click the delete button.

Editing Tasks in your Script
Select the task in the bottom table and click the edit button. The task parameters will be loaded into the window. You can then make any changes you require and click the "Update Task" button.

Positioning a Task within your Script
Select the task in the bottom table and use the move-up or move-down buttons as required.

Enable Step
You can enable or disable a step. If a step is disabled, the task will move on to the next step.

Skip to Step options
You can choose to skip to certain step in the Script, if a certain exit code criteria is met. If you wish to stop the Script, because a certain exit code criteria is met, enter a Skip to Step value, which is greater than the total steps in the Script.

Skip to Step (or Title)
1) You can enter either the step number in the Script, or the task title. This task title has to be a valid title in the Script list.
2) You can also run multiple tasks simultaneously using the following rules:
a) Separate each Task Title by &. Example: TaskTitle1&TaskTitle2&TaskTitle3 (no spaces allowed)
b) You can only enter Task Titles, NOT step numbers.
c) TaskTitle1 has to be a valid Task Title in the Script list
d) TaskTitle1 CAN NOT be the last step in the Script list. You can add a dummy delay task as the last step in the Script, if TaskTitle1 happens to be your last step.
e) TaskTitle2, TaskTitle3 etc. DO NOT have to be in the Script list. They have to be valid tasks in the task list.
f) In this simultaneous mode, all tasks (TaskTitle2, TaskTitle3. etc.) will be triggered simultaneously. The Script will then continue normally at TaskTitle1.

Rerun Options

You can choose to rerun the task, if it meets certain exit code criteria. You can also set the maximum number of times to rerun a task. The task will rerun, while the rerun criteria is valid, until the maximum rerun count has been met.

Maximum Wait Time

This is the maximum time (**T1**) the Script will wait for your task to finish. Enter 0 to wait forever until the task finishes its run. ***Please note that the Script will not kill or terminate your task because it is unsafe to do so.*** It just waits for your task to complete normally. If the task does not complete normally in the allotted time, the Script will assume that the task has failed. The email, web, ftp and most other tasks will terminate normally.
For the **DirChange** and **Command** tasks, please ensure that you can Terminate the task within the Task settings itself. Both these tasks have a safe Termination time (**T2**) that can be set within the task. Please set **T1 T2**. If you have set the Command or DirChange task to rerun on failure, please set the **T1 (T2+Td)*n**, where **Td** is the delay between reruns, and **n** is the number or reruns.

Why not just Kill a task when the wait time is exceeded

Short answer from us: In the Script, each task is run as a thread. Killing a thread is unsafe, and may corrupt your system hours or days in the future.
Long answer from Java documentation: Because it is inherently unsafe. Stopping a thread causes it to unlock all the monitors that were locked. (The monitors are unlocked as the Thread Death exception propagates up the stack.) If any of the objects previously protected by these monitors were in an inconsistent state, other threads may now view these objects in an inconsistent state. Such objects are said to be damaged. When threads operate on damaged objects, arbitrary behavior can result. This behavior may be subtle and difficult to detect, or it may be pronounced. Unlike other unchecked exceptions, Thread Death kills threads silently; thus, the user has no warning that his program may be corrupted. The corruption can manifest itself at any time after the actual damage occurs, even hours or days in the future.

Web Downloads

This feature enables you to download web pages, from your local intranet, or from the internet. The most common internet protocols http://, https://, & ftp:// can be downloaded. This feature will work with most web pages, except web pages using frames.

Download Url

Enter the full path of the URL to download (http://www.hiteksoftware.com). **You must include the protocol (i.e. http://).**
This field also supports dynamic variables. Using dynamic variables, you can select a URL based on current date.
Example: http://www.test.com/$%pp-yy%$.htm, will download http://www.test.com/07-02.htm

Many Ftp urls require username & password information to be sent in the url itself. The format may vary, but is usually of the form: ftp://username:password@host.com/folder/file
To hide your password, enter the password in the password field of the dialog box. Use the following format for the url:
ftp://username:$password$@host.com/folder/file
The string $password$ will be replaced by the value in the password field.

Examples:
1) http://www.cnn.com
2) ftp://comp.com/test/test.dat
3) ftp://username:$password$@test.com/folder/test.zip

Parameters (optional)

This is an optional field. If you did not enter input parameters in your url field, you can enter the parameter string here. You must separate each key=value pair by &
Example: name=joe&age=25&date=2/12/01
This field also supports dynamic variables. Example:
name=$%USERNAME%$&age=25&date=$%DATE::pp/dd/yy%$

Local Directory

Enter the full pathname of the local Directory to place the html file in.

Local Filename

Enter the required name of the downloaded file. Example test.htm. If you do not enter a name, a default name of "TaskTitle.htm" is used. This field also supports dynamic variables. Using dynamic variables, you can save the file to a customized name, based on date/time, or username, or other variable.

Maintain Timestamp

With this option selected, the download file will have the same timestamp, as the file on the web server.

Download Option

There are 3 download options:
1) Download always - This will overwrite the existing local file, irrespective of timestamp.
2) Download based on local file timestamp - If a local file exists in the download folder, a file will be downloaded, only if the file on the web server is newer than the existing local file.
3) Download based on archived url timestamp - Every time a file is downloaded by this task, the file timestamp on the web server, is saved locally in a file. If this option is selected, a new file is downloaded from the web server, only if its timestamp on the web server is newer than the saved value. IMPORTANT: Please note that not all web servers will provide the timestamp information. In this case, this option will not work.

Append to Filename

If you are regularly downloading a certain URL and wish to save each download to a different filename based on current time/date or custom code you set, choose the append file option which best suits your download frequency.

Proxy Parameters

If you are behind a firewall, enter the proxy settings. Your network administrator can provide your proxy settings or you may also obtain them from your web browser.

Download Embedded Images

This option allows you to download embedded images within the html page. The image formats supported are .jpg,.gif, .tif, .png, and .bmp. Only images in the same or child folder as the html page will be downloaded.

