*Welcome to AutoMate™ 4

Unisyn Software's popular AutoMate™ software enables you to teach your Windows based computer how to perform "tasks" automatically without any help or intervention. It is a fully extensible automation platform that is both powerful and easy-to-use.

A task can be set to run at specific times and intervals such as daily, weekly, only on weekends, only on weekdays, every four hours, every three weeks, etc. Furthermore, a task can optionally run as a result of many other triggers (if you specify) such as a hotkey being pressed (i.e. CTRL-F2) or a certain window appearing or even a standard Windows™ event such as a low battery power condition on a laptop.

With AutoMate™, you can set up tasks that can range from simple routine operations like running maintenance on your hard drive every Friday to extremely complex chores involving multiple applications, DDE conversations, and/or Internet file transfers.

When you are ready to take AutoMate™ to the next level, beyond single machine automation, be sure to check out our Network Automation section ... this is where AutoMate™ has the power to save IT Professionals tons of time and money.

Our goal is to insure that this product is easy for you to use without compromising available power and functionality.

Questions, comments? Contact Us.

Enjoy! -- Unisyn Software AutoMate™ Development Team

Click the Links Below to learn how to get AutoMated!

More about AutoMate™ Tasks

Features that make AutoMate™ the Windows Automation Standard

Whats New in AutoMate™ 4

Why is AutoMate™ Different from Schedulers?

Differences Between AutoMate™ Versions

Scripting In AutoMate™ Professional

AutoMate™ Tasks

Tasks are the foundation of AutoMate™…. Since AutoMate™ is designed to allow you to automate any task on your computer, creating a new task is the way to teach AutoMate™ how to perform the operation. A task is comprised of steps; and these steps are carried out in the order you specify. These steps can be comprised of a number of powerful operations, such as recording and playing back mouse movements and clicks, launching specific applications, sending keystrokes to a particular application, sending/receiving files to FTP sites, communicating via DDE commands and requests with DDE enabled server applications, playing sounds, restarting or shutting down Windows, switching focus between separate windows or applications, etc.

Occasionally when building an automated task, it is important that the task run alone and uninterrupted. Since AutoMate™ is fully multi-threaded, it has the capability of running many tasks simultaneously. Tasks can be given special priorities that allow it to interact differently with other tasks.

Related Topics:

Task Triggers

Task Actions (Steps)

Priorities And Synchronization

Importing Tasks

Exporting Tasks

Features

AutoMate™ has the most robust feature set of any automation tool on the market today! You can always count on Unisyn Software to keep you and your company on the cutting edge of automation technology.

- Easy-to-use interface with the AutoMate™ Task Wizard and Step Builder.
- TIA (Trigger Independent Architecture) allows any Windows event to trigger a task, giving you more flexibility than just simple scheduling.
- Available Windows NT service version.
- Low Memory requirements.
- VBA compatible scripting with OLE task object exposed for self-modifying tasks. Integrated Debugging Environment (IDE) included (so you don't need to use Notepad to create your scripts)
- Flexible scheduling frequencies .
- Down-to-the-second precision and fully multi-threaded operation make AutoMate™ perfect for running all your mission critical tasks.
- Fully scalable as AutoMate™ is easily scaled to fit everyone's needs from the single user home office to enterprise network environments.

There is a version of AutoMate™ for everyone

Related Topics:

Why AutoMate™ is the Automation Platform

Automation Platform

AutoMate™ versus simple job schedulers/macro programs.

Do not confuse AutoMate™ with the class of job scheduler/macro software products that have made their way into and out of the market. That assumption would not do any justice to the plethora of functions and features that make AutoMate™ much more than just another job scheduler.

AutoMate™ is the Automation Platform that provides much more that an easy way to schedule programs to run unattended - it takes that metaphor and improves on it by adding flexibility and benefits that empower users in a completely different manner than ever before. The difference between AutoMate™ and a run-of-the-mill scheduler can be explored by looking at the difference between the word "scheduling" and "automation." Scheduling is a time based firing of something. Automation is the act of automatically performing something, without one having to be there, either on a time/date based schedule or on various other dynamically trigger-based, either proactive or reactive, pseudo-schedules. Automation was, in a sense, the real reason why computers were invented – where the computer does all the redundant work for us allowing us to devote our time and energy in places that require true human interaction, ingenuity and creativity.

AutoMate™ provides user(s) with the power of automation, whether the task be something as simple as backing up important files on a weekly basis or something as complex as handling enterprise network system administration or responding to and acting according to various environmental changes in a group of systems.

Related Topics:

AutoMate™ Features

Differences between AutoMate™ Versions

Triggers

A Trigger is an event or time that causes a task you create to run. Once constructed, a task needs to have one or more triggers associated with it in order for it to perform its job. AutoMate™ utilizes Unisyn Software developed TIA (or Trigger Independent Architecture) which means simply that a task does not revolve around a single type of trigger such as a set schedule or a hotkey. Rather, multiple triggers can be assigned to a single task to cause it to launch based on a variety of circumstances.

AutoMate™ 4 offers a wide variety of triggers that will allow you to vary the times and events that will cause a given task that you construct to fire (or launch). They are:

Scheduled

Hotkey

File Change

Window Watcher

AutoKeys

Idle

Startup

Screensaver

Logout

Time Change

Low Memory

Device Change

Display Change

Palette Change

Spooler Change

Schedule Trigger

A Schedule Trigger is one of the more commonly used triggers. A schedule trigger allows you to configure a task to run at a certain date and time and optionally reschedule itself continuously based on a frequency that you specify.

AutoMate™ 4 improves on the scheduling engine of its predecessor AutoMate™ 3 in a number of ways. AutoMate™ 4 now includes seconds as a frequency option. Further adding to this new power is AutoMate's vastly improved late task handling capability.

Related Topics:

Frequency

Next Launch Date / Time

Late Task Handling

Rescheduling Options

Triggers

Schedule Frequency

If you only want your task to take place once, leave the Once option selected. If you would like to set up or change the specific interval by which the task will be launched, select the Every option.

Seconds: If you would like your task to take place every x seconds, select this option.

Minutes: If you would like your task to take place every x minutes, select this option.

Hours: If you would like your task to take place every x hours, select this option.

Days: If you would like your task to take place every x days, select this option.

Weeks: If you would like your task to take place every x weeks, select this option.

Using Weeks, you can schedule tasks to run on only specific days of the week. After selecting the Weeks option, another tab labeled Days will become available. Click this tab to choose which days of the week you want the task to run on. Upon the first rescheduling, AutoMate™ will select the next day of the week you chose.

Note that the Days option does not take effect until the first rescheduling. For example, if you want to run the task on only Mondays, Wednesdays and Fridays, but you still want to run the task on Tuesday, set the first scheduled day for Tuesday, and after the task runs, AutoMate™ will reschedule the task to run on Wednesday. After it runs on Wednesday, AutoMate™ will reschedule for Friday, and so on.

Bi-Weeks: If you would like your task to take place every x Bi-Weeks, select this option. Bi-Weeks are two complete weeks. So if you were to have a task scheduled to launch every 3 Bi-weeks, it would launch the first time, and then skip six weeks, and then launch on the same day of the week that it was originally scheduled.

The Days option is also available for Bi-Weeks. See the Weeks option for more details.

Months: If you would like your task to take place every x months, select this option.

Quarters: If you would like your task to take place every x quarters, select this option. A quarter is three months.

Setting an interval

Enter the interval by which you wish this task to be launched in the Every ___ text box.

Related Topics:

Next Launch Date / Time

Late Task Handling

Rescheduling Options

Next Launch Date and Time

The Next Launch Date and Time setting control the next time that the task will run. If this date and time has passed when AutoMate™ starts – the task will be considered late. If the task is configured to run more than once on a regular frequency, then this time will automatically be advanced by the amount specified in the frequency. An easy-to-use calendar control is provided to change the settings.

Related Topics:

Frequency

Late Task Handling

Rescheduling Options

Late Task Handling

A task can be late for processing because the computer was off or in a sleep state when the task should have been run. Since AutoMate™ is designed to run automatically when Windows starts, AutoMate™ will see that a task is late. If that occurs, there are several AutoMate™ actions that will deal with this condition --- what you will use depends on the nature of the task.

You may want to ask yourself these questions when considering what option to select for late tasks:

- Is my task time critical (meaning if it doesn't fire in the time you specify, does it still need to run even though that time has passed)?

- Do I need to know when the task is late?

- Is my computer unattended (for example on a server system you would not want it to prompt the user for action since none would be there to answer the prompt)?

There are 3 options to choose:

Immediately run the task
This option runs the task regardless (and without any prompt) if the task is late. You would use this option if the time a task runs is not important.

Don't run the task
This option does not run the task if it is late because the task is time critical and may not operate correctly if it is not within the time parameters it was meant to run.

Prompt for what to do
This option assumes a user will be at the machine to view this prompt and make a decision. If running AutoMate™ on a workstation, this is a good option to use; if running on a server, this option is not recommended, as no one would be there to deal with the prompt.

Related Topics:

Rescheduling Options

Frequency

Next Launch Date / Time

Rescheduling Options

After AutoMate™ launches a task (whether late or as scheduled), AutoMate™ will advance the date/time by the amount specified in the frequency settings. The Rescheduling options deal mainly with precision and late task handling. These options allow you to specify whether this task's next launch date time will be advanced starting from the time the task finished executing or relative to the date/time the task was originally scheduled.

Relative to its originally set Date/Time

This goes back to time-critical tasks, such as a task needing to run every hour at exactly the 15th minute of that hour. If the task is late due to the computer being shut down, choose this option.

Relative to the Date/Time the task was launched

On the other hand, if it is only important that the task run every 2 hours and not the exact time of day, then you would choose this option.

Disable the task

If the task is time critical and you would want to discontinue operation if the task is late, choose this option.

Related Topics:

Frequency

Next Launch Date / Time

Late Task Handling

Hotkey Trigger

Along with the many other triggers that can launch your task, you can easily set up ***hotkeys*** to launch tasks that you want to fire manually while at your workstation. An example of a good place to use this feature is to automate a series of keystrokes or the launching of applications or web sites. AutoMate's hotkey support allows you to attach an unlimited number of actions to a single keystroke.

This feature is an excellent time saver and allows you to increase your productivity while at your machine using AutoMate™.

Example:

To make a Hotkey launch a task, simply select that option under the triggers folder of the task properties or in the New Task Wizard below the checkbox. Press the hotkey combination that you would like to trigger the task and AutoMate™ will automatically translate the key press into the proper character recognition.

Related Topics:

Task Triggers

Window Watcher Trigger

Window Watcher Trigger

AutoMate's Window Watcher feature allows AutoMate™ to further enhance your productivity by watching for certain dialog boxes or windows and will automatically carry out your task when that dialog box or window appears.

This feature is an excellent time saver and allows you to increase your productivity while at your machine using AutoMate™.

Simple Usage Example:
This option is useful to eliminate repetitive and often annoying dialog boxes, such as "Connect to the Internet?" in earlier revisions of Windows 95. (Windows is a trademark Microsoft Corp.)

Advanced Usage Example:
Other uses for this feature may be to swipe away message boxes that may appear while other AutoMate™ tasks are running unattended. Since AutoMate™ 4 is multi-threaded, two tasks can operate simultaneously and at times assist each other in their completion.

Related Topics:

Task Triggers

Hotkey Trigger

Windows Events Trigger

As part of the TIA (Trigger Independent Architecture) of AutoMate™, you can configure tasks to launch as a result of a large variety of other Standard Windows System Events.

A Little About Windows Events (Warning… slightly technical):

Windows is an event driven environment. When a dialog box appears on your system, the system is for the most part idle. This continues until you click the OK button after which the Windows operating system generates a button click event and posts it to your applications event queue. When your application intercepts that event, it takes appropriate action.

There are many more events that most people are unaware of, that Windows™(Microsoft Corp.) posts to all applications on the system. For example when you select to shut down your computer, Windows posts a WM_QUERYSHUTDOWN message to all the applications running on the system. This gives applications the chance to save any work in progress before being shut down. There are other events that Windows generates for running applications to alert them of certain conditions. These include: when the time is changed manually on the system so that applications that are time sensitive can adjust their internal data, or when the battery power gets low on a notebook, or when the computer is about to go into a power-sever sleep mode.

AutoMate™ has the capability to hear these system-generated events and launch tasks as a result of the event.

Example:

An example of how this might be used: on system shutdown AutoMate™ could copy your document files to a server for backup or sharing purposes.

Related Topics:

Task Triggers

Hotkey Trigger

Window Watcher Trigger

Contact Unisyn Software, LLC

Unisyn welcomes user feedback and questions regarding our technology or anything else involving automation. We are located on Wilshire Blvd. in the Mid-Wilshire district of Los Angeles, California -- USA.

Mailing Address:
Unisyn Software, LLC
3440 Wilshire Blvd.
Suite 910
Los Angeles, CA 90010

Unisyn has a friendly staff of customer service and technical support representatives that are eager to assist you in finding and exploring the tremendous value in this product. Following are phone numbers that may be used to contact Unisyn during normal business hours (8:00am- 5:00pm Pacific/ West Coast Time, N. America).

Sales Toll Free (within the United States):
(888) 7-UNISYN (NOTE: remember ***not*** to dial 800! It's 888 which is also toll-free.)
Please do not use this number for Technical Support!

Sales and Customer Service Main Line:
(213) 738-1700

Real People Technical Support
(213) 738-6966

Main FAX:
(213) 738-7665

If you have a technical or sales related inquiry outside of our regular business hours, please utilize our web site:

For technical assistance, visit:
http://www.unisyn.com/automate/user/
Here you will find user resources, discussion groups and plenty of technical FAQ's that may address your issue.

For Sales information, visit:
http://www.unisyn.com/automate/

To order via our secure online order form:
http://www.unisyn.com/order/
(NOTE: By ordering online, you will receive your serial number instantaneously as your credit card will be charged and approved immediately, 24-hours a day – We accept MasterCard, Visa and American Express.)

Learn about AutoMate™ upgrades and announcements. Sign up to the newsletter! Visit:
http://www.unisyn.com/newsletters/

Related Topics:

Differences Between AutoMate™ Versions

Differences Between AutoMate™ Versions

There are several flavors available of AutoMate™, each for a distinct audience and level of automation and scalability.

AutoMate™ Standard Edition
AutoMate™ Standard is the base version of AutoMate™. This version does not contain: 1) the ability to connect to a central AutoMate™ Enterprise Server over a network for security and task deployment; 2) the BASIC scripting language capability; 3) enhanced BASIC scripting integration; 4) embedded expression in the Step Builder; 5) IF support; 6) enhanced variable support; 7) failure notification; or 8) the option to run AutoMate™ Professional as an NT Service.

AutoMate™ standard does, however, contain many features for simple automation routines on non-server systems. (EVALUATION NOTE: The trial version available for download contains features available only in AutoMate™ Professional; however, these features will be disabled upon your entering an AutoMate™ Standard serial number obtained during purchase.)

AutoMate™ Professional Edition
The Choice for IT Professionals.

If you are a "power user" involved in Information Systems or network management, AutoMate™ Professional is the version for you. The professional version of AutoMate™ contains 8 core advantages over the standard version.

1) Network Connectivity
   AutoMate™ Professional contains the ability to connect to an AutoMate™ Enterprise Server, which allows tasks to be centrally developed, deployed and securely managed on a TCP/IP network. Discount multi-user pricing is available for AutoMate™ Professional. Contact Unisyn Software for details.

2) BASIC Scripting Language
   AutoMate™ Professional combines scheduling power and application control with a full-featured Visual Basic for Applications compliant scripting language. The scripting engine integrates seamlessly into the existing AutoMate™ framework, appearing as an extra option in the macro builder called 'Run a Script'. In an automated procedure, you may have an unlimited number of 'script' steps, making decisions, handling complex File I/O routines, etc. The added flexibility of the AutoMate™ BASIC Scripting Language is that an unlimited number of customized 'AutoMate™' oriented commands can be added to the language to make life (and automation) easier for you. We have already added several new automate-centric commands, such as extensions of the BASIC scripting language and regularly release periodic AutoMate™ BASIC scripting language extension updates that are free to all registered users of AutoMate™ Professional.

3) Enhanced BASIC Scripting Integration
   Previously, the use of variables in BASIC scripts was “local” to the BASIC script in which it was used. However, beginning with AutoMate™ Professional Version 4.4, variables created in the Step Builder using the “Create Variable” action extend into the BASIC scripting language; and when the BASIC script exits, the new value of the variable is transferred back to the Step Builder for usage within the task. This allows task developers to write dramatically less code as the rest of the processing designed to take place on that value within the task can be done via the standard drag-and-drop task building process in the Step Builder.

4) Embedded Expressions in the Step Builder
   AutoMate™ Professional contains the ability to embed expressions directly into the parameters of task steps. Any valid BASIC expression can be embedded directly into any step parameter, eliminating the need to write

an entire BASIC subroutine using the "Run a BASIC Script" action for simple one line data formatting and/or retrieval operations. These operations can now be accomplished "inline" inside the AutoMate™ task.

5) IF Support

AutoMate™ Professional allows for the evaluation of variables in AutoMate™ tasks and for specific actions to occur as a result of that evaluation through the IF action. Previously, one had to use a BASIC script to obtain any level of conditional processing in a task. The IF action will make this easier and completely code-free (not to also mention it making AutoMate™ even more powerful). The IF action contains the following parameters:

- Variable: the variable to be evaluated
- Condition: the condition to evaluate
- Value: the value to compare the variable to
- Then: the action to take if the condition is true
- Else: the action to take if the condition is not true

6) Enhanced Variable Support

AutoMate™ Professional improves variable support with the new "Variable Functions" action. This action allows for several forms of variable manipulation including the incrementing and decrementing of a number and the concatenating of a text variable. Also, several actions that previously only took a number as a parameter now allow variables. This makes the new counter functions found in the "Variable Functions" action even more useful.

7) Task Failure Notification

AutoMate™ Professional contains the ability to send e-mail alerts to specific users when a task fails. All relevant information is sent (including the date, time, line number and reason for failure). Options for pager message formatting is included as well.

8) AutoMate™ Professional NT Service Edition Option

AutoMate™ Professional also contains an option to use the NT Service Edition of AutoMate™ which contains all the features of AutoMate™ Pro and allows tasks to be run on NT Servers even if those machines are locked or logged out. This version, however, does involve a few extra configuration steps and does not need to be used on NT machines unless you require the ability to run tasks on a logged out NT machine.

(NOTE: Purchasing the NT Service Edition is the same as purchasing AutoMate™ Professional. Buying one entitles you to use either AutoMate™ Professional or AutoMate™ Professional NT Service Edition on one machine)

AutoMate™ Enterprise Server

Control and Deploy AutoMate™ Tasks Throughout Your Entire Network

AutoMate™ Enterprise Server brings our successful model for automating the individual computer to the corporate network and the Internet. By incorporating AutoMate™ Enterprise into existing networks, Information Systems professionals will be able to seamlessly deploy automated tasks across one or many different machines simultaneously whether they reside on the other side of the office or on the other side of the world. It allows groups of AutoMate™ installations to be linked together and centrally administered including security rights for the various installations and task deployment to one or thousands of AutoMate™ installations.

AutoMate™ Enterprise Server is designed to make administering an AutoMate™ network painless and without the sweat. Actions, such as system maintenance, software distribution and upgrades, file transfers, system restarts or anything else AutoMate™ can do, can now be designed centrally and distributed remotely and automatically. AutoMate™ will lower the overall cost of IT maintenance and support, improve the availability and performance of computing services, and provides a scalable enterprise solution by allowing you to build solutions that are

appropriate for your organization.

Related Topics:

Contact Unisyn

License Agreement

UNISYN DISCLAIMS ALL WARRANTIES, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE SOFTWARE OR DOCUMENTATION. SHOULD THE PROGRAM PROVE DEFECTIVE, THE PURCHASER ASSUMES THE RISK OF PAYING THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION AND ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES. IN NO EVENT WILL UNISYN BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING WITHOUT LIMITATION TO DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION AND THE LIKE) ARISING OUT OF THE USE OR THE INABILITY TO USE THIS PRODUCT EVEN IF UNISYN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN PARTICULAR, UNISYN SHALL HAVE NO LIABILITY FOR ANY DATA STORED OR PROCESSED WITH THIS SOFTWARE, INCLUDING THE COSTS OF RECOVERING SUCH DATA. AS A RESULT, THIS SOFTWARE AND DOCUMENTATION ARE LICENSED AS IS AND YOU, THE LICENSEE, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE.

Information in this document is subject to change without notice and does not represent a commitment on the part of Unisyn. The software described in this document is furnished under this license agreement. The software may be used or copied only under the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license agreement. No part of this manual or software may be transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the written permission of Unisyn.

USE OF THIS PRODUCT FOR ANY PERIOD OF TIME CONSTITUTES YOUR ACCEPTANCE OF THIS AGREEMENT AND SUBJECTS YOU TO ITS CONTENTS.

US GOVERNMENT RESTRICTED RIGHTS

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subdivision (b)(3)(ii) of the Rights in Technical Data and Computer Software clause at 25 2.227-7013. Contractor / manufacturer is

UNISYN SOFTWARE, LLC
3440 Wilshire Blvd.
Suite 910
Los Angeles, CA 90010

Copyright and Trademark Acknowledgements

Actions

There are two primary components to a task , the triggers and the actions that take place when a task is triggered.

Once an AutoMate™ task is triggered, the task moves through the steps that you specified during task creation in AutoMate's built-in Step Builder.

The built-in actions that AutoMate™ has available are very powerful. Below you will find links to descriptions of each action.

AutoMate's available actions include:

General
Start an application
Open a document
Paste keys
Print a document
Message
Reminder
Play a sound
DDE commands
Send keystrokes
Comment

Window Manipulation Actions
Focus
Maximize
Minimize
Restore
Close
Hide
Unhide
Focus window containing

Internet Actions
FTP Quick Send
FTP Quick Retrieve
FTP Login
FTP Download
FTP Upload
FTP Make Directory
FTP Remove Directory
FTP Delete
FTP Rename
FTP Logout
Check for mail

Send SMTP mail
Dial-Up

File System
Copy a file
Move a file
Rename a file
Delete a file
Create a directory
Remove a directory
Change directory

Zip
Zip
Unzip

Clipboard
Copy
Cut
Paste
Clear

System Actions
Reboot
Shutdown
Quit AutoMate™
Login (NT Service Edition ONLY)
Logout
Update AutoMate™

Control Panel / Services
Start a service
Stop a service
Pause a service
Continue a service
Install a service
Remove a service

Flow Control Actions
Pause
Loop
Prompt
Wait for a window
Wait for window to disappear
Start another task
Disable this task

IF conditional

Security
Lock mouse
Lock keyboard
Unlock mouse
Unlock keyboard
Password

Recorder
Record events

Professional / Enterprise Actions
Run a BASIC script

Variables
Create variable
Set variable
Input variable
Send variable
Populate variable with clipboard
Populate clipboard with variable
Variable functions

Starting an Application

Select this action to have AutoMate™ launch a specific program. This program may be any .exe, .cmd, com, .bat, .pif or other executable file. (Special note to NT Service Edition users: this action will execute an application in the security context of the user that the task logged in as, which was set either by your default login settings or through a Login Step as the first step of your task. See the security notes on the NT Service Edition for more details.)

This action supports a number of basic parameters (located on the General tab) as well as more advanced functionality on the Advanced tab. The following parameters are located on the *General tab*:

Command Line
The command line to execute to start the application. This is usually the fully qualified path to the application you want to start (for example, C:\WINNT\Notepad.exe")

Default Directory
The directory the application should use as the default directory. Most applications, especially older ones, will use this directory to find support files it may need to execute properly. If you do not supply this parameter, AutoMate™ will use the path you supplied in the *Command Line* parameter as the application's default directory.

Parameters
Any parameters that should be supplied to the application. AutoMate™ will automatically append this line to the end of the *Command Line* parameter.

The following actions are located on the *Advanced Tab*:

Wait until application is ready for input before continuing
If this is selected, AutoMate™ will pause the task until the application you are trying to start informs the operating system that it is done starting up and is ready for user input. Use this action if you want to ensure the application is ready to take your input before moving on to the next step.

