Scheduler

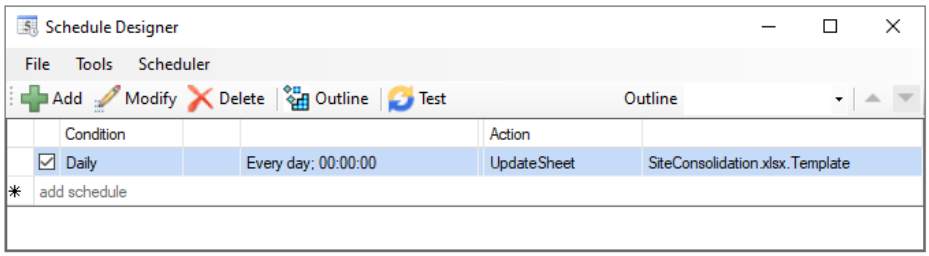
Overview

The **Scheduler** automatically processes actions on time or on event triggers (by monitoring value changes) in real-time **Connectors**. Actions range from producing reports to maintaining files on the hard drive (see **Actions**).

When multiple actions are triggered simultaneously, the **Scheduler** processes the actions in the order they are listed in the **Designer**. Since the execution of multiple schedule lines at the same time can be handled, users are urged to not stagger schedule lines that need to be performed at the same time because this is less efficient.

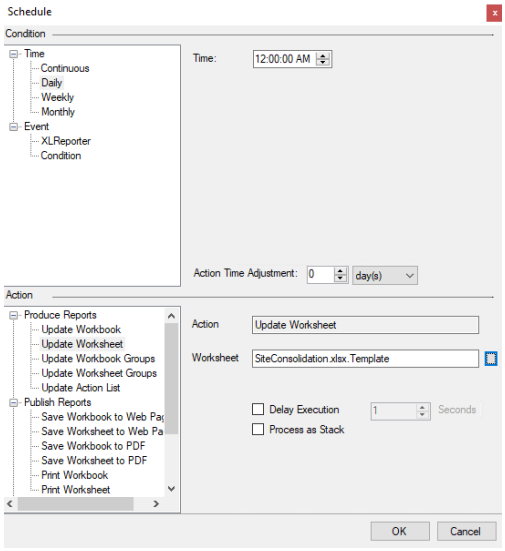
Designer

The **Schedule Designer** is opened from the **Project Explorer** or the **Template** ribbon in the **Design Studio**. From the **Project Explorer**, select the **Project** tab and choose **Schedule, Designer**.



To save any changes click **File**, **Save**. If the scheduler is running, it will automatically reload the changes.

To create a new schedule click **Add** to open the **Schedule** dialog as follows:



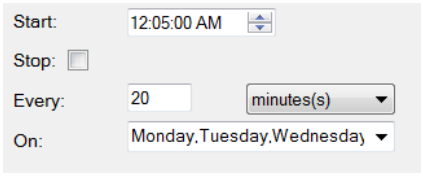
In the upper part of the display select a **Condition** (based on **Time** or **Event**) together with an **Action Time Adjustment**

Conditions

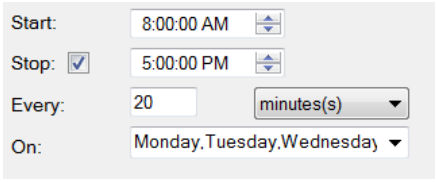
On Time

Time conditions are determined from the local operating system clock. Select one of **Continuous**, **Daily**, **Weekly** or **Monthly** and the **Time**.

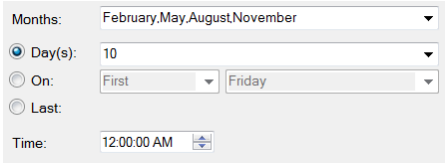
For the **Continuous** condition, the **Start** time is used as an "anchor" time.
For example: For the condition to occur 5, 25 and 45 minutes past every hour on a weekday:



However, if the **Stop** time is specified then **Start** and **Stop** times limit the execution of the action to a specific period during the day.
For example: For the condition to occur every 20 minutes between 8:00am and 5:00pm on a weekday:

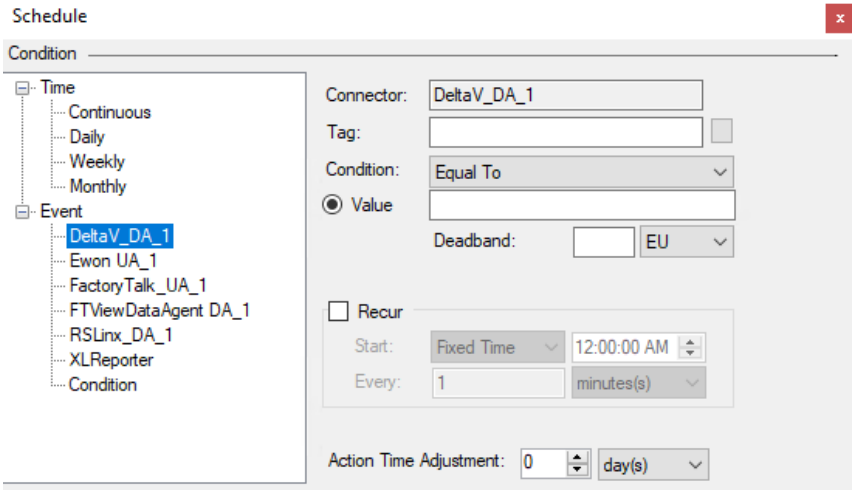


For the **Monthly** condition, the months of the year and the specific days of the month can be specified.
For example: For the condition to occur every 3 months starting on the 10th of February:

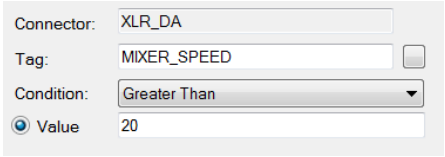


On Event

Event conditions are determined from values in real time **Data Connectors**, **XLReporter Memory Variables,** or on the **Start** or **Stop Condition** of the **Scheduler**. The event is triggered once when the scheduler detects the tag value transition into the event condition and can **Recur** if that option is enabled.