Web Interface Module

The Web Interface Module allows you to access and control the engine remotely via a web browser.

Using the Module
The URL to use is displayed on the module user interface. From a remote computer (or local for testing) enter this URL in a web browser.

Windows Command

This task is only available on Windows systems. This task allows you to run executable programs (.exe) or batch files (.bat). The task launches a native windows application called WinCommand.exe. WinCommand.exe will actually run your application, or batch file. This Windows Command Task generally provides better results, for windows batch files, than the standard Command Task. Executable programs (.exe) should perform similarly using, either the Command Task, or the Windows Command Task.

Command Line

Enter the command line. If you want to launch an executable file, or batch file, enter the full pathname to the file. Example:
c:\windows\notepad.exe
If the path to your executable file or batch file includes spaces, enter the pathname to the file in quotes. Example:
"c:\program files\mybat.bat"
You can enter a command-line parameter, to pass to your executable file, or batch file. You should enter a space between the file path, and the parameter. Leave spaces between any subsequent parameters. Example:
"c:\program files\mybat.bat" par1 par2

This field also supports dynamic variables. For example, to launch a batch file, and pass it the current time, use:
c:\data\test.bat $%DATE::hh:mm:ss%$

Working Directory

The working directory is required for your program, or script, to run correctly. Use the full pathname to the Working Directory. You do not need to enclose the working directory in quotes, even if the pathname has spaces in it.

Environment Variables

Enter an array of strings. Each element of the array, defines a single environment variable setting. The format for each environment variable is name=value. You should separate each name=value pair, by a delimiter character. You can choose the delimiter to separate each element. The default delimiter is ^. If any of your variable name=value pairs contains the ^ character, then you should set another character, as your delimiter. The delimiter cannot be "=", or any other character, that appears in your name=value pairs.
Example of correctly formatted environment variables, on windows systems:
PATH=c:\test;c:\test2;c:\test3^TEMP=c:\temp

This field also supports dynamic variables. For example, to launch a batch file and set the variable 'CUSTOM_DATE', use:
CUSTOM_DATE=$%DATE::Qqq dd, YY HH:mm:ss am_pm%$
This will set the variable CUSTOM_DATE in the batch file to: Aug 21, 2002 10:21:23 AM etc..

Termination Time

The termination time is the maximum time that your program can run. The task will terminate your program, if it is still running, after the maximum allowed time. Enter a value = 0, if you do not want to terminate your program. In this case, the task will wait indefinitely, for your task to finish.

Polling interval

While your program is running, the task continuously polls your program. The task will check if your program has completed. It will obtain the Exit Code value, when the program exits. You can select the polling interval (default = 15 seconds). The minimum interval is 1 second. We recommend using 15 seconds.

XML Loop

This task allows you to loop through data in a XML file.
An element needs to be specified. This element can be located at any level within the XML file node structure.
With every loop, the elements value and attribute names and values are set as variables.
If the element does not have a value, but has sub elements, then their names and values are set as variables.
With every loop, you can run another task.

Consider the following XML file:

```
(contacts)
(contact id="1")
(firstname)John(/firstname)
(lastname)Doe(/lastname)
(/contact)
(contact id="2")
(firstname)Jane(/firstname)
(lastname)Deer(/lastname)
(/contact)
(/contacts)
```

If element name field = contact, for each loop the following variables would be set:
AttributeName1=id , AttributeValue1=1 , SubElementName1 = firstname, SubElementValue1 = John etc..

If element name field = firstname, for each loop the following variables would be set:
ElementValue = John etc...

XML Monitor

This task allows you to Monitor data in a XML file.
You can monitor the value of 1 Element or Attribute.
If the value satisfies the comparison criteria, then a task specified in the Task To Run field will be run.

Consider the following XML file:
(contacts)
(contact id="1")
(firstname)John(/firstname)
(lastname)Doe(/lastname)
(/contact)
(contact id="2")
(firstname)Jane(/firstname)
(lastname)Deer(/lastname)
(/contact)
(/contacts)

If element/attribute name field = id, the following values of id will be found: [1,2]
If you set a criteria as: run task if any value greater than 0, then the criteria will be satisfied.
If you set a criteria as: run task if all values greater than 1, then the criteria will not be satisfied.

XML Validate

This task allows you to Validate XML files in a folder or an entire directory tree.

Source Directory

Enter the full pathname of the directory where your source files exists.

Target Directory

Enter the full pathname of the directory where you need to put the target files.

Filename

Enter the filename filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables.

Folder Path Filter

Leave this field blank if you want all folders to be processed. If you need specific folders to be processed, then enter the folder name or path,using the following wildcard rules.
If a folder passes the filter, then all its subfolders will also passthe filter.
Simple Example: Folder = c:\test , subfolders = john, john\old, sam, sam\old, joe, joe\old
Folder path filter = john . john and john\old are processed
Folder path filter = jo . john, john\old, joe, joe\old are processed
Folder path filter = old . sam\old, john\old, joe\old are processed
Folder path filter = test . all files in c:\test and its sub folders are processed
Folder path filter = john\old. only john\old is processed.
Folder path filter = c:\test\john . john and john\old are processed

Validation Method

Select the validation method required. Most commonly used validation methods are WellFormed Only and External XSD.

Validation Filepath

This field is needed for the External File validation method. You can dynamically use the Current XML filename variables and current folder variable. This allows to select a different External validation file for every XML file.
Example: $%taskTitle::CurrentFolder%$\$%taskTitle::CurrentXmlFileWithoutExtension%$.xsd
i.e. if xml file is c:\temp\test.xml, the above variables would set the validation filepath as: c:\temp\test.xsd

XML XPath

This task extracts data from an XML file based on an input XPath expression.
Every field in this task accepts variables. You can use a Directory Loop task, and for every loop, call this task and process all files in a directory!

XML data file
Enter the full path for the XML data file.

XPath Expression
XPath expression allows you to specify specific elements or list of elements within an XML file. Please search on for tutorials on XPath.