Wait until application to end before continuing
If this is selected, AutoMate™ will pause the task until the application closes. It also allows AutoMate™ to populate an AutoMate™ variable with the return code of the application, if you tell it to do so. See below for more details.

Place application return code in this variable
If this is selected, indicate which variable you want AutoMate™ to place in the application's return code. You can then use the AutoMate™ variable testing actions to perform flow control of your task based on the return value of the application.

Window Startup
Allows you to select whether the application's window should be normal, hidden, maximized, or minimized when it starts.

Related Topics:

Other Available Actions

Variable

Sending Keystrokes

This action sends keystrokes to a specific application. The application must be in focus or activated in order to receive these keystrokes. To focus a specific window, see the action Focus a Window .

All that you need to know in order to send keystrokes to an application is which keys you would press to get what you want done. One may think that some things are not possible without a mouse, but every well done Windows program has keyboard shortcuts and hot keys that will allow you to, with the keyboard, perform any action that can be performed with a mouse. The secret is simply experimentation.

Most keyboard shortcuts are made visible underlining the character necessary to press (simultaneously with <alt>) in order to execute the button / command / function, etc. For example, from this Help File, you would hold the <alt> key down and press p to Print Topic.

For further information on sending keys, please see Special Keys and Their Equivalents

To send keystrokes:

1) Select the Send keystrokes action from the list of actions in the Step Builder.
2) Type the keys that you would like to be sent into the text box labeled “Keystrokes”.
3) Click OK to add the new step to your task.

There are a few notes to keep in mind when sending keystrokes using AutoMate™:

1) Keystrokes must have a target application to execute upon. This means that you cannot use System Wide keystrokes such as Alt+Tab or Ctrl+Alt+Del etc.

2) Key codes are case sensitive. Do not use capital letters with %, ^, or + unless you specifically intend to send a capital letter. Incorrect case may cause your task to fail.

3) You may need to wait a few seconds before sending your keystrokes (while waiting for a window to appear, for example). To do this, check the “Add pause before send” and enter the number of milliseconds to pause (1000 ms = 1 second). This will cause AutoMate™ to wait the specified amount of time before beginning to “type”.

The Send Keystrokes action can emulate most of the special keys on the keyboard (such as the Scroll Lock and Caps Lock keys), and also supports “fields,” special text surround by braces that are replaced at run time by a dynamic value. Consult the help file sections on Special Keys and Their Equivalents for emulating special keys, and the section on Keystroke Fields for more information on supported AutoMate™ fields.

Example

The example below illustrates the process of launching Notepad, typing out some text into the application, and then saving that file to your hard disk.

1) While in the Step Builder, double click the “Start An Application” step. Enter the path to notepad.exe, i.e. c:\windows\notepad.exe, or select the Browse option and search for the file in your system. This will start Notepad when the task is triggered.
2) Next, we need to establish the focus on Notepad to be sure that it receives the keys being sent. To do this,

select the Focus Window step (located in the Windows submenu on the left-hand side of the Step Builder). In the text box marked “Window Title”, enter the words “Untitled – Notepad” (**NOTE: Case sensitive entry**). This will search for the window with that text and focus it. Click Ok to add the step.

3) Now we need to tell AutoMate™ what keystrokes we want to type into Notepad. Double click the “Send keystrokes” action, and in the area marked “Keystrokes”, type in the text as you want it to appear in Notepad. When you are finished entering your text, type in %f. AutoMate™ will interpret this as an Alt-f keystroke to invoke the file menu. Next enter an "s" to select save, then type the name of the file to save. Enter a "~" sign (the shortcut for the Enter key) and you are done. For example, your text may look like this:

   To whom it may concern,~~This is an example macro.%fsMy Notes.txt~

That’s it! Whenever the task is triggered, notepad will open, your text will be typed into it, and it will be saved to file entitled “My Notes.txt”.

Related Topics:

Other Available Actions

Special Keys and Their Equivalents

Each key is represented by one or more characters. To specify a single keyboard character, use the character itself. For example, to represent the letter A, use A. If you want to represent more than one character, append each additional character to the one preceding it. To represent the letters A, B, and C, use ABC. To send a space, simply press the space bar. The plus sign +, caret ^, percent sign %, tilde ~, and parentheses ( ) have special meanings. To specify one of these characters, enclose it within braces. For example, to specify the plus sign, use {+}. Brackets [ ] have no special meaning, but you must enclose them in braces as well, because in other applications, brackets do have a special meaning that may be significant. To send brace characters, use {{} and {}}. To specify characters that aren't displayed when you press a key (such as ENTER or TAB) and keys that represent actions rather than characters, use the codes shown below:

Key Codes

Key	**Code**
BACKSPACE	{BACKSPACE}, {BS}, or {BKSP}
BREAK	{BREAK}
CAPS LOCK	{CAPSLOCK}
Caps Lock On	{CAPSON}
Caps Lock Off	{CAPSOFF}
DEL	{DELETE} or {DEL}
DOWN ARROW	{DOWN}
END	{END}
ENTER	{ENTER} or ~
ESC	{ESC}
HELP	{HELP}
HOME	{HOME}
INS	{INSERT}
LEFT ARROW	{LEFT}
NUM LOCK	{NUMLOCK}
PAGE DOWN	{PGDN}
PAGE UP	{PGUP}
PRINT SCREEN	{PRTSC}
RIGHT ARROW	{RIGHT}
SCROLL LOCK	{SCROLLLOCK}
TAB	{TAB}
UP ARROW	{UP}
F1	{F1}
F2	{F2}
F3	{F3}
F4	{F4}
F5	{F5}
F6	{F6}
F7	{F7}
F8	{F8}
F9	{F9}
F10	{F10}

F11	{F11}
F12	{F12}
F13	{F13}
F14	{F14}
F15	{F15}
F16	{F16}
Windows Key	{WINKEY}
Popup Menu Key	{POPUP}
Press and release ALT	{QUICKALT}
Press and release CTRL	{QUICKCTRL}
Press and release SHIFT	{QUICKSHIFT}

Special Keys

To specify keys combined with any combination of the SHIFT, CTRL, and ALT keys,
Precede the regular key code with one or more of the following codes:

Key	Code
SHIFT	+
CTRL	^
ALT	%

To specify that any combination of SHIFT, CTRL, and ALT should be held down while several other keys are pressed, enclose the code for those keys in parentheses. For example, to specify to hold down SHIFT while "e" and "c" are pressed, use +(ec). To specify to hold down SHIFT while "e" is pressed, followed by c without SHIFT, use "+ec". To specify repeating keys, use the form {key number}. You must put a space between key and number. For example, {LEFT 42} means press the LEFT ARROW key 42 times; {h 10} means press h 10 times.

Related Topics:

Fields

Other Available Actions

Send Keystrokes Action

Focusing a Window

If AutoMate™ is working with two or more programs at once, it is necessary to bring a Window to the foreground. This is so AutoMate™ may send keystrokes to that application. If you do not need to send keystrokes, then this action may not be necessary.

To establish a focus window, simple enter the windows Caption into the text box in the Establish a Focus dialog. The caption of a window is located on the top right of the title bar of that window. For example, if Notepad is launched without a document, its caption is Notepad - Untitled.

To bring a Window into Focus (to the forefront):

1) Select the Focus a Window action from the Step Builder.
2) Type the name of the window that you would like to focus into the text box.

NOTE: Some window captions change depending on the content and actions of their application.

Related Topics:

Other Available Actions

DDE Commands

The formal definition of DDE, or dynamic data exchange is "an established protocol for exchanging data through active links between applications that run under Microsoft Windows."

Dynamic Data Exchange provides a channel for two Windows applications to communicate. Using Dynamic Data Exchange (DDE), one application -- referred to as the client application - can request information from, or send commands to, another application, referred to as the server application. The server application then processes the request from the client application. The server performs a task, such as updating data, or returning requested information to the client, such as an element of data maintained by the server application.

AutoMate™ is designed to act as a DDE client, with the ability to send DDE requests and commands to Server applications. Most popular Windows applications provide support for DDE, including Microsoft Word, Symantec's ACT!, Delrina WinFax Pro and many more.

AutoMate's DDE capabilities should only be used by an experienced individual or one who is interested in becoming experienced with the use of DDE communications.

For more information on the DDE commands supported by a particular software product, see the applications manual or contact the software's manufacturer.

Related Topics:

Other Available Actions

Playing a Sound

AutoMate™ can play a WAV file sound as part of the action steps in a task. This is useful if you would like to be alerted if a task completes.

For example you could have AutoMate™ check your POP3 mailbox as the first step and have it continue onto the second step if you have mail. The second step would be to play a sound indicating that you have mail. The net result is that you have created a mail notification tool as part of your AutoMate™ configuration!

EXPERIMENT: Set up the 3rd step to launch your mail client for you using Start an Application .

Related Topics:

Other Available Actions

Pause Action

Works on Windows NT ONLY

Select this action to have AutoMate™ wait for a certain number of seconds or minutes during/or in the middle of a task. This function is useful when performing extensive operations that require a delay between action steps.

Rather than pausing you may want to try to use Wait for a Window (if one appears when the task is ready to proceed).

NOTE: This action is frequently used to compensate for load time of an application before sending keystrokes to it.

Related Topics:

Other Available Actions

Wait for Window Action

Wait for a Window

Similar to the Window Watcher trigger, except now as an action , this option allows you to affect the execution of the task by stopping execution until a certain window appears such as Completed.

Rather than waiting for a window (if one does not appear) you may want to try to use the pause action .

NOTE: For your convenience, when inserting this step, a list of all open Windows on the system is displayed for quick selection.

Related Topics:

Other Available Actions

Wait For Window To Disappear

Open a Document Action

The open a document action works similarly to the launch an application action with the exception that it is designed to open any document file including Microsoft Word .DOC files or even web sites.

For Example:
To open a web site with your default browser, simply add the Open a document action in the Step Builder and in the document filename box, enter http://www.unisyn.com.

This would open the Unisyn Web site. If your web browser needs to connect to the Internet first, you may need to send some keystrokes to press connect or you could create another task with a Window Watcher trigger that always handles that dialog box.

Related Topics:

Other Available Actions

Print a Document Action

Print a Document Action

The Print a Document action works in a similar manner to the Open a Document action, with the exception that the document is printed to the default printer.

Related Topics:

Other Available Actions

Open a Document Action

Start an Application Action

Reminder Action

Use a Reminder action to create a task that will serve as a personal reminder to you to remind you of important events such as birthdays, appointments, etc.

When AutoMate™ executes a Reminder action, a dialog is displayed with the text of the reminder and a snooze button to tell AutoMate™ to re-remind you at a later date/time.

NOTE: This feature replaces the Reminder type task in AutoMate™ 3. Users of AutoMate™ 3 will remember that there were two types of tasks: Procedures and Reminders. In AutoMate™ 4 there is only the main task architecture which is robust enough to encompass the functionality of both.

Related Topics:

Other Available Actions

Message

The message feature simply displays a message box with the text you specify. It does not include a snooze option like the Reminder action.

Related Topics:

Other Available Actions

Reboot Action

The Reboot action allows AutoMate™ to reboot the computer as part of the task. Please note that this step should always be the last step in your task, as a system reboot will begin immediately which would cause problems or failure for any steps following that action.

NOTE: The reboot option differs from the shutdown option in that your computer will reboot and reinitialize. If AutoMate™ is selected to run in the Startup group or as an automatically starting system service on Windows NT – AutoMate™ will be able to continue processing when the reboot process has completed.

Related Topics:

Other Available Actions

Shutdown Action

Quit AutoMate™ Action

Shutdown Action

The Shutdown action shuts down the computer. As with the Reboot action, this step should always be the last step in your task, as a system shutdown will begin immediately which would cause problems or failure for any steps following that action.

NOTE: The shutdown action differs from the reboot action in that your computer will not reboot and reinitialize. Manual intervention will be required to turn the computer back on so that AutoMate™ will continue processing tasks.

ANOTHER NOTE: Sorry folks, AutoMate™ cannot function when the computer is off. AutoMate™ will be unable to turn your computer back on – however we have heard reports of people using AutoMate™ to shut their computer down at a certain time in conjunction with a power timer that plugs into the socket to turn the system back in at the appropriate time.

Related Topics:

Other Available Actions

Reboot Action

Quit AutoMate™ Action

Quit AutoMate™

The Quit AutoMate™ action causes AutoMate™ to shut down. . Please note that this step should always be the last step in your task as AutoMate™ will shut down immediately which would cause problems or failure for any steps following that action.

NOTE: AutoMate™ will be unable to continue running your automated tasks if it is shutdown.

Related Topics:

Other Available Actions

Reboot Action

Shutdown Action

Dial-Up Action

The Dial-Up action establishes a Windows RAS (or Dial Up Networking) connection. Usually this feature is used to make a connection to the Internet before performing some other Internet related activity in your task. However it can also be used to establish a connection to any Dial-Up Server that you have configured in your Dial-Up Networking connections.

Related Topics:

Other Available Actions

Check for Mail Action

The Check for Mail action allows AutoMate™ to check a POP3 (Internet standard) mailbox and to continue or exit the task based on the existence of mail.

EXAMPLE:

You could easily create a task that first silently checks to see if mail is waiting in your mailbox using this action. If there is, then the task would continue on to launch your mail client so you could retrieve it, if not, then the task would exit before getting to those steps. This eliminates launching the mail client unnecessarily if there is no mail to retrieve.

Related Topics:

Other Available Actions

Dial Up Action

Run a Basic Script Action

Arguably the most powerful feature in AutoMate™ is its ability (Professional and higher only) to create, and execute VBA™ (Visual Basic for Applications) scripts.

To create a BASIC script, use the AutoMate™ IDE located in the AutoMate™ Program group under the Start Menu. Once the script is created simply select path to the .BAS filename that contains your BASIC code.

There is much more under the hood in AutoMate's scripting engine than just VBA… for an in depth look at AutoMate's scripting language extensions and exposed objects, click here .

Related Topics:

Other Available Actions

Scripting In AutoMate™

Scripting in AutoMate™

BASIC Scripting is available in the professional edition of AutoMate™. This fully Visual Basic for Applications™ compatible scripting engine lets you integrate BASIC language support into your automated procedures.

BEGINNERS NOTE:

It is not necessary for you to know how to write basic code to use AutoMate™. Many people will be able to achieve everything they need without ever using this feature. However, learning how to write VBA™ (compatible) code will extend the power of AutoMate™ tremendously. Furthermore VBA™ appears to be the standard for embedded application scripting languages as it currently appears in many other commercial applications.

What is BASIC Scripting :

BASIC Scripting allows you to add Basic language scripts to your procedure. Examples of commercial applications that contain a similar facility include: Microsoft Excel, Word for Windows, and CrossTalk for Windows. In each of these applications you can write source code that is used to control the associated application.

With BASIC Scripting, you can now have AutoMate™ make decisions while a task is running, such as an if [a condition is met] / then [perform an action] / else [perform another action]. Scripting gives you full control of file I/O, Winsock (internet), case structures, OLE Automation, .dll support and much much more...

If you have ever used Microsoft's Visual Basic™, the scripting process in AutoMate™ should be very familiar.

The following is a brief overview of the major functions and features of AutoMate's BASIC Scripting :

- Full support for industry standard BASIC (Beginners All-purpose Symbolic Instruction Code) source code.
- Custom commands and objects such as AMTask specifically oriented toward AutoMate's function.
- Call Basic subroutines or functions directly from tasks, giving you the ability to customize the behavior of your automation.
- Supports the in-code creation of custom dialog boxes using standard Windows controls including: buttons, checkboxes, group boxes, list boxes, option buttons (also called radio buttons) and text edit controls.
- Support for OLE Automation which allows the Basic language to programmatically access software components from other applications.
- The ability to declare and call any Windows API function from within the Basic language.
- Built-in editor and Integrated Development Environment.

For BASIC language syntax, please see Basic language help from the AutoMate™ program group or from within the AutoMate™ IDE (Integrated Debugging Environment).

SEE ALSO:

Scripting Language Extensions

Inline Expressions

Scripting Language Extensions

(Please note that this information is of a technical nature designed for individuals interested in using the advanced features of AutoMate™ BASIC… it is not necessary to write BASIC code to benefit from AutoMate™.)

AutoMate™ contains custom commands and objects as a part of its BASIC implementation. These are extensions to the language that are designed to help you work with, affect and manipulate AutoMate™ specific objects and functions during the execution of your task while running a BASIC script.

The potential for automation power is tremendous. Now, not only can your AutoMate™ task do anything you can write in BASIC code, the code itself has the ability to check back with AutoMate™ -- to read and modify the task from which it is running.

An example of the functionality is shown below. Where AutoMate™ BASIC code is executing and writing back a specific time to reschedule its own task that would otherwise not be supported using AutoMate's built in re-scheduling engine.

Following are the object extensions included in AutoMate's BASIC implementation:

AUTOMATE™ OBJECTS LANGUAGE EXTENSIONS:

Task Object

AutoMate™ System Object

LEGACY COMMANDS – Carried over from previous versions of AutoMate™ for compatibility

LaunchProc

ExitProc

ConnectTo

HangUp

IsWindowPresent

SEE ALSO:

Scripting in AutoMate™

Task Object Scripting Extension

DESCRIPTION:

The Task object encapsulates an AutoMate™ task. Upon executing an AutoMate™ BASIC script, AutoMate™ automatically creates an instance of Task called CurrentTask that contains the information from current task being executed. Using the CurrentTask object, a task has the capability to modify itself from with the scripting language. Using the Task Object in the scripting language, AutoMate™ could create new tasks as well to respond to certain conditions. You may create additional instances of the Task within AutoMate™ Basic scripts as necessary.

Properties

TaskName	Read/Write	Variant	Task Name
LaunchDate	Read/Write	Variant	Launch Date in the system ShortDate format
LaunchTime	Read/Write	Variant	Launch Time
Frequency	Read/Write	Integer	Task Execution Frequency 0 = Once 1 = Every 2 = Manual
IntervalType	Read/Write	Integer	Task Execution Frequency 0 = Minute 1 = Hour 2 = Day 3 = Week 4 = Weekday 5 = Weekend 6 = Bi-week 7 = Month 8 = Quarter 9 = Year 10 = Second
Interval	Read/Write	Integer	Task Execution Frequency in IntervalTypes
RunLate	Read/Write	Integer	Controls Run Action When Task is Overdue 0 = Immediately 1 = Don't Run 2 = Prompt
ScheduleLate	Read/Write	Integer	Controls Rescheduling When Task is Overdue 0 = Reschedule Relative to Original Date/Time 1= Reschedule Relative to Launch Date/Time 2 = Do Not Reschedule
Active	Read/Write	Integer	Controls Task Execution 1 = Task is Executing / Allowed to Execute 0 = Task will Stop Executing
Triggers	Read/Write	Integer	Task Triggers; combination of the following values: 0 = None (exclusive) 1 = Scheduled 2 = Windows Event (Message) 4 = Wait for a Window 8 = Hotkey
Hotkey	Read/Write	Variant	Trigger Hotkey
WindowName	Read/Write	Variant	Trigger Window Title

StepCount	Read Only	Integer	Number of Steps in Task
TaskData	Read/Write	Variant	Task Data in Raw, Internal Format
Details	Read/Write	Variant	Task Details
FrequencyString	Read Only	Variant	Task Frequency Display String
Events	Read/Write	Integer	Windows Events to Trigger On; combination of the following values: 1 = System Time has been Changed 2 = System is Low on Memory 4 = System Device has been Changed 8 = Display Mode has been Changed 16 = Color Palette has been Changed 32 = Spooler Count has Changed

Methods

Function GetTaskStep (StepNo As Integer) As Variant

Retrieves the string containing the specified task step. The first step is numbered 0. The string contains the task step command and parameters in the format that appears in the AutoMate™ step windows.

Example:

```
Sub Main
  x = CurrentTask.GetTaskStep(0)
  MsgBox x
  'CurrentTask.DeleteTaskStep(1)
  CurrentTask.AppendTaskStep(1, x)
End Sub
```

Function GetStepCommand (StepNo As Integer) As Variant

Retrieves the string containing the specified task step command. The first step is numbered 0.

Function GetStepParamCount (StepNo As Integer) As Integer

Retrieves the number of parameters in the specified task step.

Sub SetTaskStep (StepNo As Integer, NewAction As Variant)

Changes the specified task step. The NewAction parameter must be passed in the correct format for the new step to be valid.

Example:

```
Sub Main
  CurrentTask.SetTaskStep(1,"OPEN")
End Sub
```

Function GetStepParameter (StepNo, Parameter As Integer) As Integer

Retrieves a string containing the specified task step parameter. The first step is numbered 0 and the first parameter is numbered 0.

Example:

```
Sub Main
  X = CurrentTask.GetStepParameter(1,0)
```

```
End Sub
```

Sub AppendTaskStep (Action As Variant)
Adds a new task step. The Action parameter must be passed in the correct format for the new step to be valid.

Example:
```
Sub Main
  x = CurrentTask.GetTaskStep(0)
  MsgBox x
  'CurrentTask.DeleteTaskStep(1)
  CurrentTask.AppendTaskStep(1, x)
End Sub
```

Sub InsertTaskStep (Before As Integer, Action As Variant)
Inserts a new task step before the specified step. The Action parameter must be passed in the correct format for the new step to be valid.

Example:
```
Sub Main
  x = CurrentTask.GetTaskStep(1)
  CurrentTask.InsertTaskStep(3, x)
End Sub
```

Sub DeleteTaskStep (StepNo As Integer)
Deletes the specified step.

Example:
```
Sub Main
  CurrentTask.DeleteTaskStep(3)
End Sub
```

Function GetStepStatus (StepNo As Integer) As Integer
Retrieves the status flag for the specified task step. A return value of 0 means the step is disabled (skipped), 1 means the step in enabled.

Example:
```
Sub Main
  x = CurrentTask.GetStepStatus(1)
  MsgBox x
End Sub
```

Function SetStepStatus (StepNo, Status As Integer) As Integer
Sets the status flag for the specified task step. Setting Status to 0 means the step is disabled (skipped), 1 means the step in enabled.

Example:
```
Sub Main
  CurrentTask.SetStepStatus(1,1)
```

```
End Sub
```

Sub ClearSteps
Deletes all steps from the task.

Example:
```
Sub Main
  CurrentTask.ClearSteps
  MsgBox CurrentTask.TaskData
End Sub
```

Sub SaveToFile (FileName As Variant)
Save the task data to a file.

Example:
```
Sub Main
  CurrentTask.SaveToFile("c:\testtask.txt")
End Sub
```

Sub LoadFromFile (FileName As Variant)
Loads the task data from a file.

Example:
```
Sub Main
  CurrentTask.LoadFromFile("c:\testtask.txt")
End Sub
```

Related Topics:

Other Scripting Extensions

Scripting In AutoMate™

Run a BASIC Script Action

What's New In AutoMate™ 4

The upgrade from AutoMate™ 3 to 4 is the biggest and most compelling in Unisyn history.

Previous versions of AutoMate™ have been called the Definitive Windows Task Scheduler by Windows Magazine (a CMP publication), Editors Pick by C-Net, and a Five Cow Rating (like 5-star) by TUCOWS.COM. As you might guess we had a pretty big challenge on our hands to improve on those kind of accolades. One of the approaches we took to doing this was to transform AutoMate™ into much more than a task-scheduler. Rather, we wanted to make an automation platform wherein a variety of automation tasks, including but not limited to scheduling, was possible.

At Unisyn, we pride ourselves in our attentiveness to customer feedback. AutoMate™ 4 contains all the features you have been asking for over the years. First of all, it is a complete re-write from previous versions, created using the C++ programming language and it implements fully an object oriented architecture with exposed OLE server object for easy customization and extension. You will find we have re-created and improved on the base functionality of previous versions of AutoMate™ -- creating a new high-speed, multi-threaded, industrial strength architecture that we will continue to build on for many years into the future.

And now we are proud to present the new feature list for AutoMate™ 4:

Re-written from the ground up in C++

This was done mainly for the purposes of speed, memory footprint, extensibility, and reliability.

Up to 500% speed improvements in many areas, including interface operation

Lower Memory Requirements -- Dual Module Architecture (5-10% of previous versions)

The configuration interface (SETTINGS.EXE) is a separate EXE file from the Task Processing Service (AUTOMATE.EXE). Since AutoMate™ will need to be running most of the time, you will not be making changes to the interface as there is no need for it to be always loaded. AutoMate™ now has been split into two program files so that when you are done making changes to the configuration, the interface can be completely unloaded.