For example: For the condition to occur each time the Mixer Speed exceeds 20:



XLReporter Memory Variables

There are 2 types of Memory variables available: **System** and **User**.

System

System variables provide information about the running system which can be monitored and when triggered execute specific actions. For example, the *Error Message* can be monitored so if it changes an email can be sent to alert the operator than an error has occurred in the reporting system.

The system variable **Current Action Duration** has special meaning when it is used in a schedule event condition. When an event is triggered using this variable the **Action** associated with the trigger is performed and the current action is stopped.

For example, when **Current Action Duration** is *Greater Than* and a **Value** of *60* send an email. If any Action in the Scheduler takes more than 60 seconds, an email is sent, and the current action is stopped.

User

User variables are defined in the **Variable Editor** and are used to hold numbers or text. These values are stored in system memory and are persistent as long as the system is running. These can be very useful for triggering schedule actions from other applications.

Condition

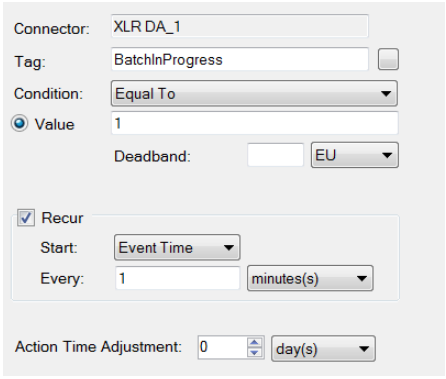
The following condition options are available for Events from real time **Data Connectors** as well as **XLReporter Memory Variables**.

- **Numeric**

  The numeric conditions are *Equal, To, Not Equal To, Equal or Greater Than, Greater Than, Equal or Less Than, Less Than* and *On Change*.

  In the case of *Equal* and *On Change* a **Deadband** can also be specified.

  For example: For the condition to occur when the Mixer Speed changes at least 5 EU (engineeering units):



- **Text**

  The text conditions are *IS, IS NOT, START WITH, END WITH, CONTAIN* and *ON CHANGE*. Text conditions are case sensitive.

Recur

When the **Condition** is satisfied, and **Recur** is unchecked, the **Action** will be performed once.

When the **Condition** is satisfied, and **Recur** is checked, the **Action** will be performed periodically according to the time settings. In this case, the start time of the recur is specified as either **Event Time** or **Fixed Time**.

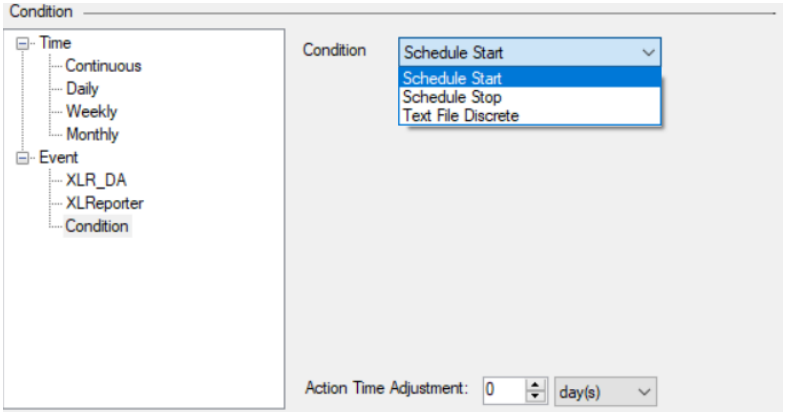
- **Event Time**
  The recur starts at the event time.
- **Fixed Time**
  The recur start at the specified time.

Event Condition

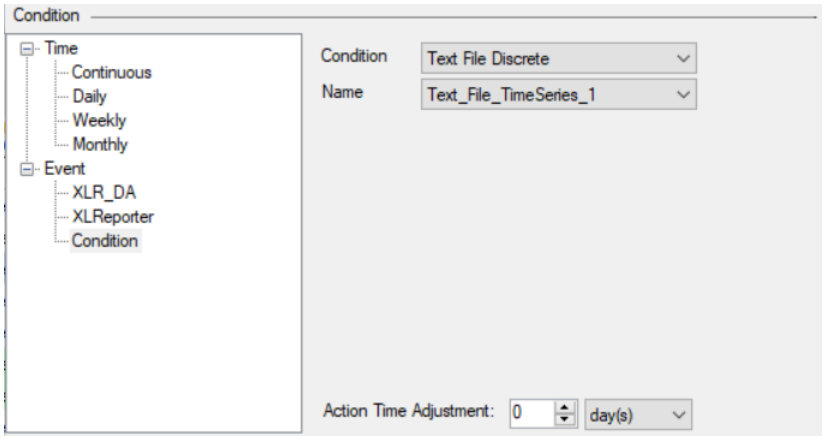
Event Conditions are application or file system occurrences that can be used to trigger actions.

Schedule Start/Stop

These options trigger actions when the scheduler itself starts or stops.



Text File Discrete



The **Text File Discrete** option provides a list of all the **Text File Discrete** and **Text File Discrete (time series)** connectors configured in the project.

If the **Source File** in the connection is set to *Variable*, any time a new file appears in the **Folder** specified that satisfies the **Filter** specified, the condition is triggered.

If the **Source File** in the connection is set to *Fixed*, any time **File** specified in the **Folder** changes (e.g., is modified), the condition is triggered.

Action Time Adjustment

The **Action Time Adjustment** adjusts the schedule time and the **Action** is then performed at this "modified" time rather than the schedule time.

This is most useful when generating reports from historical data. For example, to schedule a daily report from historical data, the report cannot be generated until the next day when all the data is available from the historian. To produce a report for the previous day set **Condition** to **Time**, **Daily**, **12:15:00AM** and the **Action Time Adjustment** to *1 day(s)*.