Output File
The result of the XPath expression is stored in an output file. The normal format is a list of values (one per row) which satisfy the XPath expression.
You can dynamically use the Current XML filename variables and current folder variable. This allows to set a different output file for every XML file.
Example: $%taskTitle::CurrentFolder%$\$%taskTitle::CurrentXmlFileWithoutExtension%$.htm
i.e. if xml data file is c:\temp\test.xml, the above variables would set the output filepath as: c:\temp\test.htm

XML Loop

This task allows you to loop through data in a XML file using XPath Expression. There are a number of tutorials online on XPath expressions.
The XPath expression entered should result in a list of values.
The task will then loop through each value satisfying the XPath expression.
With every loop, you can run another task.

Consider the following XML file:

```
(contacts)
(contact id="1")
(firstname)John(/firstname)
(lastname)Doe(/lastname)
(/contact)
(contact id="2")
(firstname)Jane(/firstname)
(lastname)Deer(/lastname)
(/contact)
(/contacts)
```

If XPath Expression = /contacts/contact/firstname, for each loop the following variable would be set:
CurrentValue = John or CurrentValue = Jane

XML XSLT transform

This task transforms XML data into HTML or other formats, based on a XSL style sheet.
Every field in this task accepts variables. You can use a Directory Loop task, and for every loop, call this task and transform all files in a directory!

XML data file

Enter the full path for the XML data file.

XSL Stylesheet

Enter the full path of the XSL style sheet.
You can dynamically use the Current XML filename variables and current folder variable. This allows to select a different XSL style sheet for every XML file.
Example: $%taskTitle::CurrentFolder%$\$%taskTitle::CurrentXmlFileWithoutExtension%$.xsl
i.e. if xml data file is c:\temp\test.xml, the above variables would select the following XSL style sheet: c:\temp\test.xsl

Output File

You can dynamically use the Current XML filename variables and current folder variable. This allows to set a different output file for every XML file.
Example: $%taskTitle::CurrentFolder%$\$%taskTitle::CurrentXmlFileWithoutExtension%$.htm
i.e. if xml data file is c:\temp\test.xml, the above variables would set the output filepath as: c:\temp\test.htm

Zip Directories

The Zip Directories task allows you to Zip an entire directory and its sub-tree to a single zip archive. Each file in the directory is compressed, before being added to the zip archive. The biggest zip file that can be created is 4 GB.

Directory to Zip
Enter the full pathname of the directory, where your source files exists.

Filename
Enter the filename filter for the files you wish to copy using the following wildcard rules. This field also supports dynamic variables. Using dynamic variables, you can select a file based on current date/time.

Folder Path Filter
Leave this field blank if you want all folders to be processed. If you need specific folders to be processed, then enter the folder name or path, using the following wildcard rules. If a folder passes the filter, then all its subfolders will also pass the filter.
Simple Example: Folder = c:\test , subfolders = john, john\old, sam, sam\old, joe, joe\old
Folder path filter = john   . john and john\old are processed
Folder path filter = jo   . john, john\old, joe, joe\old are processed
Folder path filter = old   . sam\old, john\old, joe\old are processed
Folder path filter = test   . all files in c:\test and its sub folders are processed
Folder path filter = john\old.   only john\old is processed.
Folder path filter = c:\test\john   . john and john\old are processed

Include sub directories
Select this option, if you want to zip the entire sub directory tree below the main directory.

Directory to place Zip Archive
Enter the full pathname of the directory, where you need to put the newly created zip archive.

Name for Zip Archive
Enter the filename for zip archive. This field also supports dynamic variables. Using dynamic variables, you can save the file, to a customized name based on date/time, or username, or other variable.

Append to Filename
You can save each archive to a different filename, based on current time/date, or on custom code you set.

Date Filter
If you need to filter files based on file modified date, select this option. For the between option, enter 2 values separated by '-' (2-4 etc..). Examples:
Older than 5 Minute, Newer than 2 Day, Between 3-5 Day, Between 1-4 Hour

Adding or Modifying a Task

```
Usage: clm "add::Task.TASK_TYPE=TaskTypes.SSH2|*|Task.TASK_TITLE=ssh|*|Task.COMMENT=my test task|*|Task.PARAMETERS1=mac_test|*|Task.PARAMETERS2=123Y|*|Task.PARAMETERS3=245"
```

You do not have to enter all 50 parameters.
If editing existing task, only enter modified parameters.
If adding new task, enter all required parameters.
Task.TASK_TITLE and Task.TASK_TYPE are required parameters.
Look at the task property files in the 'data\tasks' folder to see the parameters for each task type.

Also see addprop function which allows you to change a single parameter in a task, schedule or settings file.

Adding or Modifying a Schedule

Usage: clm

```
"addx::Schedule.RUN=true|*|Task.TASK_TYPE=TaskTypes.ALARM|*|Task.TASK_TITLE=alarm|*|Task.COMMENT=|*|Schedule.FREQUENCY=1|*|Schedule.PERIOD=1|*|Schedule.HOUR=12|*|Schedule.MINUTE=0|*|Schedule.RUN_HOURS=0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,|*|Schedule.RUN_DAYS=0,1,2,3,4,5,6,|*|Schedule.RUN_MONTHS=0,1,2,3,4,5,6,7,8,9,10,11,|*|Schedule.RUN_DATES=|*|Schedule.RUN_WEEKS=1,2,3,4,5|*|Schedule.START_DATE=|*|Schedule.END_DATE="
```

Schedule.PERIOD=0 (second)
Schedule.PERIOD=1 (minute)
Schedule.PERIOD=2 (hour)
Schedule.PERIOD=3 (day)
Schedule.PERIOD=4 (week)
Schedule.PERIOD=5 (month)

If editing existing schedule, only enter modified parameters.
If adding new schedule, ALL parameters are required.
Schedule.RUN,Task.TASK_TITLE,Task.TASK_TYPE are always required.
Tip: commands cannot have Enter / Return in them. Use a text editor to create your command in a single line. Then paste it here.