Fully multi-threaded operation

When a task is triggered in AutoMate™, it is contained within its own thread. AutoMate's scheduling engine remains on its own critical thread, unaffected by any problems or load placed by the task it has just started. As an example of the power of this architecture, AutoMate™ could now in theory run an unlimited number of tasks simultaneously and could launch them all at the exact same time of the day (down to the second).

Down-To-the-Second Precision

The scheduling engine in AutoMate™ now has the ability to maintain down-to-the-second precision when determining when to launch a task. This allows for extremely time-critical tasks to be carried out by AutoMate™.

TIA (Trigger Independent Architecture)

This new architecture means simply that AutoMate™ is no longer simply a task scheduler. Rather, a schedule is just one of many things that can launch a task. New triggers include, Hotkey , Window Watcher , and Window System Events (such as SHUTDOWN and LOWBATTERY)

More Actions

AutoMate™ 4 hosts a wide array of new steps, ranging from native FTP client actions for automating FTP sessions, controlling the flow of a task based on user input, and manipulating open windows. And the features and steps keep growing with each release. AutoMate's TIA architecture ensures that future triggers and steps can be added quickly and easily as they become available.

New Step Builder

The new Step Builder has been carefully designed to be more functional, faster and easier to understand then its predecessors. Actions are now grouped by their function for easy location when first entering the Step Builder. Furthermore, steps in a task can be cut, pasted, multi-selected, tested, temporary disabled and more. Also, you may now edit your BASIC script by selecting and right clicking it from within the Step Builder. You will be able to work faster with this new interface.

Windows NT Service Version

Those running Windows NT will now be able to execute their tasks while the NT machine is locked or logged out .

Improved and updated BASIC scripting support addition of New Objects

Updated and faster BASIC engine. New AutoMate™ Objects such as the task object and CurrentTask so that scripts can access AutoMate™ tasks from within the basic language. New IDE as a separate and more robust EXE file.

Better Late Task and Precision Rescheduling Options

You may now retain originally set times (down to the second) when rescheduling regardless of when the task is launched. Disable if late was also added.

Improved security

AutoMate™ provides several steps that allow tasks to be run in a more secure environment. For example, you can lock the mouse or lock the keyboard from user input while a task is running. You can add steps that prompt a user for a password while the task is running or from being modified, read or executed without authorization. The task-file has also been improved to prevent unauthorized editing and manipulation.

Related Topics:

Welcome to AutoMate™ 4

FTP Quick Send

Use FTP send to upload files through the internet to a specific FTP (File Transfer Protocol) Server. A common use of this functionality would be to automatically upload the latest versions of certain files to a web site.

The Quick means that you do not need to specify a FTPLOGIN step before using the action. Nor do you need to specify a FTPLOGOUT instruction. All the required parameters are contained within this step.

Upon Executing this step AutoMate™ will login to the server you have specified download the file and immediately log out. No further interaction with the FTP server is possible without performing an actual FTP LOGIN.

Related Topics:

FTP Quick Retrieve

Other Available Actions

FTP Quick Retrieve

Use FTP receive to download files through the internet from a specific FTP (File Transfer Protocol) Server. A common use of this functionality would be to automatically download the latest versions of certain files such as a stock update or report.

The Quick means that you do not need to specify a FTPLOGIN step before using the action. Nor do you need to specify a FTPLOGOUT instruction. All the required parameters are contained within this single step.

Upon Executing this step AutoMate™ will login to the server you have specified download the file and immediately log out. No further interaction with the FTP server is possible without performing an actual FTP LOGIN.

Related Topics:

FTP Quick Send

Other Available Actions

LaunchProc

Procedure : LaunchProc

Syntax:
LaunchProc([procedure name])

Example:

LaunchProc(Sample Procedure)

This will launch the sample procedure that is installed in AutoMate™.

Notes: If you use the LaunchProc command to start a procedure that contains one or more scripts, they will not be processed until the procedure started with LaunchProc comes to an end.

Related Topics:

Other Scripting Extensions

Scripting In AutoMate™

Run a BASIC Script Action

ExitProc

(This function has been made obsolete by functions contained in the AutoMate™ System Object . It is provided only for compatibility with previous versions of AutoMate™. Developers should use CurrentTask.Active = 0 upon exiting the script the task will immediately stop executing.)

Procedure : ExitProc

Syntax:
ExitProc

Example:

If not FileExists then ExitProc

This will exit the current AutoMate™ procedure.

Notes: ExitProc completely exits the current running task. It is the same as selecting Task->End Task form the main menu.

Related Topics:

Other Scripting Extensions

Scripting In AutoMate™

Run a BASIC Script Action

ConnectTo

(This function has been made obsolete by functions contained in the AutoMate™ System Object . It is provided only for compatibility with previous versions of AutoMate™.)

Function : ConnectTo

ConnectTo will return a string value depending on whether the internet connection succeeded or failed in the amount of time given.

Syntax:

Result$ = ConnectTo(Providers Name, {Seconds to Wait Before Timing Out})

Results:

Connected : Internet Connection Succeeded

Failed : Internet Connection did not succeed in the amount of time given.

Example:

```
Sub Main
    result = ConnectTo("The Loop", 45) ' Enter your Internet Providers name in the "The Loop" area and then the amount of time in seconds that you want to give the connection before it times out automatically (usually at least 45)
    If result = "Connected" Then ' Check to see whether we are connected
        MsgBox("Connected") ' Yes (you can add your own events here)
    Else
        MsgBox("Failed to Connect") ' No (you can add your own events here - maybe try again?)
    End If
End Sub
```

Notes: The Internet provider's name should be entered exactly as it appears in your Dial-Up Networking Folder. It is not case sensitive. If you are already connected to the Internet, Connected will immediately be returned. If you are already connected to the Internet and your connection was established with a program other than AutoMate™, the function may erroneously report Failed, thus it is a good idea to connect directly to the Internet with AutoMate™ if a decision is to be made dependent on the results of this function.

Related Topics:

Other Scripting Extensions

Scripting In AutoMate™

Run a BASIC Script Action

HangUp

(This function has been made obsolete by functions contained in the AutoMate™ System Object . It is provided only for compatibility with previous versions of AutoMate™.)

Procedure : HangUp

HangUp will disconnect from a Dial Up Networking (DUN) or RAS connection to the Internet or to another Computer.

Syntax:

HangUp

Example:

If x = 5 then HangUp

Related Topics:

Other Scripting Extensions

Scripting In AutoMate™

Run a BASIC Script Action

IsWindowPresent

(This function has been made obsolete by functions contained in the AutoMate™ System Object . It is provided only for compatibility with previous versions of AutoMate™.)

Function : IsWindowPresent

IsWindowPresent is a function that returns a true or false result dependent on whether the specified window caption exists and is visible.

Syntax:

Result = IsWindowPresent(Window Caption)

Results:

True : A window exists and is visible with the corresponding caption.

False : No window exists or is visible with the corresponding caption.

Example:

```
Sub Main
    If IsWindowPresent(AutoMate™) then msgbox(Found a window with a caption of AutoMate™)
End Sub
```

Related Topics:

Other Scripting Extensions

Scripting In AutoMate™

Run a BASIC Script Action

AutoMate™ System Object

Description

The AMSystem object contains utility functions that can be used within AutoMate™ Basic scripts. AutoMate™ automatically creates an instance of AMSystem called AutoMate™. You should not need to create additional instances of the AMSystem object within your AutoMate™ Basic scripts.

Properties

TaskCount	Read Only	Integer	The number of currently executing AutoMate™ tasks.
TaskList	Read Only	Variant	A listing of the currently executing AutoMate™ tasks. Each task is separated by a carriage-return.
LastRASErrorCode	Read Only	Integer	The error number resulting from the last RAS operation. This property will contain 0 if the last RAS operation was successful.
LastRASErrorText	Read Only	Variant	The error text resulting from the last RAS operation. This property will contain an empty string if the last RAS operation was successful.
WindowsDir	Read Only	Variant	The name and path of the Windows directory.
System Dir	Read Only	Variant	The name and path of the Windows System directory.

NOTE: The TaskCount and TaskList properties are static values. Although they will accurately represent the number and names of running AutoMate™ tasks when your script begins executing, during execution, the number and names of running tasks may change. The TaskCount and TaskList properties will not be updated automatically.

Methods

Function FindWindowByTitle(Title As Variant) As Integer

Retrieves the window handle of the top-level application window with the specified title. Returns 0 if no window with the specified title exists.

Function FindWindowByClass(ClassName As Variant) As Integer

Retrieves the window handle of the top-level application window with the specified window class. Returns 0 if no window with the specified window class exists.

Function FindWindowContaining(Title, ClassName As Variant) As Integer

Retrieves the window handle of the top-level application window or associated child window that partially matches the specified window title or classname. Returns 0 if no window with the specified window title or class exists or partially matches.

Function RASConnect(ConnectTo As Variant) As Integer

Attempts to establish a remote access connection using the connection parameters in the specified phonebook entry specified by ConnectTo. If the connection is successfully established, the function returns 0. A non-zero value indicates an error, and sets the LastRASErrorCode property to the RAS Error Code (see below).

Sub RASDisconnect

RasDisconnect is now obsolete. Use RasDisconnectEX instead.

Function RASDisconnectEx(ConnectTo As Variant) As Integer

Disconnects the connection associated with the phonebook entry name specified by ConnectTo. Returns 0 if successful, and non-zero if an error occurs. A non-zero return value indicates the RAS error from the chart below.

Error Code and Error Text for RASConnect and RASDisconnect

143 Busy performing current action.
200 Can't load the RASAPI library.
201 Can't access the functions of RASAPI library. Corrupted file?'
600 An operation is pending.
601 The port handle is invalid.
602 The port is already open.
603 Caller's buffer is too small.
604 Wrong information specified.
605 Cannot set port information.
606 The port is not connected.
607 The event is invalid.
608 The device does not exist.
609 The device type does not exist.
610 The buffer is invalid.
611 The route is not available.
612 The route is not allocated.
613 Invalid compression specified.
614 Out of buffers.
615 The port was not found.
616 An asynchronous request is pending.
617 The port or device is already disconnecting.
618 The port is not open.
619 The port is disconnected.
620 There are no endpoints.
621 Cannot open the phone book file.
622 Cannot load the phone book file.
623 Cannot find the phone book entry.
624 Cannot write the phone book file.
625 Invalid information found in the phone book file.
626 Cannot load a string.
627 Cannot find key.
628 The port was disconnected.
629 The port was disconnected by the remote machine.
630 The port was disconnected due to hardware failure.
631 The port was disconnected by the user.
632 The structure size is incorrect.
633 The port is already in use or is not configured for Remote Access dial out.

634 Cannot register your computer on on the remote network.
635 Unknown error.
636 The wrong device is attached to the port.
637 The string could not be converted.
638 The request has timed out.
639 No asynchronous net available.
640 A NetBIOS error has occurred.
641 The server cannot allocate NetBIOS resources needed to support the client.
642 One of your NetBIOS names is already registered on the remote network.
643 A network adapter at the server failed.
644 You will not receive network message popups.
645 Internal authentication error.
646 The account is not permitted to logon at this time of day.
647 The account is disabled.
648 The password has expired.
649 The account does not have Remote Access permission.
650 The Remote Access server is not responding.
651 Your modem (or other connecting device) has reported an error.
652 Unrecognized response from the device.
653 A macro required by the device was not found in the device .INF file section.
654 A command or response in the device .INF file section refers to an undefined macro.
655 The 'message' macro was not found in the device .INF file secion.
656 The 'defaultoff' macro in the device .INF file section contains an undefined macro.
657 The device .INF file could not be opened.
658 The device name in the device .INF or media .INI file is too long.
659 The media .INI file refers to an unknown device name.
660 The device .INF file contains no responses for the command.
661 The device .INF file is missing a command.
662 Attempted to set a macro not listed in device .INF file section.
663 The media .INI file refers to an unknown device type.
664 Cannot allocate memory.
665 The port is not configured for Remote Access.
666 Your modem (or other connecting device) is not functioning.
667 Cannot read the media .INI file.
668 The connection dropped.
669 The usage parameter in the media .INI file is invalid.
670 Cannot read the section name from the media .INI file.
671 Cannot read the device type from the media .INI file.
672 Cannot read the device name from the media .INI file.
673 Cannot read the usage from the media .INI file.
674 Cannot read the maximum connection BPS rate from the media .INI file.
675 Cannot read the maximum carrier BPS rate from the media .INI file.
676 The line is busy.
677 A person answered instead of a modem.
678 There is no answer.
679 Cannot detect carrier.
680 There is no dial tone.

681 General error reported by device.
682 ERROR_WRITING_SECTIONNAME
683 ERROR_WRITING_DEVICETYPE
684 ERROR_WRITING_DEVICENAME
685 ERROR_WRITING_MAXCONNECTBPS
686 ERROR_WRITING_MAXCARRIERBPS
687 ERROR_WRITING_USAGE
688 ERROR_WRITING_DEFAULTOFF
689 ERROR_READING_DEFAULTOFF
690 ERROR_EMPTY_INI_FILE
691 Access denied because username and/or password is invalid on the domain.
692 Hardware failure in port or attached device.
693 ERROR_NOT_BINARY_MACRO
694 ERROR_DCB_NOT_FOUND
695 ERROR_STATE_MACHINES_NOT_STARTED
696 ERROR_STATE_MACHINES_ALREADY_STARTED
697 ERROR_PARTIAL_RESPONSE_LOOPING
698 A response keyname in the device .INF file is not in the expected format.
699 The device response caused buffer overflow.
700 The expanded command in the device .INF file is too long.
701 The device moved to a BPS rate not supported by the COM driver.
702 Device response received when none expected.
703 ERROR_INTERACTIVE_MODE
704 ERROR_BAD_CALLBACK_NUMBER
705 ERROR_INVALID_AUTH_STATE
706 ERROR_WRITING_INITBPS
707 X.25 diagnostic indication.
708 The account has expired.
709 Error changing password on domain. The password may be too short or may match a previously used password.
710 Serial overrun errors were detected while communicating with your modem.
711 RasMan initialization failure. Check the event log.
712 Biplex port initializing. Wait a few seconds and redial.
713 No active ISDN lines are available.
714 No ISDN channels are available to make the call.
715 Too many errors occured because of poor phone line quality.
716 The Remote Access IP configuration is unusable.
717 No IP addresses are available in the static pool of Remote Access IP addresses.
718 Timed out waiting for a valid response from the remote PPP peer.
719 PPP terminated by remote machine.
720 No PPP control protocols configured.
721 Remote PPP peer is not responding.
722 The PPP packet is invalid.
723 The phone number including prefix and suffix is too long.
724 The IPX protocol cannot dial-out on the port because the machine is an IPX router.
725 The IPX protocol cannot dial-in on the port because the IPX router is not installed.
726 The IPX protocol cannot be used for dial-out on more than one port at a time.

727 Cannot access TCPCFG.DLL.
728 Cannot find an IP adapter bound to Remote Access.
729 SLIP cannot be used unless the IP protocol is installed.
730 Computer registration is not complete.
731 The protocol is not configured.
732 The PPP negotiation is not converging.
733 The PPP control protocol for this network protocol is not available on the server.
734 The PPP link control protocol terminated.
735 The requested address was rejected by the server.
736 The remote computer terminated the control protocol.
737 Loopback detected.
738 The server did not assign an address.
739 The authentication protocol required by the remote server cannot use the Windows NT encrypted password. Redial, entering the password explicitly.
740 Invalid TAPI configuration.
741 The local computer does not support encryption.
742 The remote server does not support encryption.
743 The remote server requires encryption.
744 Cannot use the IPX network number assigned by remote server. Check the event log.
745 ERROR_INVALID_SMM
746 ERROR_SMM_UNINITIALIZED
747 ERROR_NO_MAC_FOR_PORT
748 ERROR_SMM_TIMEOUT
749 ERROR_BAD_PHONE_NUMBER
750 ERROR_WRONG_MODULE
751 Invalid callback number. Only the characters 0 to 9, T, P, W, (,), -, @, and space are allowed in the number.
752 A syntax error was encountered while processing a script.

Related Topics:

Other Scripting Extensions

Scripting In AutoMate™

Run a BASIC Script Action

Logout

The logout function shuts down all processes running on the system under the context of the currently logged on user, and logs the current user off of the system. After logout, the Windows logon screen, will appear and wait for the next user.

NOTE: After a system Logout, AutoMate™ will no longer be running unless you are running the Windows NT service version which runs as a system service and does not require a user to be logged on.

Related Topics:

Shutdown Windows

Reboot Action

Quit AutoMate™

Other Available Actions

Copy File Action

Copies one or more files you specify from source to destination. Wildcards are supported.

NOTE: Be aware when using wildcards that specify c:\windows*.* as the source, it will also copy the sub-directories in the Windows directory if there are any (following Windows, not DOS, standards). As of this writing, there is no solution to this behavior except to use a legacy batch file.

Related Topics:

Move a file

Rename a file

Delete a file

Create a Directory

Remove a directory

Change Directory

Other Available Actions

Move File Action

Moves one or more files you specify from source to destination. The same as a copy operation except the source entry is deleted after the copy operation. Wildcards are supported.

NOTE: Be aware when using wildcards that specifying c:\windows*.* as the source would also move the sub-directories in the Windows directory if there were any (following Windows, not DOS, standards). As of this writing, there is no solution to this behavior except to use a legacy batch file.

Related Topics:

Copy a file

Rename a file

Delete a file

Create a Directory

Remove a directory

Change Directory

Other Available Actions

Rename File Action

Renames the file specified in source to the name specified in destination. Use fully qualified pathnames for most reliable operation.

Related Topics:

Copy a file

Move a file

Delete a file

Create a Directory

Remove a directory

Change Directory

Other Available Actions

Delete File Action

Deletes the file specified. Use fully qualified pathnames for most reliable operation. Wildcards are supported.

Related Topics:

Copy a file

Move a file

Rename a file

Create a Directory

Remove a directory

Change Directory

Other Available Actions

Create a Directory Action

Deletes the file specified. Use fully qualified pathnames for most reliable operation.

Related Topics:

Copy a file

Move a file

Rename a file

Delete a file

Remove a directory

Change Directory

Other Available Actions

Remove a Directory Action

Deletes the directory specified. Use fully qualified pathnames for most reliable operation.

Related Topics:

Copy a file

Move a file

Rename a file

Delete a file

Create a directory

Change Directory

Other Available Actions

Change Directory Action

Changes the current directory for the running task. Use this command only when necessary, otherwise utilize fully qualified pathnames such as c:\windows\desktop\myfile.txt when performing file I/O actions.

Related Topics:

Copy a file

Move a file

Rename a file

Delete a file

Create a directory

Remove a Directory

Other Available Actions

Loop -- Flow Control Action

Causes AutoMate™ to repeat a set of or all of the steps in a task. Specify how many times to loop. To adjust what steps are contained in the loop, simply move the BEGIN loop and END LOOP up or down in the Step Builder . If there are any steps after the loop, AutoMate™ will execute them after performing the loop operation the designated number of times.

Related Topics:

Scripting in AutoMate™

Other Available Actions

Maximize Action

Used to maximize any sizable window that is active on the system. Simply specify the title (or partial title) of the window to maximize.

Maximizing a window can be useful for fixing the coordinates of certain controls on a window when playing back recordings.

Related Topics:

Minimize

Restore

Close

Hide

Unhide

Find Text

Other Available Actions

Minimize Action

Used to maximize any minamizable (i.e., Message Boxes and certain dialog boxes that can be minimized) window that is active on the system. Simply specify the title (or partial title) of the window to minimize.

Related Topics:

Maximize

Restore

Close

Hide

Unhide

Find Text

Other Available Actions

Restore Action

Used to restore a window that is either minimized or maximized to its normal size. Simply specify the title (or partial title) of the window to restore.

Related Topics:

Maximize

Minimize

Close

Hide

Unhide

Find Text

Other Available Actions

Hide Action

Used to hide a window that is open on the system. Simply specify the title (or partial title) of the window to hide. Please note that an application launched from within AutoMate™ can also be hidden at launch time using the **Hide** option on the Start An Application action.

To unhide a window once it has been hidden, please use the Unhide action.

Related Topics:

Maximize

Minimize

Restore

Close

Unhide

Find Text

Other Available Actions

Unhide Action

Used to make visible, a window that is hidden and/or has its visible property set to false. The window specified must be open on the system but not visible. You should only need to use this command to Unhide windows you have previously hidden using the hide action, all other windows that are hidden are probably set that way for a reason.

Related Topics:

Maximize

Minimize

Restore

Close

Hide

Find Text

Other Available Actions

Close Action

Used to close a window on the system matching the title (or partial title) you specify. You may want to consider whether it would be better to send keystrokes to the application to use its File | Exit capability. (usually sendkeys %FX~)

Related Topics:

Maximize

Minimize

Restore

Hide

Unhide

Find Text

Other Available Actions

Focus Window Containing Action

This action searches for a Window on the system and focuses it either on title of the window and text in the dialog such as a button or label. This is extremely useful when dealing with an application that has the same title bar caption across the application, but you only want to wait for a certain dialog box. The only way to distinguish the desired window sometimes to is to specify unique text **in** the dialog box or window such as a button or a label caption. This function searches top-level windows and ALL children windows.

Related Topics:

Focus

Maximize

Minimize

Restore

Close

Hide

Unhide

Other Available Actions

Send Message Action

Used to send an Internet standard e-mail message via the specified SMTP server. This option is useful for notifying a system administrator via e-mail of the task status and whether it completed or not. As with most actions in AutoMate™, it is also accessible from within the AutoMate™ BASIC scripting language in the **Actions** object if you need the sent message to be or variable content as opposed to static text.

Multiple recipients can be supplied through the To: or Cc: parameters by separating the valid email address with a semi-colon or comma.

Currently, the action supports only one attachment per email message. To send multiple attachments, you must use multiple Send Message actions.

Related Topics:

Scripting in AutoMate™

Other Available Actions

Lock Keyboard Action

Using this action, you can prevent keystrokes received from the keyboard from being sent to any windows on your system. Keystrokes are ignored until an Unlock Keyboard action is processed, or until the task ends. You can use this action to prevent interruption to tasks, and to your workstation as a whole, while an AutoMate™ task is working.

Special Notes:

- this action prevents keystrokes globally, which means that the keyboard cannot be used by any application until the keyboard is unlocked or the task ends

- locking the keyboard also prevents AutoMate™ from sending keystrokes as well. This means that a Send Keystrokes action will fail unless the keyboard is unlocked first.

- because the action imposes a global system keyboard lock, a task that locks the keyboard may prevent another task from sending keystrokes. To prevent this, be sure the keyboard is in a known state before launching tasks that require a Send Keystrokes action

Related Topics:

Unlock Keyboard Action

Lock Mouse Action

Unlock Mouse

Password Action

Other Available Actions

Lock Mouse Action

Using this action, you can prevent mouse clicks from being processed by Windows. Clicking any mouse button will generate now response from the workstation an Unlock mouse action is encountered, or until the task ends. You can use this action to prevent interruption to tasks, and to your workstation as a whole, while an AutoMate™ task is working.

Special Notes:

- this action prevents mouse clicks globally, which means that the mouse buttons cannot be used by any application until the mouse is unlocked or the task ends

- locking the mouse also prevents AutoMate™ from moving or clicking the mouse as well. This means that a Record Mouse Movements action will fail unless the mouse is unlocked first.

- because this action imposes a global system mouse lock, a task that locks the mouse may prevent another task from playing recorded mouse movements. To prevent this, be sure the mouse is in a known state before launching tasks that require a Record Mouse Movements action.

Related Topics:

Unlock Mouse

Unlock Keyboard Action

Lock Keyboard Action

Password Action

Other Available Actions

Unlock Keyboard Action

This action will unlock a keyboard that was previously locked by AutoMate's Lock Keyboard Action, thus allowing keyboard input to once again be processed by Windows. Note that AutoMate™ will automatically unlock a locked keyboard when a task ends, regardless of whether or not an Unlock Keyboard step is present.

Related Topics:

Unlock Mouse

Lock Mouse Action

Lock Keyboard Action

Password Action

Other Available Actions

Unlock Mouse

This action will unlock a mouse that was previously locked by AutoMate's Lock Mouse Action, thus allowing button clicks to once again be processed by Windows. Note that AutoMate™ will automatically unlock a locked mouse when a task ends, regardless of whether or not an Unlock Mouse step is present.