Actions

An **Action** is performed when the **Condition** is met. These actions can also be initiated as commands from third party applications and programming languages.

An **Action** consists of two parameters and options. For example, the action **Email by Workbook** requires a **Workbook**, **Configuration** name and the optional **Wait to Complete** flag.

Options

For the option **Delay Execution**, the scheduler waits for anywhere from 1 to 3600 seconds before placing the action into the queue. Note that the *Report DateTime* of the action will still be the initial time before the delay.

For the option **Wait to Complete**, the scheduler waits for a maximum of 30 seconds for the **Action** to complete.

Produce Reports

These actions produce reports. The workbook and worksheet specified is the template name. The name of the report is specified in the **Report Names** setting of the template (see the **Report Names** section of the DESIGN, **Template Studio** document).

Update Workbook

This action updates all of the worksheets in a workbook. If the target workbook does not exist, it is first created and then updated. Note that only data connections that have a defined **Scope** will be considered in the update.

Settings

Workbook	Workbook name

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

UpdateBook 'Workbook'

Update Worksheet

This action updates a single worksheet.

Settings

Worksheet	Worksheet name

Process as Stack

When checked, the updated worksheet will be treated as a **Stack Report Template**. This should be checked automatically when a template is created using the **Stack Report Wizard** and should not be used for standard report template updates.

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

UpdateSheet 'Workbook.Worksheet'

Update Workbook Groups

This action updates all of the worksheets in a workbook that are associated with the group number or list of group numbers specified. Note that irrespective of the connection group number(s) in the action, connections belonging to group zero are included for this action.

Only connections that have a defined **Scope** are updated with this action.

Settings

Workbook	Workbook name
Group	Comma separated connection group number(s)

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

UpdateGroupBook 'Workbook' 'Group'

Update Worksheet Groups

This action updates a single worksheet that are associated with the group number or list of group numbers specified. Note that irrespective of the connection, group number(s) in the action, connections belonging to group zero are included for this action.

Settings

Workbook.Worksheet	Worksheet name
Group	Comma separated group number(s).

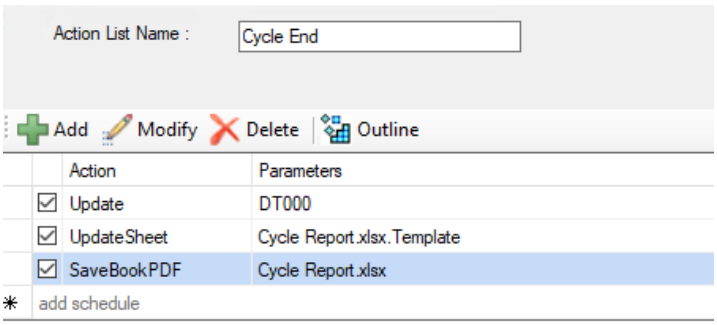
Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

UpdateGroupSheet 'Workbook.Worksheet' 'Group'

Update Action List

This action is provided to configure multiple actions to run on the same **Condition**. The browse pushbutton opens the **Action List Editor** where the list of actions can be configured.



At the top is a list of existing **Action Lists**. To create a new **Action List**, select *add new action list* from the drop down and then specify the **Name**.

Settings

Action List Name	Action list configuration name

Syntax

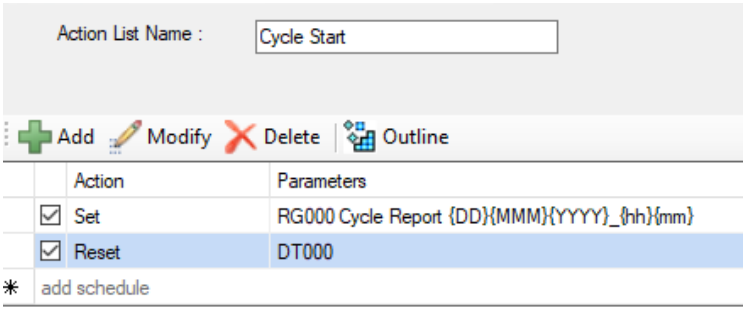
To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

UpdateActionList 'Action List Name'

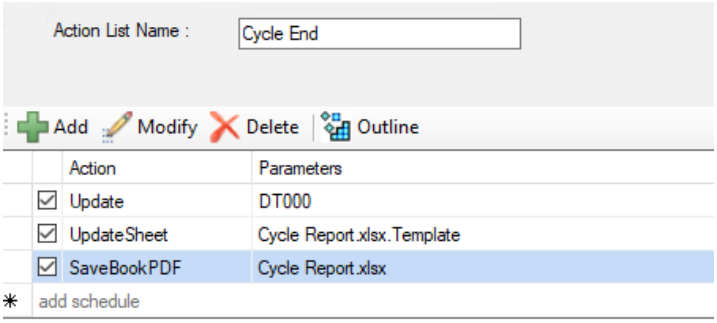
Usage

A practical usage for this action is when generating a cycle report where at the beginning of the cycle some information needs to be captured for the naming convention and the start of the time period for historical data and at the end of the cycle setting the end of the time period, generating the report and publishing to PDF.

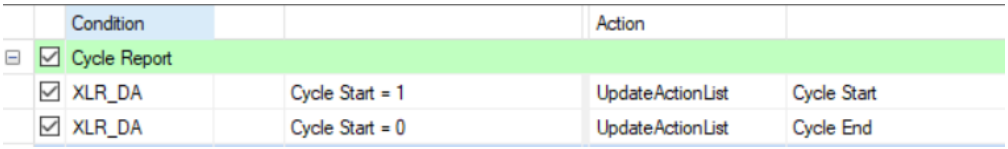
This can reduce the schedule down to 2 lines: an Update Action List when the cycle starts and when it ends. The *Cycle Start* **Action List** is configured as:



The *Cycle End* **Action List** is configured as:



The resulting schedule is:



Publish Reports

These actions publish reports to different file formats and printers. The workbook and worksheet specified is the template name.