Also see addprop function which allows you to change a single parameter in a task, schedule or settings file.

```
Usage: clm command[::par1][::par2]...

Command List
? - Displays this command reference page
?::add - Help information on how to add a new task or modify a task
?::addx - Help information on how to add a new schedule or modify a schedule
?::pars - Help information on task parameters 1-50 for each task type
?::type - Help information on TaskTypes used
init::program_name - program_name = Automize, AbleFtp, or JaSFtp etc (Case sensitive). This command is only required if you are using the clm for the first time, and also have not used the user interface yet
start - Starts scheduler engine
stop - Stops scheduler engine
status - Checks if scheduler engine is running
list - Displays Task List
list::taskTitle - Displays task data for task 'taskTitle' in a format suitable for editing
listx - Displays Schedule List
listx::taskTitle - Displays schedule data for task 'taskTitle' in a format suitable for editing
run::taskTitle - runs task 'taskTitle'.
add::... - adds/edits a task.  Use 'clm list' for information to add/edit tasks.
addx::.... - adds/edits a schedule. Use 'clm listx' for information to add/edit schedules.
addprop::filepath|*|key|*|value  - adds/modifies a value in a task, scheduler or settings file.  Example usage: clm "addprop::data/settings/ftpProfiles.jsd|*|profile|*|test" or clm "addprop::data/tasks/testtask|*|TASK.PARAMETERS1|*|testvalue"
listprops::filepath - lists all the keys/values in a settings file.  Example usage:  clm listprops::data/settings/ftpProfiles.jsd
listtaskfiles - lists the filepaths of all existing task files.  usefull when you need to use listprops or addprop functions
listschedulefiles - lists the filepaths of all existing schedule files.  usefull when you need to use listprops or addprop functions
listsettingfiles - lists the filepaths of all existing settings files in settings folder.  usefull when you need to use listprops or addprop functions
del::taskTitle - deletes taskTitle from task list
delx::taskTitle - deletes taskTitle from schedule list
sch::taskTitle - schedules taskTitle in schedule list
sus::taskTitle - suspends taskTitle in schedule list
susAll - suspends all schedules
resAll - resumes all schedules
out - Displays output log
act - Displays activity log
err - Displays debug log
task::taskTitle - Displays the Tasklog for taskTitle
listVar - Displays a list of user variables
getVar::variable_name - returns the value of user variable
setVar::value_to_set::variable_name - sets the user variable value
syncList - Displays a list of synchronization profiles
syncDis::profile - Displays the settings for the synchronization 'profile'
syncAdd::profile::taskList::option::waitTime::afterWait::priority - Adds a new
profile or modifies synchronization profile data if profile exists.
syncDel::profile - Deletes the synchronization profile
emailList - Displays a list of email server profiles
emailDis::profile - Displays the smtp server settings for the email 'profile'
emailAdd::profile::host::port::emailAddress::senderName::authenticationType::popServer::user::pwd - Adds a new profile or modifies email profile data if profile exists.
emailDel::profile - Deletes the email profile
emailNotifyList - Displays a list of email notification profiles
```

```
emailNotifyDis::profile - Displays the email notification settings for this profile
emailNotifyAdd::profile::true/false::notificationEmail::"crit"::exitCodeValue::priority -  Adds a new email notification profile or modifies profile data if
profile exists. Example: to send notification email if task exits with exit
code 0, use: clm emailNotifyAdd::true::me@test.com::"="::0
emailNotifyDel::profile - Deletes this email notification profile
ftpList - Displays a list of ftp profiles
ftpDis::profile - Displays the ftp profile settings
ftpDel::profile - Deletes this ftp profile
ftpAdd::..... - Adds/edits a ftp profile.
usage:
ftpAdd::profile::hostname/ip::port::username::password::FtpDirectoryListParser.FULL_LISTING/FtpDirectoryListParser.NAMES_ONLY_LISTING::UNIX/WINDOWS/VMS/MVS/OS2/OS400::encoding name or DEFAULT_ENCODING::account::passive true/false::timeout::firewall profile name or All.none
example:
ftpAdd::test_profile::193.13.1.1::21::john::cleartextpassword::FtpDirectoryListParser.FULL_LISTING::UNIX::DEFAULT_ENCODING:: ::true::65::All.none
ftpFireList - Displays a list of ftp firewall profiles
ftpFireDis::profile - Displays the ftp firewall profile settings
ftpFireDel::profile - Deletes this ftp firewall profile
ftpFireAdd::..... - Adds/edits a ftp firewall profile.
usage:
ftpFireAdd::profile::hostname/ip::port::username::password::USER/OPEN/SITE/Custom::custom commands or blank
example:
ftpFireAdd::test_profile::192.168.0.4::21::john::cleartextpassword::USER::-
backupDir - Displays the last backup directory
backup::dir_path - backup data to dir_path
backup::default - backup data to last backup directory
restore::dir_path - restores data from dir_path
restore::default - restores data from last backup directory
registerSched::program::code - Registers & unlocks the scheduler engine for
specific 'program' using the registration 'code'. program = Automize or
AbleFtp or JaSFtp etc..
registerCLM::code - Registers & unlocks this CommandLine Module using
'code'

NOTE: n is 0 based (i.e. first task in task or schedule list has index = 0)

More CLM help information is available in the user interface help file.
```

Back to Index

Using Ftp to set/view remote data

The easiest way to setup or view data on a remote system is via the user interface itself on your local system. See the section on managing multiple remote schedulers. **All the data files are in the 'inputs' folder within the main install folder.**

Setting (export) data to remote system
1. Telnet to and Shut down the scheduler on the remote system using: clm stop
2. Launch the user interface to create the tasks and schedules on your local system itself.
3. Use an ftp client to transfer all files in the 'inputs' folder from your local machine to the 'inputs' folder on the remote machine
4. On the remote system, start up the scheduler using: clm start