Related Topics:

Unlock Keyboard Action

Lock Mouse Action

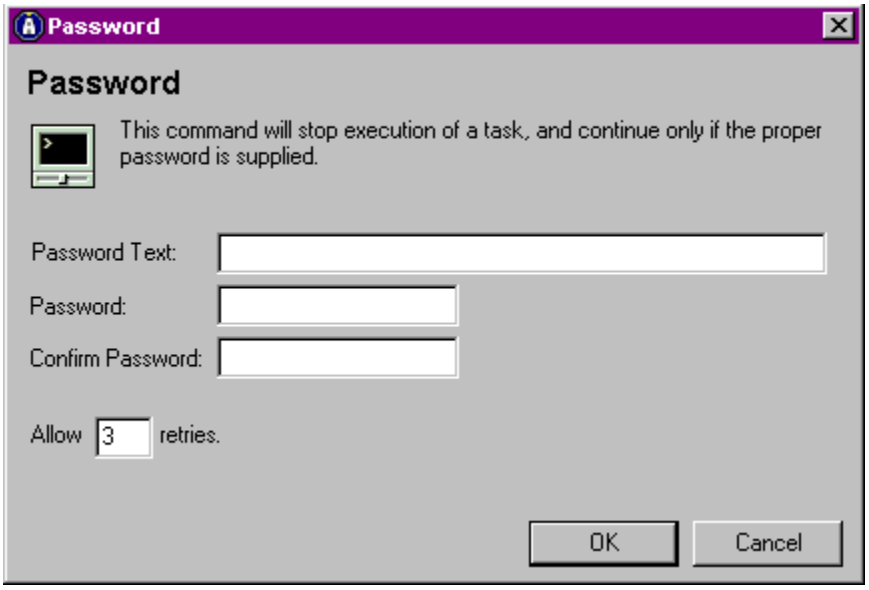
Lock Keyboard Action

Password Action

Other Available Actions

Password Action

The password action presents a dialog box that requests a user at the workstation to enter a password before task execution will proceed. The password can be any text the task designer wishes. While the password box is displayed, the task is put into a holding state, and execution of that task stops. If a user enters an incorrect password more times than allowed, the task fails itself and the task stops.



Related Topics:

Lock Mouse Action

Lock Keyboard Action

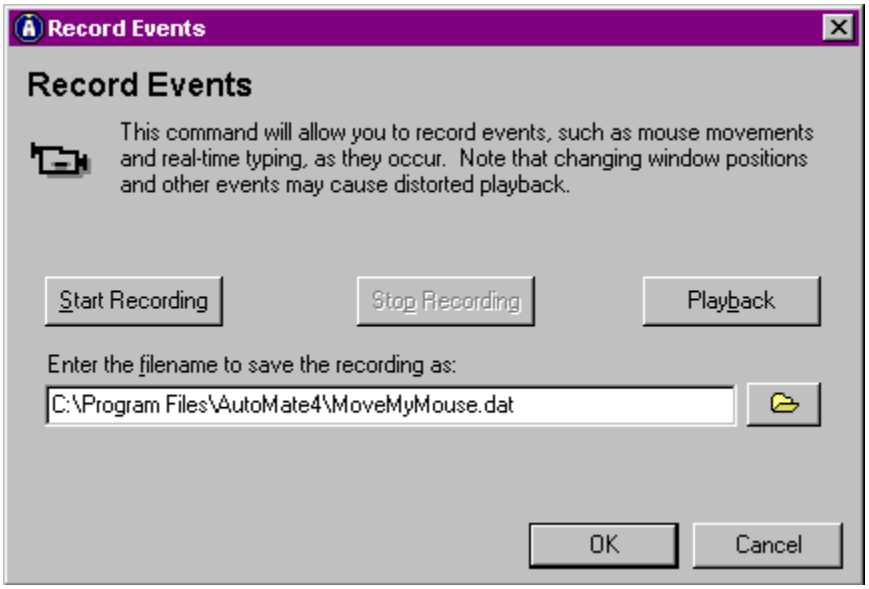
Unlock Mouse

Unlock Keyboard Action

Other Available Actions

Record Events

This action allows you to record the movements of your mouse or your keystrokes to a file, which can then be played back through a task to automate mouse movements. You can, for example, record your mouse movements as you double-click an icon, and then play these mouse movements back through AutoMate™, which would move the pointer on your screen as if you were moving the mouse physically.



To record your mouse movements, first enter a filename and path of the file you would like to use to write the mouse movements to. Then, click the Start Recording button. AutoMate™ will minimize itself, and start recording the movements of your mouse and the keys you press. When you have finished, press Ctrl-Alt-F12. AutoMate™ will reappear, and your movements will have been written to the file.

To verify your movements, you can click the Playback button, which will playback your mouse movements.

Special Notes:

- if you have a step in your task that locks the keyboard or locks the mouse, you must ensure that they are unlocked before the Record Events step is encountered, otherwise the mouse will not reproduce your mouse clicks or keystrokes.

Related Topics:

Unlock Keyboard Action

Unlock Mouse

Lock Mouse Action

Lock Keyboard Action

Other Available Actions

Password Text

The descriptive text that appears in the dialog box.

Password

A user-defined password that must be entered before the password dialog box will be cleared and task execution can continue.

Confirm Password

Type the password in again here to confirm that you have typed the password in correctly.

Start Recording

Minimizes Settings and records the movements of your mouse and your keystrokes to the filename specified below. Pressing Ctrl-Alt-F12 stops recording.

Stop Recording

Stops the current recording session. This can also be done by pressing Ctrl-Alt-F12 on your keyboard.

Playback

Playback the mouse movements that were recorded into the file specified below.

Browse

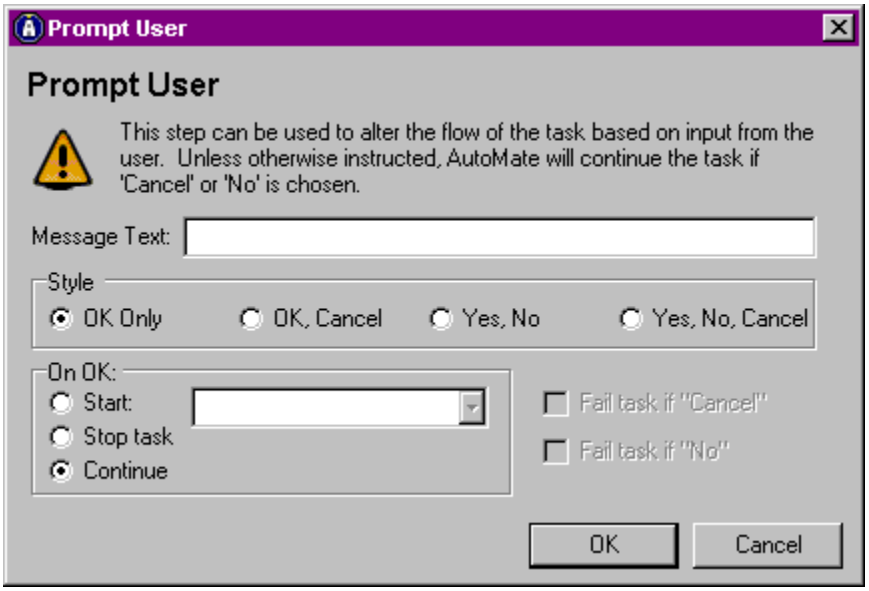
Displays an Explorer-like window from where you can select a file.

Record Movements Filename

Specifies the filename where the mouse movements will be recorded to. This must be provided before you can successfully record your mouse movements.

Prompt Action

The prompt action is used to display a dialog box with a custom message and format, and control the flow of the task based on the user's selection. When the step is executed, a Windows-like messagebox appears with a question-mark icon, and the task pauses until the user clicks a button. Based on the button that is selected and the configuration of the step, the task can stop, fail, continue or start another task.



To setup the step, type in the text you wish to appear in the dialog box. Select which buttons you wish to be available by clicking an option in the Styles section.

The On OK area sets up how the task will behave if the user clicks the Yes or OK button when the dialog appears during task execution. If Start is selected, AutoMate™ will stop the currently executing task and begin the task specified in the box to the right. If Stop Task is selected, AutoMate™ will stop the current task with a message in the Log File stating that the task has ended programmatically. (This is different than a task *failing*, which is written to the Log File as an error.) If Continue is selected, the task proceeds from that point normally.

The Fail on Cancel and Fail On No checkboxes indicate whether or not the task should fail if the user selects the Cancel and/or No buttons (depending on which are available).

Related Topics:

Other Available Actions

Message Text

This is the text that will appear in the messagebox that appears. A question mark icon is used as default.

Style

Specifies which buttons will appear on the messagebox.

Task To Start

This is used to select which task AutoMate™ will start if the user clicks OK. This is only available if the "On OK:" option is set to "Start:". You can enter your own task name, or click to reveal a list of available tasks.

On OK:

This selects how control is returned to the task after the user selects OK.

Fail On Cancel

Check this off if you wish for the task to fail when the user clicks the "Cancel" button.

Fail On No

Check this off if you wish for the task to fail if the user clicks "No".

Security

AutoMate™ contains many features aimed at security throughout the application. Each task can be individually password protected to prevent either read access or read/write access to a task. This prevents unauthorized editing or viewing of a task's contents in a multi-user environment. You can even protect the task from being launched manually through the Configuration Manager; a feature useful for time-sensitive task launching.

New actions have also been added to the Step Builder to tighten security. The new Password Action will interrupt task execution at a user-defined point and prompt for a password. If the incorrect password is entered a specified number of times, the task will fail. There are even actions to lock the workstation from keyboard and/or mouse input for a block of steps.

For even more added security, the task file itself can no longer be edited manually using a text editor, as was the case with previous versions. Only AutoMate™ itself can modify a task file. This was done to prevent possibly malicious task modifications being made directly to the task file and defeating the task password protection. If AutoMate™ detects an attempt to defeat the task file's security, it will erase every password field of each step of each task within the task file, while leaving the task-specific passwords intact.

Remember, however, that the best security is accomplished through common sense. If security is a major concern, choose your steps and tasks properly, and take every necessary precaution on your system to make it as difficult as possible for others to access sensitive parts of your system.

Password Protecting Tasks

You can password protect a task with a user-defined password to prevent the task from being modified or even read without authorization. To do this, go to the Tasks Properties, and select the Security tab. From within, select Password protect this task, and type the password you wish to use in the two spaces provided.

You can also choose to allow the task to be viewed, but not modified, by checking off the Allow read access check box.

To prevent a user from manually launching the task from Settings without first specifying the proper password, check off the Require password to launch task manually button.

If you want to prevent a task from launching in general without first entering a password, insert a Password step into your task using the Step Builder .

Related Topics:

Security

Step Builder

Password Action

AutoMate™ Task File

A data file that AutoMate™ uses to store information about the tasks to be processed and automated.

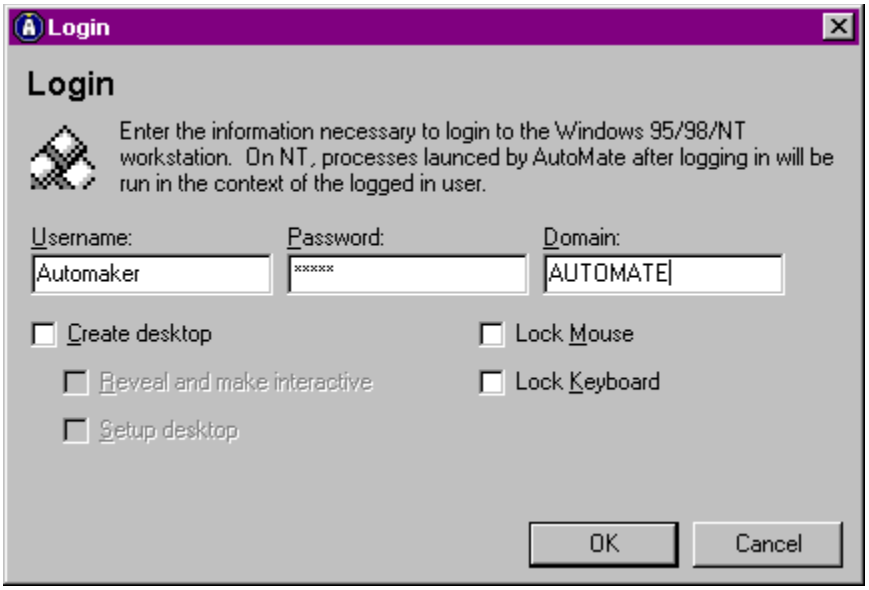
Login Action

NT Service Edition ONLY

The Login action allows AutoMate™ to log on a user into a Windows NT domain, and pass control of a task to that user, giving the task the security and access privileges of that user. This allows tasks to act on behalf of administrators for system-specific actions and tasks.

Each task that runs through the NT Service Edition must be performed on behalf of an active user on a domain. Otherwise, AutoMate™ will not have access to network or system resources that it may need to perform its tasks. The Login Action performs the login necessary to give the task its required privileges. This step should be the first step of any task that will be run by the Service Edition.

To setup the step, enter the username, password and domain of a user that has the access privileges required to successfully carry out the task. This should be lowest set of security rights possible. This will limit the susceptibility your system may encounter if it is running a task on a logged out workstation, or if the task is somehow interrupted by a user. See the NT Specific Security Notes for more information about running secure tasks on an NT workstation or server.



You can automatically lock the mouse and/or keyboard after a successful login by checking off the Lock Mouse and Lock Keyboard check boxes. This eliminates the need for a Lock Mouse Step or Lock Keyboard Step after the login step.

The remaining check boxes of the Login Step are used when you wish to create a task capable of running on a logged off workstation. Check Create Desktop to have AutoMate™ create a new, virtual desktop that the task can use to start its applications on. If you wish to send keystrokes to this new desktop, you must also check Reveal and make interactive. This will give AutoMate™ interactive access to the new desktop. If you require an Explorer interface (i.e. a fully functional desktop with a taskbar and icons), select Setup desktop.

A Login Step should be followed by a Logout step if you have created a desktop in the login command (by

checking the Create Desktop checkbox), since logging out cleanly destroys the desktop and reestablishes the previous environment.

NOTE: You can also set up a default user profile for the NT Service Edition. This will allow the Task Service to assume a certain user if a Login step is not found for a task.

Related Topics:

NT Service Edition

Security

NT Specific Security Notes

NT Service Edition

A version of AutoMate™ that runs as an NT Service on NT workstations and servers. Now available as a separate product from Unisyn Software.

NT Specific Security Notes

The NT Service Edition's security model was structured around presuming restricted access for the tasks AutoMate™ launches. This means that, unless told otherwise through the use of a LOGIN step or by means of the default user profile, AutoMate™ will not launch a task through the service module. This is because, by default, all services run in the context of a "SYSTEM" account, which has access to sensitive areas of the NT environment. Without establishing a security or user context for a task to operate in, AutoMate™ would present itself as security hole to the NT Operating System. Rather than risk a breach of security, AutoMate™ puts the onus on the user to establish the correct security context for the task before the task starts executing.

Setting up a security context is usually best achieved by selecting a user that has just enough access rights to perform the actions required by the task, and logging them onto the system by using the "LOGIN" step from the Step Builder . This should be the first step of your task. It will allow the task to have access privileges identical to that of the user you selected, including file privileges and network access. The last step of your task should be a LOGOUT step , which will cleanly log the user out. Note that using these steps will not affect a user that is currently logged onto the workstation, if there is one; only the task and the steps contained within are affected by any LOGIN and LOGOUT steps.

Alternatively, you can set up a default user profile in the Preferences window of the Settings program. If a "LOGIN" step is not found as the first step of a task, AutoMate™ will default to this default user and attempt to log them in. If successful, the task will continue in the context of the default user. If unsuccessful, the task will fail, and the reason for the failure will output to the Log File (viewable by clicking on the "View Log" button on the Settings screen).

In the Step Builder itself, LOGIN steps cannot be copied or pasted between tasks. While this may be inconvenient, it prevents an unauthorized user from gaining access to an administrator task, for example, copying out their LOGIN line, and pasting it into their own, thus creating a task with admin privileges that does what they want. LOGIN steps must therefore be entered manually, with the correct password, with each use.

AutoMate™ has also added some general security enhancements to all versions to increase security.

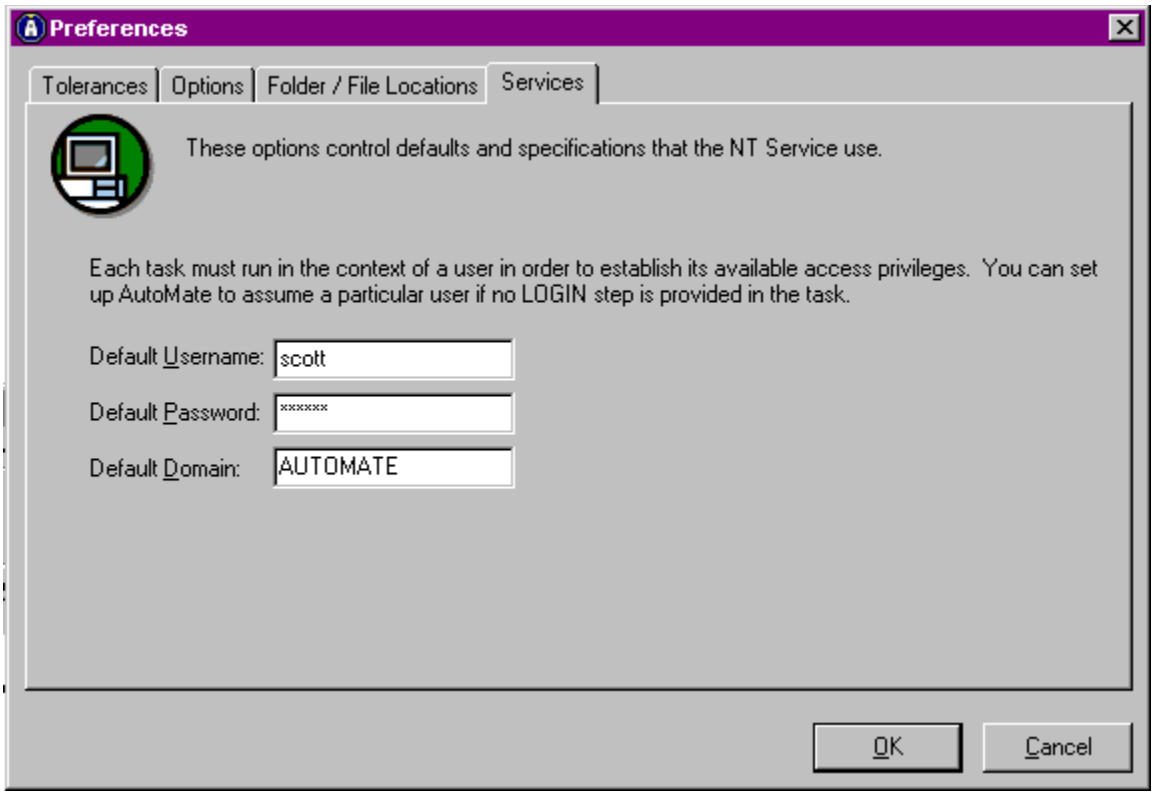
There are, however, a few other security issues of which you should be aware. Although AutoMate™ can create desktops and launch interactive tasks on logged out workstations, doing so should be done very carefully. If a user were to walk by a workstation that was performing an automated task that was logged in with powerful privileges (an admin account, for example), they could take over control from AutoMate™ and gain access to the system with the same privileges that AutoMate™ had. Careful selection of your user accounts, as well as proper use of the "Lock Keyboard" and "Lock Mouse" actions minimize the chances of this happening. Also, be aware that the only reliable way to launch a task on a logged out workstation is by a schedule, and that the only way to disable a scheduled task is to log in to the workstation and either pause or terminate the AutoMate™ service.

Setting Up A Default User Account

NT Service Edition ONLY

Because each task that is run in the NT Service Edition must be performed on behalf of an active user on a network, each task should have a step that logs a user in before the rest of the steps are executed. This can be done by placing a Login step at the beginning of each task.

But a faster way is to setup a default user profile, which AutoMate™ uses to assume a user to log in if a Login step is not present in a task. Unisyn recommends you setup a default user account with very minimal privileges (for example, using an account that has basic User group rights). This will allow most users to accomplish their needs without have to remember to enter a Login step for each task, while providing the workstation with the right level of security.



The default user can always be overridden by adding a Login step as the first step of a task, and entering in the necessary user information.

Refer to the NT Specific Security Notes for more information on default users, and general security information.

Retries

Specifies the number of times a user can enter a wrong password before the task will fail.

Tasks And Logged Out Workstations

One of the most impressive features of AutoMate™ 4 NT Service Edition is its ability to start tasks even when a workstation is logged out. Here is how you would go about doing so.

If you have not already done so, go to the Step Builder by either creating a new task or by editing an existing one (see online help for more specific instructions on how to accomplish this).

If you want the task to be interactive (that is, the task will be able to be seen and manipulated in the foreground - necessary for sending keystrokes), the first step MUST be a LOGIN step. You should only choose to make a task interactive if it is absolutely necessary that you be required to send keystrokes or mouse movements in the task. Recall that interactive tasks are of a higher security threat than non-interactive tasks. If the task is not required to be interactive, and your default user account is sufficient to run the task, you do not need a LOGIN step as your first step.

For tasks that require interactive actions, check off "Create Desktop" and "Reveal And Make Interactive" options of the LOGIN action. This will create a new desktop for your task to appear on. If you need a taskbar and icons, you should also select "Setup Desktop". This will launch the explorer interface, and setup the Active Desktop (if installed on your system). When the task starts, a new desktop similar to a login session will appear, and the task will begin to work on its own desktop. When the task is complete, the desktop will disappear and the login screen will reappear.

Security Considerations

Interactive logged-out task sessions are high security risks for your NT workstation, especially if the task you are running is logged in as a user with privileged access. An interactive desktop acts exactly as a typical user session would if a human were at the keyboard. This means anyone could interrupt AutoMate™ while it was doing task processing, interrupt that task, and take control of the system, assuming the identity of the logged in user.

This is analogous to someone strong-arming a human away from their computer and taking control of the workstation.

Because of this, you should be cautious about what tasks your perform on a logged out workstation, which user you log-in to perform the task, and take careful consideration of how long and when the task will occur. Make use of the LOCK KEYBOARD and LOCK MOUSE actions (keeping mind that your keyboard and mouse must be unlocked if you wish to send keystrokes or recorded mouse movements in your task -- see the example task "Interactive Example" in AutoMate™).

Note also that, because of the Windows NT security architecture, key-presses are normally locked out at a "Press Ctrl-Alt-Delete To Log-On" screen. Therefore, hotkeys will not work as a task trigger when the workstation is logged out. Scheduling a task to run is usually the only effective way to start an Interactive Logged-out Task.

Setting Up A Task

NT Service Edition ONLY

Creating tasks in the NT Service Edition is a little different than what you may be used to in other versions of AutoMate™. The primary difference is the requirement of a LOGIN step, either implicitly or explicitly entered into the task.

If you have setup a "Default User Account", then any task that does not have a LOGIN step as its first step will login in the default user, and start task execution from there. The steps will be performed as if the account represented by the default user was doing them. For example, guests normally do not have access to any administrator tools, such as Disk Administrator. If you setup the Default User Account as the guest account, and did not LOGIN in the administrator account as the first line of your task, then a step of that task set to launch Disk Administrator would fail, because guests cannot start that application.

If you have not set up a Default User Account, and do not have a LOGIN step as the first step of a task, the task will fail. This is because AutoMate™ will not start a task unless it has been told whom the task is running on behalf of. This prevents security breaches.

You can also use AutoMate™ to start tasks when the workstation is logged out.

Everything else is the same between the NT Service Edition and the other fine editions of AutoMate™.

Related Topics:

AutoMate™ Actions

Security

NT Specific Security Notes

Tasks And Logged Out Workstations

Username

The username of the user you wish to login. This must be a valid username at the time the task is run.

Password

The password of the user you wish to login.

Domain

The NT Domain that the user is a member of, and that you wish to login to.