The **Target** setting is an optional setting for each of these actions.

- **Target** setting is blank

  In this case, the **Report Names** settings of the template are used.

  For the *Save Workbook* actions, the **WORKBOOK Report Name** is used including any subfolders specified.

  For the *Save Worksheet* actions, the **WORKBOOK Report Name** is used including any subfolders specified and combined with the **Name** set for the worksheet specified in the format *Workbook_Worksheet*. If there is no **Name** set for the worksheet, the worksheet itself is used.

- **Target** setting is a folder

  To specify a folder, the trailing character for **Target** must be the folder separator (\), e.g., *C:\My PDF Reports*. This can contain hard coded text as well as variables and Name Types as long as they are valid characters for a Windows folder name.

  The name of the of the published report then follows the **Report Names** settings just like if the **Target** is blank.

- **Target** setting is a full file name

  In this case, the entire folder structure and file name is specified. This can contain hard coded text as well as variables and Name Types as long as they are valid characters for a Windows folder and file name. The file extension should not be specified.

Save Workbook to Web Page(s)

This action saves all visible worksheets of a workbook to web pages.

Settings

Workbook	Workbook name
Target	The web page name

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

SaveBookHTML 'Workbook' ['Target']

Save Worksheet to Web Page

This action saves a worksheet to a web page.

Settings

Workbook.Worksheet	Worksheet name
Target	The web page name

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

SaveSheetHTML 'Workbook.Worksheet' ['Target']

Save Workbook to PDF

This action saves all visible worksheets of a workbook to a PDF file according to the PDF encryption settings (see the **PDF Reports** section of the SETUP, **Customize a Project** document).

Settings

Workbook	Workbook name
Target	The PDF name

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

SaveBookPDF 'Workbook' ['Target']

Save Worksheet to PDF

This action saves a worksheet to a PDF file according to the PDF encryption settings (see the **PDF Reports** section of the **Customize a Project** document).

Settings

Workbook.Worksheet	Worksheet name
Target	The PDF name

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

SaveSheetPDF 'Workbook.Worksheet' ['Target']

Print Workbook

This action prints of its visible worksheets of the workbook to a specific printer. To print to the default printer of the system, set the **Printer** name to *Default*.

Settings

Workbook	Workbook name
Printer	Printer name

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

PrintBook 'Workbook' 'Printer'

Print Worksheet

This action prints a worksheet to a specific printer.

Settings

Workbook.Worksheet	Worksheet name
Printer	Printer name

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

PrintSheet 'Workbook.Worksheet' 'Printer'

Save Workbook to VantagePoint

This action saves the workbook to a web page then adds a link to the VantagePoint portal to view the workbook. The web page is saved under the **Web** folder of the project in a folder matching the **Report Node** setting of the VantagePoint connector.

Note this action is only shown if a VantagePoint connector is defined in the project.

Settings

Workbook	Workbook name
Target	The web page name

Update before Publishing

If checked, an **Update Workbook** action is executed before the workbook is saved as a web page.

Save Worksheet to VantagePoint

This action saves the worksheet to a web page then adds a link to the VantagePoint portal to view the worksheet. The Web Page is saved under the **Web** folder of the project in a folder matching the **Report Node** setting of the VantagePoint connector.

Note this action is only shown if a VantagePoint connector is defined in the project.

Settings

Workbook.Worksheet	Worksheet name

Target	The web page name

Update before Publishing

If checked, an **Update Worksheet** action is executed before the worksheet is saved as a web page.

Transfer Reports

These actions transfer files over the network through E-mail and FTP transfer. The actions apply to a single workbook or to content of a configuration.

FTP a Workbook

This action transfers a workbook using FTP using the FTP server settings specified in a configuration file (see **FTP Cofiguration** below).

Settings

Workbook	Workbook name
Configuration	Configuration file name

Only FTP configurations that are set up to *Upload* are available to select. In addition, if an existing configuration is selected that has **Source** settings configured, those settings are replaced with the **Workbook** specified.

Wait to Complete

If checked, the scheduler will wait for the action to complete before processing any other action.

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

FTPBook 'Workbook' 'Configuration'[-w]

FTP by a Configuration

This action using FTP according to the settings in a configuration file (see **FTP Configuration** below).

Settings

Configuration	Configuration file name (s)

Wait to Complete

If checked, the scheduler will wait for the action to complete before processing any other action.

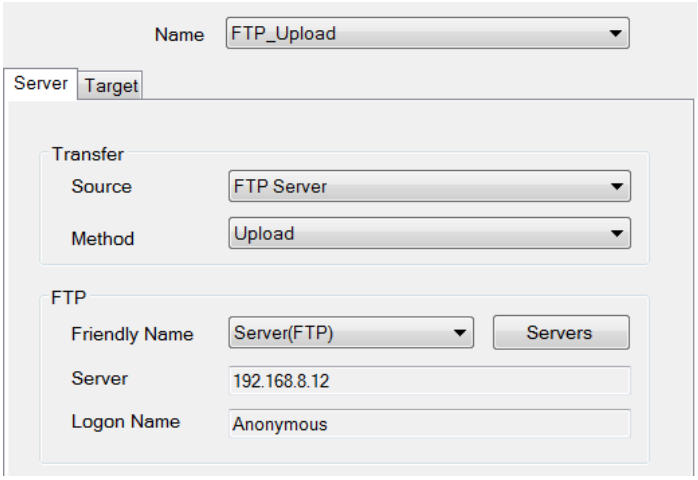
Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

FTPTransfer 'Configuration'[-w]

FTP Configuration

When the browse pushbutton […] is selected for **Configuration**, the FTP Configuration dialog is opened.



If there are any existing configurations, they are listed at the top in a drop-down list. To choose an existing configuration, select it from the list. To create a new configuration, select the *new* option. The drop-down list is then replaced by a text box where the **Name** of the configuration can be specified.