View (import) data from remote system
1. **BACKUP your local scheduler data, using the Backup feature, if the scheduler is also running on your local system!!!**
2. Shut down the scheduler on your local system.
2. Shut down the user interface on your local system.
3. Use an ftp client to transfer all files in the 'inputs' folder from the remote machine to the 'inputs' folder on your local machine.
4. Start up user interface on your local system. Shut down the scheduler engine so that it does not run any tasks on your system.
5. Make any changes that you desire. Then you can export the data back to the remote system (see section above).
6. Restore your local scheduler data, using the Restore feature, if the scheduler is also running tasks on your local system

Command Line Module

Back to Index

Introduction

The CommandLine Module (CLM) allows a user to:
1) Control the scheduler engine on a remote system
2) Allow a programmer to control the scheduler tasks and schedules from their own programs, scripts or batch files

Back to Index

Managing remote scheduler(s)

We developed the commandline module on a Windows system and tested it by controlling scheduler engines running on MacOSX, Linux and Solaris. You can manage remote scheduler(s) either by:
1. Using only the CLM
2. Using the CLM in combination with the user interface & Ftp

Managing Remote Systems using the CLM in combination with Local User Interface & Ftp
The easiest way to setup or view data on a remote system is via the user interface itself on your local system. To prevent confusion, it helps to understand and organize the data well on your local system to prevent confusion between the programs data on your local system and the remote system.

1. The main install directory contains an 'data' folder
2. All the data files are in this 'data' folder. The user interface saves all your updates to data files in this folder.
3. Create new folders in the folder called 'data_remote1' , 'data_remote2', 'data_remote3' etc.. for each remote system that you will control. In our case, we used 'data_mac', data_solaris', and data_linux'.
4. Create a new folder in the main install folder called 'data_localhost' for your local system, if you also run tasks on your local system.
5. If the scheduler is also running tasks on your local system, shut down the scheduler & shut down the user interface. **Then Backup (copy) all the data from the 'data' folder to 'data_localhost'**
6. If you wish to view data from 'remote2', you can either copy the data from 'data_remote2' to the 'data' folder or by ftp directly FROM the 'data' folder on the 'remote2' system.
7. Launch the user interface to create the tasks and schedules for 'remote2'
8. After you have modified the data for 'remote2', you can backup the 'data' folder to 'data_remote2' and ftp directly TO the 'data' folder on the remote system.
9. Then use the CLM module to STOP and then START the engine on 'remote2'. This will force the engine to load the new data.
10. If the scheduler is also running tasks on your local system, you can restore your local systems program data from 'data_localhost' to 'data'.

Task parameters - updated for version 7.x
These parameters are dependent on the task. These parameters may also change with versions, if more features are added to a task. These parameters are exactly in the same sequence in the task table of the user interface. If all 50 parameters are not used, they should be left blank. However, the correct number of |*| are required when used in the CLM.

Command and Windows Command
Parameter1 = commandline
Parameter2 = working directory
Parameter3 = polling interval in seconds
Parameter4 = terminate time in minutes
Parameter5 = environment parameters
Parameter6 = environment parameters delimiter

Copy Files
Parameter1 = source directory
Parameter2 = target directory
Parameter3 = filename
Parameter4 = append to filename (None, Date , Date/Time or custom code)
Parameter5 = include subdirectory option? (true/false)
Parameter6 = create new subdirectory (None, Date, Date/Time)
Parameter7 = date based copy? (true/false)
Parameter8 = older than/newer than
Parameter9 = time (integer number)
Parameter10 = Minute/Hour/Day
Parameter11 = ignore file timestamp? (true/false)
Parameter12 = rename filename
Parameter13 = copy/move/rename

Zip Directories
Parameter1 = source directory
Parameter2 = target directory
Parameter3 = filename
Parameter4 = append to filename (None, Date , Date/Time or custom code)
Parameter5 = include subdirectory option? (true/false)
Parameter6 = create new subdirectory (No, Date, Date/Time)
Parameter7 = date based zip? (true/false)
Parameter8 = older than/newer than
Parameter9 = time (integer number)
Parameter10 = Minute/Hour/Day
Parameter11 = path type

Delete Files
Parameter1 = source directory
Parameter2 = filename
Parameter3 = date based delete?(true/false)
Parameter4 = older than/newer than
Parameter5 = time (integer number)
Parameter6 = Minute/Hour/Day
Parameter7 = delete subdirectories(true/false)
Parameter8 = delete empty subdirectories(true/false)

Text Search
Parameter1 = source directory
Parameter2 = filename
Parameter3 = search term(s)
Parameter4 = search term separator
Parameter5 = surrounding lines to append to results file

Parameter6 = case sensitive search? (true/false)
Parameter7 = results directory
Parameter8 = results filename
Parameter9 = append to results filename (None, Date , Date/Time or custom code)
Parameter10 = replace? (true/false)
Parameter11 = replace term(s)
Parameter12 = include directories (true/false)
Ping
Parameter1 = host
Parameter2 = port
Telnet
Parameter1 = host
Parameter2 = port
Parameter3 = delay time between commands in seconds
Parameter4 = commands separated by *EOL* (Example: username*EOL*password*EOL*ls -lt*EOL*exit)
Telnet(Adv)
Parameter1 = host
Parameter2 = port
Parameter3 = username
Parameter4 = password
Parameter5 = maximum login time (seconds)
Parameter6 = login prompt (typically :)
Parameter7 = maximum command time (minutes)
Parameter8 = polling interval for each command (seconds)
Parameter9 = telnet prompt ($, % or etc..)
Parameter10 = prompt characters, default = 10
Parameter11 = commands separated by *EOL* (Example: username*EOL*password*EOL*ls -lt*EOL*exit)
Ftp
Parameter1 = remote directory
Parameter2 = local directory
Parameter3 = filename (based on wildcards, see help for details)
Parameter4 = Ascii/Binary
Parameter5 = ftp option (Put File(s)/Get File(s)/Delete File(s)
Parameter6 = append to filename (None, Date or Date/Time)
Parameter7 = log transfer? (true/false)
Parameter8 = transfer modified files only? (true/false)
Parameter9 = include subdirectories? (true/false)
Parameter10 = ftp profile name
Parameter11 = transfer based on modified date? (true/false)
Parameter12 = date value
Parameter13 = delete source file after transfer? (true/false)
Parameter14 = transfer with temporary extension? (true/false)
Parameter15 = temporary extension (ex: .tmp)
Parameter16 = backup local file? (true/false)
Parameter17 = local backup directory
Parameter18 = backup remote file? (true/false)
Parameter19 = remote backup directory
Parameter20 = maintain timestamp (true/false)
Parameter21 = transfer modified source files using Ftp log? (true/false)
Parameter22 = comparison criteria (newer than/ older than)
Parameter23 = period type (Day/Hour/Minute)
Ftp Monitor
Parameter1 = remote directory
Parameter2 = filename