Create Desktop

Check this off if you need AutoMate™ to create a virtual desktop to run a task on top of. This must be checked if you want the task to work when a workstation is logged out.

Reveal Desktop

Check this box if you want the newly created desktop to be visible to the user. A desktop on a logged out workstation must be revealed before keystrokes can be sent to it.

Setup Desktop

This will launch an Explorer interface on the newly created desktop.

Lock Mouse

Locks the mouse from button presses after a successful login. This eliminates the need for a "Lock Mouse" step after the Login step.

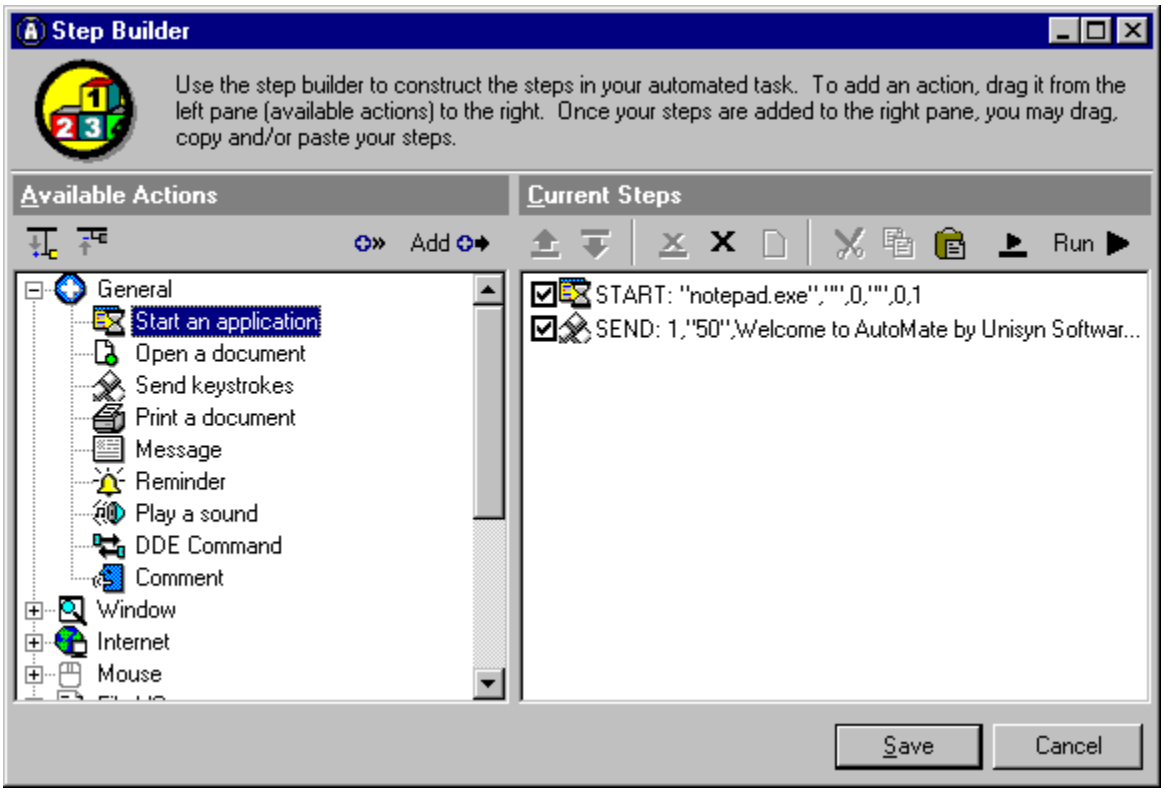
Lock Keyboard

Locks the login session from keyboard input after a successful Login.

Step Builder

The Step Builder is perhaps the most important dialog in AutoMate™. It is used to design the step by step actions you wish AutoMate™ to perform when the task is launched. The task is designed by organizing the available actions into the correct sequence you need to complete the task.

(Click a spot on the figure below for more information)



For example if you need to create a task that reboots the machine, simply expand the plus (+) sign next to the System group in the left pane of available actions, select the Reboot action, and drag it across to the list of Current Steps

The Builder is organized into two major sections: the Available Actions and the Current Steps. The Available Actions section is a list of all the actions AutoMate™ can perform for you.

Moving an action from the Available Actions section to the Current Steps section creates a step. To do this:

1) Expand the group where the desired action is contained (on the left pane)
2) Highlight the action you wish to add to the task (on the left pane)
3) Either Click the Add button, drag the action into the list current steps (on the right pane), or double click the action

Some actions require parameters, like Open a Document Action or Starting an Application, while others do not, such as Reboot Action or Quit AutoMate™. See Actions for a complete explanation of each action and their parameters.

The Step Builder provides a great environment for manipulating steps. You can copy and paste steps from one task to another, or within the same Step Builder by selecting the steps you wish to copy, and pressing Ctrl-C. Then, select the position you wish to paste the steps and press Ctrl-V. (Special note to users of the NT Service Edition: for security reasons, LOGIN steps cannot be pasted or copied; they must always be manually added). You can drag-and-drop tasks from the Available Actions section to the Current Steps section or drag-and-drop steps within the Current Steps section. You can even delete multiple steps by multi-selecting the steps and pressing the Delete key.

For debugging, you can turn off individual steps by unchecking the box to the left of a particular step in the list of current steps (on the right pane). This prevents that specific step from processing when the task is run.

Related Topics:

AutoMate™ Actions

Security

NT Specific Security Notes

New Task Wizard

Triggers

Deployables

Constants

Variables

Available Action

This is the listing of all the actions that can be added as steps to the current task. Actions are categorized based on their general functionality and purpose.

Current Steps

This lists the steps that are currently in the task. When the task is launched, these actions will be performed in the order they appear here.

Add

This adds the currently selected action from the "Available Actions" section as a step to the right. This can also be done by double-clicking the action in the "Available Actions" section.

Remove

This will remove the currently selected step from the "Current Steps" section.

Remove All

This will remove all the steps from the "Current Steps" section.

Test Task/Step

This will execute the currently selected step from the "Current Steps" section of the Step Builder. If no step is selected, AutoMate™ will execute each step from top to bottom.

Move Step Up

This will move the currently selected step from the "Current Steps" section one position upwards.

Move Step Down

This will move the currently selected step from the "Current Steps" section one position downwards.

NT Service Edition Intro

This information is provided as a basic introduction to NT security and user rights. It is by no means complete on either of these issues. For more information see the Windows NT Resource Kit, or other Windows NT books.

1. Windows NT vs. Windows 9x.

One of the differences between NT and Windows 9x is its security model. This is the structure in which Windows determines what a user can and cannot do on the computer station. In Windows 9x the user name and password tells the stand-alone station who is currently using the computer. The Windows 9x user profile also houses information concerning desktop setting and user preferences as well as an identity and access permissions for the network. Network user rights are set up by the network administrator on the server. In this system the user has full control of the 9x station, but is subject to the network permissions set by the administrator.

NT Security, on the other hand, allows users to setup their information and network permissions via the NT workstation from which they are logging on. Also, user accounts can be restricted to specific areas and functions within the individual workstation. All actions that are performed on an NT station must be performed in the context of some user.

2. Services on NT.

Services are special applications for NT that can remain active even when no user(s) is logged on. They can only be installed and disabled by an administrator. You can view the services currently installed on the NT Workstation via the services control panel.

The advantage to NT services is that they run as another user on the system when the system starts. This means that even though a person logs in, does some work and then logs out, the service never stops running. AutoMate's NT service version allows a user to set up a task (like backing up data), log out and still have the task complete itself at the desired time or on a trigger. This AutoMate™ task will still run because the AutoMate™ launcher is a user that is always logged in, therefore NT will allow this task to run.

3. AutoMate™ and NT security.

Because AutoMate™ is always running, it must be a registered user on the system just like all other services. Services log themselves in under a special account called "system". Normally "system" users have access rights to areas in Windows NT that normal user accounts do not. For that reason AutoMate's "system" rights are stripped away so that a general user can not ask AutoMate™ to do things he or she would not normally be able to do.

Because AutoMate™ has no rights, every task must have a user name and password so that NT will allow this task to continue. If you try to launch a task and there is no user defined, NT will ignore any and all requests made by that task. This is why there needs to be a login step in an AutoMate™ task.

There is a default user section under preferences that the admin for the machine can set up so that if no Login step is included in a task, the task will default to this user. See the readme_nt.txt file for more info on this.

4. NT and User Interface

When NT is either:

1) Running a screen saver, or

2) Locked, or
3) Logged out,

NT will hide the user interface of processes that are running. This prevents all keystrokes from going to a window and also prevents mouse clicks. This is why AutoMate™ has included the "Create Desktop" and "Make Interactive" choices in the Login step. There is no need to create a desktop for a task that will be run when there is a user logged in and using the system.

NOTE: When you create a desktop, even if it is for the user that is currently logged into the system, AutoMate™ will create another instance of that user. When doing this you should always use Logout as the last step so that there are not two users on the system.

If you do not use the "Create Desktop" and you include the Logout step, it will logout the current user from the system.

See Also:

Security

NT Specific Security Notes

Setting Up A Default User Account

Tasks And Logged Out Workstations

Login Action

Logout

Setting Up A Task On The NT Service Edition

Mouse To Location Action

This step will change the mouse pointer to any position on the screen that you specify. You can then click a mouse button, or send keystrokes, to interact with the object below the mouse pointer.

The computer screen can be viewed as a large grid, with each position referred to as a pixel. Pixels are given in x and y values, where x is the number of pixels from the left-hand side of the monitor, and the y value is the number of pixels from the top of the monitor. The total number of pixels on screen is the same as the resolution setting of your video card (i.e. 800x600, or 1024x728)

To specify a position, you can enter the x and y values into the "X Pos:" and "Y Pos:" boxes on the Move Mouse to Location settings window. Alternatively, you can simply move the mouse to where you want it and press the "Insert" key on your keyboard, which will place the mouse's current position into the X and Y boxes for you.

Mouse Clicking

AutoMate™ has a number of actions that simulate the clicking of a button on your mouse. In other words, AutoMate™ will click the mouse for you!

The mouse is clicked at the location where the mouse is when the step is encountered. Therefore, you should use a Move To Location step to put the mouse pointer where you want it before issuing one of the click actions.

Dragging the mouse with one of the mouse buttons down can be simulated using a combination of the following steps. First, position the mouse where you wish to begin the drag (do this using the Move To Location action). Then, use one of the “mouse down” actions (Left Mouse Down, for example). Next, use another Move To Location action to move the mouse pointer to the position when the drag should end, and finally issue a “mouse up” action (in our case, a Left Mouse Up).

The click actions are:

Left Click

Clicks the left mouse button once.

Left Double Click

Clicks the left mouse button, wait 250 milliseconds, and then press the left mouse button again (thus simulating a double-click)

Left Mouse Down

Holds down the left mouse button until a “left mouse up” step is encountered.

Left Mouse Up

Releases the left mouse button (this has no effect if a Left Mouse Down step has not been issued)

Right Click

Clicks the right mouse button once (note that this will usually make a "pop-up" menu appear, which will require another "Move Mouse To Location" action, and possibly another click, to simulate clicking the pop-up menu).

Right Double Click

Clicks the right mouse button, wait 250 milliseconds, and then press the right mouse button again (thus simulating a double-click)

Right Mouse Down

Holds down the right mouse button until a “right mouse up” step is encountered.

Right Mouse Up

Releases the right mouse button (this has no effect if a Right Mouse Down step has not been issued)

Middle Click

Clicks the middle mouse button once. If a middle button on the mouse is not available, the step is ignored.

Middle Double Click

Clicks the middle mouse button, wait 250 milliseconds, and then press the middle mouse button again (thus simulating a double-click). If a middle button on the mouse is not available, the step is ignored.

Middle Mouse Down

Holds down the middle mouse button until a “middle mouse up” step is encountered.

Middle Mouse Up

Releases the middle mouse button (this has no effect if a Middle Mouse Down step has not been issued)

These steps do not fail and cause the task to fail. Ever.

Start A Service Action

Works on Windows NT ONLY

This step will start a service already installed on the local NT server or workstation. Task execution pauses while a service is starting and does not continue until AutoMate™ receives notification from the Service Control Manager about the success of the start service command. If the service starts correctly, the task continues. If the service does not start correctly, the step will fail. Most services provide you with more details of their starting status by making an entry in the Windows NT Event Log.

To set up the action, enter in the name of the Service into the "Service To Start" drop-down list. You can also click on the down-arrow of the drop-down list to view a list of services presently installed on the machine. If the service is not in the list, you can still enter in the name of a service that you know will be installed when the task starts.

NOTE: The user running the task (or the user-context provided by the LOGIN step of the task) usually must have Administrator rights or the equivalent to start, stop, add or remove a service on most NT workstations.

See Also:

Pause a Service Action

Continue a Service

Stop a Service Action

Install A Service Action

Remove A Service

Other Available Actions

Install A Service Action

Works on Windows NT ONLY

AutoMate™ can install a service into the Windows NT Service Control Manager using this step. AutoMate™ will attempt to install the service using the information you supply. You can then use the Start A Service step to start the service or, otherwise, control it.

Services usually have specific requirements about how they are installed, started, and/or paused. You should consult the documentation regarding the service you want to install for information about the proper parameters to use when installing it. Also note that AutoMate™ installs the service with the SERVICE_ALL_ACCESS flag, which means that anyone can access the service if it is installed by AutoMate™.

If the service type is Run in own process, the username is the account name in the form of "DomainName\Username", which the service process will be logged on as when it runs. If the account belongs to the built-in domain, ".\Username" can be specified. Services of type Shared Process are not allowed to specify an account other than LocalSystem. If the username field is blank, the service will be logged on as the "LocalSystem" account, in which case, the Password parameter should also be blank.

If the service type is Kernel Driver or File System Driver, this name is the Windows NT driver object name (that is, \FileSystem\Rdr or \Driver\Xns), which the input and output (I/O) system uses to load the device driver. If the username is blank, the driver is run with a default object name created by the I/O system, based on the service name.

NOTE: The user running the task (or the user-context provided by the LOGIN step of the task) usually must have Administrator rights or equivalent to start, stop, add or remove a service on most NT workstations.

See Also:

Start A Service Action

Pause a Service Action

Continue a Service

Stop a Service Action

Remove A Service

Remove A Service

Works on Windows NT ONLY

This step will attempt to remove a service that has been previously installed on the local NT server or workstation. Task execution pauses while AutoMate™ attempts to remove the service, and does not continue until AutoMate™ receives notification from the Service Control Manager. If the service is removed correctly, the task continues. If not, the step will fail. Most services provide you with more details of their status by making entries in the Windows NT Event Log.

To set up the action, enter in the name of the Service to uninstall into the "Service To Remove" drop-down list. You can also click on the down-arrow of the drop-down list to view a list of services presently installed on the machine. If the service is not in the list, you can still enter in the name of a service that you know will be installed when the task starts. If you wish for AutoMate™ to attempt to stop the service before trying to uninstall it, check the Attempt To Stop Service option. Services usually should be stopped before attempting to uninstall them.

NOTE: The user running the task (or the user-context provided by the LOGIN step of the task) usually must have Administrator rights or equivalent to start, stop, add or remove a service on most NT workstations.

NOTE: Not all services can be removed. If the task fails for no apparent reason, check the services documentation to ensure that it is removable.

See Also:

Start A Service Action

Pause a Service Action

Continue a Service

Stop a Service Action

Install A Service Action

Wait For Window To Disappear

Using this step, you can pause task execution until a window with a user-defined title disappears. This is useful in situations where an application displays a message box or window while processing, and you want to make the task wait until that process is complete.

Rather than waiting for a window to disappear you may want to try the pause action .

NOTE: For your convenience, when inserting this step, a list of all open Windows on the system is displayed for quick selection.

Related Topics:

Other Available Actions

Wait for a Window

Stop a Service Action

Works on Windows NT ONLY

This step will stop a service already installed and running on the local NT server or workstation. Task execution pauses while a service is stopping and does not continue until AutoMate™ receives notification from the Service Control Manager about the success of the stop service command. If the service stops correctly, the task continues. If not, the step will fail. Most services provide you with more details of their status by making entries in the Windows NT Event Log.

To set up the action, enter in the name of the Service into the "Service To Stop" drop-down list. You can also click on the down-arrow of the drop-down list to view a list of services presently installed on the machine. If the service is not in the list, you can still enter in the name of a service that you know will be installed when the task starts.

NOTE: The user running the task (or the user-context provided by the LOGIN step of the task) usually must have Administrator rights or equivalent to start, stop, add or remove a service on most NT workstations.

See Also:

Start A Service Action

Pause a Service Action

Continue a Service

Install A Service Action

Remove A Service

Pause a Service Action

This step will pause a service already installed and running on the local NT server or workstation. Task execution pauses while a service is pausing and does not continue until AutoMate™ receives notification from the Service Control Manager about the success of the pause service command. If the service pauses correctly, the task continues. If not, the step will fail. Most services provide you with more details of their status by making entries in the Windows NT Event Log.

To set up the action, enter in the name of the Service into the "Service To Pause" drop-down list. You can also click on the down-arrow of the drop-down list to view a list of services presently installed on the machine. If the service is not in the list, you can still enter in the name of a service that you know will be installed when the task starts.

NOTE: The user running the task (or the user-context provided by the LOGIN step of the task) usually must have Administrator rights or equivalent to start, stop, add or remove a service on most NT workstations.

NOTE: Not all services support pausing. The service must be installed on the system with support for pausing the service in order for this step to succeed.

See Also:

Start A Service Action

Continue a Service

Stop a Service Action

Install A Service Action

Remove A Service

Continue a Service

Works on Windows NT ONLY

This step will attempt to continue a service that has been previously paused on the local NT server or workstation. Task execution pauses while a service is processing the continue command and does not continue until AutoMate™ receives notification from the Service Control Manager. If the service continues correctly, the task continues. If not, the step will fail. Most services provide you with more details of their status by making entries in the Windows NT Event Log.

To set up the action, enter in the name of the Service into the "Service To Continue" drop-down list. You can also click on the down-arrow of the drop-down list to view a list of services presently installed on the machine. If the service is not in the list, you can still enter in the name of a service that you know will be installed when the task starts.

NOTE: The user running the task (or the user-context provided by the LOGIN step of the task) usually must have Administrator rights or equivalent to start, stop, add or remove a service on most NT workstations.

NOTE: Not all services support pausing. The service must be installed on the system with support for pausing the service in order for this step to succeed.

See Also:

Start A Service Action

Pause a Service Action

Stop a Service Action

Install A Service Action

Remove A Service

Copy Action

This action will copy the selected region of text from the currently focused window to the Windows clipboard. This information can then be pasted to another application using the Paste Step, or by pressing CTRL+V.

See Also:

Copy Action

Cut Action

Paste Action

Clear Action

Cut Action

This action will cut the selected region of text from the currently focused window to the Windows clipboard. This information can then be pasted to another application using the Paste Step, or by pressing CTRL+V.

See Also:

Copy Action

Cut Action

Paste Action

Clear Action

Paste Action

This action will paste the information currently on the clipboard into the currently focused window.

See Also:

Copy Action

Cut Action

Paste Action

Clear Action

Clear Action

This action will clear the contents of the Windows clipboard.

See Also:

Copy Action

Cut Action

Paste Action

Clear Action

Start A Website

This action will go to the website you choose. AutoMate™ will attempt to use the default web browser on your system. If a web browser is not already started, AutoMate™ will start the browser first, and then connect to the website you supplied in the parameters.

Related Topics:

Other Available Actions

Auto Keys Trigger

Using the AutoKeys Trigger AutoMate™ can launch a task based on text that is typed into the system regardless of the application you are working in. AutoMate™ watches for the text you specify in the trigger properties for as long as the task is enabled.

Example:

1) To set up a quick cross application text replacement tool – use an AutoKeys trigger set to watch for the word "uni" and set the actions to send Unisyn Software, LLC using the Send Keystrokes action in the Step Builder. Now whenever you type "uni" in any document the text will be replaced with the full company name!
2) As an intuitive way of launching a task. Now if you want to start your upload files to the internet task you could simply assign an AutoKeys trigger to in and set the trigger text to upload files. Now typing that text in any document will cause file transfer to begin!

Idle Trigger

The Idle trigger is used to execute a task when the computer is unoccupied. When you are not doing anything on your computer, the system is referred to as being in an Idle state. By using the Idle trigger in AutoMate™, you can change that. Simply tell AutoMate™ in the Idle trigger properties how long the system should be idle before the trigger is fired.

Example

This trigger is very useful for sweeping document files up to a network server for backup purposes. When you leave your computer at lunchtime – AutoMate™ can be sure your data is backed up – handling it all for you while you relax.

See Also:

Triggers

Startup Trigger

The Startup trigger is useful for starting tasks when AutoMate™ first starts. When AutoMate™ first loads the tasks that are set with this trigger will fire immediately.

NOTE: If you are using AutoMate™ on Windows 95/98 and AutoMate™ is in the Startup program group -- using this trigger will cause your tasks to run when you first log on to the system since that is when AutoMate™ starts. However, if using the Windows NT Service Edition of AutoMate™ note that this trigger will behave differently since NT loads services when the system starts even before anyone logs in to the system.

See Also:

Triggers

Screensaver Trigger

The Screensaver trigger causes a task to run when Windows starts the screensaver. The task will begin execution immediately before the screensaver becomes active. AutoMate™ itself will not attempt to stop the screensaver from starting. Your task, however, may consequently prevent the screensaver from starting if the mouse is moved using the Record Movements step, or if keystrokes are sent using the Send Keystrokes action.

See Also:

Triggers

Logout Trigger

The Logout trigger causes a task to run when a user logs out of the system.

Example
This trigger is very useful for sweeping document files up to a network server for backup purposes. When you leave your computer at lunchtime – AutoMate™ can be sure your data is backed up – handling it all for you while you relax.

See Also:

Triggers

Time Change Trigger

The Time Change trigger causes a task to execute when a user or application attempts to change the system time.

This is useful on network machines where you want to protect the system time. You can create a task that displays a messagebox informing the user not to change the system time, or send the network administrator an email informing the recipient that someone has modified the system.

See Also:

Triggers

Low Memory Trigger

The Low Memory trigger causes a task to execute when Windows reports that the system is low on memory.

You should keep the task which is executed as a result of this event as short as possible. Remember that the event is sent when the system is low on resources, and may therefore effect the execution of a task that AutoMate™ tries to start. A good example of task is one that displays a messagebox telling the user to shut down some applications (or maybe even which ones to shutdown), or sending an email to a network administrator or a pager to inform someone of the situation.

See Also:

Triggers

Device Change Trigger

The Device Change trigger causes a task to execute if Windows reports that a device has been changed on the system. For example, removing a PCMCIA card on a laptop computer would cause this task to run.

See Also:

Triggers

Display Change Trigger

The Display Change Trigger causes a task to execute if the graphics resolution of the system is changed.

NOTE: for color changes use the **Palette Change** trigger

See Also:

Triggers

Palette Change

The Palette Change Trigger causes a task to execute if the graphics color depth of the system is changed. For example changing 256 colors to 16 colors would cause this task to fire.

See Also:

Triggers

Spooler Change

The Spooler Change trigger is fired whenever a new print job is added or removed from the printer queue of the system.

See Also:

Triggers

Priority

Occasionally, when building an automated task, it is important that the task run alone and uninterrupted. Since AutoMate™ is fully multi-threaded – it has the capability of running many tasks simultaneously.

The Priority feature in AutoMate™ gives you the power to instruct AutoMate™ whether or not to run a task in conjunction with other tasks, and how to handle a conflict if one arises between tasks.

NOTE: This feature is useful when a task must make sure that an application stays in the foreground (also called focused) in order to send keystrokes to it. If another task were to run and try to focus an unrelated application – it would interfere with the task that is attempting to send keystrokes.

Options on this feature are self-explanatory.

See Also:

Introduction to AutoMate™ Tasks

Exporting A Task

Tasks can be saved as individual files and then loaded in other AutoMate™ installations by transferring the file to another system and importing it. Single Task File Format files (or STFF for short) contain all the information required to run the task: task name, triggers, and steps. It will even remember your password and priority settings. The file does not, however, contain executables or data files that may be required to run your task properly. If your task depends on script files, recorded event data files, or any executables that are starting using a Start Application action, you need to ensure that these files are copied as well, or the task modified to take these occurrences into account.