If there are no FTP configurations in the project, the **Name** text box is shown immediately so that a new configuration can be created.

For details on setting up an FTP configuration, see the DISTRIBUTE, **Transfer Reports to an FTP Server** document.

Please note that for the **FTP a Workbook** action the **Transfer Method** can only be *Upload* and the **Source** tab is not available since the source is the **Workbook** specified for the action.

Email a Workbook

This action emails a workbook as an attachment using the SMTP settings specified in a configuration (see **Email Configuration** below).

Settings

Workbook	Workbook name
Configuration	Configuration file name

Only Email configurations that are set up for *Email* or *Text Message (MMS)* are available to select. In addition, if an existing configuration is selected that has **Attachment** settings configured, those settings are replaced with the **Workbook** specified.

Wait to Complete

If checked, the scheduler will wait for the action to complete before processing any other action.

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

EmailBook 'Workbook' 'Configuration'[-w]

Email by a Configuration

This action emails according to the setting in a configuration (see **Email Configuration** below.

Settings

Configuration	Configuration file name (s)
Value	Custom Variable value(s)

If the configurations contain custom variables, then their values can be specified in the Value setting in the form *Variable1=Value1, Variable2=Value2*.

Wait to Complete

If checked, the scheduler will wait for the action to complete before processing any other action.

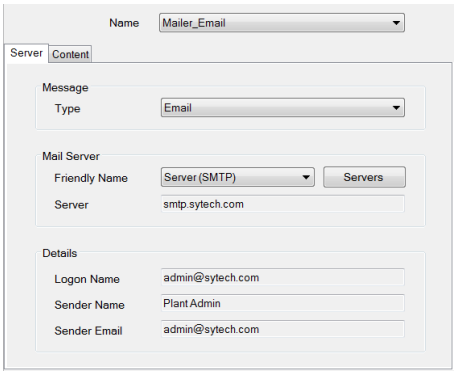
Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

EmailTransfer 'Configuration' ['Value'][-w]

Email Configuration

When the browse pushbutton […] is selected for **Configuration**, the Email Configuration dialog is opened.



If there are any existing configurations, they are listed at the top in a drop-down list. To choose an existing configuration, select it from the list. To create a new configuration, select the *new* option. The drop-down list is then replaced by a text box where the **Name** of the configuration can be specified.

If there are no Email configurations in the project, the **Name** text box is shown immediately so that a new configuration can be created.

For details on setting up an Email configuration, see the DISTRIBUTE. **Send Reports by Email and Text Messaging** document.

Please note that for the **Email a Workbook** action the **Message Type** can only be *Email* or Text *Message (MMS)* and the **Attachments** tab is not available since the attachment is the **Workbook** specified for the action.

Manage Files and Folders

These actions are used to assist in maintaining files. The actions apply to a single workbook or to content of a configuration.

Move a Workbook

This action moves a copy of an existing workbook. If the **Target** contains a folder that does not exist, it is created.

Settings

Workbook	Source Workbook name
Target	Target Workbook name

Overwrite Target

If checked, the target workbook will be overwritten (if it exists).

Delete Source

If checked, the source workbook will be deleted after it has been moved.

Note, this action is designed to move or copy a workbook file based on the source **Workbook** specified as the name of a template in the project.

To use this action for files other than those produced from a template, the source **Workbook** setting must be the full path to the file including file extension. The **Target** setting must also be a full path complete with a file name, but no file extension should be specified.

To move or copy multiple files with a single action, see **Manage by a Configuration**.

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

MoveBook 'Workbook' 'Target'[-od]

Delete Workbook

This action deletes an existing workbook.

Settings

Workbook	Workbook name

Note this action is designed to delete a workbook file based on the source **Workbook** specified as the name of a template in the project

To use this action for files other than those produced from a template, the source **Workbook** setting must be the full path to the file including file extension.

To delete multiple files with a single action, see **Manage by a Configuration**.

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

DeleteBook 'Workbook'

Manage by a Configuration

This action performs file management according to the setting in a configuration (see the DISTRIBUTE, **Manage the File System** document).

Settings

Configuration	Configuration file name

Wait to Complete

If checked, the scheduler will wait for the action to complete before processing any other action.

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

ManageFiles 'Configuration'[-w]

Manage Variables

These actions are used to modify values of **Variables**.

Set a Value to a Variable

This action assigns value(s) to variables This can be used for **Function**, **Memory** and **Custom** variables. For more information, see the DESIGN, **Variables** document.

Up to 4 variables can be assigned at once by separating the Variable and Value by tilde (~).

Settings

Variable	Variable name
Value	Value (numeric, text or variable)

Example: To assign RG000, RG011 and RG022 to the values 1, 2 and 3 set *Variable* to *RG000~RG011~RG022* and *Value* to *1~2~3.*

Alternatively, an array of variables can be set at once to a single value by specifying the first variable-the last two digits of the last variable in the array

Example: To assign RG000, RG001, RG002, RG003, RG004, and RG005 to the value of 0, set *Variable* to *RG000-05* and set *Value* to *0.*

Example: To assign RG000 with the BatchID tag of a connector called Process, set *Variable* to *RG000* and *Value* to *{<Process>BatchID}*. Note the syntax {**<connector name>item name**}

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

Set 'Variable' 'Value'

Update a Variable

This action updates the function variables **Counter** and **DateTime**.

Settings

Variable	Variable name

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

Update 'Variable'

Reset a Variable

This action updates the function variables **Counter**, **DateTime** and **Lookup**.

Settings

Variable	Variable name

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

Reset 'Variable'

Run Applications

These actions run various scripts and third-party applications.

Run a Schedule Script

This action runs a schedule script and is used when additional logic is required in the schedule (see **Scheduler Script** below).

Settings

Configuration	Script configuration name

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

RunScript 'Configuration'

Run an Application

This action runs an external executable.

Settings

Application	Name of the application to run (may require a full path).
Parameter	Command line parameters

Wait to Complete

If checked, the scheduler will wait for the action to complete before processing any other action.