Parameter3 = file exists check (true/false)
Parameter4 = file length check (true/false)
Parameter5 = ftp length option (greater than/less than)
Parameter6 = file length value in kb
Parameter7 = file date check (true/false)
Parameter8 = file date option (newer than/older than)
Parameter9 = file date value
Parameter10 = file date option (minute/hour/day)
Parameter11 = file totals check (true/false)
Parameter12 = file totals option (greater than/less than)
Parameter13 = file totals value
Parameter14 = Run task if any change is detected (true/false)
Parameter15 = Task to run (tasktitle)
Parameter16 = Ftp profile name
Ftp Commands
Parameter1 = host
Parameter2 = port
Parameter3 = password
Parameter4 = commands (Each Command should be separated by *EOL*, Ex: user test*EOL*pass mypwd*EOL*PWD)
Email
Parameter1 = recipients (format = To:email1,[Cc:email2,Bcc:email3,Reply-To:email4]
Parameter2 = subject
Parameter3 = attachments (format = filepath1,filepath2)
Parameter4 = message (newlines separated by *EOL*, Ex: this is a *EOL* new line)
Parameter5 = email profile name
Parameter6 = append to attachment (None, Date or Date/Time)
Parameter7 = headers
Bulk Email
Parameter1 = recipient list file path
Parameter2 = subject
Parameter3 = attachments (format = filepath1,filepath2)
Parameter4 = message (newlines separated by *EOL*, Ex: this is a *EOL* new line)
Parameter5 = email profile name
Parameter6 = append to attachment (None, Date or Date/Time)
Parameter7 = delay
Parameter8 = headers
Check Email
Parameter1 = criteria1 (date/sender/subject)
Parameter2 = option1 (newer than X days/older than X days/newer than X hours/older than X hours/equals/contains/starts with/ends with)
Parameter3 = value1
Parameter4 = criteria2 (date/sender/subject)
Parameter5 = option2 (newer than X days/older than X days/newer than X hours/older than X hours/equals/contains/starts with/ends with)
Parameter6 = value2
Parameter7 = and/or
Parameter8 = mail folder path
Parameter9 = append to filename (None, Date or Date/Time)
Parameter10 = server type (POP3/IMAP)
Parameter11 = host
Parameter12 = port
Parameter13 = user
Parameter14 = password

Parameter15 = delete message
Web
Parameter1 = download URL
Parameter2 = local directory
Parameter3 = filename
Parameter4 = append to filename (None, Date or Date/Time)
Parameter5 = authentication required (true/false)
Parameter6 = host
Parameter7 = port
Parameter8 = user
Parameter9 = password
Parameter10 = download images? (true/false)
Parameter11 = parameters
Parameter12 = file date check (true/false)
Parameter13 = file date criteria (greater than / less than etc..)
Parameter14 = date value
Parameter15 = date type (minute/hour/day)
Parameter16 = maintain timestamp (true/false)
Parameter17 = download option (Download always/Download based on local file timestamp/ Download based on archived url timestamp)
Url Monitor
Parameter1 = url to monitor
Parameter2 = spider depth
Parameter3 = local directory
Parameter4 = place files in subdirectory (No, Date or Date/Time)
Parameter5 = authentication required (true/false)
Parameter6 = host
Parameter7 = port
Parameter8 = user
Parameter9 = password
Parameter10 = filename
Parameter11 = download option
Parameter12 = maintain timestamp (true/false)
Database SQL
Parameter1 = sql statement
Parameter2 = results directory
Parameter3 = results filename
Parameter4 = append to filename (None, Date or Date/Time)
Parameter5 = driver class name
Parameter6 = database url
Parameter7 = authentication required (true/false)
Parameter8 = username
Parameter9 = password
File Monitor
Parameter1 = directory
Parameter2 = filename filter
Parameter3 = file exists? (true/false)
Parameter4 = file length check? (true/false)
Parameter5 = greater than/less than
Parameter6 = size in kb (integer number)
Parameter7 = file date check? (true/false)
Parameter8 = older than/newer than
Parameter9 = time (integer number)
Parameter10 = Minute/Hour/Day

```
Parameter11 = total files check? (true/false)
Parameter12 = greater than/less than
Parameter13 = number of files (integer number)
Parameter14 = Task to Run (must be valid taskname in task table)
Directory Monitor
Parameter1 = Directory to monitor
Parameter2 = Filename filter
Parameter3 = include subdirectories (true/false)
Parameter4 = Task to Run (must be valid taskname in task table)
Directory Change
Parameter1 = Directory to monitor
Parameter2 = maximum wait time in minutes (integer number)
Parameter3 = polling interval in seconds (integer number)
Parameter4 = Task to Run (must be valid taskname in task table)
Parameter5 = Use as daemon (true/false)
Variable Monitor
Parameter1 = Variable name
Parameter2 = Variable value
Parameter3 = comparison criteria (equals, contains, starts with, ends with, "=", "!=", "", "<")
Parameter4 = Task to Run (must be valid taskname in task table)
File Variable
Parameter1 = Variable name
Parameter2 = Filepath to output variable to.  (ex: c:\data\test.txt)
Parameter3 = Filing options (insert at start or append at line . etc..)
Parameter4 = Option value corresponding to Filing Option chosen
Parameter5 = Add line separator before inserting variable
Parameter6 = Add line separator after inserting variable
Archive Logs
Parameter1 = archive directory
Parameter2 = append to filename (None, Date or Date/Time)
Parameter3 = place files in subdirectory (No, Date or Date/Time)
Auto Backup
Parameter1 = archive directory
Parameter2 = not used
Parameter3 = place files in subdirectory (No, Date or Date/Time)
Email Logs
Parameter1 = email addresss to receive notification (separate multiple addresses by comma ,)
Chain
Parameter1 = total steps in chain
Parameter2 = All the Chain data in a single line
Data for each step is separated by |@|
Each step has multiple entries separated by |#|
   Entry1 = task type
   Entry2 = taskTitle
   Entry3 = maximum wait time in minutes
   Entry4 = Skip to step exit code choice (true/false)
   Entry5 = exit code comparison criteria (= , != , , < )
   Entry6 = exit code value
   Entry7 = Skip to step (step number or taskTitle)
   Entry8 = rerun step? (true/false)
   Entry9 = rerun exit code comparison criteria (= , != , , < )
   Entry10 = rerun exit code value
   Entry11 = number of times to rerun step
   Entry12 = delay time between reruns
```

```
    Entry13 = Always stop chain at this step?
Wakeup
Parameter1 = Mouse Move/Key Stroke/Mouse Click
Shutdown
Parameter1 = Logoff / Reboot / Shutdown / Poweroff
Parameter2 = Win 95/98/ME / Win NT/2000/XP
Open Documents
Parameter1 = Document path or name etc..
MSAccess
Parameter1 = Full path to MSAccess.exe
Parameter2 = Database path
Parameter3 = Option (0 = none/autoexec macro , 1 = run macro, 2 = compact database, 3 = repair database)
Parameter4 = Macro name
Parameter5 = terminate time in minutes
MSExcel
Parameter1 = Full path to Excel.exe
Parameter2 = Workbook path
Parameter3 = Option 0/1  (0 = none/autoexec macro , 1 = run macro)
Parameter4 = Macro name
MSWord
Parameter1 = Full path to winword.exe
Parameter2 = Database path
Parameter3 = Option 0/1  (0 = none/autoexec macro , 1 = run macro)
Parameter4 = Macro name
Parameter5 = terminate time in minutes
MSPowerpoint
Parameter1 = Full path to ppowerpnt.exe
Parameter2 = Presentation path
Parameter3 = Option 0/1/2  (0 = none/autoexec macro , 1 = print presentation, 2 = run slideshow)
Parameter4 = terminate time in minutes
```