Saving a task can be done through the File->Export command, or by right clicking over the task, and selecting Export. Enter a filename for the task, and click OK. The file will be saved. You can then copy this file, email it to a friend, or anything else you would do with a normal document file.

See Also:

Importing A Task

Introduction to AutoMate™ Tasks

Importing A Task

Tasks that have been saved in Single Task File Format (STFF) using AutoMate's Export command can be imported into AutoMate™ and manipulated like any other task that was created on the machine.

To do this, select Import from the File menu, and select the filename of the task you wish to import. You can also right-click and select Import from the pop-up menu. Better yet, you can drag a Single Task File into AutoMate™, and AutoMate™ will automatically import it for you!

Importing and exporting AutoMate™ tasks provides a quick way to swap tasks with friends and coworkers.

See Also:

Exporting A Task

Introduction to AutoMate™ Tasks

FTP Login Action

Use the FTP Login action to log the AutoMate™ task onto an FTP server. Once this step is successfully completed, the other AutoMate™ FTP Actions (e.g. FTP Upload, FTP Download, FTP Rename, etc.) can be issued.

Using an FTP Login step is different from using FTP Quick Send and FTP Quick Receive, because the latter logon to the server, perform their action, and then disconnect. Using FTP Login, AutoMate™ stays connected to the server until you issue the FTP Logout step. This gives you the ability to perform a series of actions on the server throughout the duration of the task.

See Also:

FTP Logout Action

FTP Quick Send

FTP Quick Receive

FTP Upload Action

FTP Download Action

FTP Delete Action

FTP Make Directory

FTP Remove Directory

FTP Rename Action

FTP Logout Action

Use the FTP Logout action to disconnect from an FTP server you previously connected to using the FTP Login step.

Although AutoMate™ will attempt to logout from a connected server at the end of the task, it is a good idea to issue the FTP Logout action before your task ends, especially if your task runs for extended periods of time.

Keep in mind that most FTP servers will forcibly disconnect you after a number of minutes of inactivity.

See Also:

FTP Login Action

FTP Quick Send

FTP Quick Receive

FTP Upload Action

FTP Download Action

FTP Delete Action

FTP Make Directory

FTP Remove Directory

FTP Rename Action

FTP Upload Action

The FTP Upload action allows you to upload one or more files from your local machine to an FTP server. You must have previously established a connection with an FTP server using the FTP Login step before using this step.

The FTP Upload action supports wildcards. Using wildcards, you can upload a series of files that match a particular mask, all in one step.

You can use as many FTP Upload steps between an FTP Login and FTP Logout as you desire.

See Also:

FTP Login Action

FTP Logout Action

FTP Quick Send

FTP Quick Receive

FTP Download Action

FTP Delete Action

FTP Make Directory

FTP Remove Directory

FTP Rename Action

FTP Download Action

The FTP Download action allows you to download one or more files from an FTP server to your local machine. You must have previously established a connection with an FTP server using the FTP Login step before using this step.

The FTP Download action supports wildcards. Using wildcards, you can download a series of files that match a particular mask, all in one step.

You can use as many FTP Download steps between an FTP Login and FTP Logout as you desire.

See Also:

FTP Login Action

FTP Logout Action

FTP Quick Send

FTP Quick Receive

FTP Upload Action

FTP Delete Action

FTP Make Directory

FTP Remove Directory

FTP Rename Action

FTP Rename Action

The FTP Rename Action allows you to rename a file on the FTP server to a different name. A connection with the server must have already been established using the FTP Login step before you can use the FTP Rename action.

The file must exist on the server for this step to work. If AutoMate™ cannot find the file you wish to rename on the connected server, the step will fail.

See Also:

FTP Login Action

FTP Logout Action

FTP Quick Send

FTP Quick Receive

FTP Upload Action

FTP Download Action

FTP Delete Action

FTP Make Directory

FTP Remove Directory

FTP Delete Action

The FTP Delete Action allows you to delete a file on the FTP server. A connection with the server must have already been established using the FTP Login step before you can use the FTP Delete action.

Please not that AutoMate™ will continue to the next step of the task regardless of whether or not the file on the server is successfully deleted.

See Also:

FTP Login Action

FTP Logout Action

FTP Quick Send

FTP Quick Receive

FTP Upload Action

FTP Download Action

FTP Make Directory

FTP Remove Directory

FTP Rename Action

FTP Remove Directory

The FTP Remove Direction Action allows you to remove an entire directory from an FTP server. A connection with the server must have already been established using the FTP Login step before you can use the FTP Remove Directory action.

In most cases, AutoMate™ can only remove empty directories from an FTP server. You should be sure that any directories you wish to remove are empty using the FTP Delete Action before issuing the FTP Remove Directory step.

Attempting to remove a directory or file that does not currently exist on the FTP server will *not* cause the task to fail. The task continues regardless of whether or not the directory is successfully removed from the server.

See Also:

FTP Login Action

FTP Logout Action

FTP Quick Send

FTP Quick Receive

FTP Upload Action

FTP Download Action

FTP Delete Action

FTP Make Directory

FTP Rename Action

FTP Make Directory

You can create directories on the FTP server by using this step. A connection with the server must have already been established using the FTP Login step before you can use the FTP Make Directory action. You must also have sufficient rights on the server in order to the step to complete successfully.

Please note that, because it is not informed of the status of the makedir command from the FTP server, AutoMate™ will continue with the task regardless of whether or not the directory is successfully created.

See Also:

FTP Login Action

FTP Logout Action

FTP Quick Send

FTP Quick Receive

FTP Upload Action

FTP Download Action

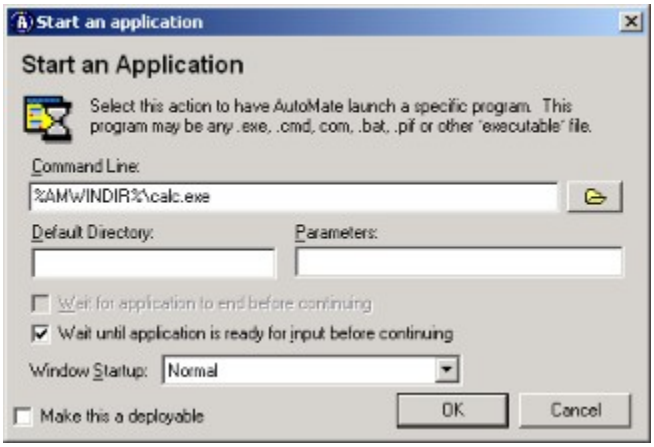
FTP Delete Action

FTP Remove Directory

FTP Rename Action

Constants

AutoMate™ tasks may be designed on one machine and run on others – either by **exporting and importing** or by deploying through AutoMate™ Enterprise Server. Because of this, sometimes it is necessary to specify a path to a file or any other piece of information that may be different from machine to machine. It is impractical to design a separate task for each machine – so AutoMate™ allows the use of Constants.



One example: If Machine 1 has Windows 98 installed to c:\WINDOWS\ and Machine 2 has Windows NT installed in

D:\WINNT\

When adding a Start an Application action you could use the AutoMate™ defined global system constant %AMWINDIR% in the path as follows:

When the task is deployed/imported and run on the individual machines, AutoMate™ resolves the value at runtime to the appropriate value – which in this case is the directory of Windows (without the trailing backslash).

Constants can act to make tasks more portable, as they can easily convey information to the task at runtime about that users system.

You can also configure your own constants, for example UserName, favorite color, or whatever else you need to use in a centrally created task that might differ from machine to machine.

Constants, unlike Variables, are system-wide, meaning that they are not specific to single tasks. They are configured in the preferences dialog, and are specific to the machine which AutoMate™ is running on.

This dialog shows the AutoMate™ System Defined Constants, and enables addition, deletion and modification of custom constants.

Constants may be used in any AutoMate™ step parameter, with the exception of the Send Keystrokes action. Within Send Keystrokes you must use the format {CONST AMWINVER}. Otherwise, the constant will be taken literally and the name of the constant will be sent as keystrokes.

An option is available from the Send Keystrokes properties dialog box on the Field button to make selection easier for you.

See Also:

Variables

Deployables

Variables

Like constants, variables can be used to represent data in a task that may be different from machine to machine. But unlike constants, variables can also be modified during a task's execution through the use of the Input Variable action. In this way, variables can be used to represent data that may be different each time the task is run. Also unlike constants, the scope of variables is limited to the context of the task they are used in. For example, creating and adding data to a variable called DATERANGE would not make the data accessible for any other task except for the task it was created in.

Example:

To use a variable, simply select it from within the Step Builder interface. Obviously, a variable is not very useful if data cannot be entered into it. When used in this way the task stops execution to ask for user input -- after the variable is populated with the users entry, other steps may reference the data by specifying the variable name anywhere in the step parameters, like this: !!VARIABLENAME!!
(substitute VARIABLENAME for the variable name you used.)

The two exclamation marks (!!) at the beginning and end of the variable name tell AutoMate™ that the word in-between the !! is a variable, and should not be taken literally. Instead, it is replace with the current contents of that variable.

Starting with the Professional version 4.4 of AutoMate™, unresolved variables (i.e., variables that AutoMate™ does not recognize) are passed to the AutoMate™ scripting language for evaluation. If the string between the !! symbols is a valid AutoMate™ Basic command or expression, the return value of that expression will be returned. This allows you to create inline expressions right inside any AutoMate™ action, without having to build a separate script and run it from a script action. See the inline functions section of the help file for more details.

See Also:

Constants

Deployables

Inline Expressions

Deployables

Deployables are used when files must be distributed along with a task to many different machines. Task distribution supporting deployables can either be accomplished through manual exporting / importing of tasks, or task deployment through an AutoMate™ Enterprise Server

Deployables are used in actions that take a file path as a parameter (such as Start an Application, Open a document, Copy a File, Print a Document, etc.) by specifying the name of the deployable surrounded by double pipe symbols (i.e. ||DEPLOYABLENAME||). All steps that support deployables have a checkbox labeled Make this a deployable. This may be used where you may either create a new deployable by specifying a name that does not already exist or select one that does from the Name dropdown combo box.

Deployables allow tasks to be easily shared even if files that the task depends on such as basic scripts (.BAS), mouse recordings (.DAT) or executable (.EXE) files do not exist already on the users machine. By specifying a file as deployable, when the task is either exported or sent via the deploy feature through AutoMate™ Enterprise Server, AutoMate™ will automatically retrieve the file from the path specified in the Local Path parameter of the deployable. The file is then encoded into the STFF (Single task file format) when the task is exported or deployed. On task import or retrieval from an AutoMate™ Enterprise Server, all deployables are extracted to the path specified in the Remote Path parameter on the remote machine before the task executes. Any directories and subdirectories that do not exist are automatically created. Constants may be used in either the local path or the system path to deal with drive or directory ambiguities from system to system. After the files are extracted onto the remote machine, the value in the Local Path is overwritten with the value in the Remote Path and the task is imported. The steps that rely on the deployables continue to use the deployables to indicate the file so the task may be exported again from that machine if needed.

When AutoMate™ executes a task and a deployable is encountered (indicated by the double pipe symbols in the file path such as, ||DEPLOYABLENAME||), a lookup is performed in the tasks deployable list for the name specified between the || symbols. When found AutoMate™ looks at the Local Path value to determine which file to use for that action. All of this work occurs transparently to the user. After task execution the path still remains the deployable name so that the task retains its portability.

See Also:

Variables

Constants

AutoMate™ Enterprise Server

Expand All

Expands all the trees in the Available Actions section, revealing all of the possible actions in AutoMate™.

Collapse All

Collapses all the trees in the Available Actions section, revealing only the action categories of the actions in AutoMate™.

Insert Step Above

Inserts the selected action from the Available Actions section above the currently selected step in the Current Steps section.

Insert Step Below

Inserts the selected action from the Available Actions section above the currently selected step in the Current Steps section.

Properties

Shows the properties of the currently selected step from the Current Steps section. Use this button to modify the properties of a step after it has been added to the task.

Cut Step

Cut the currently selected step from the Current Steps section and place it on the Windows Clipboard.

Copy Step

Copy the currently selected step from the Current Steps section and place it on the Windows Clipboard.

Paste Step

Paste a previously selected step from the clipboard into the Current Steps section.

Test Task

Runs each step of your task, ignoring the checked state of the steps.

Save Task

Saves the changes made in the Step Builder to the task, and closes the Step Builder.

Cancel

Discards any changes made to the steps of the task and closes the Step Builder.

(type popup caption here)
(type popup definition text here)

New Task Wizard

The AutoMate™ Task Wizard is designed make the task creation process easy by walking you through the required steps. Creating a task is comprised of three primary steps:

1) Actions - Specify what the task should do
2) Triggers - Specify when the task should start
3) Name - A unique name for the task

After defining your actions in the Step Builder, the wizard will prompt you for the triggers you wish to use. Depending on the check boxes you select, the wizard will modify its content to request more specific information for each of the triggers you selected.

At the end of the wizard you are prompted to enter a name for the task, and then asked a few additional questions:

1) Whether you would like a icon shortcut on the desktop to start this task.
2) Whether you would like to open the advanced properties dialog box afterwards to modify more task properties.
3) Whether the task should delete itself after it runs once.

After using the task wizard you can modify your tasks later from the Configuration Manager dialog box by either double clicking the task, by selecting the task and selecting the menus File | Properties, or by selecting the task and clicking the Properties toolbar button.

You may disable the Task Wizard if you prefer not to go through the wizard. Tasks will be created as Untitled-1 when you press the New toolbar button.

See Also:

Actions

Triggers

Step Builder

Preferences

The Preferences dialog allows you to adjust aspects of AutoMate's performance to suit your individual system, requirements and tastes.

The dialog is separated into five separate tabs, each dealing with a separate category:

Tolerances

Task Speed: Adjusts the speed at which AutoMate™ executes each line of a running task. Occasionally, on slower machines, adding a delay between steps improves task reliability when interacting with other applications that may not be able to react to AutoMate™ fast enough. By adjusting this slider, it may not be necessary to add an explicit Pause step in your task.

Task Scan Rate: Adjusts the frequency by which AutoMate™ checks the system clock to see if a scheduled task should be triggered. This will increase or decrease the accuracy of a scheduled task, but if task timing accuracy is not important to you, then setting this lower could decrease the amount of CPU time that AutoMate™ uses (although it is already very low).

SendKeys Speed: Adjusts how much time AutoMate™ should wait in-between each key-press when sending keystrokes. On some faster machines, AutoMate™ may type the keys too quickly, flooding the keyboard buffer and causing unpredictable results. Increasing this setting will prevent this from occurring.

Logging Level: AutoMate™ does not issue error message boxes during task execution as it is designed to operate properly in an unattended fashion. Instead the errors and task results are sent to a log file. This setting adjusts how much information is sent to the log file – you should keep this setting low unless you are debugging a problem with the task, in which it may be useful to log All Events, since this will show the status of each individual step as they execute. To view the task log select View | Task Log from the Configuration Manager window, or click the View Log button from the toolbar.

Options

Prompt to run task service at Configuration Manager startup:
This option selects whether the Configuration Manager stops to prompt to user if they want to start the AutoMate™ Task Service (if it is not running) when the Manager starts. When unchecked, AutoMate™ will not ask you this and you must start the Task Service manually by selecting File | Start AutoMate™ Task Service.

Display confirmation on Task Service shutdown:
When you attempt to shutdown AutoMate™, the program will confirm that you really want to stop AutoMate™ and suspend all automated processing. Unchecking this will option will prevent this confirmation box from appearing and close AutoMate™ immediately.

Display confirmation on delete task:
Disables the prompt on deletion of a task from within the Configuration Manager.

Enable drag and drop operations in the Step Builder :
When unchecked this will disable drag and drop support in the Step Builder window. If drag and drop support is disabled, add and move steps in the Step Builder by using the add button and the up and down arrow buttons.

Display indicator window when task is running

When unchecked, the small window that appears in the lower right of the screen when AutoMate™ runs a task that says AutoMate™ is currently running a task will not appear. Regardless of this setting, the AutoMate™ tray icon will be a green color when a task is executing.

Display AutoMate™ tray icon:

This is a security feature to disable the A icon in the system tray. Preventing the display of the tray icon would prevent users from shutting down the AutoMate™ Task Service.

Enable wizard when creating tasks

By default, AutoMate™ will guide you through the process of creating a new task, stepping through the available triggers and configuring them, and provide the opportunity to set advanced options at the end. You can disable the wizard by checking or unchecking this option. When the wizard is not used, AutoMate™ creates a task named "Untitled Task", which you can then edit and configure manually by double clicking on it.

Display errors from running tasks when Configuration Manager is open

If this option is selected, errors generated by a running task will be displayed in a dialog box while the AutoMate™ Configuration Manager is running. This greatly aids debugging. When this option is not selected, the error messages are quietly displayed in the status bar of the AutoMate™ Configuration Manager. All errors and messages are written to the log file regardless of this setting.

Disable Windows "Foreground Window Timeout" Feature

Windows 98 and Windows 2000 operating systems now implement a "Window Lock Timeout" feature, which prevents an application that a user is not currently interacting with from giving itself the keyboard focus. This can cause problems with AutoMate™, since functions like "Focus A Window" and "Message" will not have the authority to carry out its action unless you have directly interacted with AutoMate™ within the timeout period. Checking this option disables this feature for the entire operating system. This allows AutoMate™ to properly focus windows when the machine is unattended. Unchecking this option restores the timeout functionality to its original state. The AutoMate™ installation by default disables the "Foreground Window Timeout."

Folder/File Locations

Task File: Specifies the location of the task file that the Task Service and the Configuration Manager should use.

Log File: Specifies the location of the log file that AutoMate™ will log events to.

Default Script Folder: Specifies the default path for AutoMate™ BASIC scripts to be saved to and opened from when adding a script step into a task via the Step Builder (note that this only affects the default script path in the Step Builder, not during actual task execution).

Services (NT Service Edition Only)

This tab allows modification of the default username, password and domain information that AutoMate™ will use when a login step is not present in a task. This should be set to a user on the NT workstation or domain that does not need to change its password (configurable via the Windows NT User Manager). **WARNING: If you are using the NT Service Edition and this information is not set and there are no login steps in your task, your tasks will not run! Also, the user specified in this entry must have rights on the system to perform the desired task (i.e. launch the requested programs, etc.)**

Constants

Allows the viewing, addition, deletion, and modification of constants – both system defined and user defined. See the Constants section of the help file for more details.

AutoMate™ Enterprise

Used to configure AutoMate™ to connect to an AutoMate™ Enterprise Server. AutoMate™ Enterprise Server is a separate software product used to connect together large network installations of AutoMate™ for easy tasks deployment and AutoMate™ administration network wide. Includes various Server log on options dealing with the servers IP address and the information to identify this machine and user to the network.

Security

Used to password protect a task list. When this option is configured, AutoMate™ will prompt the user for a password when the Configuration Manager is loaded

Event Logging

Limit Log file size

If this is selected, AutoMate™ will take action to stop the log file from exceeding a specific size you set using the *Log file should not exceed* parameter. When the log file reaches this size, you can select to either disable logging until the log file is cleared (at which point logging will automatically resume), or have AutoMate™ "trim" (i.e. remove from the top of the file, the oldest messages) the log file by a percentage you specify.

Log File Viewer

By default AutoMate™ will use the Windows Notepad application to display the log file. In some cases this may not be satisfactory; you can enter a path to a different text file viewer (for example, Wordpad.exe) here.

Failure Notification

NOTE: This option is only available on Professional versions of AutoMate™

Send email on task failure

When selected, AutoMate™ will attempt to send an email message containing details about the error that occurred to the email recipient. AutoMate™ must have access to a functioning SMTP mail server (enter the server address at the *Server* parameter). Recipients should be supplied to the *Recipients* parameter. The email can be sent to multiple people by separating their email address with a semi-colon.

Message Length

AutoMate™ offers two different email message types: verbose, for standard email clients, and brief, for limited text clients (such as pagers and cell phones). The verbose email will send the following information: the task that failed, the error message, the line the error occurred on, the computer name the error occurred on and when the error occurred. The brief email sends only the task name that failed, when it failed, and on what machine.

See Also:

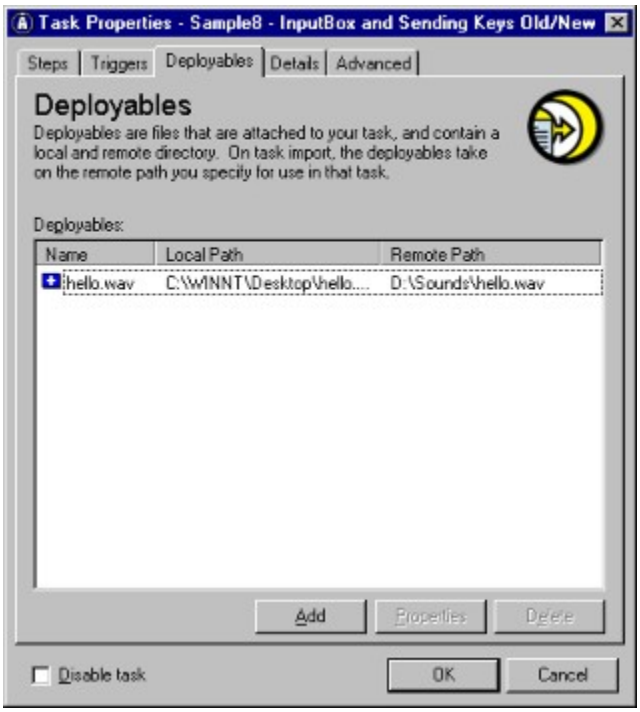
Step Builder

Using Variables

An example of using a variable in a "Send Keystroke" action. When the text is printed out, {VAR thename} will be replaced with the result of the INPUTTOVAR step before it (not shown).

Deployables

Lists all of the deployables that are to be attached to this task on export or deployment to a remote machine. The listbox shows the AutoMate™ name of the deployable, its local path (the path to the file on the local machine), and its remote path (the path to where the file will be written on import and receipt of the task).



Deployables: Lists all the Deployables currently attached to this task. To modify or delete an item, highlight the item and select Properties or Delete.

Add: Adds a deployable for use in the task.

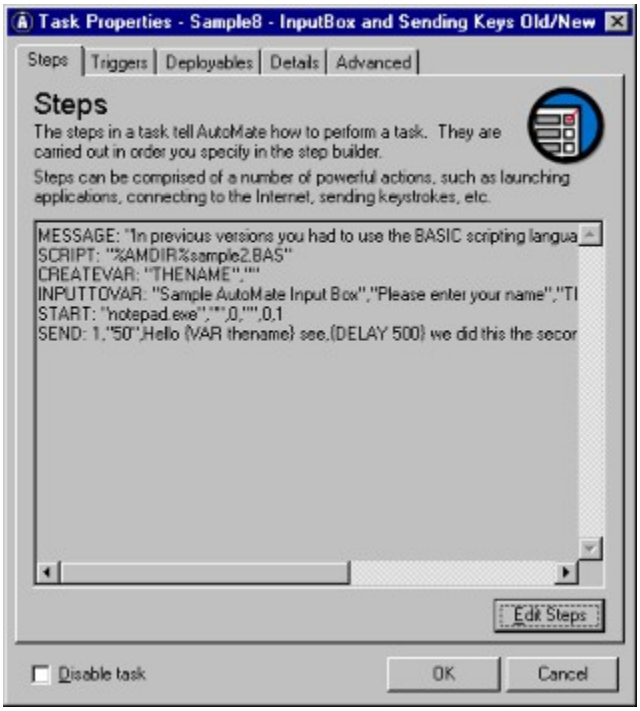
Properties: Modifies the properties of a deployable

Delete: Removes a deployable from a task.

NOTE: This action does not delete the actual file specified in the local path only the reference to it as a Deployable.

Steps

Displays current general information about the selected task.