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

RunApplication 'Application' ['Parameter'][-w]

Write Value to Process

This action writes a specified value to real time connector. This action can be used as a "heart beat" or a handshake.

Settings

Tag	Tag used for the write
Value	Value to write

Wait to Complete

If checked, the scheduler will wait for the action to complete before processing any other action.

Write as Text

If checked, the value will be written as text.

Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

WriteValue 'Tag' 'Value'[-s]

Run a Workbook Macro

This action runs a macro defined in a workbook.

Note: This action requires Microsoft Excel to be installed and the macro name must be 31 characters or less.

Settings

Macro	Macro name in the form **Workbook.Macro**
Parameter	Up to 3 comma separated parameters

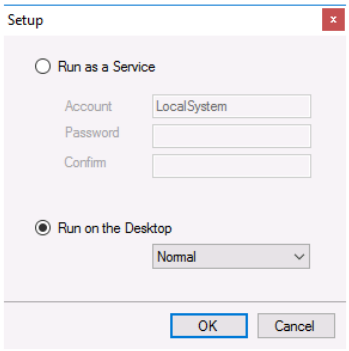
Syntax

To use in a **Scheduler Script**, command line or third-party applications, use the syntax:

RunBookMacro 'Workbook.Macro' ['Parameter']

Scheduler Setup

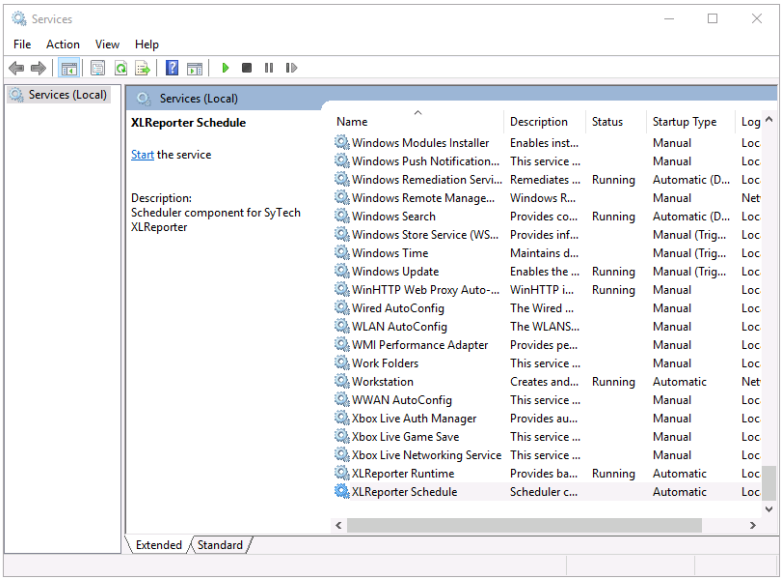
The scheduler can be setup to run on the desktop, in the background or as a windows service. From then menu option, select **Scheduler**, **Setup**.



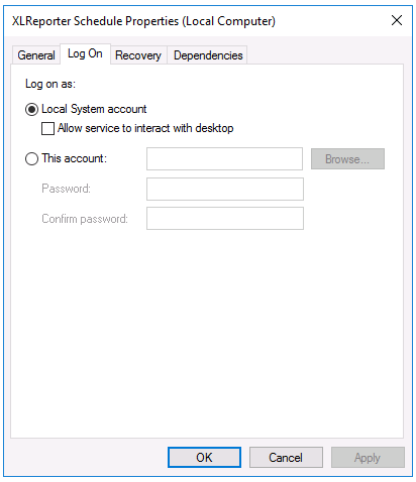
Run as a Service

Select this to run the **Scheduler** as a Windows Service. As a service, the **Scheduler** is automatically started by the operating system using the **Account** specified. By default, the **Account** is *LocalSystem* which would be suitable for most cases. However, if the action requires specific rights e.g., running a book macro or moving the workbook to a protected file server, then use an **Account** that has the appropriate rights.

The **Scheduler** service can also be set from **Services** inside the Windows **Control Panel, Administrative Tools**.



Navigate to the **XLReporter Schedule** service and right-click, select **Properties**.



Under the **Log On** tab, select **This Account**. Use the **Browse** button to select the account name. Once selected, enter the **Password** and confirm it. Click **Apply** at the bottom to retain this change. When the service is started it will run under the new account.

A specific **Account** should be an *Administrator* and have a **Password**.

Run on the Desktop

Select this option to run the **Scheduler** on the desktop. The startup can be *Normal*, *Minimized* on the taskbar or *Background* with no icon on the taskbar.

A benefit of running **Normal** on the desktop is that the progress of the schedule is visible providing valuable insight during development.

Use this option only in special cases when the **Scheduler** cannot run as a service due to the limitations of the **Connectors**.

Automatic Startup

When running on the desktop, to start the **Scheduler** automatically, add a shortcut for the **Scheduler** to the *Windows Startup* folder. The easiest way to do so is to drag the **Scheduler Engine** item in the **XLReporter** program group (*Start Menu, Programs, XLReporter*) to the *Windows Startup* folder.

Stopping the Scheduler

When the scheduler is running, it can be stopped from the **Scheduler** menu option or from the **Project** tab of the **Project Explorer**.

Diagnostics

If the **Scheduler** is set to run on the desktop visible, the font color of the schedule line indicates the current state of the action.

Color	State
Grey	The schedule line is enabled but not currently executed.
Red	The schedule line is not enabled.
Blue	The schedule line is currently being executed.
Brown	There is an error reading the event trigger tag for the schedule line.

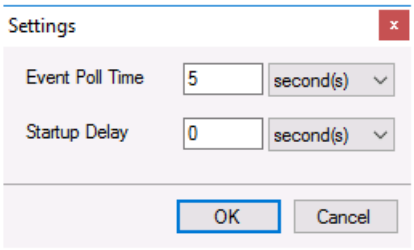
The leftmost column in the **Scheduler** indicates the last time the schedule line was triggered to execute.