Back to Index

Programmatically controlling the scheduler

There are various methods available to control the Automize scheduler engine from your programs

1) **Need to run Automize tasks only.** Use the TaskRunner feature from the Automize user interface to generate the commandline code. Then use this code in your batch files, unix scripts, java exec function, c/c++ exec/spawn functions etc... to run the tasks. The task will return an exit code that can be used in your programs for conditional processing. Please see the help file on Exit Codes for details.

2) **Full control of engine via commandline module:** Adding, modifying, suspending the tasks/schedules, start/stop the engine, or run tasks, and many more features are available via the commandline module. You would have to format and send the desired commands via the commandline module using your program. Please use the commandline module manually to see how it works, before trying to program through it. *This is the recommended method of programatically controlling the scheduler engine*

3) **Manually parse the data files and send socket commands to the engine server:** Here you would manually write the task data or schedule data to the required data files, and then send a socket command to the engine server to load the data or run tasks etc.. *This is the more difficult approach and is not recommended. Method 2, does this automatically for you.*
The task data is in the file ..Automize/inputs/taskData.jsd
The schedule data is in the file ..Automize/inputs/scheduleData.jsd

The engine server understands the following socket commands:
ConfirmRunning
ShutdownEngine
RunTask^Tasktitle
ReloadTaskData
ReloadScheduleData
ReloadRegSettings
ReloadUserSettings

The following is an example to send socket command to the server using java. The command is a simple string denoted by msg, where msg = "ReloadTaskData" etc...

```
public static boolean sendEngineServerMsg(String msg)
{
      try
     {
         Socket s = new Socket("localhost",enginePort);  //enginePort = 1965 for version 5.x
         PrintStream ps = new PrintStream(s.getOutputStream());
         ps.println(msg);
         ps.flush();
         s.close();
         return true;
     }
     catch(Exception e)
     {
         String err = "Scheduler engine is either shut down or is not responding";
          // Output (err);
         return false;
     }
```

```
    }
```

[Back to Index](#)

Programmatically controlling the scheduler

There are 2 ways to control the Automize scheduler engine from your programs

1) **Need to run Automize tasks only.** Use the TaskRunner feature from the Automize user interface to generate the commandline code. Then use this code in your batch files, unix scripts, java exec function, c/c++ exec/spawn functions etc... to run the tasks. The task will return an exit code that can be used in your programs for conditional processing. Please see the help file on Exit Codes for details.

2) **Full control of engine via commandline module:** Adding, modifying, suspending the tasks/schedules, start/stop the engine, or run tasks, and many more features are available via the commandline module. You would have to format and send the desired commands via the commandline module using your program. Please use the commandline module manually to see how it works, before trying to program through it.

Back to Index

Remote Install

Some users need to install the software onto remote computers, which may be miles away. This is also useful for IT administrators who need to install on multiple systems quickly and monitor and control multiple systems. Others may need the commandline module because their system does not have display capabilities or only serve as servers.

If you need to install remotely, you can simply copy over the entire main install folder from your Local system to the Remote system using Ftp (or simple Copy if the remote folder is accesable as a shared drive to you). Then access the CLM using Telnet (or rlogin, ssh etc..) to start/stop and perform other scheduler operations. NOTE: when you install remotely, the normal system shortcuts are not created on the remote systems desktop or other folder since the installation software is not used.