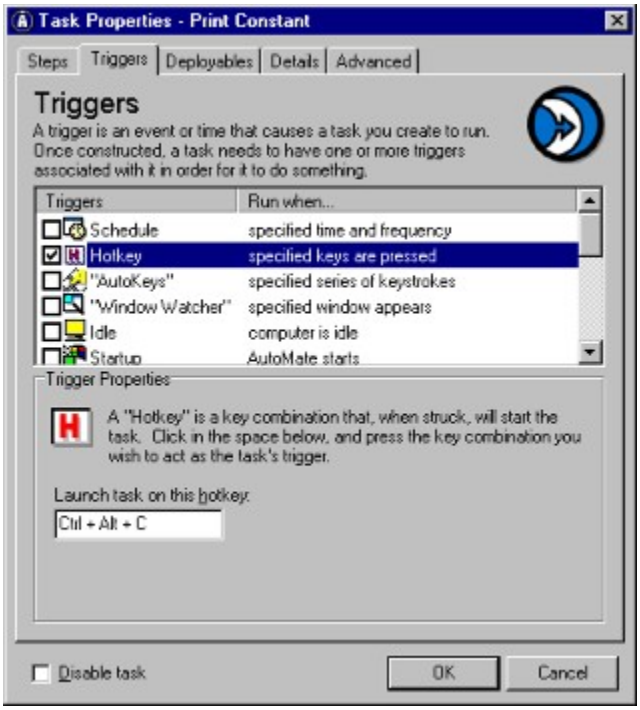


Current Steps: Show the current steps that will be executed when the task is triggered or launched.

Edit Steps: Starts the Step Builder task construction window, an easy-to-use task-editing environment.

Triggers

Displays the current choices of triggers that will be used to launch the task.

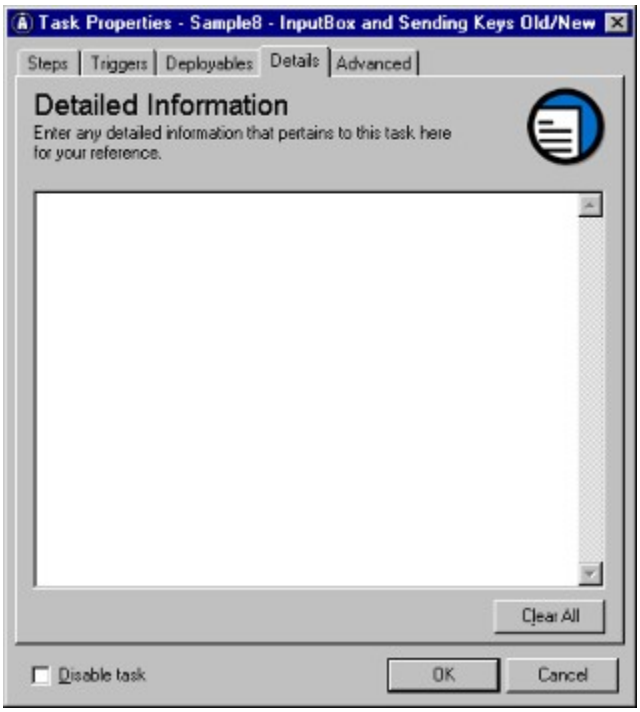


Trigger List: This is a list of available triggers for this task. To activate a trigger an set its properties, select the check box to the left of the trigger item in the list. Its properties will appear below.

Trigger Properties: The properties for the currently selected trigger in the trigger list appear here.

Details

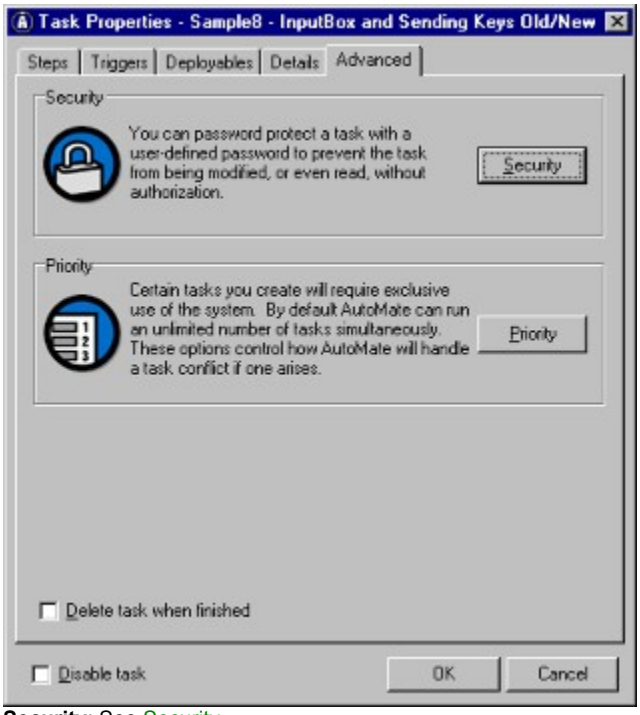
Provides a common area for your own notes about the selected task.



Details: Use this space to store any extra information in free form text format about the task.

Advanced Properties

Configuration of advanced features of the task, such as password protection and task prioritization.



Security: See Security

Priority: See Priority

Active Step

Selects whether or not the step is to be executed when the task is run. If the checkbox is unselected, the step will be ignored during testing and running of the task. Usually used for debugging.

Start Another Task Action

As a part of your task execution you may wish to branch from one task to another, either for organization of various automation routines or for error checking so that you can see which tasks may be failing. Start another task enables you to branch task execution to another task and contains several options to control this task flow during the branching process.

Available options for Start Another Task:

Task To Start:
Specify the name of the task to branch to start. (Variables are supported i.e. !!VARIABLENAME!!)

Continue on with this task immediately:
When this option is checked, the execution of the current task will continue asynchronously along with the new task, which will be spawned at this point in the task. Both tasks will run in parallel.

Wait for the task to finish:
When this option is checked the execution of the current task will not continue until the spawned task is complete or has failed.

Fail this task if the started task fails:
This option is only available when wait for the task to finish is checked. It is useful when a task's continued successful execution is dependent on another tasks successful completion. When this option is enabled, the current task will fail also if the spawned task has a failure during its execution.

Update AutoMate™ Action

Although most users will update AutoMate™ to the latest version via the Tools | Net-Update menu option, the Update AutoMate™ task action is provided so that users may build tasks to accomplish one or both of these two objectives:

1) Build and deploy a task through an AutoMate™ Enterprise Server to update entire networks of AutoMate™ clients.
2) Schedule the update process (i.e. every month, download the latest version of AutoMate™)

Most of the options for the Update AutoMate™ action control how the file is obtained. The file to point to is a regular AutoMate™ installation file (the same that is obtained at our web site). Every AutoMate™ installation has special command line options that allow it to run and configure itself in silent mode. Once the file is obtained on the machine running the task the action handles shutting AutoMate™ down, starting the install in silent mode (the only thing users will see is a setup icon in the system tray (next to the system clock). When the install is complete the AutoMate™ helper process will restart the AutoMate™ Task Service. This greatly enhances the ease with which network administrators may upgrade AutoMate™ installations on entire networks and/or multiple machines.

Available options for Update AutoMate™ Action:

Download from Unisyn:
This options causes AutoMate™ to download the current installation directly from Unisyn's FTP Servers. A version will only be downloaded if it is newer than the one running on the client attempting to download.

Install from Network Drive Path:
This causes Automate to use the file specified in the Path to Installation Field

Path to Installation:
When Install From a Network Drive is checked. This field should be populated with the path and filename to the AutoMate™ installation. The path should be visible to all the clients to whom this task is going to be deployed and run on.

Show Progress Window:
When checked AutoMate™ will show a progress window while the file is being downloaded, when uncheck the install will be completely silent.

Force Reboot:
If the AutoMate™ installation needs to reboot when complete, when this option is checked it will force a machine reboot. The user will not be prompted, so use with caution. If this installation does not need to reboot then no reboot will be performed.

See Also:

Net-Update

Net Update

Net Update is a quick and easy way to keep your AutoMate™ installation up to date. Net-Update handles checking to see if a newer version is available, downloading the AutoMate™ installation from Unisyn's FTP servers, shutting AutoMate™ down, starting the install in silent mode, and restarting AutoMate™ when complete.

It's easiest and fastest way to obtain the latest AutoMate™ installation!

See Also:

Update AutoMate™ Action

Inline Expressions

NOTE: This feature is only available in the registered Professional version

Inline expressions provide a quick and easy way to evaluate a simple AutoMate™ Basic expression right inside a step's parameters, instead of writing an entire script. Inline expressions work by taking any unresolved variables it finds between double-exclamation points and passing them to the AutoMate™ Basic interpreter. The return value is then passed back to AutoMate™, which replaces your variable with the return value of the expression.

For example, consider the following trivial example: we want a task to do a simple mathematical calculation and display the results in a message-box. In the past, it would have been necessary to make an AutoMate™ BASIC script that carried out the calculation, and displayed the result in using BASIC's MsgBox command. The script would then need to be called from an AutoMate™ SCRIPT step. Now, the exact same function can be done using the Message step, and using !!4+4!! as the Message step's "Message to display" parameter. Since the phrase 4+4 does not exist as a variable name that was created inside AutoMate™, it is passed to the scripting language, where it is processed and returned.

Inline expressions can also contain AutoMate™ variables imbedded inside the expression. Taking the previous example a step further, assume we now want the task to give the user the ability to enter a number he would like squared (that is, multiplied by itself). Because AutoMate™ passes its own variable list to the scripting language *before* the step is evaluated, it is aware of any variables you have already used in AutoMate™. Therefore, your task could look like this (hint: you can copy and paste this into the step builder and press F5 to see how it works!):

```
REM: Create the variable
CREATEVAR: "NUMTOSQUARE","1"
REM: Ask the user to enter a number
INPUTTOVAR: "Enter a number","Which number would you liked squared?","NUMTOSQUARE",0
REM: Perform the calculation using an inline expression
MESSAGE: "!!Int(NUMTOSQUARE) * Int(NUMTOSQUARE)!!",1
```

That's it!

There are a few things here to note:

1) Notice that we first had to create the variable we intend to use. All variables to be passed into a scripting language, either explicitly through the use of a SCRIPT step or implicitly through the use of inline expressions, need to be declared first. You can accomplish this using the Create Variable action .
2) Notice the last line of our example. A*ll* variables are passed into the scripting language, and all return values are returned from the scripting language, as *strings*. This is why it was necessary to use the BASIC command Int() before squaring the numbers.
3) Notice that when we are using the AutoMate™ variables in the string to be evaluated that we *did not* enclose them in the AutoMate™ variable format (i.e. enclosing the variable name between double exclamation points). This is because once the string is passed to the scripting language, they become *scripting* variables, and not AutoMate™ variables. Variable names are *not* case sensitive in the AutoMate™ BASIC language; the variables are capitalized here for clarity.

Because each expression is passed to the scripting language engine, any command available from a common AutoMate™ BASIC script is available to you through inline functions. Refer to the AutoMate™ BASIC Language Reference for more details on these powerful commands.

See Also:

Scripting in AutoMate™

Run a BASIC Script Action

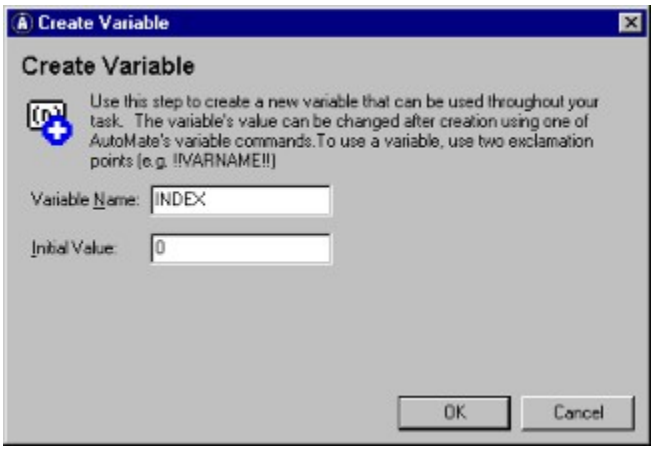
Inline Expressions

Creating Variables

Variable Functions

Create Variable Action

Variables allow your task to store and manipulate dynamic data. Before you use a variable in a task or script, you must first create it. The Create Variable action allows you to do this.



This action has two parameters that must be supplied:

Variable Name
This is the name of the variable to be created. It must be unique to this task (i.e. you cannot have two variables with the same name in the same task). You can then reference the contents of this variable in subsequent steps by enclosing the variable name is double exclamation points (for example, in the figure above, we are creating a variable called INDEX with an initial value of 0. To retrieve the number 0 from this variable in another step, we would use !!INDEX !!)

Initial Value
The initial value to be stored in this variable. You can always change the contents of the variable later on using the Set Variable action, or through the scripting language or an inline expression.

Users of AutoMate™ Professional can also use the Create Variable function to create variables that are automatically passed into any scripts they run from a task. All AutoMate™ variables that are created using the Create Variable action before entering an AutoMate™ BASIC Script (by using the SCRIPT step) are available inside the script using its variable name. Any changes made to these variables are also automatically passed back to AutoMate™ when the script has completed. This allows great flexibility in your tasks, since data can seamlessly be exchanged between your executing task and AutoMate™ scripts.

An advanced feature of the Create Variable action, available only to users of the Professional version, is to use an inline expression to create "dynamically named variables." Using this technique, you can create a variable name that is the result of an inline expression. For example, say we have a Create Variable step that creates a variable called INDEX with an initial value of 0. We then enter a loop set for 3 iterations, and inside this loop is another Create Variable step, which has its *variable name* parameter set to "NEWVAR!!INDEX+1!!" (without the quotes). When the loop exits, AutoMate™ will have created five new variables: NEWVAR0, NEWVAR1, and NEWVAR2.

See Also:

Inline Expressions

Set Variable Action

The Set Variable Action is used to change the contents of a previously created AutoMate™ variable. The action replaces the current contents of the variable with those specified by the *New Value* parameter of the step.

The step has the following parameters:

Variable Name
The name of the variable to populate with the new value. This variable must have already been created using the Create Variable action.

New Value
The new value to place inside the variable. All values are converted to strings before being placed inside the variable.

See Also:

Inline Expressions

Creating Variables

Inputing user data into a variable

Sending variable contents as keystrokes

Variable Functions

Input To Variable Action

The Input To Variable Action provides a way for a user at the computer to enter data into a variable at runtime. A dialog box is displaying with a title and descriptive text of your choice. The user can enter text into the edit box provided. If the user clicks OK, their input is automatically placed inside the variable specified by *variable name*. If the user clicks the Cancel button, the step and task fails.

The step has the following parameters:

Variable Name
The name of the variable to which the user's input should be stored. This variable must have already been creating using the Create Variable action.

Title
The text that should appear in the title bar of the dialog box prompting the user for input.

Text
Any descriptive text you wish to be displayed on the dialog box. This text will appear above the edit box where the user can enter their input.

If the user's input is successfully applied to the variable, the task will proceed to the next step. If not, the task will fail. The step fails if the user clicks the Cancel button on the dialog box, or if the variable name supplied in the *variable name* parameter does not exist.

See Also:

Inline Expressions

Creating Variables

Setting Variables

Sending variable contents as keystrokes

Variable Functions

Send Variable Action

The Send Variable action transmits the contents of the variable you specify as a series of keystrokes to the currently focused window.

The step has the following parameter:

Variable Name
The name of the variable which contents you want “typed out”. The variable must have already been created using the Create Variable action.

The step fails if the variable name you specify does not exist, or if the variable contents contain illegal characters (which is not possible unless the variable has been populating using the scripting language or an invalid inline expression).

See Also:

Inline Expressions

Creating Variables

Setting Variables

Inputing user data into a variable

Variable Functions

Populate Variable With Clipboard Action

The Populate Variable With Clipboard Action will populate the variable you specify with the current text on the system clipboard.

The step has the following parameter:

Variable Name
The name of the variable where you want the clipboard's contents to be stored. The variable must have already been created using the Create Variable action.

The step fails if the variable name you specify does not exist, or if the clipboard does not contain any valid text.

See Also:

Inline Expressions

Creating Variables

Setting Variables

Inputing user data into a variable

Sending variable contents as keystrokes

Variable Functions

Variable Functions Action

The Variable Functions action provides a number of functions that change the value of an AutoMate™ variable.

The step has the following parameters:

Function
A drop down list of the variable functions AutoMate™ supports. Depending on the function you choose, the use of the *parameter* may change (or not even be available). Consult the chart below for what each function does, and what the *parameter* does for each.

Variable
The first (or primary) variable to use in the function. This is the variable that the function will be performed upon. This variable must have already been created using the Create Variable action.

Dest. Variable
The variable where the completed function contents should go. This variable can (and usually is) the same as the primary variable. By default, AutoMate™ uses your primary variable as the destination variable, unless you enter a different value here. This variable must have already been created using the Create Variable action.

Function	What It Does	What Parameter Does
Increment	Increments the primary variable by the amount specified in *parameter*	The amount to increment the variable by
Decrement	Decrements the primary variable by the amount specified in *parameter*	The amount to decrement the variable by
Concatenate	Combines the primary variable and the *parameter* together	The text (or variable) to append to the end of the primary variable

The step fails if the function cannot be performed on the variable you specify (trying to increment a variable that does not contain a number, for example), or if the primary or destination variables don't exist.

See Also:

Constants

Deployables

Inline Expressions

Creating Variables

Setting Variables

Inputing user data into a variable

Sending variable contents as keystrokes

IF Action

NOTE: This feature is only available in the registered Professional version

The IF Action provides a means of altering the flow of the task based on a true or false comparison of two variables, or based on an evaluation of a test.

The IF Action dialog is laid out with three input areas side by side along the middle of the window. The first input box is referred to as the "first parameter", box in the middle is the "IF type", and the last box on the far right in the "second parameter". The remaining two parameters towards the bottom the dialog box, the *Then* and *Else* parameters, are used to tell AutoMate™ what to do when the IF statement you create is true (the *then* parameter is executed) and when it is false (the *else* parameter).

These parameters are explained below:

First Parameter

This is the first parameter you supply to your IF Action. This can either be straight text (as in a string literal), a variable, or an inline expression. What you put here will ultimately depend on what you select for your *IF Type*.

IF type

This is where you specify what kinds of conditional testing you wish to do. Consult the chart below for a list of what each IF type does, and what each parameter does as a consequence of your choice.

Second Parameter

The is the second parameter of the IF Action. This can either be straight text (as in a string literal), a variable, or an inline expression. This parameter has different meanings depending on what *IF Type* you select. For some *IF Types*, the second parameter is not even available. Consult the chart below for a list of all the supported *IF Types* and what their parameters do.

Then

This is where you select what AutoMate™ should do when the IF Action you specify evaluates to true (i.e. the expression you build is correct):

Continue the current task	The step returns okay, and the task continues to the next step
Stop the current task	The step returns stop, and the task *does not* continue to the next step, but rather exits gracefully at this step
Start another task	Allows you to specify another task to start. The selected task starts, and the current task stops.

Else

This is where you select what AutoMate™ should do when the IF Action you specify evaluates to false (i.e. the expression you build is incorrect):

Continue the current task	The step returns okay, and the task continues to the next step
Stop the current task	The step returns stop, and the task *does not* continue

	to the next step, but rather exits gracefully at this step
Start another task	Allows you to specify another task to start. The selected task starts, and the current task stops.

These are the available *IF Types*, and a description of each:

IF Type	**Description**
Equals	Tests whether parameter 1 is exactly the same as parameter 2.
Does not equal	Tests whether parameter 1 is *not* exactly the same as parameter 2.
Greater than	Tests whether parameter 1 is greater than parameter 2. If these parameters can be converted to numbers, AutoMate™ will do so before comparison. Otherwise, an ANSI text comparison will be done instead.
Less than	Tests whether parameter 1 is less than parameter 2. If these parameters can be converted to numbers, AutoMate™ will do so before comparison. Otherwise, an ANSI text comparison will be done instead.
Is longer than	Tests whether the length of parameter 1 is greater than the length of parameter 2.
Is shorter than	Tests whether the length of parameter 1 is shorter than the length of parameter 2.
Is same length as	Tests whether the length of parameter 1 is the same as the length of parameter 2.
Is numeric	Tests whether parameter 1 is a valid number (parameter 2 is unused)
Is date	Tests whether parameter 1 is in a valid date format, based on the regional settings of the system (parameter 2 is not used)
Is empty	Tests whether parameter 1 contains anything
Contains	Tests whether parameter 2 is a sub-string (i.e. contained within) parameter 1.

The step fails if either parameter 1 or parameter 2 contains an unknown variable or otherwise invalid text. The step may also fail if AutoMate™ is set to start another task on a *Then* or *Else* condition, and that task is not available (for example, if it was removed from the task list).

See Also:

Variables

Inline Expressions

Paste Keys Action

The Paste Keys action uses the focused application's paste command to rapidly paste text into the focused window. This function is similar to Send Keystrokes, except this action is much faster because it uses the system clipboard. It does not interfere with the keyboard because it does not emulate a user at the keyboard. On the other hand, it does not provide the functionality of Send Keystrokes; only valid, plain text can be used with this action.

The action works by saving the textual contents of the clipboard, then writing the keys to be pasted to the clipboard. After the paste is complete, the previous contents of the clipboard are restored.

This step requires the following parameter:

Text
The text to be pasted to the focused application when the step is executed. Note that variables and fields are allowed in this section, just as they are in Send Keystrokes. The variables and fields are interpreted and parsed before the contents are pasted to and retrieved from the clipboard.

Comment Action

The Comment Action is really not an action at all, but rather a way to comment your task. Comment actions are not executed by the task when it is run. Use the comment action to provide internal documentation for your task steps.

Commenting your task is beneficial for a number of reasons. First, it makes it clear to others who may need to examine your task what the task is supposed to be doing, and the reasons why it is doing what it is doing. Secondly, it provides a convenient place to explain workarounds or rationale behind certain approaches you may take in a task that may not be intuitively clear. Finally, it is indispensable for those times when you come back to edit a large task you made months ago, and can't remember why you did what where.

Open Webpage Action

The Open Webpage action opens a webpage or an HTML document using the system's default web browser.

To access a webpage from the internet, you have to either be on the internet at the time the step is executed, or use a Dial-Up action to establish a valid connection to the internet. Attempting to access an inaccessible website or a local HTML file that does not exist at the path specified in the *Webpage To Open* parameter will cause the step to fail.

The step requires the following parameter:

Webpage to open
The URL of the webpage to open, or the local HTML file to open using the default web browser (note that, if you are attempting to open a local file, you must prefix the path of the file with file://)

FTP Change Directory Action

Changes the current working directory on the remote FTP server. Once this step is executed., all subsequent FTP calls that do not use an absolute path will use this new working directory as the relative path.

The step is especially useful when logging onto an FTP server with a complicated directory structure. For example, if all the files you need to access were in the path /usr/me/docs/html/mywebsite, you could first issue an FTP Change Directory action, and from that point forward, use just the proper filenames relative to this path. AutoMate™ will automatically place the proper /usr/me/docs/html/mywebsite path in front of the files you specify in subsequent FTP steps.

The step will fail if it cannot issue the FTP CD command, which usually occurs if a valid FTP session has not been established using the FTP Login step.

This step requires one parameter:

Directory
The directory to change to as the relative path. Unisyn suggests using the UNIX FTP format (using forward slashes instead of backward slashes for directories) for maximum compatibility with the widest range of public FTP servers.

Ping Machine Action

The Ping Machine action attempts to ping a machine on a TCP/IP network (such as the Internet) and waits the specified amount of time for a response. AutoMate™ will then take further action as specified based on whether or not the machine responded to the ping request.

This step requires the following parameters:

Hostname or IP
The hostname or IP address of the machine to ping.

Time to wait (ms)
The number of milliseconds AutoMate™ should wait for a response before assuming it is not going to get one. Usually 1500ms (1.5 seconds, the default) is sufficient for most networks to respond. If the machine does not respond within this time frame, it is down or the network route to the machine is heavily congested.

Var to put relay time (optional)
If you wish, you can place the amount of time the ping request response took in a variable. The variable must already exist before the step is executed if you place a variable here. You can create a variable by using the Create Variable action.

If machine responds
Here is where you select what AutoMate™ should do if the machine responds to the network ping within the time specified by *Time to wait*.

Continue the current task	The step returns okay, and the task continues to the next step
Stop the current task	The step returns stop, and the task *does not* continue to the next step, but rather exits gracefully at this step
Start another task	Allows you to specify another task to start. The selected task starts, and the current task stops.

If machine does NOT respond
The action AutoMate™ should take if no response is returned to the machine within the time specified by *Time to wait.* See above for the available options.

The task will fail if it cannot issue a ping command to the requested machine (if the hostname cannot be resolved, or a valid network connection is not available when the step is issued) or if an unrecognized response is heard from the target machine.