Event Trigger Tag

If there is an issue either with either a connector or trigger tag in the **Scheduler**, the **Status Log** is updated to indicate an error has occurred. In addition, once the error is resolved and data can be read again, the **Status Log** is again updated to indicate the tag is back in a good state.

Scheduler Settings

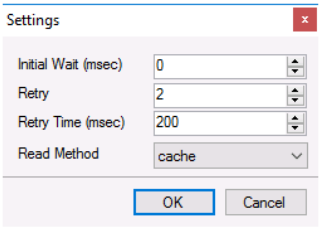
From then menu option, select **Scheduler**, **Settings**.



Event Poll Time

The **Event Poll Time** is the rate at which event schedules are scanned to determine if the trigger condition is satisfied.

When configuring this setting be conscious of the **Initial Wait** setting configured for any real time connector being used for event triggering. **Initial Wait** setting is the amount of time (in milliseconds) the **Connector** waits after making a request and before retrieving data. This can be useful for communications that do not respond immediately.



For example, if the **Initial Wait** is set to *1000* milliseconds the minimum **Event Poll Time** that should be set is *2* seconds because there is a 1 second overhead whenever data is read from the connector.

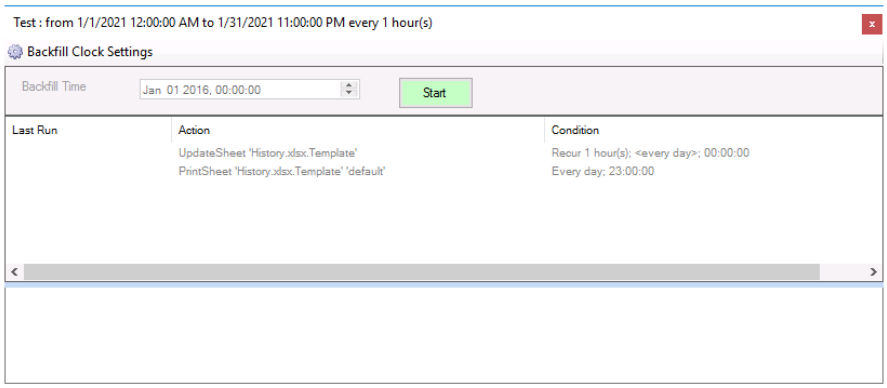
Startup Delay

When the **Scheduler** is started, it immediately connects to any connectors that have been configured. This behavior may cause problems for certain connectors in which case the **Startup Delay** is used to make the **Scheduler** wait before connecting.

Backfilling Reports

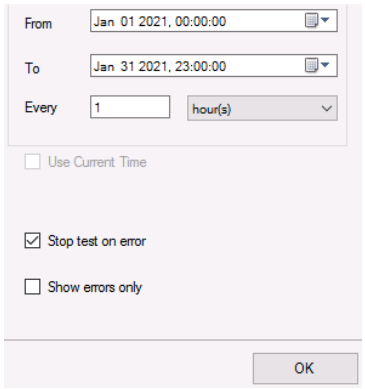
Scheduler **Backfill** provides a powerful **Virtual** clock to rerun a set of schedules over a period of time. The feature can be used to create reports from data collected prior to the purchase of **XLReporter** or those that were missed due to unexpected problems.

Select the schedule lines by holding down the *Ctrl* key and clicking on a row(s). Note that any schedules that use **Event** based triggers are not considered. Click **Tools**, **Report Backfill**.



The **Virtual** clock setting of the start/end times and interval are show in the caption of the display.

Click the **Backfill Clock Settings** to provide a new start/end and interval for the **Virtual** clock and click **OK**.



From the main display, click **Start**. Using the **From** time, the **Virtual** clock increments until it reaches **To** time with the actions being processed at each increment. The processing is shown by the schedule lines changing color and with the results of the processing is shown in the results panel.

Enabling Schedules

By default, a schedule is enabled when it is **Add**ed to the schedule list, this is indicated by a check mark.

By unchecking, the schedule line will not be considered when the **Scheduler** is started. If the scheduler is running on the desktop, a disabled line will appear with red font.

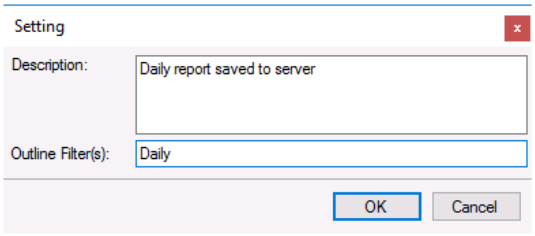
Toolbar Items

The toolbar items are used to **Add, Modify,** and **Delete** schedule lines, insert a schedule **Outline** or **Test** a schedule line.

Outline

An **Outline** is a way of commenting and grouping schedule lines. With outlines, groups of schedule lines can be accessed using a filter configured for the **Outline**.

To add an **Outline**, highlight the row where it is to be placed and click **Outline**.

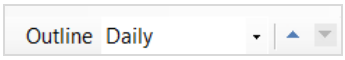


Enter a **Description** and a list of comma-separated **Filter**(s).

From the schedule list notice the outline inserted and that all the schedule line below it can be displayed/hidden by clicking the leftmost +/- buttons.

	Condition			Action	
⊟	Daily report saved to server				
	Continuous		Recur 1 hour(s); <every day>; 08:00:00-17:00:00	UpdateSheet	SiteColidation.xlsx.Template
	Daily		Every day; 00:00:00	SaveBookPDF	SiteColidation.xlsx
	Daily		Every day; 00:00:00	MoveBook	SiteColidation.xlsx s:\Reports
⊞	Weekly reports emailed to management				

Alternatively, the outlines can be displayed/hidden by selecting a **Filter** that was specified when the outline was created.



Test

A schedule line can be "simulated" to run on a specific date/time range. To perform a **Test**, highlight the row and click **Test**. For time-based actions, the test is performed between the date range specified otherwise it is performed at the current date/time.