The program does not require any system wide properties to be set. On Windows, nothing is read/written to the registry or system folders/files etc... On Unix systems, the system classpath or path is not set or accessed. Hence, only a simple copy or Ftp of the entire main install folder to the Remote System is required.

Steps for Remote Install:
1) Install a master copy of the software on the same platform as the target remote systems. i.e. if you want to perform remote installs on Windows systems, Install a master copy on a Windows system.
2) Ftp the entire main install folder to the target remote systems.

Back to Index

TaskTypes

The taskType field is case sensitive as well as space sensitive.
For Version 7.x, the following are correct taskTypes entries:
usage in add or addx command:

```
....Task.TASK_TYPE=TaskTypes.CHAIN....
```

```
CHAIN  = TaskTypes.CHAIN
COPY  = TaskTypes.COPY
SYNCHRONIZE  = TaskTypes.SYNCHRONIZE
ZIP  = TaskTypes.ZIP
UNZIP  = TaskTypes.UNZIP
ENCRYPT  = TaskTypes.ENCRYPT
DECRYPT  = TaskTypes.DECRYPT
TEXT_SEARCH  = TaskTypes.TEXT_SEARCH
PING  = TaskTypes.PING
TELNET  = TaskTypes.TELNET
TELNET_ADVANCED  = TaskTypes.TELNET_ADVANCED
PRINT  = TaskTypes.PRINT
DELETE_FILE  = TaskTypes.DELETE_FILE
COMMAND  = TaskTypes.COMMAND
MACLAUNCH  = TaskTypes.MACLAUNCH
WINCOMMAND  = TaskTypes.WINCOMMAND
OPEN_DOCUMENTS  = TaskTypes.OPEN_DOCUMENTS
MACRO  = TaskTypes.MACRO
SHUTDOWN  = TaskTypes.SHUTDOWN
ACCESS  = TaskTypes.ACCESS
EXCEL  = TaskTypes.EXCEL
POWERPOINT  = TaskTypes.POWERPOINT
WORD  = TaskTypes.WORD
FTP  = TaskTypes.FTP
FTP_COMMAND  = TaskTypes.FTP_COMMAND
FTP_SYNCHRONIZE  = TaskTypes.FTP_SYNCHRONIZE
FTP_MONITOR  = TaskTypes.FTP_MONITOR
EMAIL  = TaskTypes.EMAIL
EMAIL_CHECK  = TaskTypes.EMAIL_CHECK
EMAIL_BULK  = TaskTypes.EMAIL_BULK
WEB  = TaskTypes.WEB
URL_MONITOR  = TaskTypes.URL_MONITOR
ALARM  = TaskTypes.ALARM
MESSAGE  = TaskTypes.MESSAGE
ECHO  = TaskTypes.ECHO
DELAY  = TaskTypes.DELAY
KEY_STROKES  = TaskTypes.KEY_STROKES
WAKE_UP  = TaskTypes.WAKE_UP
DIRECTORY_CHANGE  = TaskTypes.DIRECTORY_CHANGE
DIRECTORY_MONITOR  = TaskTypes.DIRECTORY_MONITOR
FILE_MONITOR  = TaskTypes.FILE_MONITOR
LOG_MONITOR  = TaskTypes.LOG_MONITOR
DIRECTORY_LOOP  = TaskTypes.DIRECTORY_LOOP
VARIABLES  = TaskTypes.VARIABLES
```

```
VARIABLES_OUTPUT  = TaskTypes.VARIABLES_OUTPUT
DATABASE_SELECT  = TaskTypes.DATABASE_SELECT
SSH2  = TaskTypes.SSH2
SFTP  = TaskTypes.SFTP
SFTP_MONITOR  = TaskTypes.SFTP_MONITOR
SFTP_SYNCHRONIZE  = TaskTypes.SFTP_SYNCHRONIZE
RESTART  = TaskTypes.RESTART
AUTO_BACKUP  = TaskTypes.AUTO_BACKUP
ARCHIVE_LOGS  = TaskTypes.ARCHIVE_LOGS
EMAIL_LOGS  = TaskTypes.EMAIL_LOGS
```

[Back to Index](#)

Using the CommandLine Module

You can control the scheduler engine using the Commandline Module (CLM) via Telnet (or rlogin, ssh etc..).

Windows:
1. Open a DOS command window.
2. Telnet to the remote system where the program is installed
3. 'cd' to the main install folder (Example: cd "c:\program files\hiteksoftware\automize" or cd "c:\program files\Automize")
4. The main install folder includes the following file: clm.bat
5. Use following command to verify that clm is setup correctly: clm ? . If clm is setup correctly, you will get a listing of all commands.

NOTE: You may get the following error: "The system cannot find the path specified". This means that you do not have the Java machine installed into the main install folder. You probably used the zip installer and are using a 'java' machine which is in your system path. In this case you will have to edit 'clm.bat' and replace the commandline:
jre\bin\java -Dpass="%pass%" -cp .;hitek10.jar com.hitek.gui.CLM %*
with:
java -Dpass="%pass%" -cp .;hitek10.jar com.hitek.gui.CLM %*

Unix & MacOSX:
1. Open a Terminal window.
2. Telnet to the remote system where the program is installed
3. 'cd' to the main install folder
4. The main install folder includes the following file: clm
5. Run the command: chmod +x clm
6. Use following command to verify that clm is setup correctly: ./clm ? . If clm is setup correctly, you will get a listing of all commands.
7. You can add the appropriate shell header line to the file 'clm' to avoid having to type in './' everytime you run a command

NOTE: You may get a "command not found" error. This means that you do not have 'java' in your system path. You probably used the Install including 'JVM' and have the Java machine (jre folder) installed into the main install folder. In this case, you will have to edit 'clm' and replace the commandline.
java -Dpass="%pass%" -cp .:hitek10.jar com.hitek.gui.CLM $*
with:
jre/bin/java -Dpass="%pass%" -cp .:hitek10.jar com.hitek.gui.CLM $*