Disable This Task Action

The Disable This Task Action disables the currently running task in the AutoMate™ Configuration Manager to prevent it from running again until it has been manually stet to active.

Use this action when you want to programmatically inactivate the task, thereby preventing it from starting itself.

Fields

AutoMate™ also supports a number of fields, which are placeholders for dynamic data (i.e. data that may change over time). When AutoMate™ encounters a field during task execution, it replaces the field with the proper information. Fields are always surrounded by braces ({ })
The following fields are supported:

Field	**What AutoMate™ Types**
SHORTDATE	The current date, in short format, according to your regional settings (e.g. 01/05/1999 for January 5, 1999 in US format)
LONGDATE	The current date, in long format, according to your regional settings (e.g. Sunday, February 21, 1999 in US format)
TIME	The current time, in 12-hour format.
TIME24	The current time, in 24-hour format.
DELAY x	Waits x number of milliseconds before continuing with the keystrokes that follow. (e.g. DELAY 500 would make AutoMate™ pause 500 milliseconds before proceeding).
{FORMATDATETIME "MMDDYY HH:MM"}	Display the current date in a custom format. Replace the information between the quotes with the format you desire.
{VAR varname}	Output the variable with variable name *varname*
{CONST constname}	Output the value of constant *constname*

AutoMate™ also supports date shifting with any of the date fields (SHORTDATE, LONGDATE and FORMATDATETIME). Append a + or – at the end of the field, along with the number of days you wish to add or subtract, and AutoMate™ will send the current date offset by the number of days you requested.

Zip Action

The Zip Action allows you to automatically compress any number of files into one file in standard .zip format. The file can then be unzipped using AutoMate™ or any popular compression utility that supports the zip file type.

The step has the following parameters:

Files To Zip
The list of files you wish to compress into the .zip file. This line can contain one or more files, separated by semi-colons. Wildcards are fully supported. You can include multiple wildcards by separating each one with a semi-colon (for example, to compress all the files with a file extension of .txt and .cpp in the c:\tocompress directory, the parameter would be: c:\tocompress*.txt;c:\tocompress*.cpp)

Files To Exclude (optional)
A list of one or more files that you wish to *exclude* from being included in the compression. This is usually used in conjunction with a wildcard used for the *Files To Zip* parameter to above. For example, if we wanted to compress all the files in the c:\tocompress directory as indicated in the example above, but we *did not* want to compress the leavemeout.cpp file in that directory, we would put "c:\tocompress\leavemeout.cpp" in the *File To Exclude* parameter. Wildcards and multiple files separated by semi-colons is also accepted here.

Name of Archive
The name, including full path, of the zip file to create.

Include subdirectories
If this is selected, AutoMate™ will automatically recurse into the subdirectories of the directories you include in *Files To Zip*.

Preserve directory structures
If this is selected, AutoMate™ will include directory information into the compressed file so that the files can be restored to the same directories when you unzip the file. If this is not checked off, AutoMate™ will not store the directory information, and the directory structure of your files will be ignored.

The step fails if the zip file can not be created (if there is insufficient disk space for either the final zip file *or* the temporary files that may be created during the compression process) or if one or more of the files indicated in *Files To Zip* cannot be found.

See Also:

Unzip Action

Unzip Action

The Unzip Action will unzip any valid .zip file into the directory you specify.

The step requires the following parameters:

File to unzip
The filename (including path) to unzip/decompress.

Extract path
The directory where the zip file's contents should be extracted to. Note that the directory you supply here must already exist before the step is executed.

Overwrite existing files
If this is selected, AutoMate™ will replace any files that already exist with the contents of the zip file without warning. If this is not selected, AutoMate™ will not decompress (i.e. extract) a file if it already exists in the *extract path*.

Preserve directory structure
If this is selected, AutoMate™ will preserve the directory structure as it was in the zip file, creating an necessary directories and placing the unzipped files into them to duplicate the structure that was present when the zip file was created (the zip file must have been created with the *preserve directory structure* flag selected). If this is not selected, AutoMate™ will unzip all the files in the zip file to the same directory as set by the *extract path*.

The step fails if AutoMate™ cannot unzip the file, either because the zip file cannot be found., there is insufficient disk space, or the file is not in a valid zip format.

See Also

Zip Action

File Change Trigger

TO DO: Enter topic text here!

Populate Clipboard With Variable Action

The Populate Clipboard With Variable Action will overwrite the current textual contents of the system clipboard with the contents of the variable you select.

The step has the following parameter:

Variable Name
The name of the variable you want to use to populate the system clipboard. The variable must have already been created using the Create Variable action.

The step fails if the variable name you specify does not exist, or if the clipboard cannot be written to for any reason.

See Also:

Inline Expressions

Creating Variables

Setting Variables

Inputing user data into a variable

Sending variable contents as keystrokes

Variable Functions

What is AutoMate Enterprise Server?

AutoMate Enterprise Server (AES) is a separately available server software package when coupled AutoMate client software, is a complete software solution for achieving total Network Automation. It is designed to save IT Professionals time and money in controlling small to large groups of PC's on a network. And to eliminate running from machine to machine performing rudimentary system administration tasks.

The AutoMate Enterprise system is comprised of 2 key components:

AutoMate Professional

This runs on the client machines and is used as both the sending and receiving device when configured to connect to an AutoMate Enterprise Server. (For an in depth description of AutoMate please see the AutoMate help file)

AutoMate Enterprise Server

This system service runs on a Windows NT Server and handles request brokering and security rights between AutoMate installations.

IT Professionals can use AutoMate Professional to easily create AutoMate tasks (sometimes called macros) and deploy them through the ***AutoMate Enterprise Server*** which ***acts as the connector and the traffic cop*** enforcing a rights system for allowing/disallowing users from performing certain tasks. The key benefit that AutoMate Enterprise allows is a secure controlled way for a network administrator to deploy AutoMate tasks to large groups of machines simultaneously.

Deploy to groups of machines simultaneously

Forget remote control. AutoMate Enterprise Server allows you to deploy tasks to many machines simultaneously.

One of the aims of AutoMate Enterprise Server is to allow the IT Professional to just as quickly package a task and deploy it out to other machines on the network. Basically any administrative task that can be performed in AutoMate can now be sent and executed network wide (or selectively) with just a few mouse clicks. Since AutoMate's native protocol is TCP\IP (the language of the Internet) the stations on an AutoMate Enterprise Network can be either across the office or across the world.

Central Rights Management

Another aim of AutoMate Enterprise Server was to address concerns of central management of large installations of AutoMate. Utilizing AutoMate Enterprise Server, the IT Professional now has the ability to control many aspects of AutoMate's operation on each installed client – such as whether the user will see the trademark blue AutoMate icon and/or be able to create and or deploy tasks to other AutoMate users. Rights are assigned / removed from a user via the AutoMate Enterprise Server Security Manager utility.

See Also:

Installing and Configuring the Software

Deploying Tasks

Attaching Files to Deploy with Tasks

AutoMate Enterprise Security System

AutoMate Basics

AutoMate Enterprise Server Security System

The AutoMate Enterprise Security system is a transaction-based system that utilizes two components:

1) The Enterprise Server (AES.EXE)

2) The Security Database (AESDB.exe)

These two executables are installed and located on the same machine. These are both implemented as NT System services and they communicate with one another.

In future versions, these services may be able to be located on different machines for scalability and high transaction volume.

The Enterprise Server Service is responsible for handing the TCP/IP communication between clients, it acts as a traffic director, accepting requests to deploy tasks and or change items in the security databases. For example, when the Enterprise Server Service receives a request for a list of machines that are running on the network it requests the Security Database service to perform a lookup. Based on the information it is given at that time it determines whether the user has rights to do so, if it does, it will request that the security database enumerate it's user list and will in turn pass this information back to the client who requested it.

The communication between client and server is performed over native TCP/IP, so you may be located in the same office or on the other side of the world. Most transactions (with the exception of the request to be added to the user list) must be performed by a logged in User and the requesting machine must be registered on that AutoMate Enterprise Server. For a machine to be "registered" the requesting client must have completed the "registration wizard" which guides the user through this process including requesting an Administrator username and password. If an appropriate Administrator user name and password is not given the machine is not allowed on the server. If the Administrator username and password is correct the machine is added to the database and if the username is not already in the list then it is added to the guest group and it inherits whatever rights are assigned to the guest group. Users may later be moved to another group with more rights later should the Administrator wish it using the Security Manager Tool that is included with the server package (under the AutoMate 4 | Server Tools program group). This dual level of security (Users and Machines) provides a strong model to prevent unauthorized access to run tasks on your network.

This is an important issue as someone who is able to gain access to deploy tasks to the network could build a malicious task and in theory do anything he/she wants to all the machines that are logged into the Enterprise Server. Unisyn has worked hard to provide a robust security model that assures this does not happen.

IMPORTANT NOTE:

It is important to note that the AutoMate Enterprise Server security system only allows assignment of rights to groups (of either machines or users). It is not possible to assign rights to individual users. If you would like to establish a new set of rights for an individual, you must create a new group, assign the desired rights to it and add the desired user(s) to the group.

See Also:

AutoMate Enterprise Rights

Defining the Initial Administrator Account

Upon installation of AutoMate Enterprise Server you will be prompted for an Administrator user name and password that you would like to use. It is extremely important that you do not misplace this Username and Password. If you loose it – you will need to rebuilt your entire Security database (Users, User Groups, Machines, Machine Groups, Associated Rights) which can take some time to configure.

IMPORTANT: DO NOT LOOSE YOUR ADMINISTRATOR USER NAME AND PASSWORD

See Also:

Installing and Configuring the Software

AutoMate Enterprise Rights

The purpose of rights is to enable or disable certain transactions or features of either the AutoMate Enterprise Server Security Manager or the AutoMate client for groups of users. There are many rights that may be applied or removed from groups of users, these include:

Login on a machine

Deploy a composed task

View a list from a client

Delete a task from a client

Run a task on a client

Add a task on a client

View the list of users

View the list of Clients

View the client group list

View the user group list

View the list of Rights

Add a Right

Create a User

Create a client

Create a group of Clients

Create a group of users

Remove a right

Remove a Client

Remove a group of Clients

Remove a group of users

Remove a user

View the user's details

View the client's details

Add a user in a group

Add a client in a group

Modify a user

Modify a client

Remove a User from a Group

Remove a Client from a Group

Enumerate the Groups of a user

Enumerate the users of a group

Enumerate Groups of a client

Enumerate clients of a group

Modify a user's group

Modify a clients group

Delete the invalid rights

Deploy a simple task (no step)

Login on the Security Manager

Start Settings

See Icon

IMPORTANT NOTE:

It is important to note that the AutoMate Enterprise Server security system only allows assignment of rights to groups (of either machines or users). It is not possible to assign rights to individual users. If you would like to establish a new set of rights for an individual, you must create a new group, assign the desired rights to it and add the desired user(s) to the group.

AutoMate Basics

AutoMate is designed to assist IT Professionals and others in creating automated tasks to handle repetitive or mundane work for you such as starting stopping programs at certain times, backing up files, transferring data from databases, sending files through the Internet, and more. Unlike scripting and batch files, these tasks are designed visually using drag and drop building blocks. The building blocks that comprise AutoMate tasks are called Triggers and Actions.

Triggers control what causes the task to execute on a user's machine (when running AutoMate). Examples of available triggers include Schedule, Hotkey, or Window Watcher.

Actions (sometimes called steps) control what the task actually does upon running, actions such as Starting Programs, copying files, connecting to the Internet, starting and stopping services and more (over 100 available) are arranged sequentially in the order which the are meant to be executed via AutoMate's visual design tool – the Step Builder.

This building block approach allows extremely rapid development of AutoMate tasks for standalone machines.

See Also:

What is AutoMate Enterprise?

Configuring AutoMate to Connect to AES

In order to work with an AutoMate Enterprise Server, AutoMate is configured at installation or in the preferences to connect to an AutoMate Enterprise Server (as opposed to standalone). When configured this way, at startup, AutoMate will contact the AutoMate Enterprise server and if the machine is not added to the security database AutoMate will prompt the user with the "Registration Wizard". This wizard is responsible for gathering the information for how the user and the machine will be named on the Enterprise Server, furthermore in order to be added the user is requested to enter the user name and password of a AutoMate Enterprise Administrator. This is the username and password that was entered into the AutoMate Enterprise Server at the time of installation. For more information see AutoMate Enterprise Security Introduction .

See Also:

Installing and Configuring the Software

Deploying Tasks

Tasks are created and deployed from inside AutoMate Professional. Upon selecting the Deploy toolbar option in AutoMate, the user is given options to control whether the task will run immediately upon receipt, whether the task should stay in the recipient users task list, and whether to request authorization from the receiving party on receipt of the task.

Provided the user attempting to deploy tasks has rights to do so, deploying a task is easy – the user simply performs the following simple steps:

Steps to Deploy a Task

1) Select the task(s) to be deployed

2) Press the Deploy button on the toolbar

3) A list of computers will be displayed, select the computers you wish to receive the task

4) Press Deploy.

(A status screen will be displayed showing the progress of the deployment to each user you selected)

See Also:

Attaching Files to Deploy with Tasks

Attaching Files to Deploy with Tasks

Introduction

Dependant files, **called "Deployables"** -- such as BASIC scripts, Application Installations, or any other file you would like sent to another AutoMate client or client(s) – may be attached to tasks through the *Deployables* section of the task properties. **Deployables have powerful properties that allow the user to build tasks very flexibly so that they are portable across machines.** To define a Deployable, the user must specify three important fields:

a) Name – The name by which this deployable should be referenced in inside the task

b) Local Path – The path and filename of the file to be included (system directory constants may be used)

c) Remote Path – The path and filename the file should be extracted to upon receipt by a remote client (system directory constants may be used)

Multiple Deployables may be attached to an AutoMate task, if a task step needs to reference a deployable (in any of the actions that utilize files) simply check the box on the action labeled make this a Deployable and select the deployable name from the dropdown list.

Important points about *Deployables*:

a) When a task with Deployables runs *locally* -- the Deployables *Local Path* is used.

b) When a task is received by another AutoMate client on an AutoMate Enterprise Server network from a 'deployer' (or imported manually) -- the file sent with the task and is automatically extracted to the *remote path* (all non-existing directories are created automatically). The remote path value in the local path field is then overwritten by the value in the remote path field (since this is where the file now exists on this machine).

The Task steps themselves still reference the Deployable by its name, so that it may be again deployed from that location.

Several Ways to Add Deployables

You add a deployable to a task either by selecting the deployables tab in the task properties and selecting add or by clicking the make this a deployable checkbox in an action as you add it in the step builder. If the name used does not already exist, a new deployable will be created for that task. Please not that if you need to attach many files to a task as deployables at once that the easiest way to add them is in the task properties via drag and drop from Windows explorer.

Tutorial Example

For example, assume you wanted to send a task that plays a WAV file sound to 5 workstations on your AutoMate Enterprise Network. On your machine the file is located in C:\Documents\sounds\ and is called TEST.WAV. Since some of these machines are located in other offices around the world it is not possible or practical to play the file on a centrally located Drive and/or directory on the network. You have decided that the best place to put this file when received on their computers is into a directory called c:\SOUNDS\. Not to fear! AutoMate allows you to send as many files as you need to with a deployed task and tell the task where to put the file on their machine when it arrives!

The steps to accomplish the task are easy (this looks long but it is a thorough treatment of deployables and their operation).

1) Click New (in the AutoMate Configuration Manager) on the toolbar

2) Click Next to proceed through the Task Wizard, when prompted, click “Create Steps”.

3) The Step Builder will appear this is where we build our task and tell AutoMate what actions to perform

4) We decided we only want to play a sound, so, simply drag the Play a sound action from the left pane (Available Actions) to the right (Current Steps).

5) When prompted for the path information, we cannot use a local path (C:\Documents\sounds\test.wav) because it will not exist on the target machine! So instead, we will check the box on the lower left called “Make this a Deployable”.

6) When this box is checked the dialog expands to reveal 3 additional fields these are: Deployable Name, Local Path, and Remote Path

7) Type WAVFile1 for the deployable name. The name really could be anything or you could select from the drop down to choose a Deployable that has previously been added to the task (via the Task Properties, Deployables tab). This field does not need to match the file name although should describe it somewhat as it will be the unique identifier for the Deployable file.

8) For the local path simply type the path that you would have if this was only going to be running on your machine (C:\Documents\sounds\test.wav). When the task is run, exported, or deployed from your machine, this is the path that will be used to obtain the file for inclusion in the task. In other words, when this task is run, and AutoMate senses that instead of a filename, you have specified a Deployable name (this is indicated by a word surrounded by dual pipe symbols, i.e. ||DEP1||) -- AutoMate will perform a lookup for that Deployable name in that task's list of Deployables.

9) The Remote Path should be set to whatever path you would like the file to be extracted to upon receipt by another AutoMate client either by importing or when you deploy the task. You may use system constants in this field such as %AMDIR%, or %AMWINDIR% to deal with different drive letters or directories from machine to machine. But for our example we will use a static path and filename, enter c:\sounds\test.wav into the Remote Path field. It is important to remember to also enter the filename.

10) Now add the step to the task, you will notice upon saving the step that the SOUND step only references the deployable name, all resolution of the actual filename and path will be performed at runtime by looking at the Local Path of the named deployable.

11) Save your task. Proceed through the wizard, we are going to make the task run immediately so you do not need any triggers... however you could just as easily send it to their task list to run on a schedule or any other trigger.

12) After completing the New Task wizard, highlight the newly created task and press “Deploy” on the Configuration Manager toolbar. If you are logged on to an AutoMate Enterprise Server with rights to perform this action, you will see a list of other machines that are registered with this AutoMate Enterprise Server. The computer icons in the list will be blue for those users that are logged in to the, for those who are not (the machine is off/AutoMate is not running) server the icons will be gray.

13) Highlight the 5 computers you wish to deploy the task to (to multi-select, hold down the CTRL key and click on each user once). When the are all selected... press deploy.

14) You will see a status display that will show that status of the deployment to each machine. The file TEST.WAV specified as a Deployable has gone with the task and upon receipt has been extracted to the path specified in the Remote Path field. Since on that machine the file resides only in that path, the Local Path information is discarded by the receiving machine and the Local Path is over written with the value in remote path. The task steps still reference the Deployable. As the task begins to run on the remote machine, you will see real-time feedback on its progress and whether it successfully ran or not. That's it, we are done!

See Also

Deploying Tasks

Installing and Configuring the Software

Introduction

This section is designed to be a quick and easy step by step reference to getting your AutoMate Enterprise Network Going as rapidly as possible.

First of all, you must own a few licenses of AutoMate and an AutoMate Enterprise Server license that matches the number of users you wish to connect together concurrently. You can buy these either separately or together in the "AutoMate Enterprise Complete" package.

For more information on this call Unisyn Software at (888)7-UNISYN Ext. 1. Assuming you have these things this step by step guide should help you get up and running with AutoMate Enterprise as quickly as possible by going through the installation for both server and AutoMate client and establishing connectivity between them.

Installing the Server

1) Proceed to a Windows NT/2000 Workstation or server that will be used as the designated AutoMate Enterprise Server. You may run other processes on it, but you will want it to be a machine that has a minimum of downtime as any server application would require. Be advised that it does **not** require you use Windows NT Server or Windows 2000 Server it can be the Workstation or Professional version of the operating system.

2) To begin the installation, start the AutoMate Enterprise Server installation program. Follow the steps to install to the desired folder of your choice. If the machine already has programs that use the Borland Database Engine (BDE) you may need to adjust the BDE installation settings – otherwise leave those settings at their default.

3) As the last step in the server install wizard you will be prompted to select an Administrator user name and password for your AutoMate Enterprise Network. When you choose the username and password – it is extremely important that you write this information down and store it in a safe place. If you loose or forget this password, you will need to start over.

4) Once the server installation has completed – check the Services Control Panel applet to see of the two required services are up and running. The services are called: AutoMate Enterprise Server and AutoMate Security Server. In order for your AutoMate Enterprise network to function – both or these services must be operational. If either or both of these services do not start you will probably need to reinstall / reboot and/or try different BDE settings.

5) That is all there is to it for the server! The remainder of the setup will be a matter of setting up and configuring the individual client machines with AutoMate.

6) OPTIONAL – CONFIGURING THE RIGHTS OF THE GUEST GROUP: By default all machines that you configure to connect to this server will automatically go into the GUEST group. At this point you may optionally change the rights of the GUEST group using the Security Manager Tool. The shortcut to start this tool is installed in the Start menu program group AutoMate | Server Tools | Security Manager. Since it uses TCP/IP, the Security Manager does not need to be run on the same machine. Upon startup of the Security Manager, you will be prompted to logon with your Administrator username and password (the one you selected during the server install) -- you also have a choice of the server to logon to. If you are running on the same machine as the server is installed on, type "localhost" for the Server, otherwise enter the server machine's IP Address, URL, or machine name (if on the same network). Much of the Security Manager interface design is modeled after the NT User Manager utility so to NT administrators it should feel at home. You may modify the rights of the guest group by selecting the Users Tab | right clicking on the GUEST group and selecting properties. Once again, this is optional. Nobody will be able to register a client machine on your AutoMate Enterprise

Server unless they have the Administrator user name and password that you selected during the server install. When you are done modifying the rights to the GUEST group you may begin installing the clients.

Installing the AutoMate Client Software (Manual Mode)

1) Obtain the latest version of the AutoMate installation from Unisyn Software. You must be using 4.3e or later.

2) Run the AutoMate installation program. After specifying the standard information (folder location, etc.) you will prompted as to whether you would like to run this AutoMate in standalone mode or "Connect to an AutoMate Enterprise Server", select Connect to an AutoMate Enterprise Server

3) At this point the installation will begin. When complete, reboot if you are prompted.

4) When AutoMate starts and you are prompted to run the Task service answer "Yes". Upon Startup of the AutoMate Configuration Manager, because you chose during the install to "Connect to an AutoMate Enterprise Server", a wizard will appear to assist you in connecting this computer to the AutoMate Enterprise Server. The wizard will gather information and when complete it will register this machine and create the username in the security Database.(which will automatically be added to the Guest group in your AutoMate Enterprise Server). All machines that will be controlled by the AutoMate Enterprise Server must be registered and a username must be used so that rights may be applied.

5) Follow the instructions on the wizard, for the server name enter either the servers IP address or the URL (if it has one), or if it is on the local network use the machine name.

6) You will also be prompted for the username, password, and machine name that you would like to use on this. As the last step you will need to enter the administrator username and password.

7) Upon completion of the wizard an attempt will be made to connect to the server and register with the server. After proper authorization, an account will automatically be created in the GUEST group.

8) Repeat for all machines that you wish to connect to the AutoMate Enterprise Server.

9) When complete, you may connect to the server using the Security Manager and move the newly created users out of the GUEST group and into the USER group or news group(s) of your choosing. You can assign rights to the group by editing the properties of the group. Rights may only be assigned to groups, not individual users.

Installing the AutoMate Client Software (Silent Mode)

The AutoMate client software can also be installed silently in automatic mode. The installation supports several command line options to facilitate this:

/s – Silent Mode

/r – Register with AutoMate Enterprise Server mode (causes the installation to watch for the following additional parameters)

if "/r" is specified:

/sn = server name

/ct = connection type (0=LAN, 1 = RAS)

/mn = machine name

/md = machine desc

/un = username

/up = user password

/ud = user description

/an = admin name

/ap = admin password

/rn = RAS Name

/al = AutoLogon (0,1)

/pl = PromptLogon (0,1)

For AutoMate NT Service Edition there are three additional parameters, which are used regardless of the use of /r:

/du = default username

/dp = default password

/dd = default domain/ machinename

Example:

C:\AUTOMATE.EXE /s /r /sn=server2 /ct=0 /mn=hercules /md=athome /un=scott /up=unisynrules /an=Administrator /ap=passwordhere /al=1 /pl=0

Ready to Deploy Tasks!

That's all there is to it! At this point you can start AutoMate on your machine logging in with the Administrator login you selected daring the install to the server and begin building tasks and deploying them. Once a task is built, to deploy it simply select the "Deploy" button. You may deploy the task to as many workstations as you wish... they can either run the task immediately or you can schedule it for later.

See Also:

Deploying Tasks