Scheduler Script

A script file can be created and modified by selecting **Tools**, **Script Editor**. All the configured scripts (file extension .scs) are listed. For an example of the script syntax, see the *XLRprocess* script in the project. Note that a line starting with a semi-colon (;) is treated as a comment.

Schedule Scripts are for backwards compatibility. For new applications, an **Update Action List** configuration is the recommended approach.

Scripts are run using the *Run a Schedule Script* **Action**.

A script can contain two sections, **Command** and **Trigger**. Each section is written in the script by enclosing it in [].

Command Section

The **Commands** section is defined by **[Commands]** in the script file followed by a list of commands. Any command listed under this section is executed when the script file is ran. This is useful when is there are a number of actions that need to be processed collectively on a common schedule condition regardless of any configured **Trigger** sections in the script.

Example: A script to update and print a worksheet.

[Commands]

command1 = UpdateSheet 'DailyReport.Template'

command2 = PrintSheet 'DailyReport.Template'

In addition, to actions, the **Command** section supports simple script keyword as follows:

- Wait n
  Pause the script for n seconds
- Init v n
  Initialize the variable v to n. Variables supported are x, y, and z.
- Incr v n
  Increment the variable v by n.
- Start Loop n
  Indicates the start of a loop to be executed n times.
- End Loop
  Indicate the end of the loop.
- Goto x
  Jump to the script commandx

Example: Update the groups 1 to 5 in a worksheet and the print the command. Note that without the script this can be accomplished by six schedule commands.

[Commands]

; Initialize x to 1

command1=Init x 1

; Set up a loop to iterate 5 times

command2=Start Loop 5

; Update the Daily Report for the specific group (1-5)

command3=UpdateGroupSheet 'DailyReport.Template' '{x}'

command4=Wait 2

command5=Incr x 1

; End the Loop

command6=End Loop

; Print the worksheet

command7=PrintSheet 'DailyReport.Template' 'default'

Trigger Section

The **Trigger** section is defined by **[TriggerX]** in the script file where *X* is a numeric value starting at 1 allowing multiple **Trigger** sections in a single script file. The script is used to monitor the content of a folder for a new file and perform commands when that happens.

Example: Monitor a folder for a CSV file and run report when that happens.

In addition to actions, the **Trigger** section supports simple script keyword as follows:

- **Type**, either *1* or *2*.

  Type *1* is a file trigger where the **FileSource** directory is monitored and when one or more files are found in the folder the most recent file (based on modified date or created date depending on the **FileLock** setting) is processed based on the other settings in the section.

  Type *2* is a file trigger where the File Source directory is monitored and when one or more files are found in the folder every file found is processed based on the other settings in the section.

- **FileSource**

  The folder to monitor for new files. This setting must be specified.

- **FileFilter**

  The filter to apply when monitoring for new files in FileSource. For example, to monitor for CSV files, set to **.csv*. To monitor for every Excel workbook file that starts with *Flow*, set to *Flow*.xls**. If this is not specified, the default is **.** which processes any file in the directory.

- **FileLock**, either *0* or *1*

  This setting determines how files are considered for the trigger. If this is set to *0*, every file that exists in the **FileSource** directory and satisfies the **FileFilter** is considered. If this is set to *1*, each file must also be accessible (e.g., the file is not locked) in order to be considered.

  If **Type** is set to *1* and this setting is *0*, the most recent file in the folder is determined using the file creation date. If this setting is *1*, the most recent file is determined using the file modified date.

  If this setting is not specified, the default behavior is *1*.

- **FileMaster**

  When the file is found in the folder, it can be copied to the **FileMaster** setting so that subsequent commands can deal with a consistent file name. If specified, it must be specified with a full path. If this functionality is not required, this keyword may be omitted.

- **FileArchive**

  Once a file is processed it can be moved to another folder so it is not processed again. This setting is the folder the processed files are moved to. This is an optional setting.

  If the file to process already exists in the folder specified for this setting, when archived, a number is appended to the file name (starting at 1) so it can be archived. For example, if the file is *ABC.txt* and *ABC.txt* has already been archived, it is archived as *ABC_1.txt*. If another *ABC.txt* is processed, it is archived as *ABC_2.txt* and so on.

  If this setting is not specified, after the file is processed, a file named _lastprocessed plus the script file name and the section is created in the **Data** folder of the project under the **XLRprocess** folder. For example, if the script file is named *CSVMonitor* and the section is *Trigger1*, the file name is:

  _lastprocessed_CSVMonitor_Trigger1.ini

  This file contains 2 timestamps, one for the last modified time and one for the last created file time. If **FileLock** is *0*, the last created file time is stored, whereas if the **FileLock** is set to *1* the last modified file time is stored.

  If you wish to have all the files in the **FileSource** directory considered, delete this file.

- **FileTarget**
  The register variable (e.g., RG000) to set with short name of the file found in FileSource. The name does not include the file path or extension. If this functionality is not required, this keyword may be omitted.
- **FileFullTarget**
  The register variable (e.g., RG001) to set with the full name of the file found FileSource. The name includes the file path and extension. If this functionality is not required, this keyword may be omitted.

Example: Monitor the folder *c:\Data* for a CSV file. When found, set *RG000* with the short name, *RG001* with the long name, copy it to *c:\mydata\mymaster.csv* and then move it to *c:\Data\archive*. Run command actions to update report and save it to PDF (using the register *RG000*, *RG001* and the file *mymaster.csv*).

[Trigger1]
Type=1
FileSource=C:\data
FileFilter=.csv*
FileLock=1
FileTarget=RG000
FileFullTarget=RG001
FileMaster=C:\mydata\mymaster.csv
FileArchive=C:\data\archive
Command1=UpdateSheet 'MyTemplate.Template'
Command2=SaveSheetPDF 'MyTemplate.Template' ''

Note that if all the files in c:\Data need to be processed, set the *Type* to *2*.